

## General description

The DA14680 is a flexible System-on-Chip combining an application processor, memories, cryptography engine, power management unit, digital and analog peripherals and a Bluetooth® Smart MAC engine and radio transceiver.

The DA14680 is based on an ARM® Cortex®-M0 CPU delivering up to 84 DMIPS and provides a flexible memory architecture, enabling code execution from embedded memory (RAM, ROM) or non-volatile memory (FLASH, OTP).

The advanced power management unit of the DA14680 enables it to run from primary and secondary batteries, as well as provide power to external devices. The on-chip charger and state-of-charge fuel gauge allow the DA14680 to natively charge rechargeable batteries over USB.

An on-chip PLL enables on-the-fly tuning of the system clock between 32 kHz and 96 MHz to meet high processing requirements. Several optimised sleep modes are available to reduce power dissipation when there is no activity.

## Features

- Complies to *Bluetooth v4.2*, ETSI EN 300 328 and EN 300 440 Class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan)
- Flexible processing power
  - 0 Hz up to 96 MHz 32-bit ARM Cortex-M0 with 4-way associative cache
  - Three optimised power modes (Extended sleep, Deep sleep and Hibernation) reducing current to 1.4 uA
- Memories
  - 8 Mbit FLASH memory
  - 64 kB One-Time-Programmable (OTP) memory
  - 128 kB Data SRAM with retention capabilities
  - 16 kB Cache SRAM with retention capabilities
  - 128 kB ROM (including boot ROM and BLE stack)
- Power management
  - Integrated Buck DC-DC converter (1.7 V to 4.5 V)
  - Three power supply pins for external devices
- Supports Li-Polymer, Li-Ion, coin, NiMH and alkaline batteries
- Charger (up to 5.0 V) with programmable curves
- High accuracy state-of-charge fuel gauge
- Programmable threshold for brownout detection
- Digitally controlled oscillators and PLL
  - 16/32 MHz crystal oscillator
  - 16 MHz RC oscillator
  - 32 kHz crystal and RC oscillator
  - 11.7 kHz RCX oscillator
  - Low power PLL up to 96 MHz
- Three general purpose timer/counters with PWM
  - One 16-bit up/down timer/counter with PWM available in Extended sleep mode
- Application cryptographic engine with ECC, AES-256, SHA-1, SHA-256, SHA-512 and True Random Number Generator
- Digital interfaces
  - 31 general purpose I/Os with programmable voltage levels
  - Two UARTs, one with hardware flow control
  - Two SPI+™ interfaces
  - Two I2C-bus interfaces at 100 kHz, 400 kHz
  - Three-axes capable Quadrature Decoder
  - PDM + HW decimator (2 mics or 2 speakers)
  - I2S/PCM master/slave interface up to 8 channels
  - Keyboard scanner with debouncing
  - Infrared (IR) interface (PWM)
  - USB 1.1 Full Speed (FS) device interface
- Analog interfaces
  - 8-channel 10-bit ADC with averaging capability
  - Three matched white LED drivers
  - Temperature sensor
- Radio transceiver
  - 2.4 GHz CMOS transceiver with integrated balun
  - 50 Ω matched single wire antenna interface
  - 0 dBm transmit output power
  - -94 dBm receiver sensitivity (BLE)
  - Supply current at VBAT1 (3 V):
    - TX: 3.4 mA
    - RX: 3.1 mA (with ideal DC-DC converter)
- Packages:
  - aQFN with 60 pins, 6 mm x 6 mm

## System diagram



|                                                          |           |                                                       |           |
|----------------------------------------------------------|-----------|-------------------------------------------------------|-----------|
| <b>1.0 Block diagram</b> .....                           | <b>6</b>  | <b>8.0 AMBA Bus</b> .....                             | <b>42</b> |
| <b>2.0 Package and pinout</b> .....                      | <b>7</b>  | <b>9.0 Memory Controller</b> .....                    | <b>43</b> |
| <b>3.0 System overview</b> .....                         | <b>16</b> | <b>10.0 OTP Controller</b> .....                      | <b>45</b> |
| 3.1 INTERNAL BLOCKS .....                                | 16        | 10.1 OPERATING MODES .....                            | 45        |
| 3.2 FUNCTIONAL MODES .....                               | 17        | 10.2 AHB MASTER INTERFACE .....                       | 46        |
| 3.3 SYSTEM CONFIGURATION .....                           | 17        | 10.3 AHB SLAVE INTERFACES .....                       | 46        |
| 3.4 SYSTEM STARTUP PROCEDURE .....                       | 20        | 10.4 ERROR CORRECTING CODE (ECC) .....                | 46        |
| 3.4.1 Power/Wakeup FSM .....                             | 21        | 10.5 BUILD-IN SELF REPAIR (BISR) .....                | 46        |
| 3.4.2 Goto Sleep FSM .....                               | 21        | <b>11.0 Quad SPI Controller</b> .....                 | <b>47</b> |
| 3.4.3 BootROM sequence .....                             | 22        | 11.1 ARCHITECTURE .....                               | 48        |
| 3.5 POWER CONTROL AND MODES .....                        | 25        | 11.1.1 Interface .....                                | 48        |
| 3.5.1 System Power Control .....                         | 25        | 11.1.2 Initialization FSM .....                       | 48        |
| 3.5.2 Power domains .....                                | 25        | 11.1.3 SPI modes .....                                | 49        |
| 3.5.3 Power modes .....                                  | 27        | 11.1.4 Access modes .....                             | 49        |
| <b>4.0 Reset</b> .....                                   | <b>29</b> | 11.1.5 Endianess .....                                | 49        |
| 4.1 POR, HW AND SW RESET .....                           | 29        | 11.1.6 Erase Suspend/Resume .....                     | 49        |
| 4.2 BROWN OUT DETECTION .....                            | 30        | 11.2 PROGRAMMING .....                                | 51        |
| <b>5.0 Clock generation</b> .....                        | <b>31</b> | 11.2.1 Auto Mode .....                                | 51        |
| 5.1 CLOCK TREE .....                                     | 31        | 11.2.2 Manual Mode .....                              | 51        |
| 5.2 CRYSTAL OSCILLATORS .....                            | 32        | 11.2.3 Clock selection .....                          | 51        |
| 5.2.1 Frequency control (16 MHz crystal) .....           | 32        | 11.2.4 Received data .....                            | 52        |
| 5.2.2 Automated trimming and settling notification ..... | 32        | <b>12.0 DMA Controller</b> .....                      | <b>53</b> |
| 5.2.3 Using a 32 MHz crystal .....                       | 33        | 12.1 DMA PERIPHERALS .....                            | 54        |
| 5.3 RC OSCILLATORS .....                                 | 34        | 12.2 INPUT/OUTPUT MULTIPLEXER .....                   | 54        |
| 5.3.1 Frequency calibration .....                        | 34        | 12.3 DMA CHANNEL OPERATION .....                      | 54        |
| 5.4 PLL .....                                            | 34        | 12.4 DMA ARBITRATION .....                            | 55        |
| <b>6.0 ARM Cortex M0</b> .....                           | <b>35</b> | 12.5 FREEZING DMA CHANNELS .....                      | 55        |
| 6.1 SYSTEM TIMER (SYSTICK) .....                         | 36        | <b>13.0 AES/Hash Engine</b> .....                     | <b>56</b> |
| 6.2 WAKEUP INTERRUPT CONTROLLER .....                    | 36        | 13.1 DESCRIPTION .....                                | 56        |
| 6.3 REFERENCE .....                                      | 36        | 13.2 ARCHITECTURE .....                               | 57        |
| 6.4 INTERRUPTS .....                                     | 37        | 13.2.1 AES/HASH engine .....                          | 57        |
| <b>7.0 Cache Controller</b> .....                        | <b>39</b> | 13.2.2 AES .....                                      | 57        |
| 7.1 CACHABLE RANGE .....                                 | 39        | 13.2.3 Modes .....                                    | 58        |
| 7.2 RUNTIME RECONFIGURATION .....                        | 40        | 13.2.4 HASH .....                                     | 58        |
| 7.2.1 Cache Line reconfiguration .....                   | 40        | 13.3 PROGRAMMING .....                                | 59        |
| 7.2.2 TAG memory word .....                              | 40        | <b>14.0 ECC Engine</b> .....                          | <b>61</b> |
| 7.2.3 Associativity reconfiguration .....                | 40        | 14.1 ARCHITECTURE .....                               | 61        |
| 7.3 2 AND 4 WAY REPLACEMENT STRATEGY .....               | 40        | 14.1.1 Supported curves .....                         | 62        |
| 7.4 CACHE RESETS .....                                   | 40        | 14.1.2 Supported high level algorithms .....          | 62        |
| 7.5 CACHE MISS RATE MONITOR .....                        | 41        | <b>15.0 True Random Number Generator (TRNG)</b> ..... | <b>63</b> |
|                                                          |           | 15.1 ARCHITECTURE .....                               | 63        |
|                                                          |           | 15.2 PROGRAMMING .....                                | 63        |
|                                                          |           | 15.2.1 Latency .....                                  | 63        |

|                                                            |           |                                                  |            |
|------------------------------------------------------------|-----------|--------------------------------------------------|------------|
| <b>16.0 Temperature Sensor</b> .....                       | <b>64</b> | 22.6 SHADOW REGISTERS .....                      | 87         |
| 16.1 PROGRAMMING .....                                     | 64        | 22.7 DIRECT TEST MODE .....                      | 87         |
| <b>17.0 Wakeup Timer</b> .....                             | <b>65</b> | <b>23.0 SPI+ Interface</b> .....                 | <b>88</b>  |
| <b>18.0 General purpose ADC</b> .....                      | <b>66</b> | 23.1 OPERATION WITHOUT FIFOS .....               | 88         |
| 18.1 ARCHITECTURE .....                                    | 66        | 23.2 9 BITS MODE .....                           | 89         |
| 18.2 INPUT CHANNELS AND INPUT SCALE .....                  | 68        | <b>24.0 I2C</b> .....                            | <b>92</b>  |
| 18.3 STARTING THE ADC .....                                | 68        | 24.1 I2C BUS TERMS .....                         | 92         |
| 18.4 ADC CONVERSION MODES .....                            | 68        | 24.1.1 Bus Transfer Terms .....                  | 93         |
| 18.4.1 Manual Mode .....                                   | 68        | 24.2 I2C BEHAVIOUR .....                         | 93         |
| 18.4.2 Continuous Mode .....                               | 68        | 24.2.1 START and STOP Generation .....           | 94         |
| 18.5 NON-IDEAL EFFECTS .....                               | 68        | 24.2.2 Combined Formats .....                    | 94         |
| 18.6 OVERSAMPLING .....                                    | 69        | 24.3 I2C PROTOCOLS .....                         | 94         |
| 18.7 CHOPPING .....                                        | 69        | 24.3.1 START and STOP Conditions .....           | 94         |
| 18.8 OFFSET CALIBRATION .....                              | 69        | 24.3.2 Addressing Slave Protocol .....           | 94         |
| 18.9 ZERO-SCALE ADJUSTMENT .....                           | 70        | 24.3.3 Transmitting and Receiving Protocol ..... | 95         |
| 18.10 COMMON MODE ADJUSTMENT .....                         | 70        | 24.4 MULTIPLE MASTER ARBITRATION .....           | 97         |
| 18.11 INPUT IMPEDANCE, INDUCTANCE AND INPUT SETTLING ..... | 70        | 24.5 CLOCK SYNCHRONIZATION .....                 | 98         |
| <b>19.0 Sample rate Converter (SRC)</b> .....              | <b>71</b> | 24.6 OPERATION MODES .....                       | 99         |
| 19.1 ARCHITECTURE .....                                    | 71        | 24.6.1 Slave Mode Operation .....                | 99         |
| 19.1.1 I/O channels .....                                  | 71        | 24.7 MASTER MODE OPERATION .....                 | 101        |
| 19.1.2 I/O multiplexers .....                              | 71        | <b>25.0 InfraRed Generator</b> .....             | <b>102</b> |
| 19.1.3 Input and Output Sample rate conversion .....       | 72        | 25.1 ARCHITECTURE .....                          | 102        |
| 19.1.4 SRC conversion modes of operation .....             | 72        | 25.2 PROGRAMMING .....                           | 103        |
| 19.1.5 DMA operation .....                                 | 72        | <b>26.0 Quadrature Decoder</b> .....             | <b>104</b> |
| 19.1.6 Interrupts .....                                    | 72        | 26.1 ARCHITECTURE .....                          | 104        |
| 19.1.7 SRC use cases .....                                 | 72        | 26.2 PROGRAMMING .....                           | 104        |
| <b>20.0 PDM interface</b> .....                            | <b>73</b> | <b>27.0 Keyboard Scanner</b> .....               | <b>105</b> |
| <b>21.0 PCM Controller</b> .....                           | <b>75</b> | 27.1 ARCHITECTURE .....                          | 105        |
| 21.1 ARCHITECTURE .....                                    | 76        | 27.2 PROGRAMMING .....                           | 107        |
| 21.1.1 Interface Signals .....                             | 76        | <b>28.0 Software Timers</b> .....                | <b>108</b> |
| 21.1.2 Channel ACCESS .....                                | 76        | 28.1 TIMER0 .....                                | 108        |
| 21.1.3 Channel delay .....                                 | 76        | 28.2 TIMER1 .....                                | 110        |
| 21.1.4 Clock generation .....                              | 76        | 28.3 TIMER2 .....                                | 111        |
| 21.1.5 DATA FORMATS .....                                  | 77        | 28.4 BRTH_TIMER .....                            | 112        |
| 21.1.6 IOM mode .....                                      | 79        | <b>29.0 Watchdog Timer</b> .....                 | <b>113</b> |
| 21.1.7 External synchronisation .....                      | 79        | <b>30.0 USB Interface</b> .....                  | <b>114</b> |
| <b>22.0 UART</b> .....                                     | <b>81</b> | 30.1 SERIAL INTERFACE ENGINE .....               | 115        |
| 22.1 UART (RS232) SERIAL PROTOCOL .....                    | 82        | 30.2 ENDPOINT PIPE CONTROLLER (EPC) .....        | 115        |
| 22.2 IRDA 1.0 SIR PROTOCOL .....                           | 83        | 30.3 FUNCTIONAL STATES .....                     | 116        |
| 22.3 CLOCK SUPPORT .....                                   | 84        | 30.3.1 Line Condition Detection .....            | 116        |
| 22.4 INTERRUPTS .....                                      | 85        | 30.4 FUNCTIONAL STATE DIAGRAM .....              | 117        |
| 22.5 PROGRAMMABLE THREE INTERRUPT .....                    | 85        | 30.5 ADDRESS DETECTION .....                     | 119        |

|                                                                |                                                |
|----------------------------------------------------------------|------------------------------------------------|
| 30.6 TRANSMIT AND RECEIVE ENDPOINT FIFOS<br>120                | 34.1.5 Biasing ..... 139                       |
| 30.7 BIDIRECTIONAL CONTROL ENDPOINT FIFO0<br>121               | 34.1.6 Control ..... 139                       |
| 30.8 TRANSMIT ENDPOINT FIFO (TXFIFO1 TO<br>TXFIFO5) ..... 123  | 34.2 DYNAMIC CONTROLLED FUNCTIONS .. 139       |
| 30.9 RECEIVE ENDPOINT FIFO (RXFIFO2 TO<br>RXFIFO6) ..... 123   | 34.3 DIAGNOSTIC SIGNALS ..... 139              |
| 30.10 INTERRUPT HIERARCHY ..... 124                            | <b>35.0 Power management ..... 141</b>         |
| 30.11 USB POWER SAVING MODES ..... 126                         | 35.1 ARCHITECTURE ..... 141                    |
| 30.11.1 Freezing USB node ..... 126                            | 35.1.1 SIMO DC-DC converter ..... 142          |
| 30.11.2 Integrated resistors ..... 126                         | 35.1.2 LDOs ..... 142                          |
| <b>31.0 Input/Output ports ..... 127</b>                       | 35.1.3 Switching from DC-DC to LDOs. .... 143  |
| 31.1 PROGRAMMABLE PIN ASSIGNMENT ... 127                       | 35.1.4 PMU configurations in Sleep modes . 143 |
| 31.2 GENERAL PURPOSE PORT REGISTERS 127                        | 35.1.5 Wake/Power up - Sleep Timing ..... 144  |
| 31.2.1 Port Data Register ..... 127                            | 35.1.6 Charger ..... 147                       |
| 31.2.2 Port Set Data Output Register ..... 128                 | 35.1.7 Fuel gauge ..... 149                    |
| 31.2.3 Port Reset Data Output Register .... 128                | 35.1.8 USB charger detection. .... 151         |
| 31.3 FIXED ASSIGNMENT FUNCTIONALITY .. 128                     | <b>36.0 Memory map ..... 154</b>               |
| 31.4 STATE RETENTION WHILE SLEEPING .. 128                     | <b>37.0 Registers ..... 156</b>                |
| 31.5 SPECIAL I/O CONSIDERATIONS ..... 128                      | 37.1 OTPC REGISTER FILE ..... 158              |
| <b>32.0 BLE Core ..... 129</b>                                 | 37.2 QSPIC REGISTER FILE ..... 165             |
| 32.1 ARCHITECTURE ..... 129                                    | 37.3 BLE REGISTER FILE ..... 174               |
| 32.1.1 Exchange Memory ..... 129                               | 37.4 AES_HASH REGISTER FILE ..... 193          |
| 32.2 PROGRAMMING ..... 129                                     | 37.5 CACHE REGISTER FILE ..... 197             |
| 32.2.1 Wake up IRQ ..... 129                                   | 37.6 CRG REGISTER FILE ..... 201               |
| 32.2.2 Switch from Active Mode to Deep Sleep<br>Mode ..... 130 | 37.7 DCDC REGISTER FILE ..... 212              |
| 32.2.3 Switch from Deep Sleep Mode to Active<br>Mode ..... 130 | 37.8 WAKEUP REGISTER FILE ..... 222            |
| 32.2.4 Switching on at anchor points ..... 130                 | 37.9 TIMER1 REGISTER FILE ..... 225            |
| 32.2.5 Switching on due to an external event. 132              | 37.10 UART REGISTER FILE ..... 228             |
| 32.3 DIAGNOSTIC SIGNALS ..... 133                              | 37.11 SPI REGISTER FILE ..... 290              |
| 32.4 POWER PROFILE ..... 134                                   | 37.12 I2C REGISTER FILE ..... 295              |
| 32.4.1 Advertising Event ..... 134                             | 37.13 KEYBOARD SCAN REGISTER FILE .... 330     |
| 32.4.2 Connection Event ..... 135                              | 37.14 IR REGISTER FILE ..... 338               |
| <b>33.0 CoEx interface ..... 136</b>                           | 37.15 USB REGISTER FILE ..... 341              |
| 33.1 ARCHITECTURE ..... 136                                    | 37.16 GPADC REGISTER FILE ..... 372            |
| 33.2 PROGRAMMING ..... 136                                     | 37.17 QUADRATURE DECODER REGISTER FILE.<br>375 |
| <b>34.0 Radio ..... 138</b>                                    | 37.18 ANAMISC REGISTER FILE ..... 376          |
| 34.1 ARCHITECTURE ..... 138                                    | 37.19 CRG REGISTER FILE ..... 384              |
| 34.1.1 Receiver ..... 138                                      | 37.20 COEX REGISTER FILE ..... 386             |
| 34.1.2 Synthesizer ..... 138                                   | 37.21 GPIO REGISTER FILE ..... 394             |
| 34.1.3 Transmitter ..... 139                                   | 37.22 WDOG REGISTER FILE ..... 411             |
| 34.1.4 RFIO ..... 139                                          | 37.23 VERSION REGISTER FILE ..... 412          |
|                                                                | 37.24 GPREG REGISTER FILE ..... 414            |
|                                                                | 37.25 TIMER0/2 AND BREATH REGISTER FILE 418    |
|                                                                | 37.26 DMA REGISTER FILE ..... 421              |
|                                                                | 37.27 APU REGISTER FILE ..... 440              |
|                                                                | 37.28 TRNG REGISTER FILE ..... 446             |

|                                          |            |
|------------------------------------------|------------|
| 37.29 ELLIPTIC CURVE CONTROLLER REGISTER |            |
| FILE.....                                | 447        |
| <b>38.0 Specifications.....</b>          | <b>452</b> |
| <b>39.0 Package information .....</b>    | <b>479</b> |
| 39.1 MOISTURE SENSITIVITY LEVEL (MSL) .. | 479        |
| 39.2 SOLDERING INFORMATION.....          | 479        |
| 39.3 PACKAGE OUTLINES .....              | 480        |

## 1 Block diagram

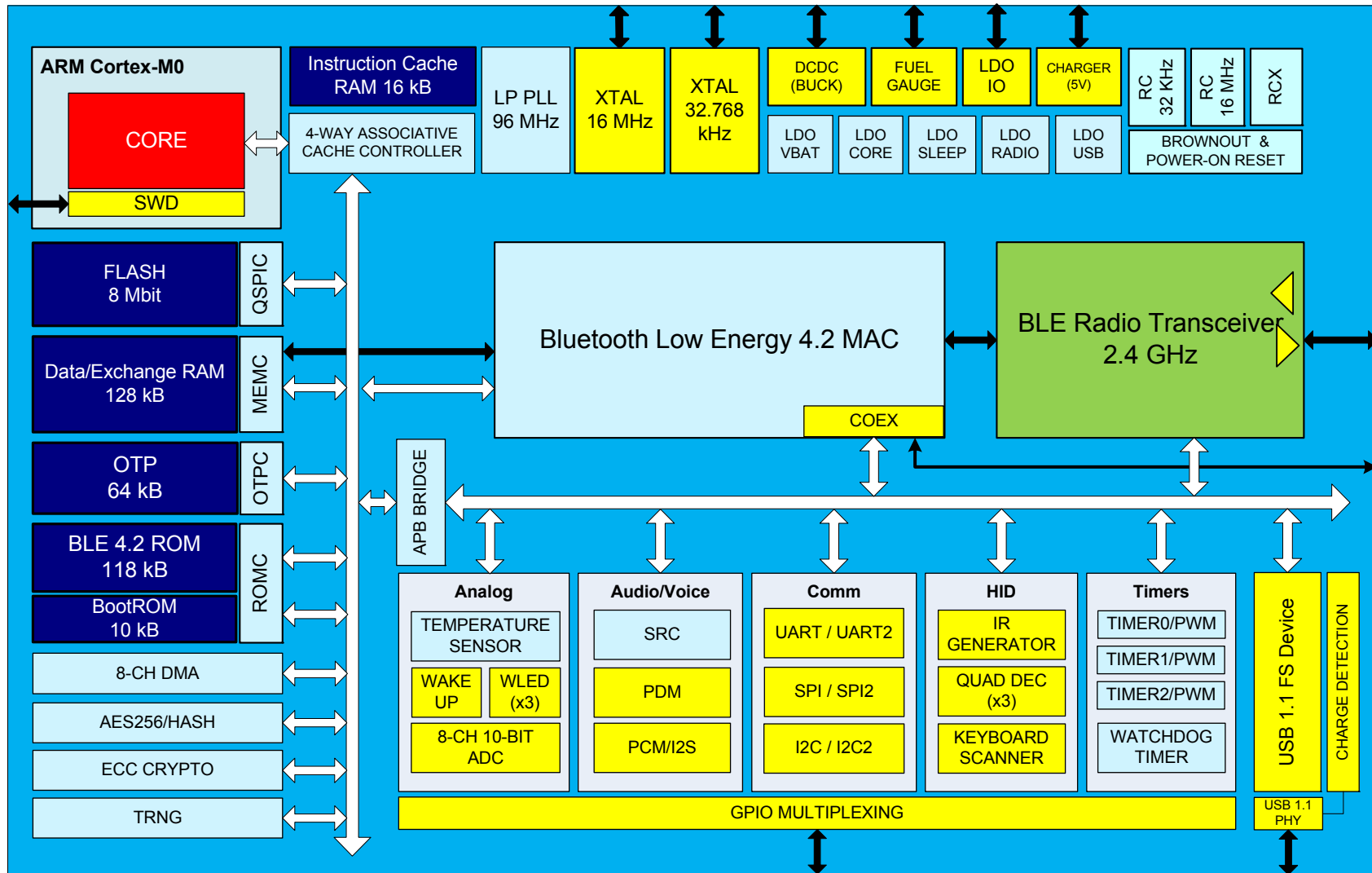


Figure 1: DA14680 block diagram

## 2 Package and pinout

The DA14680 comes in a 6 mm x 6 mm aQFN package with 60 pins. The pin assignment is shown in the following figure.

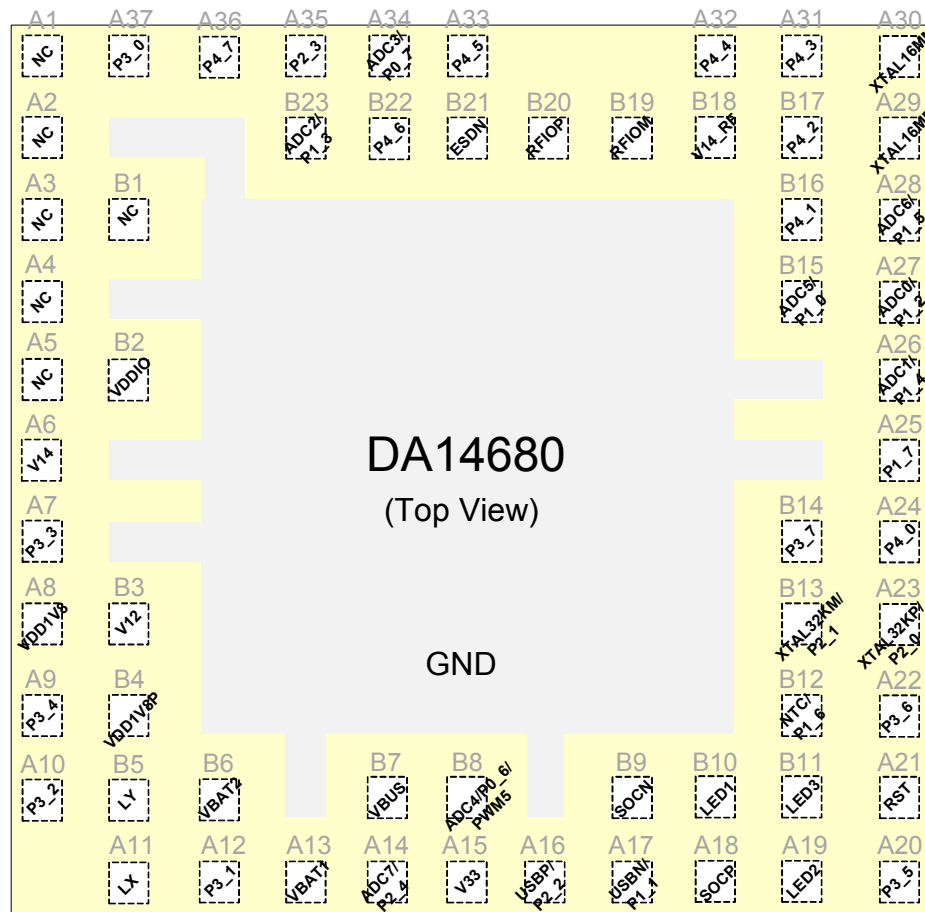


Figure 2: aQFN60 pin assignment

Table 1: Ordering information

| Part number      | Package | Size (mm) | Shipment form | Pack quantity                           |
|------------------|---------|-----------|---------------|-----------------------------------------|
| DA14680-01F08A92 | AQFN60  | 6 x 6     | Reel          | 100/1000 (samples)<br>4000 (production) |

Table 2: Pin description

| Pin no.                                                                                                    | Pin name                                 | Type                           | Drive (mA) | Reset state | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------|------------------------------------------|--------------------------------|------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>General Purpose I/Os (fixed pin assignment; additional functions are programmable via Pxx_MODE_REG)</b> |                                          |                                |            |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| B8                                                                                                         | P0_6/<br><br>SWDIO/<br>PWM5/<br><br>ADC4 | DIO<br><br>DIO<br>DO<br><br>AI | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT/OUTPUT. JTAG data I/O signal.<br>OUTPUT. Timer 1 PWM output (PWM5) in Sleep mode.<br><b>Note:</b> This is the only pin with output capability in Extended/ Deep Sleep mode.<br>INPUT. Analog input for ADC channel 4. |
| A34                                                                                                        | P0_7/<br><br>ADC3                        | DIO<br><br>AI                  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input for ADC channel 3.                                                                                                                                                                                      |
| B15                                                                                                        | P1_0/<br><br>ADC5                        | DIO<br><br>AI                  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down/up enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input for ADC channel 5.                                                                                                                                                                                   |
| A17                                                                                                        | P1_1/<br><br>USBN                        | DIO<br><br>AIO                 | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br><br><b>Note:</b> to use this pin in GPIO mode<br>USBPAD_REG[USBPAD_EN] must be set<br><br>INPUT/OUTPUT. Analog USB Full Speed D- signal.                                                                                    |
| A27                                                                                                        | P1_2/<br><br>ADC0                        | DIO<br><br>AI                  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down/up enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input for ADC channel 0.                                                                                                                                                                                   |
| B23                                                                                                        | P1_3/<br><br>ADC2                        | DIO<br><br>AI                  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down/up enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input for ADC channel 2.                                                                                                                                                                                   |
| A26                                                                                                        | P1_4/<br><br>ADC1                        | DIO<br><br>AI                  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down/up enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input for ADC channel 1.                                                                                                                                                                                   |
| A28                                                                                                        | P1_5/<br><br>ADC6                        | DIO<br><br>AI                  | 4.8        | I-PU        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down/up enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input for ADC channel 6.                                                                                                                                                                                   |



Table 2: Pin description

| Pin no. | Pin name                    | Type                | Drive (mA) | Reset state | Description                                                                                                                                                                                                                                                                                                                                                      |
|---------|-----------------------------|---------------------|------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| B12     | P1_6/<br><br>NTC            | DIO<br><br>AI       | 4.8        | I-PU        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-up enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input for external NTC resistor for battery temperature sensing.                                                          |
| A25     | P1_7                        | DIO                 | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.                                                                                                                                          |
| A23     | P2_0/<br><br>XTAL32KP       | DIO<br><br>AI<br>DI | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. Analog input of the XTAL32K crystal oscillator.<br>INPUT. Digital input for an external clock (square wave).                   |
| B13     | P2_1/<br><br>XTAL32KM       | DIO<br><br>AO       | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>OUTPUT. Analog output of the XTAL32K crystal oscillator.                                                                              |
| A16     | P2_2/<br><br>USBP           | DIO<br><br>AIO      | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br><br><b>Note:</b> to use this pin in GPIO mode USBPAD_REG[USBPAD_EN] must be set<br><br>INPUT/OUTPUT. Analog USB Full Speed D+ signal. |
| A35     | P2_3                        | DIO                 | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.                                                                                                                                          |
| A14     | P2_4/<br><br>SWCLK/<br>ADC7 | DIO<br><br>DI<br>AI | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.<br>INPUT. JTAG clock signal.<br>INPUT. Analog input for ADC channel 7.                                                                   |
| A37     | P3_0                        | DIO                 | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.                                                                                                                                          |
| A12     | P3_1                        | DIO                 | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.                                                                                                                                          |
| A10     | P3_2                        | DIO                 | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down.                                                                                                                                          |

Table 2: Pin description

| Pin no.                | Pin name | Type | Drive (mA) | Reset state | Description                                                                                                                                                                                                             |
|------------------------|----------|------|------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A7                     | P3_3     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A9                     | P3_4     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A20                    | P3_5     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A22                    | P3_6     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| B14                    | P3_7     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A24                    | P4_0     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| B16                    | P4_1     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| B17                    | P4_2     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A31                    | P4_3     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A32                    | P4_4     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A33                    | P4_5     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| B22                    | P4_6     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| A36                    | P4_7     | DIO  | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. State retention during power down. |
| <b>Debug interface</b> |          |      |            |             |                                                                                                                                                                                                                         |

Table 2: Pin description

| Pin no.       | Pin name | Type | Drive (mA) | Reset state | Description                                                                                         |
|---------------|----------|------|------------|-------------|-----------------------------------------------------------------------------------------------------|
| B8            | SWDIO    | DIO  | 4.8        | I-PU        | INPUT/OUTPUT. JTAG Data input/output. Bidirectional data and control communication. Mapped on P0_6. |
| A14           | SW_CLK   | DIO  | 4.8        | I-PD        | INPUT JTAG clock signal. Mapped on P2_4.                                                            |
| <b>Clocks</b> |          |      |            |             |                                                                                                     |
| A29           | XTAL16MP | AI   |            |             | INPUT. Crystal input for the 16 MHz XTAL oscillator.                                                |
| A30           | XTAL16MM | AO   |            |             | OUTPUT. Crystal output for the 16 MHz XTAL oscillator.                                              |
| A23           | XTAL32KP | AI   |            |             | INPUT. Crystal input for the 32.768 kHz XTAL oscillator. Mapped on P2_0.                            |
| B13           | XTAL32KM | AO   |            |             | OUTPUT. Crystal output for the 32.768 kHz XTAL oscillator. Mapped on P2_1.                          |

Table 2: Pin description

| Pin no.                                                 | Pin name | Type         | Drive (mA) | Reset state | Description                                                                                                |
|---------------------------------------------------------|----------|--------------|------------|-------------|------------------------------------------------------------------------------------------------------------|
| <b>Quadrature decoder (mapped on port Px_y)</b>         |          |              |            |             |                                                                                                            |
|                                                         | QD_CHA_X | DI           |            |             | INPUT. Channel A for the X axis.                                                                           |
|                                                         | QD_CHB_X | DI           |            |             | INPUT. Channel B for the X axis.                                                                           |
|                                                         | QD_CHA_Y | DI           |            |             | INPUT. Channel A for the Y axis.                                                                           |
|                                                         | QD_CHB_Y | DI           |            |             | INPUT. Channel B for the Y axis.                                                                           |
|                                                         | QD_CHA_Z | DI           |            |             | INPUT. Channel A for the Z axis.                                                                           |
|                                                         | QD_CHB_Z | DI           |            |             | INPUT. Channel B for the Z axis.                                                                           |
| <b>SPI bus interface (mapped on port Px_y)</b>          |          |              |            |             |                                                                                                            |
|                                                         | SPI_CLK  | DIO          |            |             | INPUT/OUTPUT. SPI clock.                                                                                   |
|                                                         | SPI_DI   | DI           |            |             | INPUT. SPI data input.                                                                                     |
|                                                         | SPI_DO   | DO           |            |             | OUTPUT. SPI data output.                                                                                   |
|                                                         | SPI_EN   | DI           |            |             | INPUT. SPI clock enable.                                                                                   |
|                                                         | SPI2_CLK | DIO          |            |             | INPUT/OUTPUT. SPI 2 clock.                                                                                 |
|                                                         | SPI2_DI  | DI           |            |             | INPUT. SPI 2 data input.                                                                                   |
|                                                         | SPI2_DO  | DO           |            |             | OUTPUT. SPI 2 data output.                                                                                 |
|                                                         | SPI2_EN  | DI           |            |             | INPUT. SPI 2 clock enable.                                                                                 |
| <b>I2C bus interface (mapped on port Px_y)</b>          |          |              |            |             |                                                                                                            |
|                                                         | SDA      | DIO/<br>DIOD |            |             | INPUT/OUTPUT. I2C bus data with open drain port.                                                           |
|                                                         | SCL      | DIO/<br>DIOD |            |             | INPUT/OUTPUT. I2C bus clock with open drain port. Supports bit stretching by a slave in open drain mode.   |
|                                                         | SDA2     | DIO/<br>DIOD |            |             | INPUT/OUTPUT. I2C bus 2 data with open drain port.                                                         |
|                                                         | SCL2     | DIO/<br>DIOD |            |             | INPUT/OUTPUT. I2C bus 2 clock with open drain port. Supports bit stretching by a slave in open drain mode. |
| <b>UART interface (mapped on port Px_y)</b>             |          |              |            |             |                                                                                                            |
|                                                         | UTX      | DO           |            |             | OUTPUT. UART transmit data.                                                                                |
|                                                         | URX      | DI           |            |             | INPUT. UART receive data.                                                                                  |
|                                                         | UTX2     | DO           |            |             | OUTPUT. UART 2 transmit data.                                                                              |
|                                                         | URX2     | DI           |            |             | INPUT. UART 2 receive data.                                                                                |
|                                                         | URTS2    | DO           |            |             | OUTPUT. UART 2 request to send.                                                                            |
|                                                         | UCTS2    | DI           |            |             | INPUT. UART 2 clear to send.                                                                               |
| <b>Infrared (IR) interface (mapped on port Px_y)</b>    |          |              |            |             |                                                                                                            |
|                                                         | IR_OUT   | DO           |            |             | OUTPUT. Infrared data.                                                                                     |
| <b>Keyboard scanner interface (mapped on port Px_y)</b> |          |              |            |             |                                                                                                            |
|                                                         | KSC_ROWx | DO           |            |             | OUTPUT. Keyboard rows driven by the scanner.                                                               |
|                                                         | KSC_COLx | DI           |            |             | INPUT. Keyboard columns sampled by the scanner.                                                            |
| <b>PDM interface (mapped on port Px_y)</b>              |          |              |            |             |                                                                                                            |
|                                                         | PDM_CLK  | DO           |            |             | OUTPUT. PDM clock output.                                                                                  |
|                                                         | PDM_DATA | DIO          |            |             | INPUT/OUTPUT. PDM data.                                                                                    |

Table 2: Pin description

| Pin no.                                    | Pin name | Type | Drive (mA) | Reset state | Description                                                                    |
|--------------------------------------------|----------|------|------------|-------------|--------------------------------------------------------------------------------|
| <b>PCM interface (mapped on port Px_y)</b> |          |      |            |             |                                                                                |
|                                            | PCM_DO   | DO   |            |             | OUTPUT. PCM data output.                                                       |
|                                            | PCM_DI   | DI   |            |             | INPUT. PCM data input.                                                         |
|                                            | PCM_CLK  | DIO  |            |             | INPUT/OUTPUT. PCM bus clock.                                                   |
|                                            | PCM_FSC  | DIO  |            |             | INPUT/OUTPUT. PCM frame sync.                                                  |
| <b>PWM interface (mapped on port Px_y)</b> |          |      |            |             |                                                                                |
|                                            | PWM0     | DO   |            |             | OUTPUT. Pulse Width Modulated output of Timer 0.                               |
|                                            | PWM1     | DO   |            |             | OUTPUT. Pulse Width Modulated output of Timer 0.                               |
|                                            | PWM2     | DO   |            |             | OUTPUT. Pulse Width Modulated output of Timer 2.                               |
|                                            | PWM3     | DO   |            |             | OUTPUT. Pulse Width Modulated output of Timer 2.                               |
|                                            | PWM4     | DO   |            |             | OUTPUT. Pulse Width Modulated output of Timer 2.                               |
| B8                                         | PWM5     | DO   |            |             | OUTPUT. Pulse Width Modulated output of Timer 1. Mapped on P0_6 in Sleep mode. |
| <b>Analog interface</b>                    |          |      |            |             |                                                                                |
| A27                                        | ADC0     | AI   |            |             | INPUT. Analog to Digital Converter input 0. Mapped on P1_2.                    |
| A26                                        | ADC1     | AI   |            |             | INPUT. Analog to Digital Converter input 1. Mapped on P1_4.                    |
| B23                                        | ADC2     | AI   |            |             | INPUT. Analog to Digital Converter input 2. Mapped on P1_3.                    |
| A34                                        | ADC3     | AI   |            |             | INPUT. Analog to Digital Converter input 3. Mapped on P0_7.                    |
| B8                                         | ADC4     | AI   |            |             | INPUT. Analog to Digital Converter input 4. Mapped on P0_6.                    |
| B15                                        | ADC5     | AI   |            |             | INPUT. Analog to Digital Converter input 5. Mapped on P1_0.                    |
| A28                                        | ADC6     | AI   |            |             | INPUT. Analog to Digital Converter input 6. Mapped on P1_5.                    |
| A14                                        | ADC7     | AI   |            |             | INPUT. Analog to Digital Converter input 7. Mapped on P2_4.                    |
| <b>USB FS interface</b>                    |          |      |            |             |                                                                                |
| A16                                        | USBP     | AIO  |            |             | INPUT/OUTPUT. USB Full Speed D+ signal. Mapped on P2_2.                        |
| A17                                        | USBN     | AIO  |            |             | INPUT/OUTPUT. USB Full Speed D- signal. Mapped on P1_1.                        |
| <b>Radio transceiver</b>                   |          |      |            |             |                                                                                |
| B20                                        | RFIOP    | AIO  |            |             | INPUT/OUTPUT. RF transceiver signal. Impedance 50 Ω.                           |
| B19                                        | RFIOM    | AIO  |            |             | INPUT/OUTPUT. RF transceiver ground.                                           |
| <b>Miscellaneous</b>                       |          |      |            |             |                                                                                |
| A21                                        | RST      | DI   |            |             | INPUT. Reset signal (active HIGH). Maximum voltage: V33.                       |
| B10                                        | LED1     | AO   | 20         |             | OUTPUT. White LED driver 1 (open drain).                                       |
| A19                                        | LED2     | AO   | 20         |             | OUTPUT. White LED driver 2 (open drain).                                       |
| B11                                        | LED3     | AO   | 20         |             | OUTPUT. White LED driver 3 (open drain).                                       |
| A18                                        | SOC_P    | AIO  |            |             | INPUT/OUTPUT. Battery fuel gauge sense signal.                                 |
| B9                                         | SOC_N    | AIO  |            |             | INPUT/OUTPUT. Battery fuel gauge reference ground. Connect as star point.      |

### Table 2: Pin description

| Pin no.                      | Pin name | Type     | Drive (mA) | Reset state | Description                                                                                           |
|------------------------------|----------|----------|------------|-------------|-------------------------------------------------------------------------------------------------------|
| <b>Power supply</b>          |          |          |            |             |                                                                                                       |
| A13                          | VBAT1    | AI       |            |             | INPUT. Battery supply voltage for LDO.                                                                |
| B6                           | VBAT2    | AI       |            |             | INPUT. Battery supply voltage for DC-DC converter.                                                    |
| B7                           | VBUS     | AI<br>AI |            |             | INPUT. USB bus voltage.<br>INPUT. Battery charge voltage.                                             |
| B18                          | V14_RF   | AI       |            |             | INPUT. Radio supply voltage. Connect to V14 externally.<br>4.7 $\mu$ F decoupling capacitor required. |
| A11                          | LX       | AIO      |            |             | INPUT/OUTPUT. External inductor for DC-DC converter.                                                  |
| B5                           | LY       | AIO      |            |             | INPUT/OUTPUT. External inductor for DC-DC converter.                                                  |
| A15                          | V33      | AO       | 100        |             | OUTPUT. 3.3 V power rail.                                                                             |
| A6                           | V14      | AO       | 20         |             | OUTPUT. 1.4 V power rail.<br>4.7 $\mu$ F decoupling capacitor required.                               |
| B3                           | V12      | AO       | 50         |             | OUTPUT. 1.2 V power rail.                                                                             |
| A8                           | VDD1V8   | AO       | 75         |             | OUTPUT. 1.8 V power rail. Supply for external devices.                                                |
| B4                           | VDD1V8P  | AO       | 75         |             | OUTPUT. 1.8 V power rail. Supply for external devices.                                                |
| B2                           | VDDIO    | AI       |            |             | INPUT. FLASH interface supply voltage (1.8 V).                                                        |
| B21                          | ESDN     | -        |            |             | Connect to ground.                                                                                    |
| die pad                      | GND      | -        |            |             | Common ground plane for radio, analog and digital circuits.                                           |
| <b>Unconnected pins</b>      |          |          |            |             |                                                                                                       |
| A1, A2,<br>A3, A4,<br>A5, B1 | NC       | -        |            |             | Internally not connected. Leave open or connect to ground.                                            |

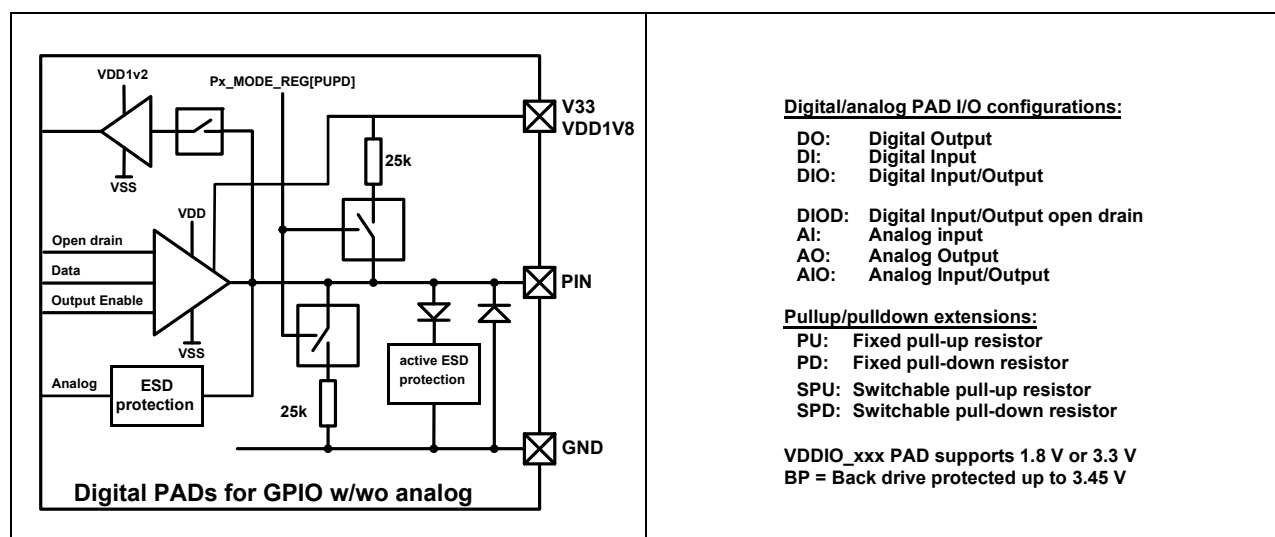
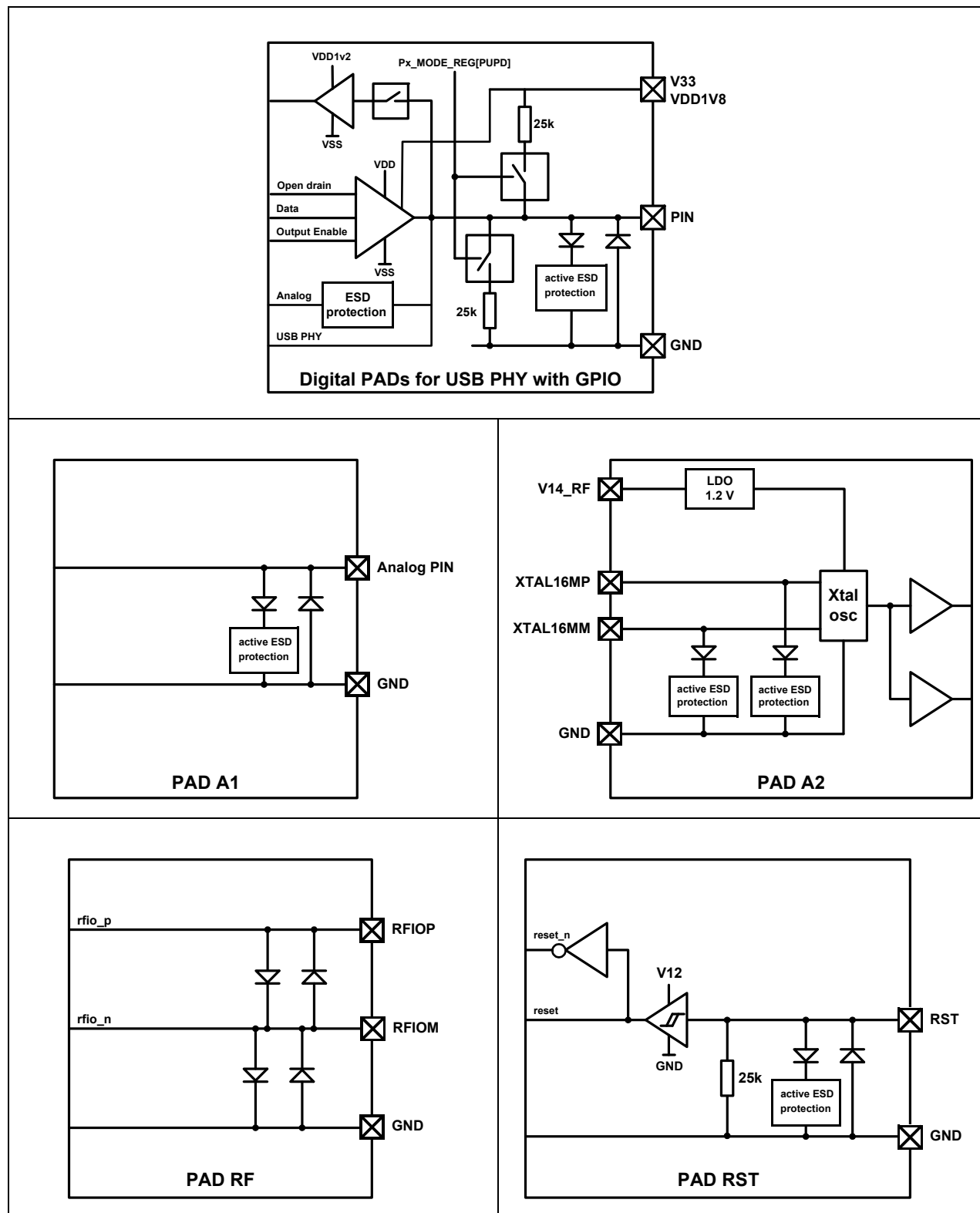
**Table 3: Pin type definitions**

Table 3: Pin type definitions





### 3 System overview

#### 3.1 INTERNAL BLOCKS

The DA14680 contains the following blocks:

**ARM Cortex™ M0 CPU** with Wake-up Interrupt Controller (WIC). This processor provides 0.9 dMIPS/MHz and is used for implementing the higher layers of the Bluetooth Low Energy protocol. It is also used for the application requirements including controlling of the power scheme of the system, reaching up to 86 dMIPs if required. It is accompanied by a powerful cache controller with configurable associativity, cache line size and RAM size.

**BLE 4.2 Core.** This is the baseband hardware accelerator for the Bluetooth Low Energy protocol.

**Co-existence** The CoEx sub-block implements a co-existence interface with external collocated modules interfering with the 2.4GHz ISM band. A three wire interface is realized to sync with the external modules about the priority and the activity of the internal radio.

**FLASH.** This is an 8 Mbit low power Quad-SPI FLASH which is used to directly execute code from (using the CPU cache) or mirror the contents into the data RAM.

**ROM.** This is a 128 kB ROM containing the Bluetooth Low Energy protocol stack as well as the boot code sequence.

**OTP.** This is a 64 kB One Time Programmable memory array, used to store the application code as well as the Bluetooth Low Energy profiles. It also contains system's configuration and calibration values.

**Data RAM.** This is a 128 kB Data RAM (DataRAM) which can be used for mirroring the program code from the OTP when the system wakes/powers up or as a normal data RAM when the system executes code directly from OTP or QSPI FLASH. It also serves as Data RAM for variables and various data that the protocol requires to be retained when system goes to sleep. It comprises 5 physical RAM cells, all with content retaining capability.

**Cache/Tag RAM.** This is a 16 kB data RAM used primarily by the cache controller (CacheRAM). It is accompanied by a Tag RAM. In mirrored mode the CacheRAM can be used as an extension of the DataRAM, increasing the available memory to 144 kB.

**Cache controller.** This is an instruction cache controller used for code execution directly from OTP or QSPI FLASH, thus reducing accesses to these memories.

**UART and UART2.** Asynchronous serial interfaces. UART2 implements hardware flow control with a FIFO of 16 bytes depth.

**SPI and SPI2.** These are the serial peripheral interfaces with master/slave capability with a 16-bit wide FIFO of 16 places.

**I2C and I2C2.** These are Master/Slave I2C interfaces used for sensors and/or host MCU communication. Each controller includes a FIFO of 4, 9-bit locations

**General purpose (GP) ADC.** This is a 10-bit analog-to-digital converter with 8 external input channels and averaging capabilities, which increase the effective number of bits (ENOB) to 12.

**Radio transceiver.** This block implements the RF part of the Bluetooth Low Energy protocol at 2.4 GHz.

**Clock generator.** This block is responsible for the clocking of the system. It contains two crystal oscillators: one running at 16 MHz (XTAL16M), which is used for the active mode of the system, and one running at 32.768 kHz (XTAL32K), which is used for the sleep modes of the system.

There are also three RC oscillators available: a 16 MHz and a 32 kHz oscillator (RC16M and RC32K) with low precision ( $> 500$  ppm) and a 11.7 kHz oscillator (RCX) with higher precision ( $< 500$  ppm).

The RCX oscillator can be used as a sleep clock replacing the XTAL32K oscillator to further improve the power dissipation, while reducing the bill of materials of the system. The RC16M oscillator is used to provide a clock for running SW already before the XTAL16M oscillator has settled after power/wake up.

Additionally, a low power, short lock time PLL can be activated to increase system's speed to 96 MHz.

**Software timers.** This block contains a 16-bit general purpose timer (Timer0) with PWM capability, a 16-bit general purpose up/down timer (Timer1) with PWM capability, which can operate at any clock even when in sleep/deep sleep mode, and a 14-bit timer (Timer2), which controls three PWM signals with respect to frequency and duty cycle. The timer block also comprises a dedicated timer implementing an LED breathing function with 256 steps granularity.

**Wake-up controller.** This is a timer for capturing external events, that can be used as a wake-up trigger based on a programmable number of external events on any of the GPIO ports, or as a GPIO triggered interrupt generator when the system is awake.

**Quadrature decoder.** This block decodes the pulse trains from a rotary encoder to provide the step size and the direction of movement of an external device. Three axes (X, Y, Z) are supported.

**Keyboard scanner.** This circuit implements scanning and debouncing of a keyboard matrix and generates an interrupt upon a configurable action without the need of CPU.

**Infrared (IR) generator.** This controller implements a very flexible, low power, microcode based scheme for IR protocols primarily used for remote controls.

**AHB/APB bus.** Implements the AMBA Lite version of the AHB and APB specifications. Two different AHB busses are used, one for the CPU and one for the DMAs of the system. APB32 is implemented for the Audio peripherals while APB16 is used for the other peripheral blocks.

**USB 1.1 FS Device.** This is a 12 Mbit/s only USB



device controller, which is mainly used for software upgrades. It is also used for recharging the system's battery.

**Cryptography blocks.** The cryptography blocks consist of a AES/HASH controller and an Elliptic Curve Controller (ECC), accelerating any application security requirements. A True Random Number Generator (TRNG) is also provided enabling secure key generation.

**DMA Engine.** This is a general purpose DMA engine with 8 channels that can be multiplexed to support data fetching between peripherals and DataRAM.

**Audio blocks.** This part enables audio streaming by means of a Pulse Density Modulation (PDM), a Sample Rate Converter (SRC) and a Pulse Code Modulation (PCM) interface. It can support 2 digital microphones or 2 digital loudspeakers using the PDM interface or connect an external CoDec at the PCM/I2S interface.

**Power management.** A sophisticated power management circuit with a Single Input Multiple Output (SIMO) Buck DC-DC converter and several LDOs that can be turned on/off via software. Extra pins are provided for supplying external devices, even when the DA14680 is in sleep/deep sleep mode. It also comprises a Constant Current/Constant Voltage (CCCV) charger for the battery charging and a state-of-charge fuel gauge circuit.

A more detailed description of each of the components of the DA14680 is presented in the following sections.

### 3.2 FUNCTIONAL MODES

The DA14680 is optimised for embedded applications such as health monitoring, sports measuring, human interaction devices, etc. Customers are able to develop and test their own applications. Upon completion of the development, the application code can be programmed into the embedded OTP or QSPI FLASH memory.

In principle, the system has two functional modes of operation:

**A. Mirrored mode.** Application, profiles etc. are all included in the OTP or QSPI FLASH. They will be mirrored at boot or wake-up time into the unified RAM, which consists of both the CacheRAM and the DataRAM cells in a single, continuous memory space. Next, the CPU starts executing from the unified RAM, which is used for code as well as data.

During Mirrored mode the cache controller is totally bypassed while all RAM cells (except for the Tag RAM) are virtually moved into a continuous memory space.

**B. Cached mode.** This mode uses the memory resources of the system as described in the block diagram. The cached area can be OTP and/or the QSPI FLASH memory space. Code is executed directly from the OTP/FLASH through the cache controller, while DataRAM is used for intermediate variables, stacks, heaps and application data.

Mirrored mode or Cached mode should be configured during initialisation of the system and not dynamically.

There are several different ways of executing code and mapping the data segment of the system. The following table provide an overview of the different possibilities for BLE product use cases.

**Table 4: Memory configurations**

| Use case | BLE stack code | Application and profile code | BLE Exchange RAM and data | Functional mode |
|----------|----------------|------------------------------|---------------------------|-----------------|
| BLE_01   | ROM            | OTP                          | DataRAM                   | Mirrored        |
| BLE_02   | ROM            | FLASH                        | DataRAM                   | Mirrored        |
| BLE_03   | ROM            | OTP                          | DataRAM                   | Cached          |
| BLE_04   | ROM            | FLASH                        | DataRAM                   | Cached          |
| BLE_05   | FLASH          | FLASH                        | DataRAM                   | Cached          |

In addition, it is also possible to use parts of the DataRAM as code segments. In that case, parts of the code can be placed in the DataRAM (a non-cacheable area), while the system is operating in cached mode.

### 3.3 SYSTEM CONFIGURATION

The DA14680 contains a 64 kB One Time Programmable (OTP) memory, which is used for storing the code (as explained in Table 4) and for retaining the system's configuration data in a special OTP space called the "OTP header".

The OTP header occupies the last 712 words (64 bits

wide) in the OTP memory space. It is partitioned into four sections that contain vital information for the system, as illustrated in the following table.

Table 5: OTP header details

| Address          | Size (B)   | Field name                              | Description                                                                                                                                                                                                                                                         |
|------------------|------------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>0x7F8E9C0</b> | <b>184</b> | <b>Chip Configuration Section (CCS)</b> |                                                                                                                                                                                                                                                                     |
| 0x7F8E9C0        | 8          | Mirrored/Cached At startup              | 0: Mirrored<br>1: Cached                                                                                                                                                                                                                                            |
| 0x7F8E9C8        | 8          | Non-Volatile Memory                     | 0: FLASH<br>Anything else: OTP                                                                                                                                                                                                                                      |
| 0x7F8E9D0        | 8          | Product Ready                           | 0x00: OTP or FLASH not programmed<br>0xAA: OTP or FLASH programmed                                                                                                                                                                                                  |
| 0x7F8E9D8        | 8          | Redundancy                              | 0: Used<br>1: Not used                                                                                                                                                                                                                                              |
| 0x7F8E9E0        | 8          | Reserved                                |                                                                                                                                                                                                                                                                     |
| 0x7F8E9E8        | 8          | Shuffle RAMs                            | Define the sequence of the RAM cells in a continuous memory space<br>0x0: DataRAM1, DataRAM2, DataRAM3<br>0x1: DataRAM2, DataRAM1, DataRAM3<br>0x2: DataRAM3, DataRAM1, DataRAM2<br>0x3: DataRAM3, DataRAM1, DataRAM1<br>DataRAM1=8KB, DataRAM2=24KB, DataRAM3=32KB |
| 0x7F8E9F0        | 8          | JTAG                                    | 0x0: Enabled<br>0x1: Disabled                                                                                                                                                                                                                                       |
| 0x7F8E9F8        | 8          | Sleep Clock                             | 0x0: XTAL32<br>0x1: RCX                                                                                                                                                                                                                                             |
| 0x7F8EA00        | 8          | Position/Package                        | B7-B4: Reserved. Keep these values to 0<br>B3:<br>0x00 – Reserved<br>0x55 – aQFN60<br>0x99 – KGD<br>B2: Wafer number<br>B1: Y coord,<br>B0: X coord.                                                                                                                |
| 0x7F8EA08        | 8          | Tester/Timestamp                        | B7: Reserved<br>B6: Tester ID (MSByte)<br>B5: Tester ID (LSByte)<br>B4: Tester Site<br>B3: TimeStamp Byte 3<br>B2: TimeStamp Byte 2<br>B1: TimeStamp Byte 1<br>B0: TimeStamp Byte 0                                                                                 |
| 0x7F8EA10        | 8          | Mirror Image Length                     | Contains the size of the image to be mirrored (unit: 32-bit words)                                                                                                                                                                                                  |
| 0x7F8EA18        | 8          | Reserved                                |                                                                                                                                                                                                                                                                     |
| 0x7F8EA20        | 8          | Chip Unique ID                          | ASCII code for “14680AE “                                                                                                                                                                                                                                           |

Table 5: OTP header details

| Address   | Size (B) | Field name                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|----------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x7F8EA28 | 8        | Cache architecture           | <p>Defines the Cache architecture to be programmed at SW reset:</p> <p>Bits[3:0] Cache Line Size</p> <p>0x0: 8 bytes</p> <p>0x1: 16 bytes</p> <p>0x2: 32 bytes</p> <p>0x3 - 0x7: RESERVED</p> <p>Bits[7:4] Associativity</p> <p>0x0: Direct Mapped</p> <p>0x1: 2-way set</p> <p>0x2: 4-way set</p> <p>0x3 - 0x7: RESERVED</p> <p>Bits[11:8] Cache Size</p> <p>0x0: RESERVED</p> <p>0x1: 8 KBytes</p> <p>0x2: 16 KBytes</p> <p>0x3-0x7: RESERVED</p> <p>Bits[15:12] RESERVED</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 0x7F8EA30 | 8        | Serial Configuration Mapping | <p>B0[7:4]: Serial signal 1, port number</p> <p>B0[3:0]: Serial signal 1, bit number</p> <p>B1[7:4]: Serial signal 2, port number</p> <p>B1[3:0]: Serial signal 2, bit number</p> <p>B2[7:4]: Serial signal 3, port number</p> <p>B2[3:0]: Serial signal 3, bit number</p> <p>B3[7:4]: Serial signal 4, port number</p> <p>B3[3:0]: Serial signal 4, bit number</p> <p>B4: Booting Method</p> <p>0xAA: booting from a specific serial port (B5) and at a specific location (B0 to B3)</p> <p>0x00: normal booting sequence</p> <p>B5: Serial Interface:</p> <p>0x0: None</p> <p>0x1: UART</p> <p>0x2: UART2</p> <p>0x3: SPI</p> <p>0x4: SPI2</p> <p>0x5: I2C</p> <p>0x6: I2C2</p> <p>B6:</p> <p>if UART/UART2 is selected:</p> <p>0x0: 115 kBaud,</p> <p>0x1: 57.6 kBaud,</p> <p>0x2: 38.4 kBaud,</p> <p>0x3: 19.2 kBaud,</p> <p>0x4: 9.6 kBaud</p> <p>SPI is not applicable since it is a slave interface</p> <p>if I2C/I2C2:</p> <p>0x0: Standard Mode (100 kbps)</p> <p>0x1: Fast Mode (400 kbps)</p> <p>B7: RESERVED</p> |
| 0x7F8EA38 | 8        | Image CRC                    | CRC16 checksum for the programmed image                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 0x7F8EA40 | 8        | Reserved                     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

Table 5: OTP header details

| Address          | Size (B)    | Field name                                      | Description                                                                                                                                                                                                                                                                                                      |
|------------------|-------------|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x7F8EA48        | 8           | QSPI Functions                                  | Bit0<br>0: Reset Function of QSPI FLASH is in BootROM<br>1: Reset Function of QSPI FLASH is in OTP<br>Bit1<br>0: Find "qQ" Function of QSPI FLASH is in BootROM<br>1: Find "qQ" Function of QSPI FLASH is in OTP<br>Bit2<br>0: QSPI loader of QSPI FLASH is in BootROM<br>1: QSPI loader of QSPI FLASH is in OTP |
| 0x7F8EA50        | 8           | UART STX timing                                 | Defines the delay for booting from UART in units of 10 ms each.                                                                                                                                                                                                                                                  |
| 0x7F8EA58        | 32          | BDADDR                                          | Bluetooth Address                                                                                                                                                                                                                                                                                                |
| <b>0x7F8EA78</b> | <b>384</b>  | <b>Trim and Calibration Section (TCS)</b>       |                                                                                                                                                                                                                                                                                                                  |
| 0x7F8EA78        | 8           | Trim and Calibration Register Address           | B7 to B5: Inverted address<br>B3 to B0: Address                                                                                                                                                                                                                                                                  |
| 0x7F8EA80        | 8           | Trim and Calibration Register Value             | B7 to B5: Inverted data value<br>B3 to B0: Data value                                                                                                                                                                                                                                                            |
| 0x7F8EA88        | 368         | Trim values                                     | Contains all trim values and calibration values in word pairs (Address, Value)                                                                                                                                                                                                                                   |
| <b>0x7F8EBF8</b> | <b>3072</b> | <b>Elliptic Curve Contents Section (ECS)</b>    |                                                                                                                                                                                                                                                                                                                  |
| 0x7F8EBF8        | 8           | ECC image length and CRC                        | B7 to B5: Inverted value of B3-B0<br>B3 to B2: Image CRC<br>B1 to B0: Image length in 32-bit words                                                                                                                                                                                                               |
| 0x7F8EC00        | 3064        | ECC microcode                                   | Contains all ECC microcode for the Curves implementation                                                                                                                                                                                                                                                         |
| <b>0x7F8F7F8</b> | <b>2048</b> | <b>QSPI FLASH Initialization Section (QFIS)</b> |                                                                                                                                                                                                                                                                                                                  |
| 0x7F8F7F8        | 8           | Address for the QSPI Reset code                 | B7-B5: Section length (Bytes)<br>B3-B0: Address                                                                                                                                                                                                                                                                  |
| 0x7F8F800        | 8           | Address for the QSPI "pP" identification code   | B7-B5: Section length (Bytes)<br>B3-B0: Address                                                                                                                                                                                                                                                                  |
| 0x7F8F808        | 8           | Address for the QSPI Loader code                | B7-B5: Section length (Bytes)<br>B3-B0: Address                                                                                                                                                                                                                                                                  |
| 0x7F8F810        | 8           | Address for the QSPI wake up uCode              | B7-B5: Section length (Bytes)<br>B3-B0: Address                                                                                                                                                                                                                                                                  |
| 0x7F8F818        | 2016        | Contains all QSPI related code segments         |                                                                                                                                                                                                                                                                                                                  |

Integrity of the data in the OTP header is guaranteed in various ways. The OTP controller has an embedded Error Correction Code hardware block, which can correct 1 bit error and detect 2 bit errors.

Furthermore, the *Chip Configuration Section* contains mostly flags with redundancy over the whole 64-bit word to ensure no mistaken value will be read. The flag value is repeated over all bytes of the word.

The *Trim and Calibration Settings Section* comprises the addresses and data values of the registers to be configured after power/wake up. The OTP contains the inverted values of both the address and the data values in the most significant 32-bit word. Reading from

OTP, checking and then storing the value into the respective register is considered to be a fast and easy task for software.

The *Elliptic Curve Contents Section* contains its own CRC-16 checksum, while the actual OTP image (not present in the header) is also optionally protected by a CRC-16 checksum (Image CRC).

The *QSPI FLASH Initialisation Section* relies on the OTP controller reports for integrity.

### 3.4 SYSTEM STARTUP PROCEDURE

After power-on or wake-up, a hardware state machine is started, which resides in the Power Management

Unit. Following this, the CPU will start executing code from address 0x0. If the system is just powered-up, then ROM resides at 0x0 hence the bootROM sequence will be triggered. If the system was just waken-up, then address 0x0 is remapped to either the Data-RAM or one of the Non-Volatile resources of the chip hence code is directly executed from there.

### 3.4.1 Power/Wakeup FSM

The hardware FSM is responsible for starting the main LDOs of the system and power the main rails used for supplying the digital and analog resources of the chip. The flow chart of this state machine is presented in [Figure 3](#). System clock after Power On Reset is released, is the 32 kHz coming from an on-chip RC oscillator (RC32K). The FSM will initially compare the voltages between pins VBUS and VBAT. This embedded PMU feature provides a digital signal for deciding which LDO to start so that Vsys is powered up. Please refer to [Figure 6](#) for an overview of the LDOs.

When VBUS is present (the system is connected to the USB for recharging or software upgrading), the PMU powers the whole system from VBUS instead of VBAT. Therefore a dedicated LDO\_USB will be started. The LDO\_USB has its own reference and will switch to the Bandgap reference voltage as soon as this has settled.

When no VBUS is detected (no USB connection), the LDO\_VBAT will be turned on, followed by the Bandgap. The LDO\_VBAT also has its own reference for starting up and will automatically switch to the Bandgap reference.

The next step is to start the LDO\_IO2 and LDO\_IO, which provides power to the external rails, mostly targeting QSPI FLASH if available. Then the LDO\_CORE is enabled to supply the VDD voltage (1.2 V) for the digital core to start operating. From that point onwards and provided that the LDOs are settled, the digital system is up and running.

Next, both 16 MHz oscillators are started. The system clock switches to the 16 MHz RC clock to start OTP mirroring or any other initialisation procedure that has to do with an QSPI FLASH.

Then the CPU can take over and either start executing code from RAM or ROM. In the case of executing code from RAM, it can switch on the SIMO DC-DC converter, disable the LDOs to lower the power consumption of the digital part and operate the radio. For a detailed overview of the PMU, please refer to [Figure 8](#) and for a representation of the timing of the power up/wake up process refer to [Figure 17](#).

The latency of the hardware FSM is not always the same. It is depending on power up or wake up and more specific, in the case of wake up, it depends on the time the system has been sleeping before.

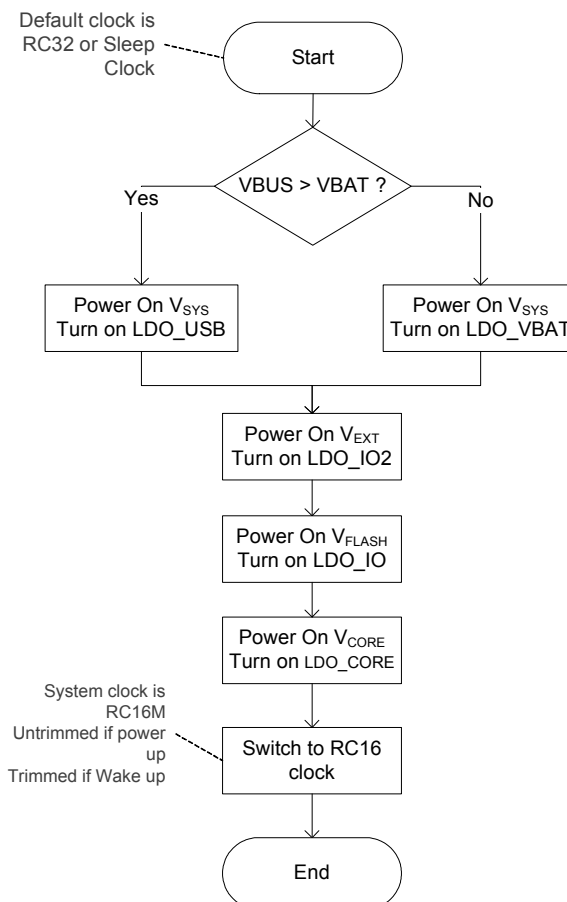


**Figure 3: Power/Wakeup FSM flow diagram**

When powered up the default clock is the RC32 which is close to 32 kHz. The time required for the completion of the HW FSM is 16 clock cycles i.e. 0.5 ms. If the system wakes up, then the sleep clock is used: either the XTAL32K (32 kHz) or the RCX (11.7 kHz). Depending on the amount of time slept, there might be some energy in the LDOs left or not, hence the settling time might be less than expected. This process might take 11 to 16 clock cycles i.e. minimum 0.35 ms, maximum 1.6 ms.

### 3.4.2 Goto Sleep FSM

After the sleep command has been issued (WFI), the system will switch to operating on RC16M and the HW FSM described in the flow chart of will take over:

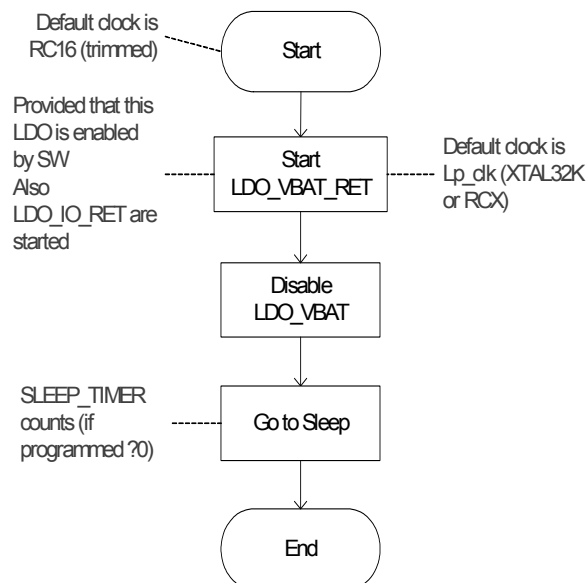


**Figure 4: Go to Sleep FSM flow diagram**

Depending on the programming of the respective control registers the FSM will switch to the `Lp_clk` and then start (or not) the `LDO_VBAT_RET` to provide power at the `Vsys` rail. `LDO_IO_RET` and `LDO_IO_RET2` will also be enabled (or not, depending on SW programming) during this state. Additionally, the `LDO_CORE` is disabled letting the `LDO_SLEEP` take over the supply of the always on logic.

Following that, the FSM will disable the `LDO_VBAT` and the `LDO_IO/LDO_IO2`.

Finally during the `Go to Sleep` state, the `SLEEP_TIMER` starts counting if there has been a value programmed in the `SLEEP_TIMER_REG` and the system goes into sleep.

A detailed timing diagram of the go to sleep procedure is illustrated at [Figure 17](#).

### 3.4.3 BootROM sequence

The BootROM sequence will be triggered right after a power-up or when the latest remapping of address 0 is pointing to the ROM.

The booting process of the DA14680 is presented in [Figure 5](#).

The BootROM code starts with the RC16 oscillator active but untrimmed, which provides an average frequency of 10 MHz in typical conditions. The BootROM code starts the watchdog timer, which will fire only after 2 seconds if not re-initialised.

Next, the OTP controller is initialised and two important configuration flags are read and evaluated: the sequence of the RAM cells and whether JTAG should be enabled or not. Especially for the latter, if it is dis-

abled, there is no way of enabling it again even by the application.

Following that, trim and calibration values are read from the OTP and stored into the respective retention registers. Note that the *TCS Section* of the OTP header (see [Table 5](#)) contains all register addresses and values that are being measured during production testing or any other values that are required to be retained. These values are all stored into their respective registers using a 'while' loop, which stops only when an empty word is found.

From that point onwards, the trimmed RC16 oscillator outputs a frequency very close to 16 MHz.

The TCS values are protected using inverted redundancy. When a voltage dropout occurs while reading or writing the value, an incorrect redundancy check will re-initiate the copy action. When copying is still unsuccessful after 5 attempts, a hardware reset will be triggered.

The "Product Ready" flag defines whether the system should follow the 'NVM' or the 'SERIAL' booting paths of the flow chart. In the NVM case, the system is supposed to start executing code from a Non-Volatile memory (NVM), which can either be the OTP or the QSPI FLASH in any of the functional modes.



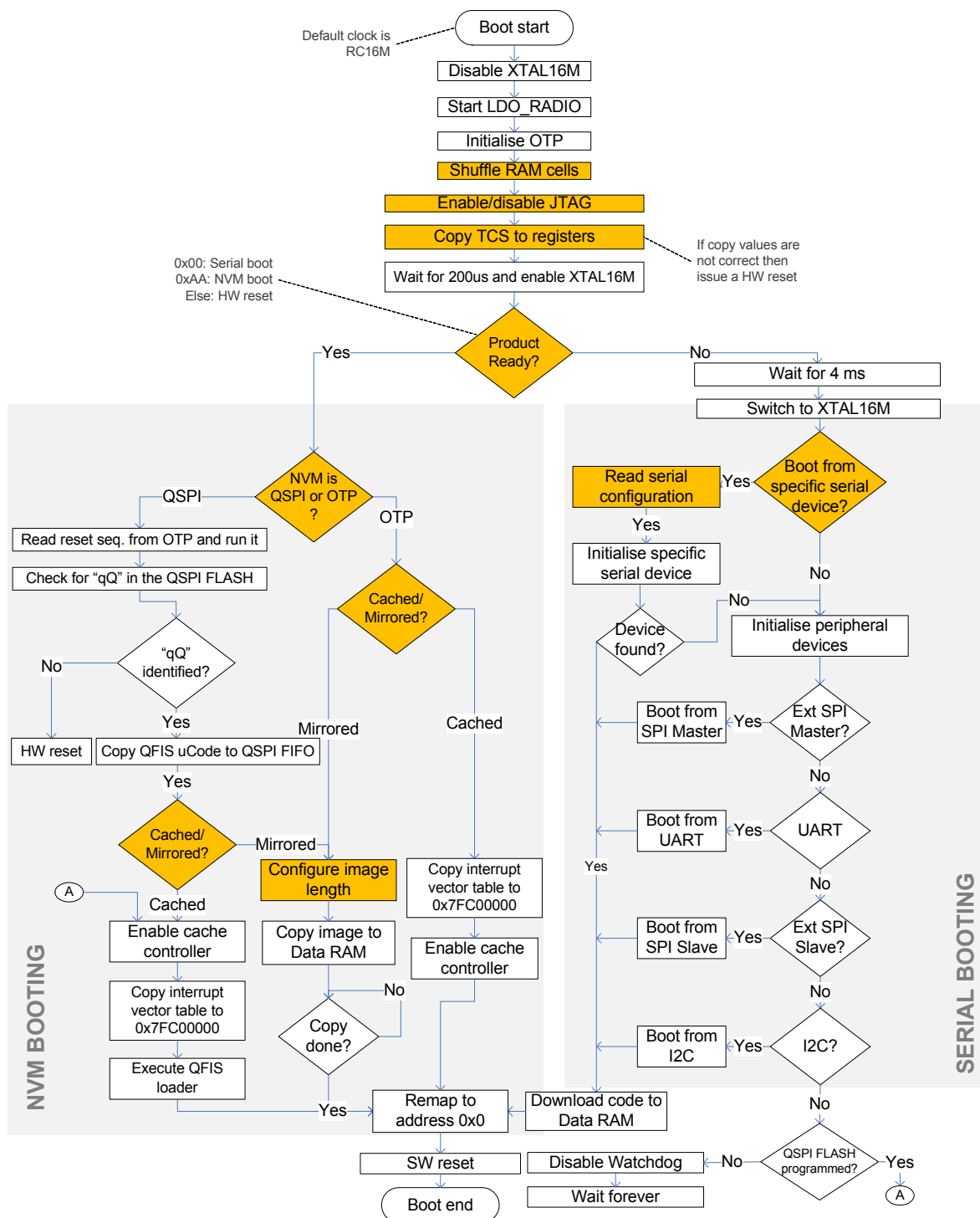


Figure 5: BootROM sequence

Colored boxes indicate OTP header access.

The NVM booting sequence is as follows:

1. If the NVM is the QSPI FLASH:

- Read the reset sequence from the OTP and apply it to the QSPI FLASH.
- Initialise the FLASH.
- Check whether there is a magic word written in

the FLASH (ASCII for "qQ").

- Download uCode for FLASH into the controller from the QFIS segment of the OTP header.
- Identify in which memory mode the system is operating (Cached or Mirrored):
- *Mirrored mode*: the Application code is copied into the DataRAM and the cache controller is bypassed, attaching the cache RAM to the DataRAM memory space.
- *Cached mode*: the cache controller is initialised as specified by the architectural parameters and the interrupt vectors are copied at the beginning of the DataRAM.

2. If the NVM is the OTP, then the same happens in Mirrored mode, while the Cached mode only requires the interrupt vector copy.
3. Remap address 0 to QSPI or OTP.

4. End the booting sequence with a software reset.  
In the case of a non-'Product Ready' device, the system clock is switched to the XTAL16M. From this point there are two options in the BootROM code:

1. **Booting from a specific serial interface.** This provides the ability to directly download code from a specific serial interface without scanning for a connected device first. This is to be used in cases where an external MCU will boot the DA14680. The configuration of the serial interface in terms of pin location, controller and speed is to be found in the OTP header as explained in [Table 5](#).
2. **Booting from any connected device** by scanning a predefined number of GPIOs and interfaces. When the respective flag is not set in the OTP header, booting from any serial interface will occur. This provides the flexibility to initially boot from a UART or an SPI at a totally blank device to start development of applications. All serial interfaces will be exercised once using the protocols described in *AN-B-046*. When no connection has been established, then a final attempt is performed for identifying a valid QSPI FLASH and if that is also unsuccessful, the system gets into a while forever loop.

The sequence of the steps that the booter takes while scanning for an external device is presented in the following table:

**Table 6: Scanning steps for booting from serial**

| Step | Boot from (Note 1)                                                                    | Speed             |
|------|---------------------------------------------------------------------------------------|-------------------|
| 0    | <b>SPI Master</b><br>MISO => P0_1<br>MOSI => P0_2<br>SPI_CLK => P0_3<br>SPICS => P0_4 | Defined by Master |

**Table 6: Scanning steps for booting from serial**

| Step | Boot from (Note 1)                                                                   | Speed      |
|------|--------------------------------------------------------------------------------------|------------|
| 1    | <b>UART</b><br>UTX => P0_1<br>URX => P0_2                                            | 115.2 kbps |
| 2    | <b>UART</b><br>UTX => P0_5<br>URX => P0_3                                            | 57.6 kbps  |
| 3    | <b>UART</b><br>UTX => P1_0<br>URX => P1_5                                            | 57.6 kbps  |
| 4    | <b>UART</b><br>UTX => P1_2<br>URX => P1_4                                            | 57.6 kbps  |
| 5    | <b>UART</b><br>UTX => P1_3<br>URX => P2_3                                            | 57.6 kbps  |
| 6    | <b>SPI Slave</b><br>MISO => P0_2<br>MOSI => P0_1<br>SPI_CLK => P0_0<br>SPICS => P0_5 | 2 MHz      |
| 7    | <b>I2C</b><br>SCL => P0_1<br>SDA => P0_2                                             | 400 kbps   |
| 8    | <b>I2C</b><br>SCL => P0_5<br>SDA => P0_3                                             | 400 kbps   |
| 9    | <b>I2C</b><br>SCL => P1_0<br>SDA => P1_5                                             | 100 kbps   |
| 10   | <b>I2C</b><br>SCL => P1_3<br>SDA => P2_3                                             | 100 kbps   |

**Note 1:** Subject to the availability of ports Px\_y.

### 3.5 POWER CONTROL AND MODES

#### 3.5.1 System Power Control

The PMU supports operation from coin-cell, 2x AAA and rechargeable Lithium-Ion batteries. An overview diagram is shown in [Figure 6](#).

There are three main supply (input) pins: VBUS, VBAT1 and VBAT2. VBUS is only used in case a USB supply is connected to the device. In all other cases, VBAT1 and VBAT2 are used, the first being the supply of the LDOs and the second the DC-DC converter supply. VBAT1 and VBAT2 should be shorted together.

From VBAT1 or VBUS, the system supply (Vsys) is generated by either LDO\_USB or LDO\_VBAT. Vsys is used to power an accurate bandgap reference, the internal sleep oscillator (RCX) and the I/O pins. In case the DC-DC converter is not activated, Vsys is also used to generate the Vcore for the digital domain and the I/O supply.



The Radio, ADC, PLL and the Xtal16M oscillator can be supplied by either the 1.4 V DC-DC converter output or by the LDO\_Radio.

The Single Inductor Multiple Outputs (SIMO) DC-DC converter has four dedicated outputs and has an average efficiency of 82% when activated. Two of its outputs (VDD1V8 and VDD1V8P) deliver power to external devices, even when the system is in sleep mode, by using the LDO\_RET\_IOx. When the VBAT1 voltage is too low to achieve a correct conversion, the LDO\_IOx will still keep the outputs powered.

In addition to the main supplies described above, the PMU has several features to support ultra low power sleep modes, where large parts of the system are turned off. There is a dedicated supply domain that is

always active and is generated either from VBUS or VBAT. This power domain is only used for the retention circuits. The LDO\_SLEEP generates the Vcore when all the other power supplies are off. This supply is used for the digital wake-up state machine and retention during sleep mode. A low voltage 32 kHz oscillator (RC32) is used as a clock for the digital state machine.

External devices might be powered by the VDD1V8, VDD1V8P or even V33 pins, exploiting the DC-DC converter efficiency and further optimising the system's power dissipation. Embedded FLASH is connected to the VDD1V8 supply, while a number of sensors can be connected to VDD1V8P (1.8 V) or V33 (3.3 V).

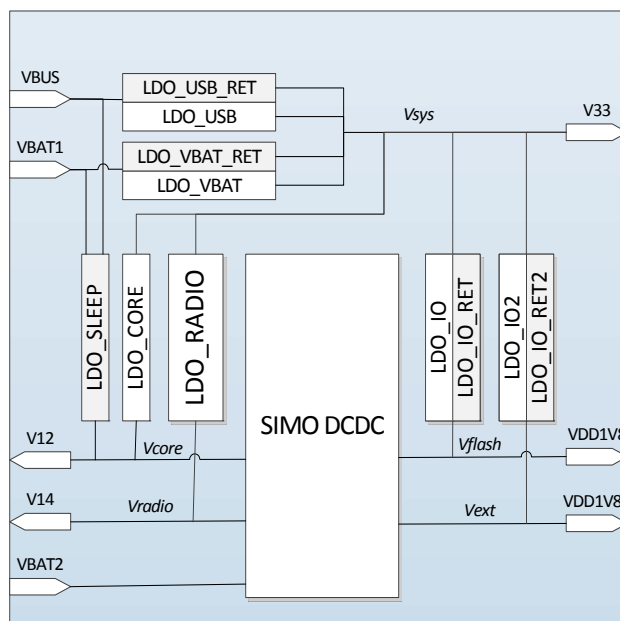


Figure 6: Power Management Unit overview

### 3.5.2 Power domains

The DA14680 comprises several different power domains, that are controlled by power switching elements, thus eliminating leakage currents by totally powering them down.

The partitioning of the DA14680's resources with respect to the various power domains is presented in the following table.

Table 7: Power domains

| Power domain | Description                                                                                                                                                                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PD_AON       | <b>Always On.</b> This power line connects to all resources that must be powered constantly: the ARM/WIC, the BLE Timer, Timer1, the Retention SRAM, the PMU/CRG, the Wake-up controller, the pad ring and various registers required for the Wake-Up sequence.                                                  |
| PD_SYS       | <b>System.</b> This power line connects to all resources that should be powered only when the ARM Cortex M0 is running: the AMBA bus, the OTP cell and controller, the QSPI controller, the ROM, the DataRAM, the Watchdog timer, the SW timers, the crypto controllers, the USB and the GPIO port multiplexing. |

Table 7: Power domains

| Power domain | Description                                                                                                                                                                                                                                                                |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PD_PER       | <b>Peripherals.</b> This power line connects to the peripherals that can be switched off after completing their operation: the UARTs, the SPI, the I2C the Keyboard scanner, the ADC etc.                                                                                  |
| PD_RAD       | <b>Radio.</b> This is the power island that contains the digital part of the Radio: the Modulator/Demodulator, the RF control unit and register file. The power management of the analog Radio subsystems is done within the Radio itself, since it contains several LDOs. |
| PD_BLE       | <b>BLE.</b> This is a separate power island that only contains the Bluetooth Low Energy Lower Mac hardware block.                                                                                                                                                          |

An illustration of the power domains on the chip block diagram is presented in the following figure.

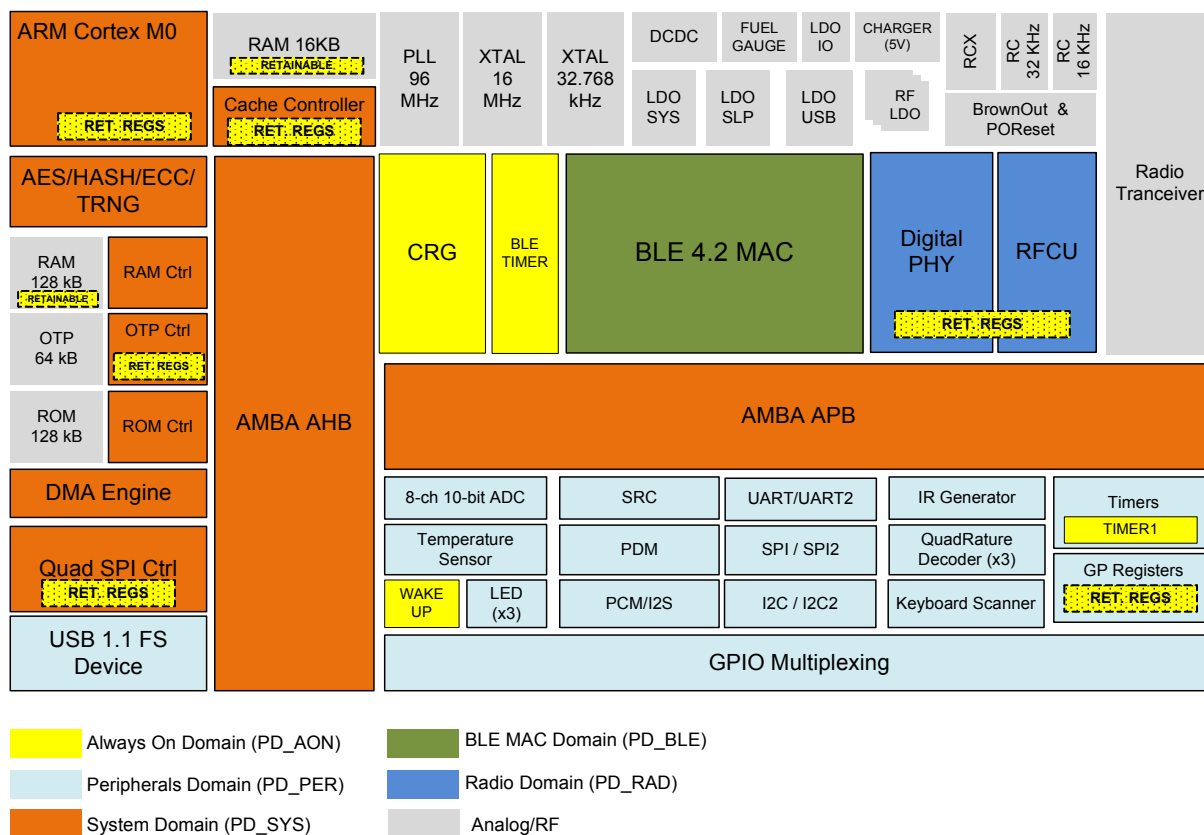


Figure 7: DA14680 digital power domains

There are specific blocks containing retention registers. These are register that keep their contents even when their power domain has been switched off.

Moreover, all DataRAM and CacheRAM blocks have their own retaining mechanism, i.e. they can be programmed to retain their content independently of the status of the power domains.

### 3.5.3 Power modes

The DA14680 has four main power modes, which are distinguished by the power domains and clocks that are active:

1. Active mode
2. Extended Sleep mode
3. Deep Sleep mode
4. Hibernation mode

However, there are several different configurations for each power mode, depending on the amount of RAM being retained and whether the DC-DC converter is kept powered on or not.

The different configurations are presented in [Table 8](#).

Table 8: Power modes and configurations

| Power mode     | Power domains                                                                 | Analog blocks                                                                                                                                                                                    | Clocks available                                                                                  | Retained memory                                                                                                               | Wake-up mechanism                                                                                                                                                                                                                                                                    | Description                                                                                                                                                                                                          |
|----------------|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active         | AON_PD=ON<br>SYS_PD=ON<br>BLE_PD=OPT <sup>1</sup><br>RAD_PD=OPT<br>PER_PD=OPT | LDO_USB = OFF<br>LDO_RET=OFF<br>LDO_VBAT=ON<br>LDO_IO=OPT<br>BandGap=ON<br>LDO_CORE= OPT<br>SIMO= OPT<br>LDO_RAD=OPT<br>LDO_VBAT_RET=OPT<br>LDO_VBUS_RET=OFF<br>LDO_IO_RETx=OPT<br>LDO_SLEEP=OFF | XTAL16M<br>RC16<br>XTAL32K (optional)<br>RCX<br>RC32<br><br>Radio activity requires XTAL16M clock | All memories are powered up and accessible                                                                                    | No wake up required. System is up and running                                                                                                                                                                                                                                        | During this mode, RAM (mirrored memory) is cached mode). The Radio, the BLE domains can operate according to the anchor point. If the system is idle, it can be turned off except for the WFI command.               |
| Extended Sleep | AON_PD=ON<br>SYS_PD=OFF<br>BLE_PD=OPT<br>RAD_PD=OPT<br>PER_PD=OPT             | LDO_USB=OFF<br>LDO_RET= ON<br>LDO_VBAT=OFF<br>LDO_IO=OPT<br>BandGap=OFF<br>LDO_CORE=OFF<br>SIMO=OFF<br>LDO_RAD= OPT<br>LDO_VBAT_RET=ON<br>LDO_VBUS_RET=OFF<br>LDO_IO_RETx=OPT<br>LDO_SLEEP=ON    | Low power clock: XTAL32K (optional)<br>RCX<br><br>Either of the above                             | 8KB to 128KB<br><br>Granularity is:<br>8KB<br>24KB<br>32KB<br><br>16KB Cache will be automatically retained if in cached mode | <b>Synchronously</b> to the BLE anchor points using the Low power clock (XTAL or RCX) BLE timer<br><b>Asynchronously</b> to the BLE anchor points from an external device powered from battery via any GPIO<br><b>Asynchronously</b> to the BLE anchor points from Timer1 interrupt. | The Extended Sleep mode is used for low power operation to the protocol. In case of Mirror mode, the code then OTP or wake-up. In case of code is required, In case of external system, there are externally or from |
| Deep Sleep     | AON_PD=ON<br>SYS_PD=OFF<br>BLE_PD=OPT<br>RAD_PD=OPT<br>PER_PD=OPT             | As above                                                                                                                                                                                         | RC32.<br>Before switching into Deep Sleep mode, <b>RC32 has to be active.</b>                     | As above                                                                                                                      | <b>Asynchronously</b> to the BLE using any GPIO. The toggling activity on any of the selected pins enables RC32. <sup>2</sup>                                                                                                                                                        | The Deep Sleep mode is used for inactivity for the system. The RAM can still happen from an external                                                                                                                 |
| Hibernation    | AON_PD=ON<br>SYS_PD=OFF<br>BLE_PD=OPT<br>RAD_PD=OPT<br>PER_PD=OPT             | As above                                                                                                                                                                                         | RC32.<br>Before switching into Deep Sleep mode, <b>RC32 has to be active.</b>                     | None                                                                                                                          | <b>Asynchronously</b> to the BLE using any GPIO. The toggling activity on any of the selected pins enables RC32.                                                                                                                                                                     | The Hibernation mode is used for inactivity for the system product shipping. There is no RAM                                                                                                                         |

1 : OPT means optionally on or off, configured by software

2 : The wake up GPIO must be configured at WKUP\_SELECT\_Px\_REG before going to sleep

## 4 Power management

The DA14680 has a complete integrated power management unit (PMU) which comprises a Single Inductance Multiple Output (SIMO) DC-DC converter with 4 outputs, various LDOs for the different power domains of the system, a Constant Current Constant Voltage (CCCV) charger for battery recharging, a charge detection circuit and a fuel gauge revealing the remaining battery charge. The PMU is capable of supplying external devices even when the DA14680 is in sleep mode.

The system diagram of the analog Power Management Unit (PMU) is presented in Figure 8.

### Features

- Synchronous Single Inductance Multiple Output Buck DC-DC converter with 4 output power rails
- Programmable DC-DC converter output charging sequence

- Two DC-DC converter outputs at 1.8 V with 75 mA load capability for powering external devices
- One LDO output up to 3.6 V with up to 110 mA load capability
- DC-DC converter on/off control per output
- Active and Sleep mode current limited LDOs
- Use of small external components
- Supply of external rails (V33, VDD1V8, VDD1V8P) while DA14680 is in Sleep mode
- Fuel gauge to indicate state-of-charge
- CCCV charger with battery/die-temperature protection
- Interrupt line for the DCDC converter

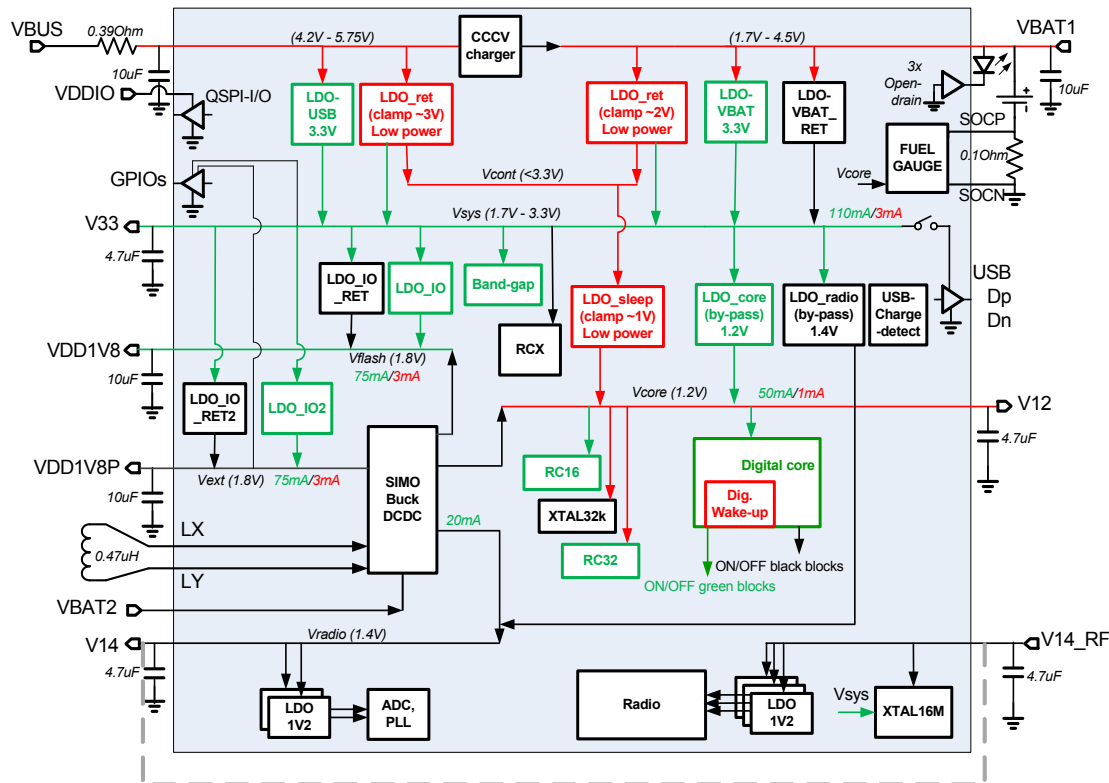


Figure 8: Power management unit block diagram

**Note 1:** Quality of the V14\_RF de-coupling capacitor is affecting the radio performance

**Note 2:** Proposed de-coupling capacitor values provide best system performance

**Note 3:** All de-coupling capacitors should be placed as close to the respective pin as possible

### 4.1 ARCHITECTURE

There are 3 main power inputs, namely VBUS, VBAT1

and VBAT2. The VBUS is connected when charging the battery through the USB connector. VBAT1 should be shorted with VBAT2 externally and supplies the LDOs, while VBAT2 supplies the SIMO DCDC converter. There are certain parts of the PMU which are always powered. They are designated in red in Figure 8. The always on power circuitry consists of 2 clamps and LDO\_SLEEP which provides the necessary voltage when the system is in extended sleep, deep sleep or hibernation mode. When the system wakes up, then

many of the blocks of the PMU are activated automatically (in green) as explained in the power/wake up sequence flow charts. Finally, SW is responsible for activating the SIMO DCDC and any other block which is in black in Figure 8.

The current that the PMU can deliver on each rail in Active/Sleep mode is presented in the following table. Note that V14 and V12 are used for the radio and the core respectively. Therefore it is recommended that

they are not used to supply any external devices. Especially V14 should be connected via the PCB to the V14\_RF as displayed in Figure 8.

$I_{DIG}$  and  $I_{RAD}$  define the currents dissipated by the digital core and the radio, respectively, while  $I_{V18\_ACT}$  represents the current load on both 1.8 V power rails.

**Table 9: PMU current supply capabilities**

| Parameter            | Description                                                    | Conditions                            | Maximum value                                                                           | Unit |
|----------------------|----------------------------------------------------------------|---------------------------------------|-----------------------------------------------------------------------------------------|------|
| $I_{V33\_ACT\_LDO}$  | Current drawn from pin by external devices when in Active mode | Power from LDO_VBAT                   | $100 - I_{V18\_ACT} - I_{DIG} - I_{RAD}$<br>$I_{V18\_ACT} = I_{LDO\_IO} + I_{LDO\_IO2}$ | mA   |
| $I_{V33\_SLEEP}$     | Current drawn from pin by external devices when in Sleep mode  | Power from LDO_VBAT_RET               | $2 - I_{V33}$                                                                           | mA   |
| $I_{V18\_ACT\_LDO}$  | Current drawn from pin by external devices when in Active mode | Power from LDO_IO/LDO_IO2             | $100 - I_{V33\_ACT} - I_{DIG} - I_{RAD} - I_{V18\_ACT}$ . Max 75 mA                     | mA   |
| $I_{V18\_ACT\_SIMO}$ | Current drawn from pin by external devices when in Active mode | Power from DCDC                       | 75                                                                                      | mA   |
| $I_{V18\_SLEEP}$     | Current drawn from pin by external devices when in Sleep mode  | Power from LDO_IO_RET/<br>LDO_IO_RET2 | For each LDO: $2 - I_{V33\_SLP}$                                                        | mA   |

#### 4.1.1 SIMO DC-DC converter

The heart of the PMU is the SIMO Buck DC-DC converter. The block diagram is displayed in the following figure:

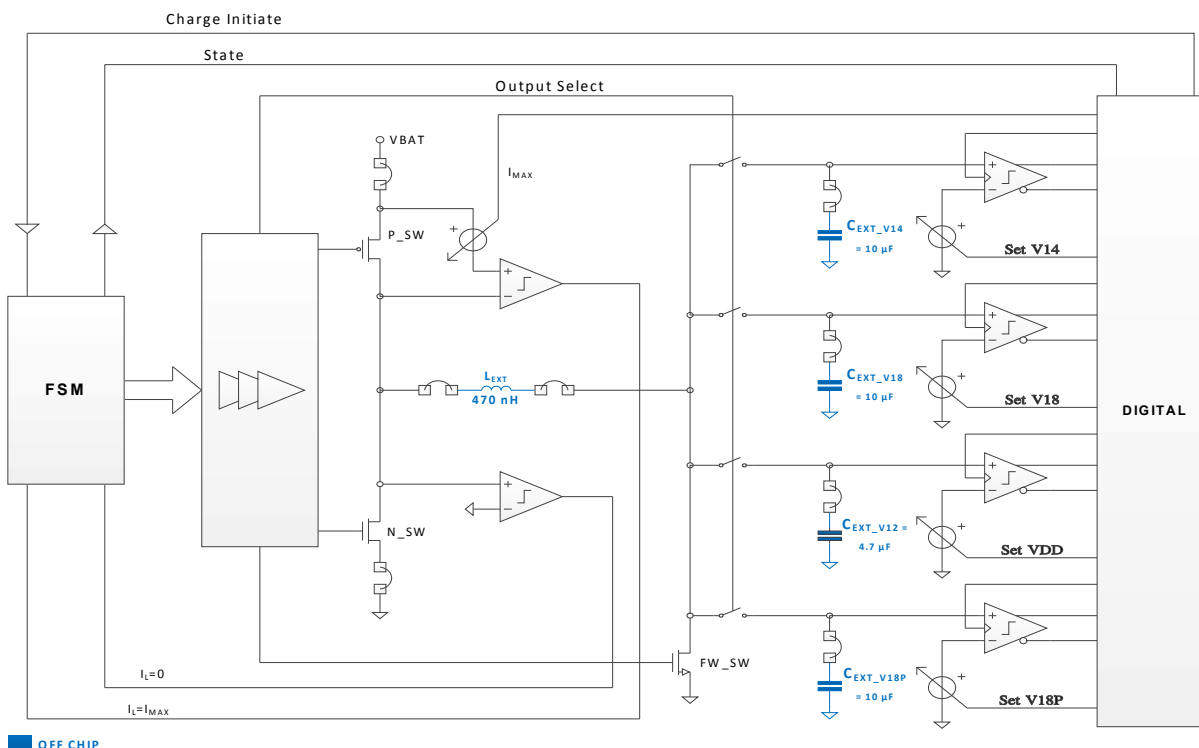


**Figure 9: SIMO DCDC Converter block diagram**

the DCDC converter comprises 4 outputs:

- **Vext** which connects to the VDD1V8P pin and delivers 75 mA when DA14680 is in active. The voltage range of this power rail is 1.8 V +/- 5%. External devices, such as Power Amplifiers (PA), Front End Modules (FEMs) or sensors can be powered by this rail.
- **Vflash** which is used for supplying external devices, such as QSPI FLASH or sensors. This rail's characteristics are identical to the Vext one.
- **Vradio** which powers the RF circuits via a number of dedicated LDOs. This rail delivers up to 20 mA at 1.4 V and should not be used for supplying external devices.
- **Vcore** which supplies the digital core of the DA14680 and delivers up to 50 mA at 1.2 V when in Active mode. This rail should not be used for supplying external devices.

The converter has an asynchronous architecture, i.e. the on-time of the switches is not determined by an external clock. Instead, the on-time is determined by a (dynamically varying) current limit in the external inductor. If one of the output voltages is too low (determined using clocked comparators) a charge cycle is triggered.

However, it is possible for more than one outputs to be below minimum at the same time, in which case the system has to decide which output to charge first. This

is done using a priority select register, which holds the sequence in which the outputs will be charged. Each time one or more outputs require a charge cycle, the system sorts the outputs based on this priority register and loads this sequence in a four tab shift register.

In order to minimize the ripple voltage on the outputs, the current limit is dynamically set during operation in Active mode. This is done by measuring how long each output is above its minimum value after a charge cycle. When this time is very long, more charge than required (given the load current) was stored on the output capacitor, so the current limit is reduced by one bit (LSB). However, when the output voltage drops too quickly, not enough charge was delivered to the output capacitor, so the current limit is increased by one bit (LSB).

The efficiency of the DCDC converter on the VDD1V8 or VDD1V8P is illustrated in the following figure. It is measured at three battery voltages (VBAT). The load on V14 and V12 is kept constant at 2mA and 5mA respectively.

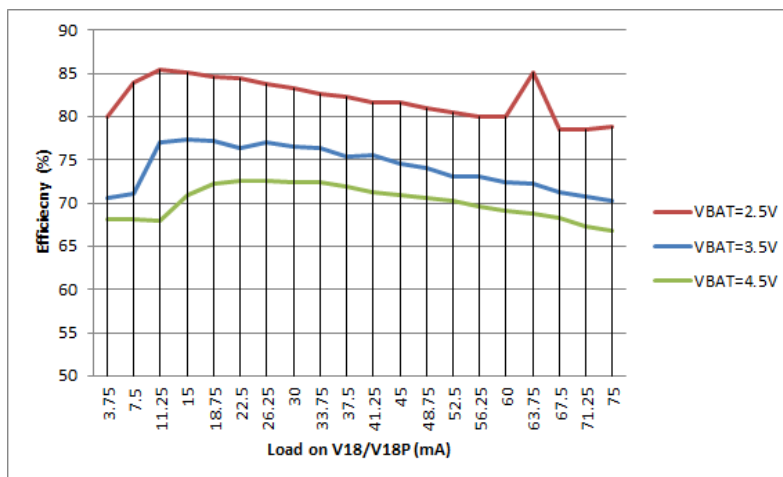


Figure 10: 1.8V rails DCDC efficiency vs load

Efficiency for sub-mA loads, i.e. between 100  $\mu$ A and 1 mA is over 70% for VBAT > 3.5 V and over 80% for VBAT < 3.5 V.

the load as well as different VBAT voltages, is presented in the following figures:

The efficiency of the V12 and V14 rails with respect to

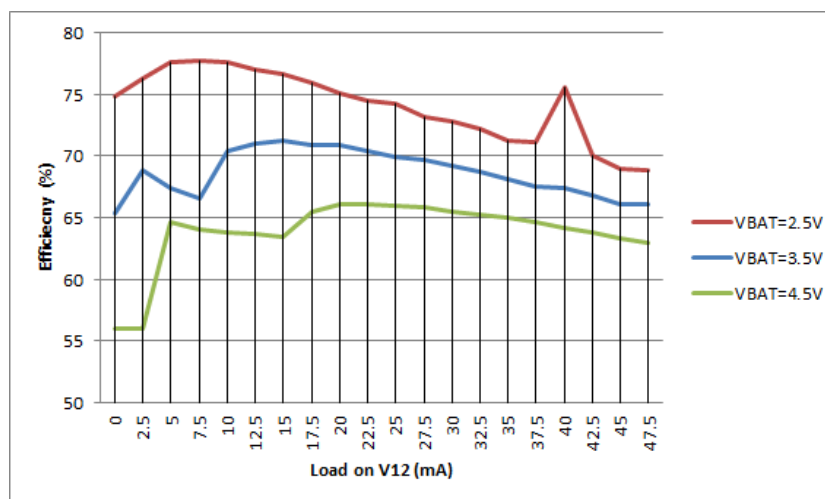


Figure 11: 1.2V rail DCDC efficiency vs load



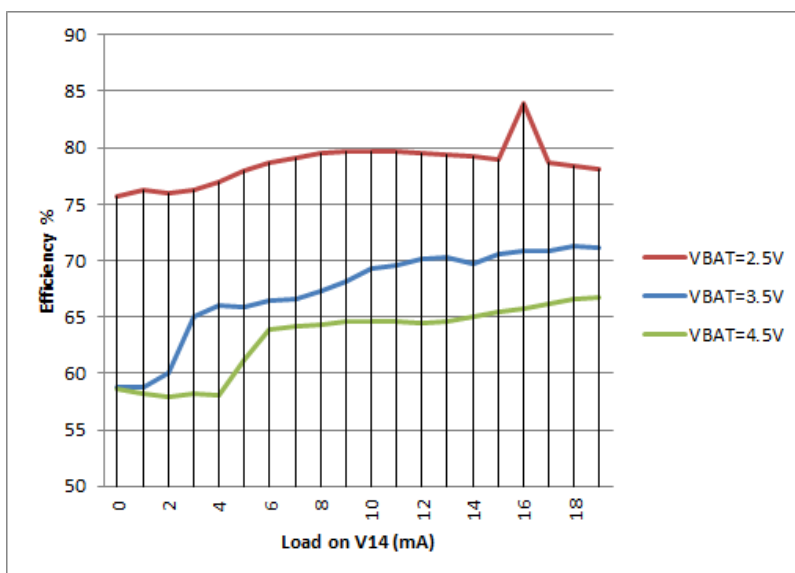


**Figure 12: 1.4V rail DCDC efficiency vs load**

1.2V and 1.4V efficiency is measured with no load on the 1.8V.

#### 4.1.2 LDOs

Several LDOs are used to provide a stable power supply to all rails, when the SIMO DCDC is not active (e.g. in Sleep mode or during start up) or when the device is plugged onto a USB charger. Furthermore, bypassing the DCDC is also considered, when the external voltage on pin VBAT2 is at the edge of enabling an efficient step-down activity (i.e. < 2.3V).

Two low power LDOs (LDO\_ret) one connected to VBUS and one to VBAT, provide power to the Vcont power line and hence to the LDO\_sleep (which is a clamp actually). This LDO is responsible for providing the VDD supply during Sleep mode, which can be trimmed down to 0.85 V. This is basically the supply of the Always On power domain (PD\_AON) which is always on, independently of active or any sleep mode.

In Sleep modes the retention LDOs might take over and make sure that the system is properly powered without the need of the DC-DC converter. The LDO\_VBAT\_RET provides power to the System supply line the LDO\_SLEEP at Vcore, and LDO\_IO\_RET/LDO\_IO\_RET2 to the external 1.8 V power rails. There is no need to power the Vradio since it is not enabled in any of the sleep mode. The LDO\_VBAT\_RET and LDO\_IO\_RETx circuits are identical and operate in a sample & hold manner: They contain a reference voltage capacitance which is used to regulate the output voltage. However, due to leakage, this internal reference capacitor is discharged. To keep a stable voltage reference, a mechanism is built to start the Bandgap, sample the voltage reference in the LDOs and shut it

down again. This periodic operation is programmable in terms of timing with use of the SLEEP\_TIMING\_REG which counts sleep clock ticks.

In active mode, when external supply is between 1.7 V and 2.4 V and the DC-DC converter is bypassed (step-down conversion not feasible due to low voltage), the LDO\_VBAT provides power to the Vsys line and the LDO\_IO/LDO\_IO2, LDO\_Core and LDO\_radio to the 1.8 V rails, the Vcore and the Vradio, respectively.

Finally, when the system is connected to a USB charger, pin VBUS is the source of the power instead of pin VBAT1/VBAT2. The same path is used as with VBAT2, but the LDO\_USB is responsible for providing the System supply line with power. This LDO is automatically switched on as soon as a VBUS > VBAT1 voltage is sensed.

##### 4.1.2.1 LDOs Loadstep Response

This section summarizes the loadstep response of the important LDOs of the PMU:

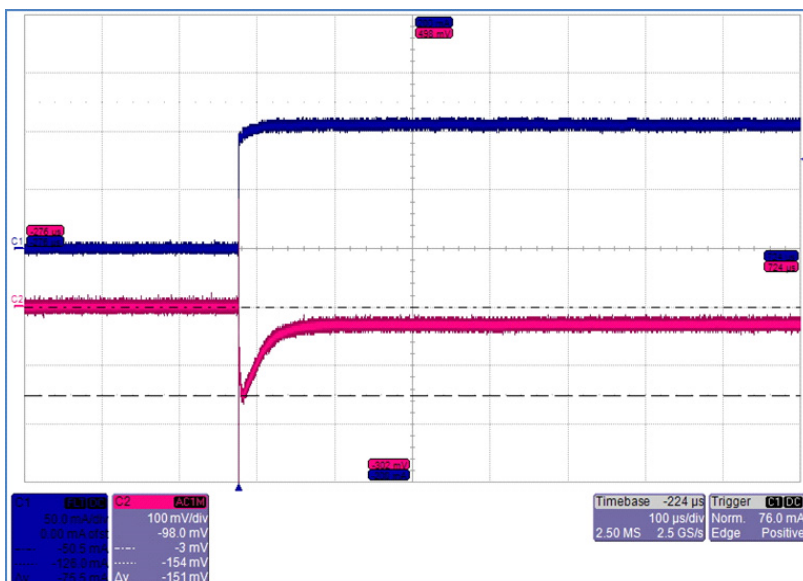


Figure 13: LDO\_VBAT loadstep 0 to 100mA (Cload=4.7uF, Vout=3.3V)

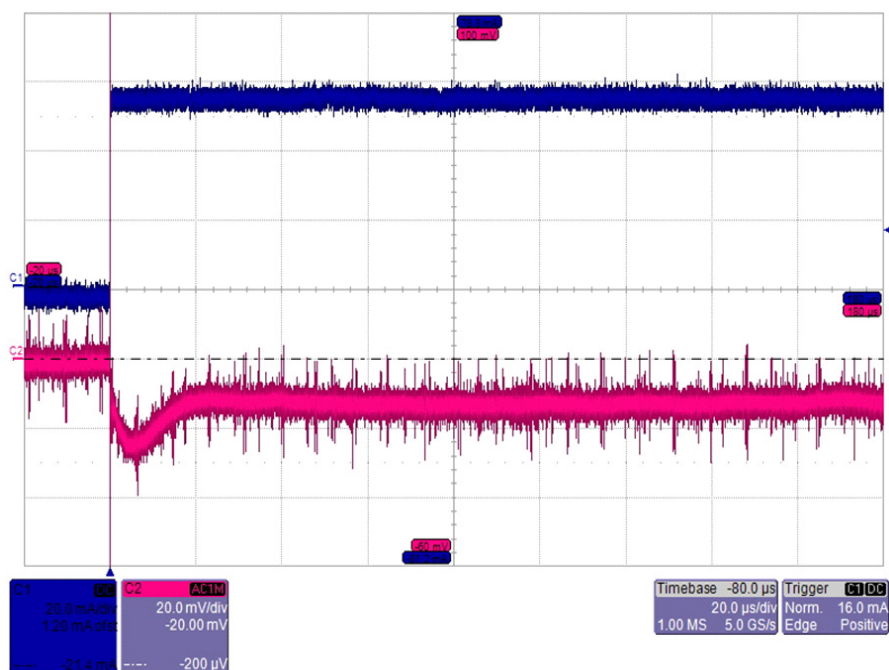


Figure 14: LDO\_IO loadstep 0 to 60mA (Cload=10uF)

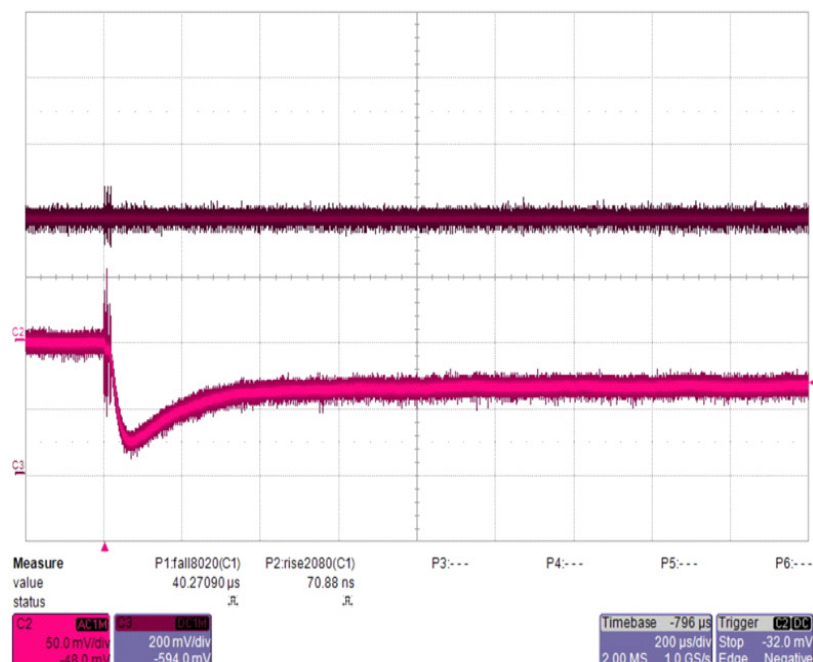


Figure 15: LDO\_VBAT\_RET loadstep 0 to 5mA (Cload=4.7 $\mu$ F, Vout=2.4V)

#### 4.1.3 Switching from DC-DC to LDOs

The DCDC converter can supply the maximum load current for all four outputs down to a battery voltage of approximately 2.4 V. When the voltage drops below this, the 1.8 V outputs can no longer be supplied by the DCDC and current will have to be provided by LDO's. There are two methods for switching from DCDC to LDO's, one based on hardware and one based on software. Both modes use a special brown out, which signals that the battery voltage has dropped below 2.45 V (+/- 50 mV) as a detection mechanism. This brown out has to be enabled by setting BOD\_CTRL2\_REG[BOD\_VBAT\_EN]=1. In hardware mode, the DCDC controller will automatically disable the 1.8 V outputs when this brownout occurs and it will set the maximum current limits for the other outputs to a lower value (which is programmable). It will assert an internal signal that indicates that the converter is working in Low Voltage mode to signal that the 1.8 V LDO's have to be enabled. This mode can only be disabled by software in order to avoid the system oscillating between High and Low Voltage modes when the battery voltage is close to the brownout threshold.

The automatic Low Voltage mode can be fully disabled by DCDC\_CTRL\_0\_REG[DCDC\_BROWNOUT\_LV\_MODE]=0. This also forces the converter back into High Voltage mode once it is operating in Low Voltage mode so when the battery voltage is once again sufficient the software needs to clear and then set the aforementioned register field to switch back to High Voltage

mode and re-enable automatic switching over to Low Voltage mode.

If automatic switching is disabled, software will have to handle the appropriate settings for a low battery. In order to alert the software to such a situation the DCDC controller can generate an interrupt upon a brownout event. The software can then handle all settings and it can choose to either clear and/or mask the interrupt (using the interrupt clear and interrupt mask registers respectively). Note that the interrupt and automatic Low Voltage mode settings are independent, so it is possible to generate interrupts when automatic Low Voltage mode is enabled.

The respective LDOs should be switched on before disabling the DCDC converter (i.e. just before entering a sleep mode). Each power rail (except Vcont and Vsys) is covered by both a DCDC output and an LDO. To switch off the DCDC, the user must enable the LDO\_IO\_RETx (if external components need to stay powered). If Vradio is required on, then LDO\_RADIO should be activated as well.

#### 4.1.4 PMU configurations in Sleep modes

Every power line can be supplied by either a DC-DC output or an LDO. Only Vcont and Vsys is not powered by a DC-DC output. There is quite a number of different ways of configuring the PMU to supply the internal rails as well as external devices, while in sleep mode, depending on the load, the sleep time etc. For supplying the Vsys rail a stable voltage level is needed hence the LDO\_VBAT\_RET has to be used, a sample-and-

hold type of LDO which samples the bandgap voltage on a regular basis and regulates accordingly. The regular wake-up is based on the SLEEP\_TIMER\_REG value.

The recommended power configuration is presented in the following table:

**Table 10: Sleep PMU configuration**

| Power Rail | Power Configuration          |
|------------|------------------------------|
| Vcont      | always on powered by LDO_RET |
| Vsys       | LDO_VBAT_RET                 |
| Vflash     | LDO_IO_RET                   |
| Vext       | LDO_IO_RET2                  |
| Vcore      | LDO_SLEEP                    |
| Vradio     | OFF                          |

There are many ways of configuring the PMU, how-

ever, there is one recommended configuration that covers all cases.

Vsys is powered by LDO\_VBAT\_RET, which is a sample and hold LDO, thus guaranteeing a stable voltage on this rail. This enables connecting external components on the V33 pin of the system. SLEEP\_TIMER\_REG has to be programmed with a value to trigger the period wakeup of the Bandgap. This covers systems with external components on 1.8V and/or 3.3V rails kept alive while the DA14680 is in Extended Sleep mode.

#### 4.1.5 Wake/Power up - Sleep Timing

There are two HW controlled FSMs which run at power/wake up and when going to sleep as explained in [section 3.4.1](#) and [section 3.4.2](#) respectively. Their timing regarding the enabling/disabling of the various resources of the PMU is explained in the following figures:

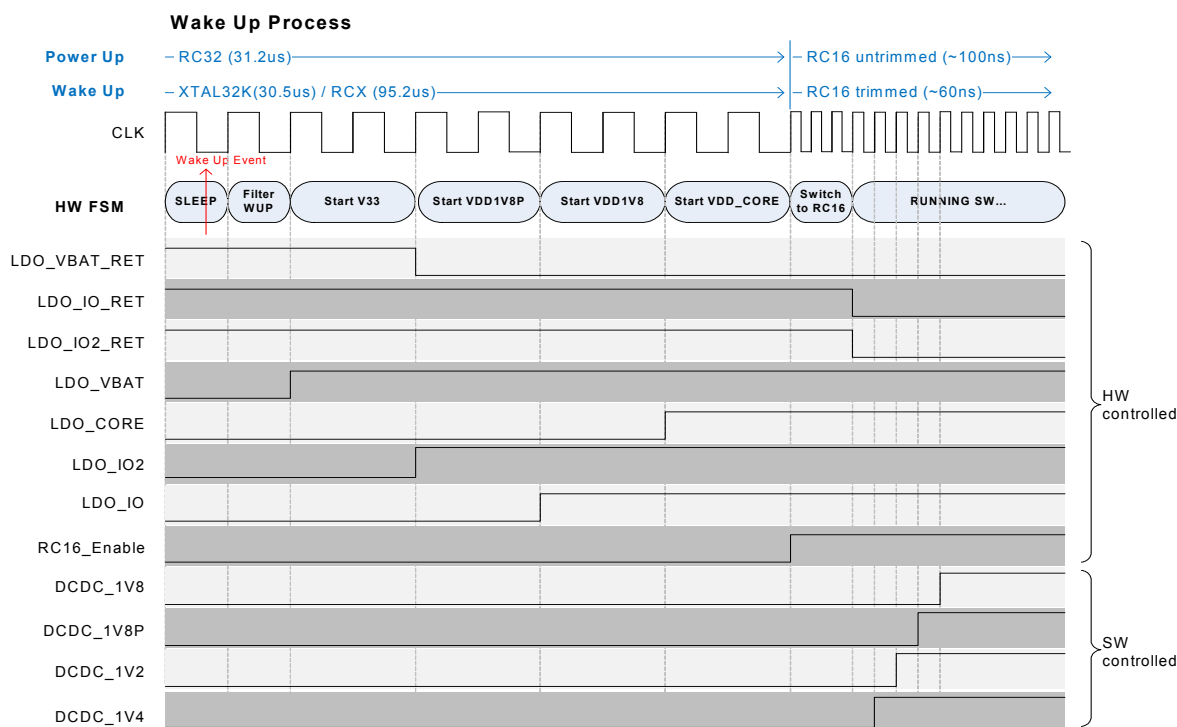


**Figure 16: Wake/Power-Up timing and PMU operations**

In the case of powering up the system, the HW FSM as depicted in [Figure 3](#), enables the various LDOs while being clocked by the RC32K. Each LDO requires a certain amount of time to settle depending on the load of the power rail. The LDOs are ready within 12 clock cycles (or 380 us) and then the clock is switched to the RC16 which has not yet been trimmed, hence the frequency is near 10 MHz.

In the case of waking up from any of the sleep modes,

the HW FSM operates at either the RCX or the XTAL32K clock hence the actual frequency may vary from 11.7 kHz to 32,768 kHz. The FSM filters the wake up event at the first state (an interrupt coming from a GPIO or an internal timer) and then it performs the same steps as in the power up case. The only difference here is that the amount of time required for the LDOs to settle might be less then before since the sleep time might have been short enough to allow a total discharge of the capacitances, internally as well

as externally. However, do the frequency variation the wake up time might reach 1.2 ms if the RCX is used.

In both cases, as soon as the RC16 clock is enabled, the FSM hands control to the CPU. The CPU will start executing code from address 0x0. If it is a power up, the BootROM code resides at address 0x0. If it is a wake up, the RAM is remapped at address 0x0. The actual application code is responsible for starting the DC-DC controller by setting `DCDC_CTRL_0_REG[DCDC_MODE]=Active`. Furthermore, the respective rails might or might not be enabled by respectively programming the `DCDC_<rail>_ENABLE_xV` bits.

Preparing the system for any of the sleep modes, requires that the application switches back the clock to the RC16 and switches off the XTAL16M (see Figure 17). After executing the WFI command (which puts the

chip into sleep), the clock is switched to the `lp_clk` (RCX or XTAL32K) the retention LDOs are enabled (`LDO_VBAT_RET`, `LDO_IO_RET` and `LDO_IO_RET2`) and the core (VDD) voltage LDO is disabled so that the `LDO_sleep` takes over. The actual amount of clock cycles since the WFI is issued up to the point where the `lp_clk` starts and the FSM changes state is 7. Depending on whether a value has been programmed in the `SLEEP_TIMER_REG`, the system is automatically shortly waken up every so many ticks (`lp_clk`) to quickly sample the bandgap voltage and get back to sleep. The DC-DC and the `LDO_VBAT_RET` require a sample-and-hold operation, hence if these two elements are not part of the sleep strategy, `SLEEP_TIMER_REG` should be kept to 0. The overall FSM latency is 4 `lp_clk` clock cycles

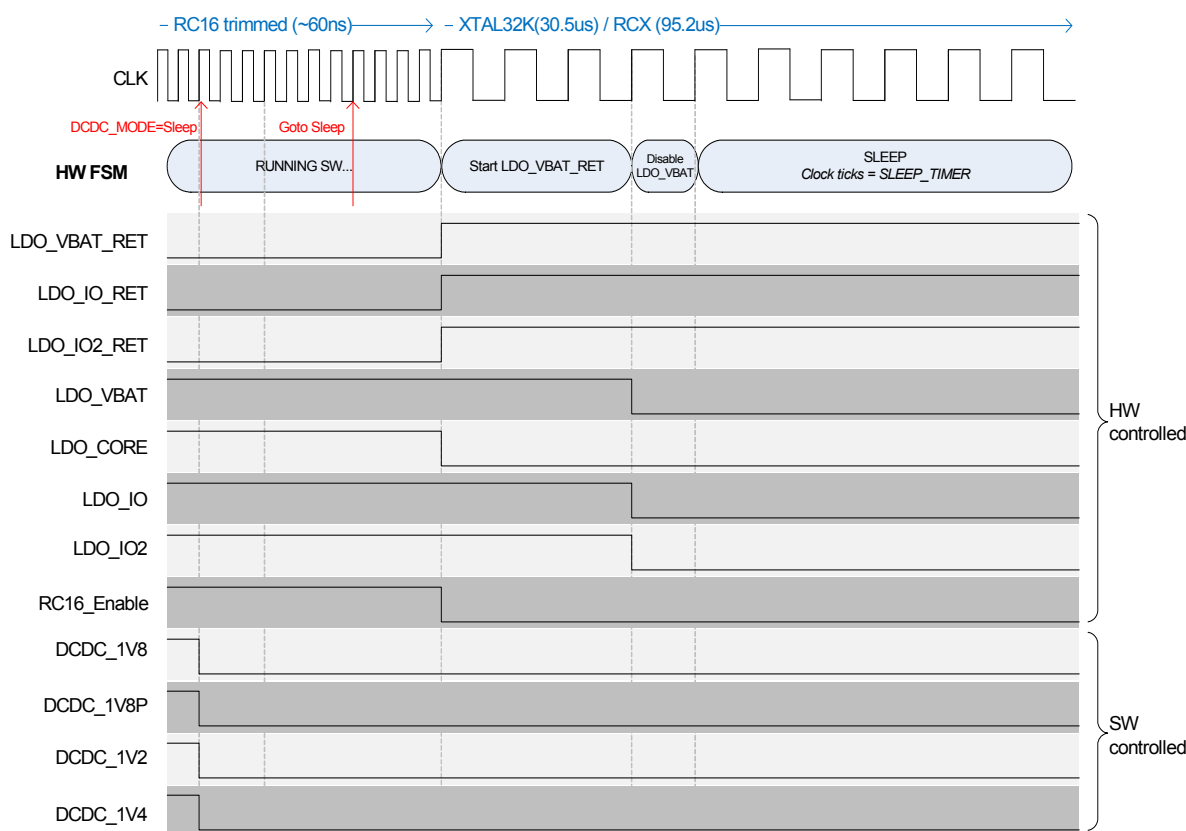


Figure 17: Goto Sleep timing and PMU operations

If the `SLEEP_TIMER_REG` is programmed to wake up the system for rails charging (DCDC operation) and bandgap sampling, then the timing of the HW FSM looks as illustrated in Figure 18. The state machine is clocked by the `lp_clk` and starts with enabling the Bandgap. On the next state, it enables the RC16 clock (`WPS_CLK`) and following that, a sampling of the Brown Out Detection circuit on the various power rail occurs (`WPS_BOD`). The sampling is performed with the `BOD_clk` at 1 MHz (see Figure 26).

Next state is dedicated to the DCDC converter (`WPS_SIMO`). This is when the charging of the different power rails will happen if the DCDC is set at sleep mode by programming the `DCDC_CTRL_0_REG[DCDC_MODE]` flag. During the last 3 states the sampling and hold of the bandgap voltage reference occurs, required by the `LDO_VBAT_RET`, the `LDO_IO_RET` and `LDO_IO_RET2`.

Charging of the DCDC rails as well as sample-and-hold functionality of the LDOs are both displayed in Figure 18 to allow illustrating the waveform for all cases. In reality, depending on the power configuration, only the

pre-defined resources will be part of the intermediate wake up FSM.

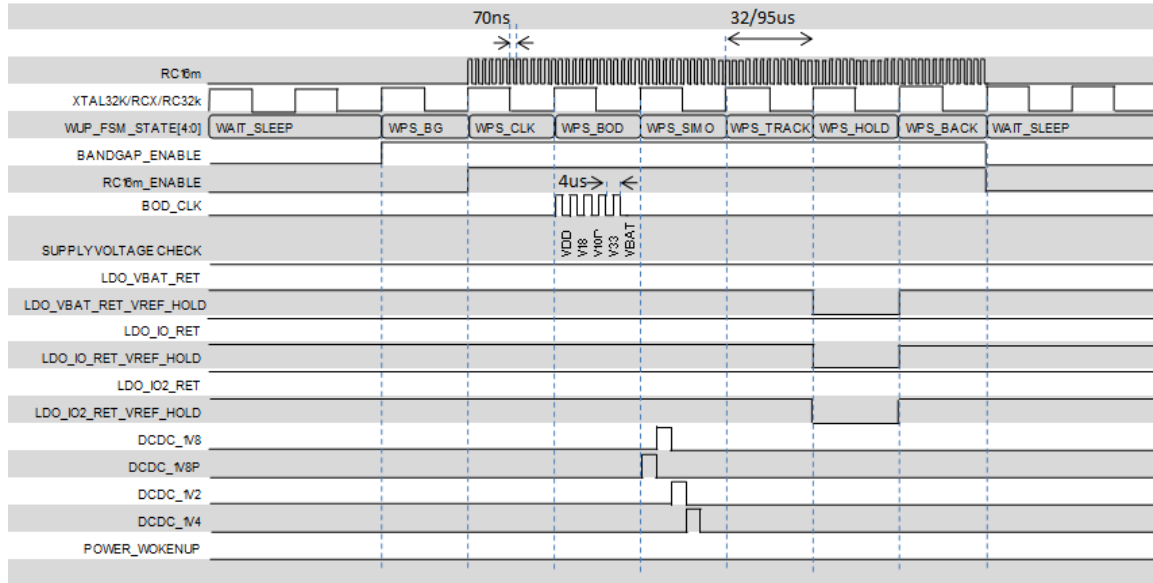


Figure 18: SLEEP\_TIMER based wake up timing

#### 4.1.6 Charger

The charging circuit operates as a constant current/constant voltage source. The control circuit keeps the charge voltage or the charge current at the predefined values (whichever of the two is reached first). The values can be changed via register CHARGER\_CTRL1\_REG.

The charge control circuit is initially supplied from pin VBUS. The complete charging circuit is powered down, when the VBUS voltage is low for more than 10 ms or by setting bit CHARGER\_CTRL1\_REG[CHARGE\_ON] = 0. In a Li-ion application the charger is also disabled, when the voltage on pin NTC is in the “too hot” or “too cold” region.

Note that, the use of the charger requires P1\_4 as a power pin and P1\_6 for controlling the NTC if required (as shown in Figure 21).

##### Pre-Charging and Charging

The Charger is supporting pre-charging and normal charging as explained in the diagram below. The charger starts at constant current mode (CC) to get the battery voltage to the predefined voltage level. It then switches to constant voltage mode and continues until the current drops below the expected threshold indicating that the Battery has reached the charging limits.

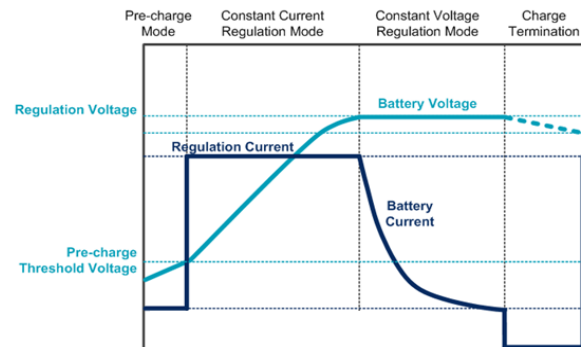


Figure 19: Pre-Charge and Charge Voltage/Current Diagram

The Charger circuitry supports a large range of current levels for pre-charge and charging. If the range is within 5 mA and 400 mA, then the respective level has to be programmed in the CHARGER\_CTRL1\_REG[CHARGE\_CUR] field. If the range of pre-charge currents is between 0.2 mA and 15 mA then programming the CHARGER\_CTRL2\_REG[CHARGER\_TEST]=0x6 in addition to the CHARGER\_CTRL1\_REG is required. This will divide the current level by ~27 hence produce sub-miliamps levels.

The pre-charge and charge state machine as imple-



mented in software is presented in the following flow chart:

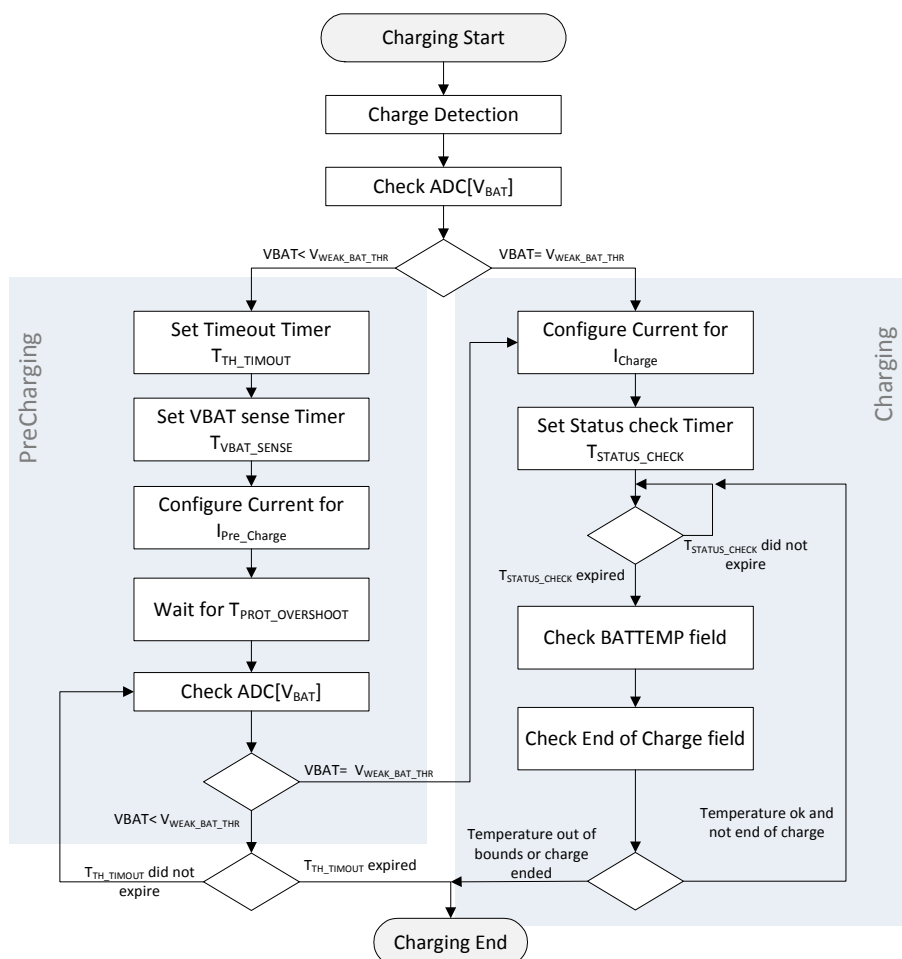


Figure 20: Pre-Charge and Charge SW FSM

The Charge detection is performed by HW according to the “Battery Charging Specification, Rev 1.2, December 2010”. There are several timers used in this flow chart as explained below:

- $T_{TH\_TIMEOUT}$  is the time-out threshold which stops the pre-charging if voltage has not increased within a certain time interval due to possibly a bad battery.
- $T_{VBAT\_SENSE}$  defines the period of checking the voltage level of the battery via the ADC channel. It is set at 10 ms.
- $T_{PROT\_OVERSHOOT}$  defines a programmable interval to avoid sampling a wrong VBAT voltage right after disabling a Battery protection IC. In these cases a small overshoot might be observed which could trick the sampling routine that battery is already charged, hence the charging sequence would be terminated. Default time is 10 ms.
- $T_{STATUS\_CHECK}$  is the time period after which, the

status bits of the charging circuitry are checked: `CHARGER_STATUS_REG[END_OF_CHARGE]` and `CHARGER_STATUS_REG[CHARGER_BATTEMP_OK]`

$V_{WEAK\_BAT\_THR}$  is the voltage threshold below which, a battery is considered as “weak” as explained in the BCS, Rev1.2 specification while  $I_{Charge}$  and  $I_{Pre\_charge}$  the charging current limits configured as already explained.

#### Auto shut-off

The charger auto shut-off circuits for Li-ion/polymer batteries automatically switches off the charger circuit, when the NTC input voltage goes outside the specified voltage ranges as shown in Figure 21.

The charge disable function will be activated, when the voltage ratio (NTC voltage/P1\_4 voltage) is below 1/2 (too hot) and above 7/8 (too cold).



The Auto shut-off control and status bits can be found in registers `CHARGER_CTR1_REG` and `CHARGER_STATUS_REG`.

R1 can be dimensioned as follows:

$$R1 = R_{NTC}@T_{critical}$$

E.g. if the NTC is 4.7 k $\Omega$  at  $T_{critical} = \sim 40^{\circ}\text{C}$ , R1 must also be 4.7 k $\Omega$ , charging will stop at the "too hot" limit. The "too cold" temperature is not critical and will in practice be around  $0^{\circ}\text{C}$ .

If the NTC auto shut-off feature is not required, bit `CHARGER_CTR1_REG[NTC_DISABLE]` must be set to '1'.

### End Of Charge detection

This is an indicator when the charge current is decreased to 10% of the programmed value and can be read at

`CHARGER_STATUS_REG[END_OF_CHARGE]`. The charge cycle is not affected directly. This indicator can be used by higher-level control to manipulate the charge process.

Cold battery protection can be switched-off independently from the `NTC_DISABLE` control.

### Non-rechargeable batteries

A 3.74 V charge level is provided to test for non-rechargeable NiMH batteries. For two NiMH batteries the maximum charging voltage is expected to be 3.4 V. When 3.74 V is selected and an ADC measurement indicates a quick rise above the 3.4 V level, most likely non-rechargeable batteries are present and the charger must be disabled.

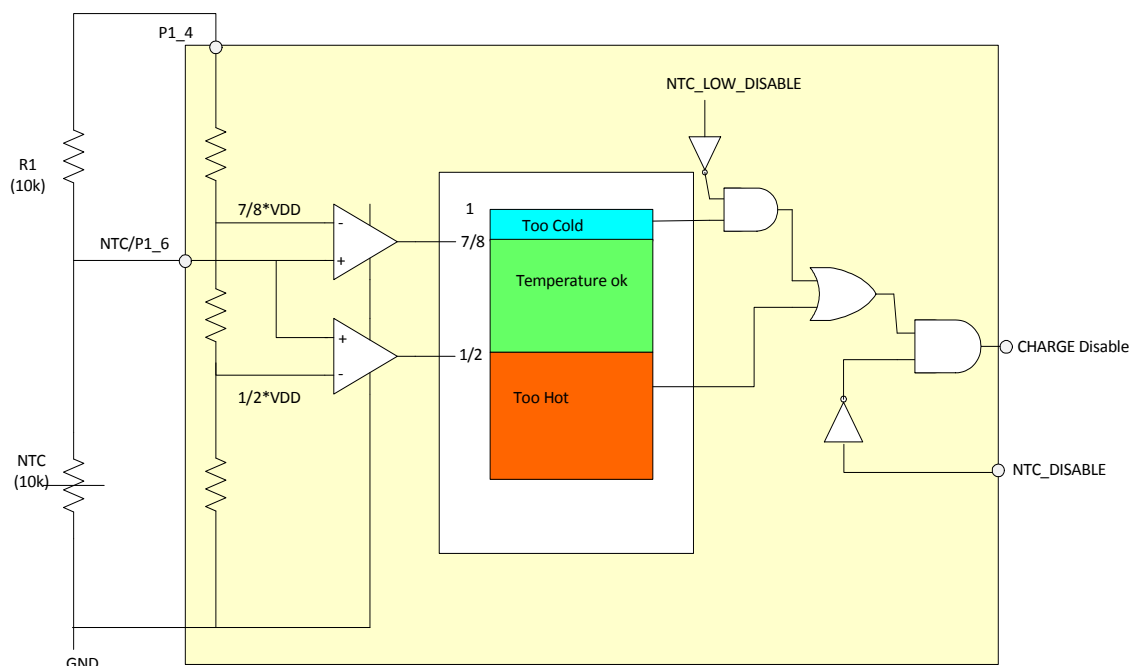


Figure 21: Charger auto shut-off circuit

### 4.1.7 Fuel gauge

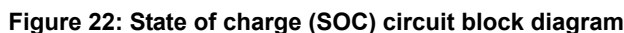
The Fuel Gauge, State of Charge (SOC) circuit is used to accurately determine the actual amount of charge (Coulombs) in rechargeable batteries as well as the discharge state of non-rechargeable batteries. This information can be used as battery status indication to the user.

#### Features

- Coulomb Counter fuel gauge with high accuracy  
Sigma delta ADC 0.05% at 1A Max
- Integrates charge and discharge currents
- Measures average currents
- Offset cancelling by chopper amplifier
- Power saving mode with auto increment

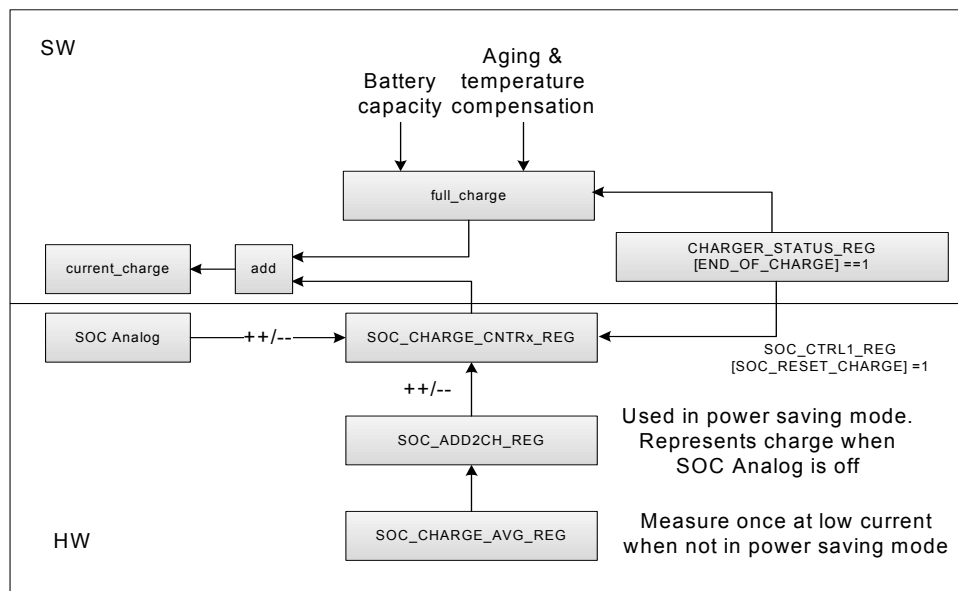
During operation (charge or discharge) the SOC continuously integrates the voltage on SOCn/SOCp using a Sigma Delta ADC and integrating counter which is

All SOC\_\* registers can be read out via APB16 interface.



Reversely, if the counter is reset at the lowest battery voltage, the increasing counter represents the current actual battery charge.

For power saving the SOC analog part can be disabled and the SOC\_ADD2CH\_REG will be used to decrement the SOC\_CHARGE\_CNTRx\_REG at the default sample frequency of 1 MHz. The value of the SOC\_ADD2CH\_REG can be determined from the SOC\_CHARGE\_AVG\_REG when not in power saving.



### Figure 23: Fuel Gauge operation

#### 4.1.8 USB charger detection

The USB controller has built-in hardware to determine the charger type to which it is connected. Depending on the charger type, battery state and USB connection state a defined current can be drawn from the charger.

## Features

- Complies to "Battery Charging Specification" Revision V1.2 December 7, 2010 (BC1.2)

- Charger type detection: Dedicated Charging Port (DCP), Charging Downstream Port (CDP), Standard Downstream Port, PS2 port and Proprietary charger
- Dead battery provision
- Compatible with various smartphone chargers

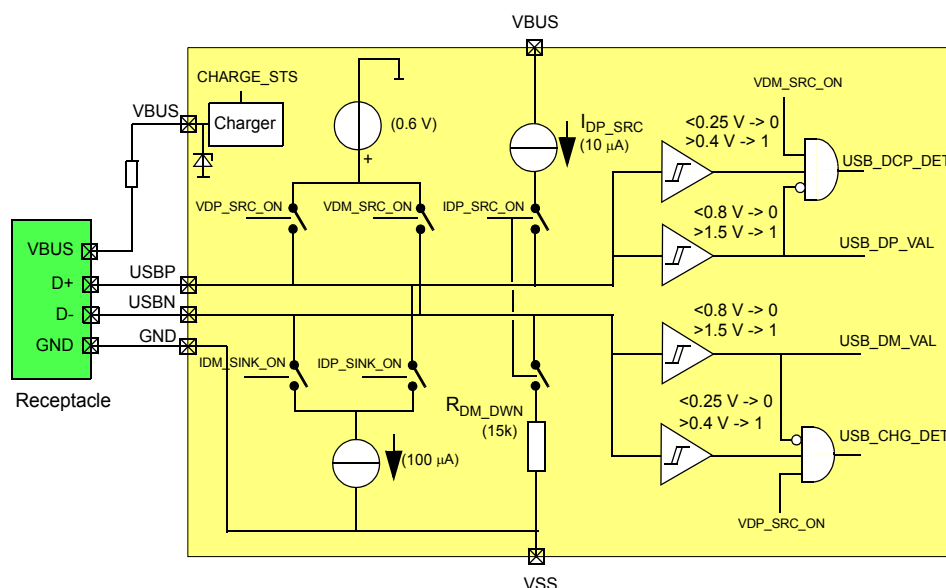


**Figure 24: USB charger detection block diagram (BC1.2)**

The USB interface supports the battery charging with the following hardware blocks as shown in Figure 24.

- A voltage source of 0.6 V can be switched to USBP or USBN with USB\_CHARGER\_CTRL\_REG bits VDP\_SRC\_ON resp. VDM\_SRC\_ON.
- A current sink of 100  $\mu$ A can be switched to USBP or USBN with USB\_CHARGER\_CTRL\_REG bits IDP\_SINK\_ON resp. IDM\_SINK\_ON. (Note that internal logic prevents to both switches can be enable at the same time)
- A current source  $I_{DP\_SRC}$  and  $R_{DM\_DWN}$  can be enabled with USB\_CHARGER\_CTRL\_REG[IDP\_SRC\_ON]
- A logic level Schmitt trigger for USBN, USBP read in USB\_CHARGER\_STAT\_REG bits USB\_DM\_VAL, resp. USB\_DP\_VAL logic level (0: <0.8 V 1: >2 V)
- A comparator output CHG\_DET read in USB\_CHARGER\_STAT\_REG[USB\_CHG\_DET] to detect a level of  $0.4\text{ V} < \text{USBN} < 1.5\text{ V}$  to indicate that a DCP or CDP is connected.
- A comparator output DCP\_DET read in USB\_CHARGER\_STAT\_REG[USB\_DCP\_DET] to detect a level of  $0.4\text{ V} < \text{USBP} < 1.5\text{ V}$  to indicate that a DCP is connected.

Initially all bits of USB\_CHARGER\_CTRL\_REG must be set to reset value '0'.

The presence of VBUS can be checked by reading bit ANA\_STATUS\_REG[VBUS\_AVAILABLE].

Optionally, in case the receptacle's VBUS is connected to the CHARGE pin via a resistor, bit BAT\_STATUS\_REG[CHARGE\_STS] is set when  $V_{\text{charge}} > V_{\text{ih\_charge}}$  (1.5 V). This value meets the VOTG\_SESS\_VLD value. With the keyboard interface configured to detect a level change of the CHARGE pin, bit KEY\_STATUS\_REG[0] is set and a KEYB\_INT is generated. Optionally, an accurate value of VBUS can be determined with a resistor divider on VBUS to VSS, connected to one of the general purpose ADC inputs.

### Detection negotiation

When a downstream port drops VBUS, the bus will discharge and ANA\_STATUS\_REG[VBUS\_AVAILABLE] goes to 0.

### Contact detection

If USB\_CHARGER\_CTRL\_REG[IDP\_SRC\_ON] is set, bit USB\_CHARGER\_STAT\_REG[USB\_DP\_VAL] indicates that the data pins make contact (see Table 11).

It is the responsibility of the SW to wait until the USBN and USBP contact bouncing has finished before the register is read.

**Table 11: USBP, USBN contact detection**

| Port                | USBP   | USBN   | USB_DP_VAL |
|---------------------|--------|--------|------------|
| Nothing connected   | >1.5 V | 0      | 1          |
| Standard downstream | <0.8 V | 0      | 0          |
| Dedicated charger   | <0.8 V | <0.8 V | 0          |
| Charging downstream | <0.8 V | <0.8 V | 0          |

### Primary charger detection

Primary charger detection is used to detect whether the downstream port has charging capabilities or not. The detection is initiated by setting bits USB\_CHARGER\_CTRL\_REG[VDP\_SRC\_ON] and USB\_CHARGER\_CTRL\_REG[IDM\_SINK\_ON]. This enables the voltage source  $V_{DP\_SRC}$  on USBP and the current source  $I_{DM\_SINK}$  on USBN. The measured levels on USBP and USBN shown in Table 12 determine the value of bit USB\_CHARGER\_STAT\_REG[USB\_CHG\_DET].

**Table 12: Charger type detection**

| Port                | USBP  | USBN                        | USB_CHG_DET                              |
|---------------------|-------|-----------------------------|------------------------------------------|
| Dedicated charger   | 0.6 V | >0.4 V<br><1.5 V            | 1                                        |
| Charging downstream | 0.6 V | First <0.25 V<br>Then 0.6 V | 0 (Note 2)<br>then 1 after 1 ms to 20 ms |
| Standard downstream | 0.6 V | <0.25 V                     | 0                                        |
| PS2                 | 2 V   | 2 V                         | 0                                        |

**Note 2:** After 20 ms a valid comparator signal is found, after 40 ms the comparator signal can be read in USB\_CHG\_DET.

Note that when the charger detection is done before the charging downstream port is enabled its  $V_{DM\_SRC}$  (so before 20 ms) a standard downstream port is detected, which is safe but incorrect. After the  $V_{DM\_SRC}$  has been enabled in the charging downstream port, a charger port is detected.

### Secondary charger detection

Secondary charger detection can be used to distinguish a dedicated charger or a charging downstream port. The detection is initiated by setting bits `USB_CHARGER_CTRL_REG[VDM_SRC_ON]` and `USB_CHARGER_CTRL_REG[IDP_SINK_ON]`. This enables  $V_{DM\_SRC}$  and  $I_{DP\_SINK}$ . The difference between a DCP and a CDP is shown [Table 13](#).

**Table 13: Secondary charger detection**

| Port                | USBP             | USBN  | USB_DCP_DET |
|---------------------|------------------|-------|-------------|
| Dedicated charger   | >0.4 V<br><1.5 V | 0.6 V | 1           |
| Charging downstream | <0.25 V          | 0.6 V | 0           |

Finally  $V_{DP\_SRC}$  shall be enabled as defined in the Good Battery Algorithm.

### Interrupts

The charger detection hardware can operate in polling mode or can generate a USB interrupt to the ARM Cortex™ M0. A change in one of the bits [3:0] of register `USB_CHARGER_STAT_REG` sets bits `USB_MAEV_REG[USB_CH_EV]` if the corresponding bits [7:4] are set to '1'. If `USB_CHARGER_STAT_REG` is read, bit `USB_CH_EV` interrupt is cleared. The interrupt "set" conditions have priority over the "clear" condition of the read access.

Once VBUS is inserted, CHARGE generates a `KEYB_INT`. A debounce timer for Contact Detection and wait time to detect a standard downstream port must be done by software polling or by using system timer interrupts `SWTIM_INT`.

### USB V1.1 compatibility

In USB V1.1 the integrated USBP and USBN resistors were used for a dedicated charger detection. Although not very convenient, these resistors can still be used to force High/Low levels on the USB lines.

### Smartphone charger detection

The battery charger detection circuit is able to detect smartphone chargers with characteristics shown in [Table 14](#).

**Table 14: Smartphone charger characteristics**

| USBP  | USBN  | Load current |
|-------|-------|--------------|
| 2.0 V | 2.0 V | up to 500 mA |
| 2.0 V | 2.8 V | up to 1 A    |
| 2.8 V | 2.0 V | up to 2 A    |
| 2.8 V | 2.8 V | Not defined  |

**Table 15: Smartphone charger detection**

| USBP/<br>USBN | USB_DP_VAL /<br>USB_DM_VAL | USB_DP_VAL2 /<br>USB_DM_VAL2 | Voltage          |
|---------------|----------------------------|------------------------------|------------------|
| 2 V           | 1                          | 0                            | >1.5 V<br><2.3 V |
| 2.8 V         | 1                          | 1                            | >2.5 V           |

## 5 Reset

The DA14680 comprises an RST pad which is active HIGH. It contains an RC filter for spike suppression. It also contains a 25 k $\Omega$  pull-down resistor. The typical latency of the RST pad is in the range of 2  $\mu$ s.

Furthermore, a separate programmable Brown-Out block will issue a Power On Reset (POR) upon voltage reaching the minimum threshold on each of the three main power rails of the system.

### Features

- RC spike filter on RST to suppress external spikes (400 k $\Omega$ , 2.8 pF)
- Three different reset lines (SW, HW and POR)
- Power On Reset (POR) issued if voltage threshold reached on VDD, VDD1v8 or V33 rails
- Programmable POR voltage threshold for the VDD rail in sleep (0.7V or 0.8V)
- Configurable POR enables for the VDD1v8 and the VDD3v3 rails

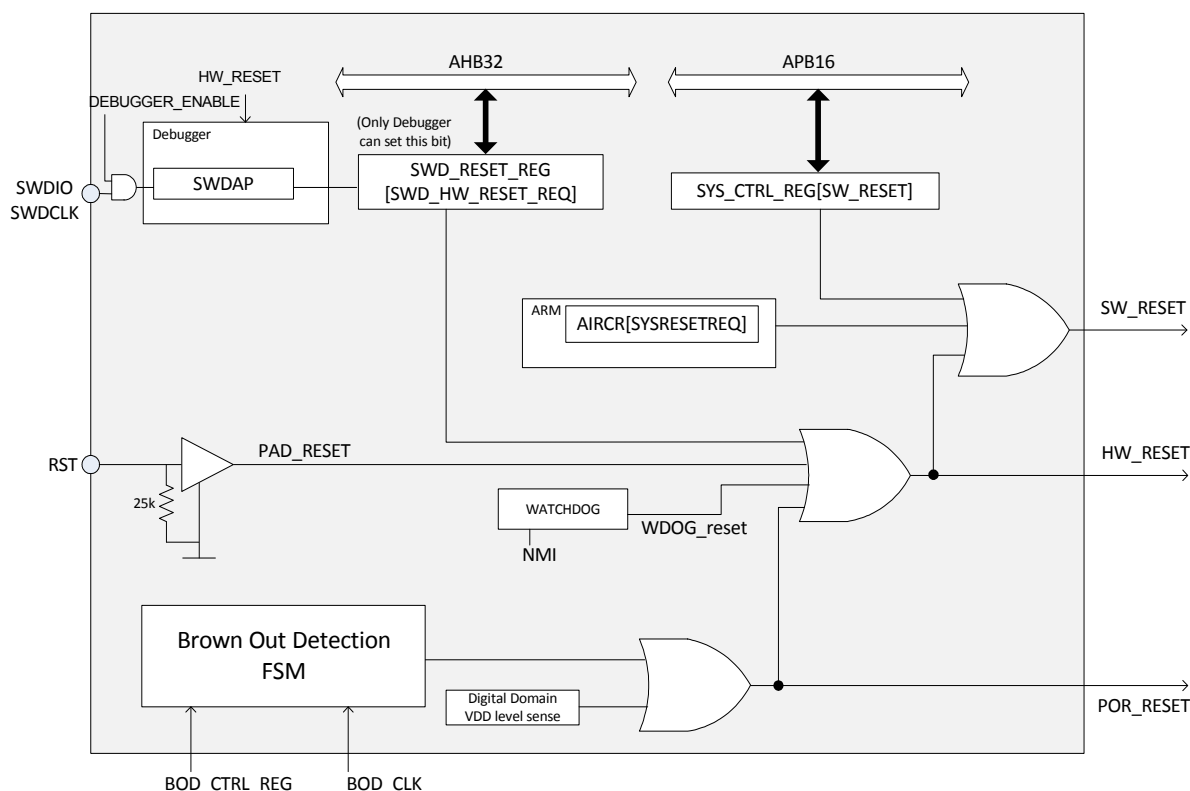


Figure 25: Reset and Brown-Out block diagram

### 5.1 POR, HW AND SW RESET

The Power On Reset (POR) signal is generated internally and will release the system's flip-flops as soon as the VDD voltage crosses the minimum threshold value at 0.65 V and the Brown-Out Detection FSM senses the various internal voltage levels to be higher than the programmed thresholds as explained in [section 5.2](#). The POR resets everything in the chip.

There are two other main reset signals in the DA14680, namely:

- the HW reset which is basically triggered by the RST pad, the Watchdog expiration, the POR and the debugger (by writing SWD\_RESET\_REG)

- the SW reset which is triggered by application software, writing the SYS\_CTRL\_REG [SW\_RESET] bit, the HW reset, and a special ARM command.

The HW reset can also be automatically activated upon waking up of the system from an Extended Sleep or Deep Sleep mode by programming bit PMU\_CTRL\_REG [RESET\_ON\_WAKEUP]. The HW reset will basically run the cold startup sequence and the BootROM code will be executed.

The SW reset is the logical OR of a signal from the ARM CPU (triggered by writing SCB -> AIRCR = 0x05FA0004) and the SYS\_CTRL\_REG [SW\_RESET] bit. This is mainly used to reboot the system after the base address has been remapped.

All registers are reset by POR, a few are not reset by the HW reset signal, and even more are not reset by the SW reset. These registers are listed in the following table:

**Table 16: Reset signals and registers**

| Reset by POR only                                                       | Reset by HW reset                                                                                                                                                                                                                                                                 | Reset by writing to the SW_RESET bit only |
|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| all DCDC_* registers                                                    | all QSPIC_* registers                                                                                                                                                                                                                                                             | The rest of the register file             |
| BANDGAP_REG                                                             | all PLL_* registers                                                                                                                                                                                                                                                               |                                           |
| AON_SPARE_REG                                                           | all CACHE_* registers except for CACHE_MRM_* registers                                                                                                                                                                                                                            |                                           |
| SYS_STAT_REG                                                            | All other OTPC_* registers                                                                                                                                                                                                                                                        |                                           |
| ANA_STATUS_REG                                                          | BLE_CNTL2_REG<br>BLE_EM_BASE_REG                                                                                                                                                                                                                                                  |                                           |
| BOD_STATUS_REG                                                          | DEBUG_REG.<br>DEBUGS_FREEZE_EN                                                                                                                                                                                                                                                    |                                           |
| OTPC_MODE_REG.MODE<br>OTPC_NWORDS_REG<br>OTPC_TIM1_REG<br>OTPC_TIM2_REG | CLK_AMBA_REG[12:0]<br>CLK_FREQ_TRIM_REG<br>CLK_RADIO_REG<br>CLK_CTRL_REG<br>PMU_CTRL_REG<br>SYS_CTRL_REG<br>CLK_32K_REG<br>CLK_16M_REG<br>CLK_RCX32K_REG<br>TRIM_CTRL_REG<br>BOD_CTRL_REG<br>BOD_CTRL2_REG<br>LDO_CTRL1_REG<br>LDO_CTRL2_REG<br>SLEEP_TIMER_REG<br>POWER_CTRL_REG |                                           |

## 5.2 BROWN OUT DETECTION

The DA14680 contains a brown out detection circuitry which is based on sensing selected voltages in the chip every clock cycle. If one of the voltages is found to be below the pre-configured threshold, then a POR is issued.

The BOD FSM is running on the BOD\_clk which is 1 MHz coming from a fixed division of the RC16 clock by 16. The pulses generated for moving from one state to another are 25% duty cycled, hence 250 kHz.

The threshold values for the comparison are configured in the BOD\_CTRL\_REG. Issuing a POR upon sensing a lower voltage than the pre-defined threshold is controlled by BOD\_CTRL2\_REG[BOD\_RESET\_EN]. Furthermore, each sensed voltage can be masked out with use of the BOD\_CTRL2\_REG enable bits. The FSM steps are the following:

1. BOD\_IDLE: idle state. It will proceed to the next state when the system wakes up.
2. BOD\_VDD: senses the VDD voltage. it will issue a HW reset if the respective bits are set.
3. BOD\_V18: senses the VDD1V8 voltage. it will

issue a HW reset if the respective bits are set.

4. BOD\_V18P: senses the VDD1V8P voltage. it will issue a HW reset if the respective bits are set.
5. BOD\_V33: senses the V33 voltage. it will issue a HW reset if the respective bits are set.
6. BOD\_VBAT: senses the VBAT1 voltage. it will not issue a HW reset. Instead, it will trigger the DCDC\_IRQ line as soon as the VBAT1 voltage reaches 2.45V. The reason is to notify SW that from this point and below, DCDC is not providing better efficiency than the LDOs, hence SW should switch the PMU operation to the LDO\_VBAT.
7. BOD\_READY: The RC16 will be turned off during this state and the state machine is set back to BOD\_IDLE.

During the extended sleep mode, the BOD FSM is only triggered if the SLEEP\_TIMER\_REG is programmed with a specific value different than zero. The timing is explained in [Figure 18](#) of [section 4](#).



## 6 Clock generation

### 6.1 CLOCK TREE

The generation of the system's clocks is described in detail in the following figure:

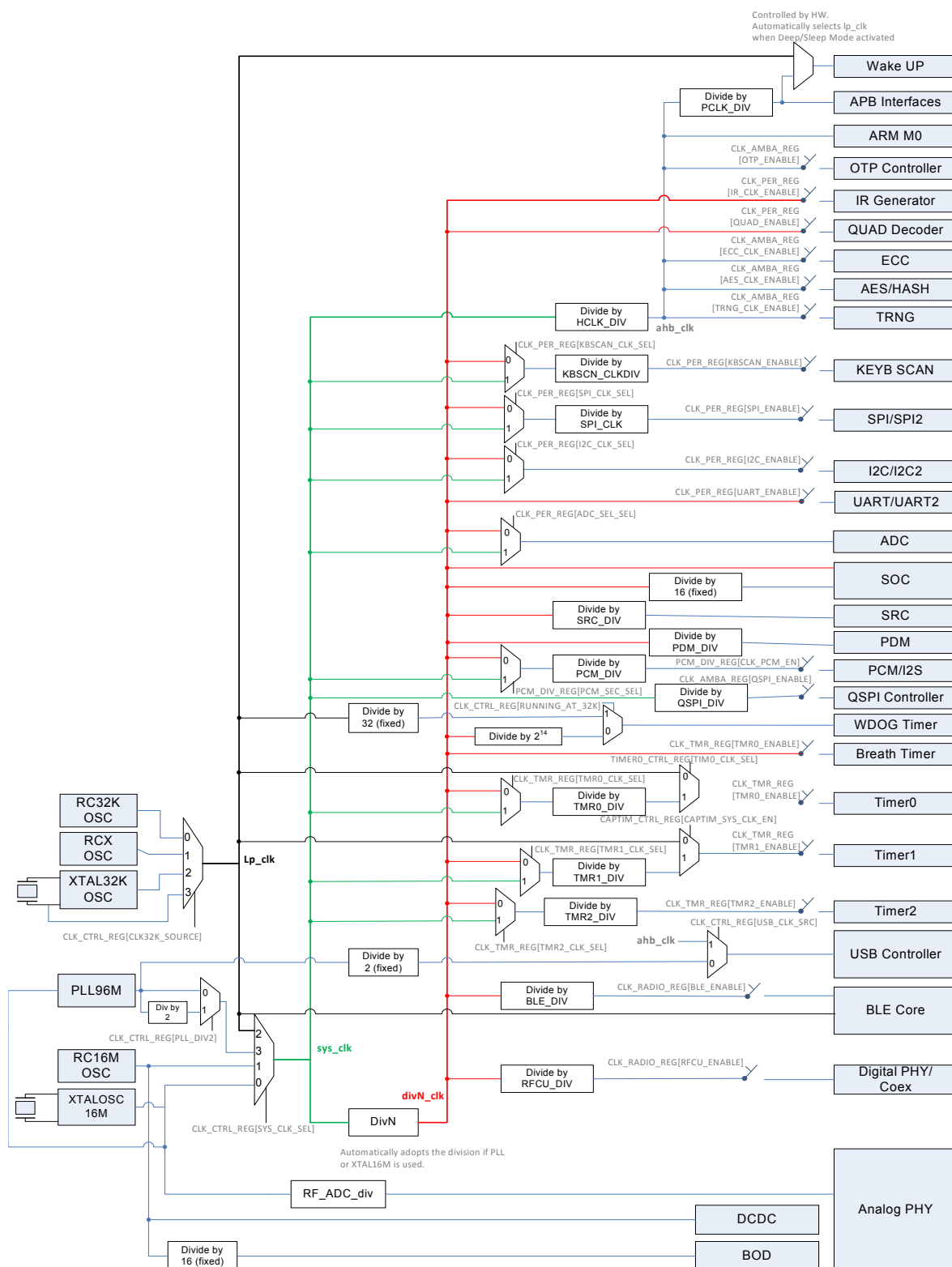


Figure 26: Clock tree diagram

The diagram depicts the possible clock sources as well as all different divisions and multiplexing paths towards the generation of each block's clock. Furthermore, the required registers that have to be programmed are also designated on the same diagram.

There are some main clock lines which are of interest:

- **lp\_clk** (black bold line): this is the low power clock used for the sleep modes and can only be either the RCX, the RC32K or the XTAL32K.
- **sys\_clk** (green line): this is the system clock, used for the AMBA clock (hclk) which runs the CPU the memories and the bus. This clock sources can be any oscillator, the PLL or even an externally supplied digital clock.
- **divn\_clk** (red line): this is a clock which automatically adjusts the division factor on the sys\_clk to always generate 16 MHz. This enables the dynamic activation of the PLL to provide more processing power at the CPU, without affecting the operation of blocks designed for 16 MHz.

## 6.2 CRYSTAL OSCILLATORS

The Digital Controlled Xtal Oscillator (DXCO) is a Pierce configured type of oscillator designed for low power consumption and high stability. There are two such crystal oscillators in the system, one at 16 MHz (XTAL16M) and a second at 32.768 kHz (XTAL32K). The 32.768 kHz oscillator has no trimming capabilities and is used as the clock of the Extended/Deep Sleep modes. The 16 MHz oscillator can be trimmed.

The principle schematic of the two oscillators is shown in [Figure 27](#): below. No external components are required other than the crystal itself. If the crystal has a case connection, it is advised to connect the case to ground.

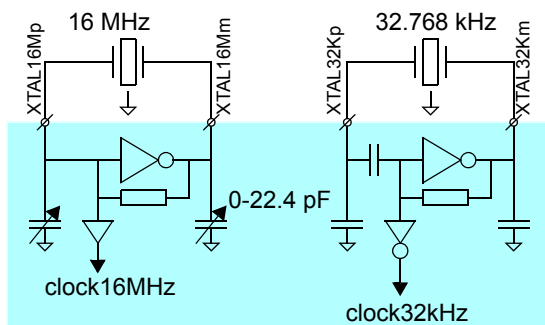


Figure 27: Crystal oscillator circuits

### 6.2.1 Frequency control (16 MHz crystal)

Register CLK\_FREQ\_TRIM\_REG controls the trimming of the 16 MHz crystal oscillator. The frequency is

trimmed by two on-chip variable capacitor banks. Both capacitor banks are controlled by the same register.

The capacitance of both variable capacitor banks varies from minimum to maximum value in 2048 equal steps. With CLK\_FREQ\_TRIM\_REG = 0x000 the maximum capacitance and thus the minimum frequency is selected. With CLK\_FREQ\_TRIM\_REG = 0x7FF the minimum capacitance and thus the maximum frequency is selected.

The eight least significant bits of CLK\_FREQ\_TRIM\_REG directly control eight binary weighted capacitors, as shown in [Figure 28](#). The three most significant bits are decoded according to [Table 17](#). Each of the seven outputs of the decoder controls a capacitor (value is 256 times the value of the smallest capacitor).

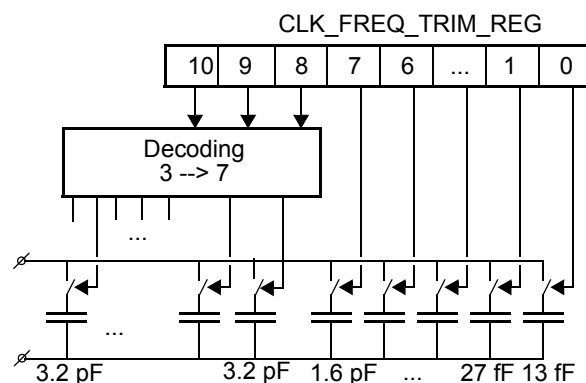


Figure 28: Frequency trimming

Table 17: CLK\_FREQ\_TRIM\_REG Decoding 3 --> 7

| Input[2:0] |   |   | Output[6:0] |   |   |   |   |   |   |
|------------|---|---|-------------|---|---|---|---|---|---|
| 2          | 1 | 0 | 6           | 5 | 4 | 3 | 2 | 1 | 0 |
| 0          | 0 | 0 | 0           | 0 | 0 | 0 | 0 | 0 | 1 |
| 0          | 0 | 1 | 0           | 0 | 0 | 0 | 0 | 0 | 1 |
| 0          | 1 | 0 | 0           | 0 | 0 | 0 | 0 | 1 | 1 |
| 0          | 1 | 1 | 0           | 0 | 0 | 0 | 1 | 1 | 1 |
| 1          | 0 | 0 | 0           | 0 | 0 | 1 | 1 | 1 | 1 |
| 1          | 0 | 1 | 0           | 0 | 1 | 1 | 1 | 1 | 1 |
| 1          | 1 | 0 | 0           | 1 | 1 | 1 | 1 | 1 | 1 |
| 1          | 1 | 1 | 1           | 1 | 1 | 1 | 1 | 1 | 1 |

### 6.2.2 Automated trimming and settling notification

There is provision in the DA14680 for automating the actual trimming of the 16 MHz crystal oscillator. This is a special hardware block that realizes the XTAL trimming in a single step. Notification about the XTAL oscillator being settled after applying the trim value is also

provided in form of an interrupt, namely the `xtal16rdy_irq` line.

and signalling that the oscillator is settled is described in the following figure:

The automated mechanism for applying the trim value

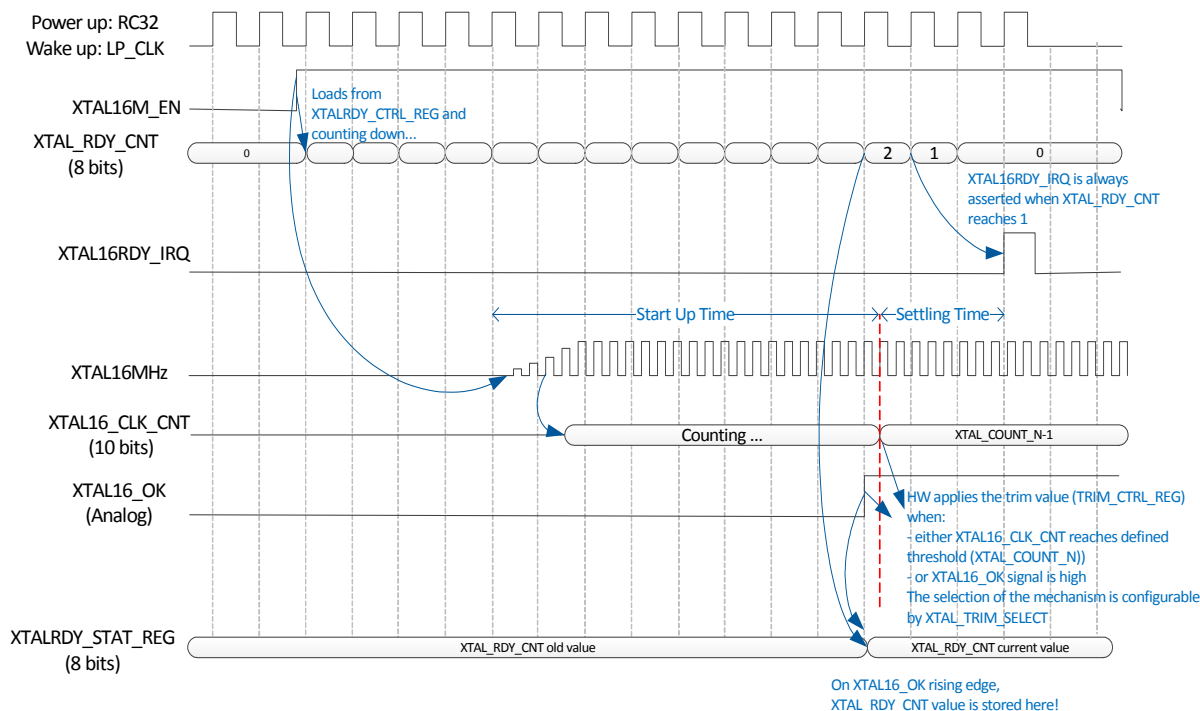


**Figure 29: Automated mechanism for XTAL16M trim and settling**

The XTALRDY\_IRQ is always triggered as soon as an internal counter reaches the value 1 counting down from starting value programmed at XTALRDY\_CTRL\_REG. This counter runs on the RC32 clock if the system is powering up, or the sleep clock if the system is waking up. Enabling of the XTAL16M oscillator is done by HW. There are two intervals until the interrupt notifies the CPU that the XTAL16 can be used:

- The Start-Up, where the XTAL16M oscillator is slowly converging towards the initial frequency of the crystal. This section ends with the application of the trim value to achieve a <50ppm, 16 MHz clock.
- The Settling, where the oscillator settles to the preferred frequency after the application of the trim value

There are two ways of deciding when the start-up section ends. This decision is controlled by TRIM\_CTRL\_REG[XTAL\_TRIM\_SELECT] bitfield:

- Use an analog filter which asserts an internal signal (XTAL16\_OK)
- Use a digital counter which counts TRIM\_CTRL\_REG[XTAL\_COUNT\_N] number of XTAL16M clocks

In any case, the trim value is applied by HW. At the rising edge of the XTAL16\_OK signal, the interrupt coun-

ter current value (XTAL\_RDY\_CNT) is stored in a shadow register XTALRDY\_STAT\_REG. Note that, this register keeps the previous value and will not be reset unless a HW reset occurs.

The settling section usually takes not more than 5 to 10 clock cycles. Using the above, fine tuning and reducing the XTAL16 settling latency is feasible.

The XTAL16\_TRIM\_READY bit indicates that the trimming process is done. This bit is reset after the XTAL16 oscillator is disabled (turning off the system power domain) if XTAL\_TRIM\_SELECT= 00 or 01.

### 6.3 RC OSCILLATORS

There are 3 RC oscillators in the DA14680: one providing 16 MHz (RC16M), one providing 32 kHz (RC32K) and one providing a frequency in the range of 11.7 kHz (RCX).

The 16 MHz RC oscillator is powered by the Digital LDO i.e. the VDD = 1.2 V which is available for the core logic during Active or Sleep Mode. The output clock is significantly slower than 16 MHz if untrimmed and is used to clock the CPU and the digital part of the chip during power up or wake up, while the XTAL16M oscillator is settling.

The simple RC oscillator (RC32K) operates on VDD = 1.2 V and provides 32 kHz. The main usage of the RC32K oscillator is for internal clocking during

power up or startup. It clocks the HW state machine which brings up the power management system of the chip.

The enhanced RC oscillator (RCX) provides a stable 11.7 kHz. The RCX oscillator can be used to replace the 32.768 kHz crystal, since it has a precision of < 500 ppm, while its output frequency is quite stable over temperature.

### 6.3.1 Frequency calibration

The output frequency of the 32 kHz crystal oscillator and the three RC-oscillators can be measured relative to the DivN clock using the on-chip reference counter.

The measurement procedure is as follows:

- REF\_CNT\_VAL = N (the higher N, the more accurate and longer the calibration will be)
- CLK\_REF\_SEL\_REG[REF\_CLK\_SEL] = 0 (RC32K) or CLK\_REF\_SEL\_REG[REF\_CLK\_SEL] = 1 (RC16M) or CLK\_REF\_SEL\_REG[REF\_CLK\_SEL] = 2 (XTAL32) or CLK\_REF\_SEL\_REG[REF\_CLK\_SEL] = 3 (RCX)
- CLK\_REF\_SEL\_REG[REF\_CAL\_START] = 1 Start the calibration
- Wait until CLK\_REF\_SEL\_REG[REF\_CAL\_START] = 0
- Read CLK\_REF\_VAL\_H\_REG and CLK\_REF\_VAL\_H\_REG = M (32-bits values)
- Frequency = (N/M) \* 16 MHz

In the case of using the RCX as a sleep clock, the frequency calibration should be implemented on each active time of a connection interval to guarantee correct operation.

### 6.4 PLL

This low power PLL multiplies the XTAL16M clock to produce a 96 MHz clock with 1% precision within a few us reaching 200 ppm precision in 30 us.

Changing the system's clock in to the PLL output can be done dynamically without affecting the operation of the chip. Its main purpose is to:

- Provide a divided by 2, 48 MHz required for the operation of the USB Controller
- Provide more processing power to the CPU, enabling 80 dMIPS, for computational hungry applications

This PLL dissipates 1.2 mA when operating at 96 MHz while the leakage current can reach 1 uA when the PLL is disabled.

## 7 ARM Cortex M0

The Cortex-M0 processor is a 32-bit Reduced Instruction Set Computing (RISC) processor with a von Neumann architecture (single bus interface). It uses an instruction set called Thumb, which was first supported in the ARM7TDMI processor; however, several newer instructions from the ARMv6 architecture and a few instructions from the Thumb-2 technology are also included. Thumb-2 technology extended the previous Thumb instruction set to allow all operations to be carried out in one CPU state. The instruction set in Thumb-2 includes both 16-bit and 32-bit instructions; most instructions generated by the C compiler use the 16-bit instructions, and the 32-bit instructions are used when the 16-bit version cannot carry out the required operations. This results in high code density and

avoids the overhead of switching between two instruction sets.

In total, the Cortex-M0 processor supports only 56 base instructions, although some instructions can have more than one form. Although the instruction set is small, the Cortex-M0 processor is highly capable because the Thumb instruction set is highly optimized.

Academically, the Cortex-M0 processor is classified as load-store architecture, as it has separate instructions for reading and writing to memory, and instructions for arithmetic or logical operations that use registers.

A simplified block diagram of the Cortex-M0 is shown in Figure 30.

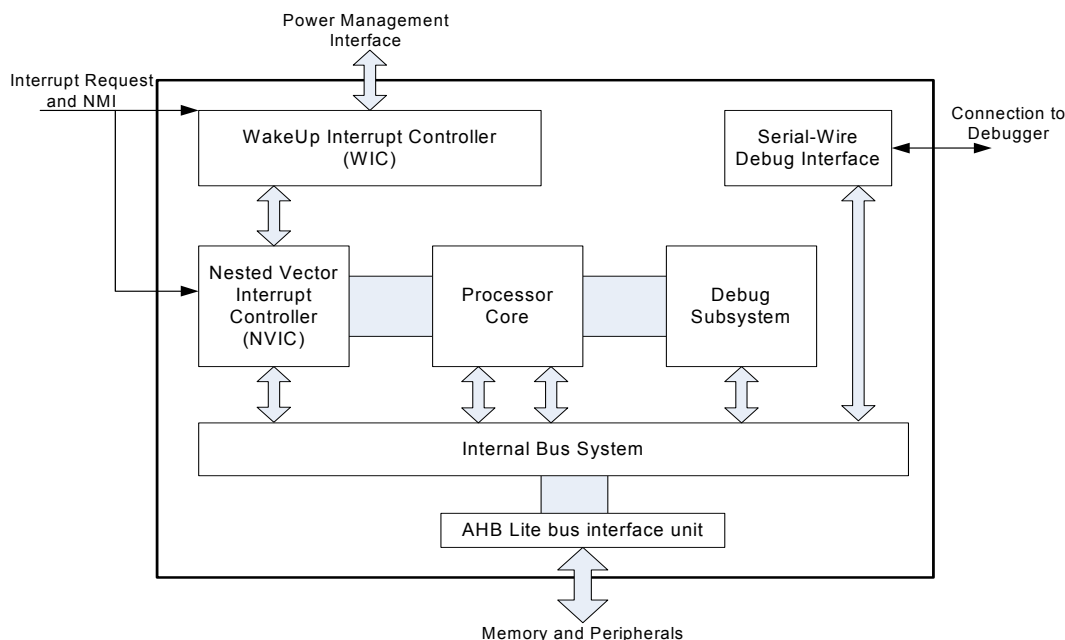


Figure 30: ARM Cortex M0 Block Diagram

### Features

- Thumb instruction set. Highly efficient, high code density and able to execute all Thumb instructions from the ARM7TDMI processor.
- High performance. Up to 0.9 DMIPS/MHz (Dhrystone 2.1) with fast multiplier.
- Built-in Nested Vectored Interrupt Controller (NVIC). This makes interrupt configuration and coding of exception handlers easy. When an interrupt request is taken, the corresponding interrupt handler is executed automatically without the need to determine the exception vector in software.
- Interrupts can have four different programmable priority levels. The NVIC automatically handles nested interrupts.
- The design is configured to respond to exceptions (e.g. interrupts) as soon as possible (minimum 16 clock cycles).
- Non maskable interrupt (NMI) input for safety critical systems.
- Easy to use and C friendly. There are only two modes (Thread mode and Handler mode). The whole application, including exception handlers, can be written in C without any assembler.
- Built-in System Tick timer for OS support. A 24-bit timer with a dedicated exception type is included in the architecture, which the OS can use as a tick timer or as a general timer in other applications without an OS.
- SuperVisor Call (SVC) instruction with a dedicated SVC exception and PendSV (Pendable SuperVisor

service) to support various operations in an embedded OS.

- Architecturally defined sleep modes and instructions to enter sleep. The sleep features allow power consumption to be reduced dramatically. Defining sleep modes as an architectural feature makes porting of software easier because sleep is entered by a specific instruction rather than implementation defined control registers.
- Fault handling exception to catch various sources of errors in the system.
- Support for 32 interrupts.
- Little endian memory support.
- Wake up Interrupt Controller (WIC) to allow the processor to be powered down during sleep, while still allowing interrupt sources to wake up the system.
- Halt mode debug. Allows the processor activity to stop completely so that register values can be accessed and modified. No overhead in code size and stack memory size.
- CoreSight technology. Allows memories and peripherals to be accessed from the debugger without halting the processor.
- Supports Serial Wire Debug (SWD) connections. The serial wire debug protocol can handle the same debug features as the JTAG, but it only requires two wires and is already supported by a number of debug solutions from various tools vendors.
- Four (4) hardware breakpoints and two (2) watch points.
- Breakpoint instruction support for an unlimited number of software breakpoints.
- Programmer's model similar to the ARM7TDMI processor. Most existing Thumb code for the ARM7TDMI processor can be reused. This also makes it easy for ARM7TDMI users, as there is no need to learn a new instruction set.

## 7.1 SYSTEM TIMER (SYSTICK)

The Cortex-M0 System Timer (SysTick) can be configured for using 2 different clocks. The SysTick Control & Status (STCSR) register specifies which clock should be used by the counter.

STCSR[CLKSOURCE]=0; use the (fixed) external reference clock STCLKEN of 1 MHz.

STCSR[CLKSOURCE]=1; use the (HCLK\_DIV dependent) processor clock SCLK (e.g. 2, 4, 8 or 16 MHz).

The default SysTick Timer configuration will be using the (fixed) external reference clock STCLKEN (STCSR[CLKSOURCE]=0). When necessary, higher clock frequencies can be used with STCSR[CLKSOURCE]=1 but the software should take the HCLK\_DIV dependent core clock SCLK into account w.r.t. the timing.

## 7.2 WAKEUP INTERRUPT CONTROLLER

The Wakeup Interrupt Controller (WIC) is a peripheral that can detect an interrupt and wake the processor from deep sleep mode. The WIC is enabled only when the DEEPSLEEP bit in the SCR is set to 1 (see System Control Register on page 4-16 of the Cortex-M0 User Guide Reference Material).

The WIC is not programmable, and does not have any registers or user interface. It operates entirely from hardware signals. When the WIC is enabled and the processor enters deep sleep mode, the power management unit in the system can power down most of the Cortex-M0 processor. This has the side effect of stopping the SysTick timer. When the WIC receives an interrupt, it takes a number of clock cycles to wakeup the processor and restore its state, before it can process the interrupt. This means interrupt latency is increased in Deep Sleep mode.

## 7.3 REFERENCE

For more information on the ARM Cortex M0, see a.o. the ARM documents listed in [Table 18](#).

**Table 18: ARM documents list**

|   | Document title                             | ARM Document number                                           |
|---|--------------------------------------------|---------------------------------------------------------------|
| 1 | Cortex™-M0 User Guide Reference Material   | ARM DUI 0467B (available on the ARM website)                  |
| 2 | Cortex™-M0 r0p0 Technical Reference Manual | ARM DDI 0432C (available on the ARM website)                  |
| 3 | ARMv6-M Architecture Reference Manual      | ARM DDI 0419C (can be downloaded by registered ARM customers) |

## 7.4 INTERRUPTS

This section lists all 32 interrupt lines, except the NMI interrupt, and describes their source and functionality. The overview of the interrupts is illustrated in the following table:

**Table 19: Interrupt list**

| IRQ number (inherent priority) | IRQ name          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                              | ble_wakeup_lp_irq | Wakeup from Extended Sleep by BLE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 1                              | ble_gen_irq       | BLE Interrupt. Sources:<br>- finergtim_irq: Fine Target Timer interrupt generated when Fine Target timer expired. Timer resolution is 625µs base time reference<br>- grosstgtim_irq: Gross Target Timer interrupt generated when Gross Target timer expired. Timer resolution is 16 times 625µs base time reference<br>- cscnt_irq: 625µs base time reference interrupt, available in active modes<br>- slp_irq: End of Sleep mode interrupt<br>- error_irq: Error interrupt, generated when undesired behavior or bad programming occurs in the BLE Core<br>- rx_irq: Receipt interrupt at the end of each received packets<br>- event_irq: End of Advertising / Scanning / Connection events interrupt<br>- crypt_irq: Encryption / Decryption interrupt, generated either when AES and/or CCM processing is finished<br>- sw_irq: SW triggered interrupt, generated on SW request |
| 2                              | RESERVED          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 3                              | RESERVED          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 4                              | rfcal_irq         | RF Calibration Interrupt. Generated by the DPHY.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 5                              | RESERVED          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 6                              | crypto_irq        | Crypto interrupt. Sources:<br>- aes_hash_irq: AES or HASH function interrupt.<br>- ecc_irq: Elliptic Curve interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 7                              | mrm_irq           | Cache Miss Rate Monitor interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 8                              | uart_irq          | uart interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 9                              | uart2_irq         | uart2 interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 10                             | i2c_irq           | I2C interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 11                             | i2c2_irq          | I2C2 interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 12                             | spi_irq           | SPI interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 13                             | spi2_irq          | SPI2 interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 14                             | adc_irq           | ADC interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 15                             | keybrd_irq        | Keyboard scanner interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 16                             | irgen_irq         | IR generator interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 17                             | wkup_gpio_irq     | Wakeup or GPIO interrupt. Will be triggered in Deep Sleep or Hibernation modes (clock-less).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 18                             | swtim0_irq        | Timer0 interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 19                             | swtim1_irq        | Timer1 interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 20                             | quadec_irq        | Quadrature decoder interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 21                             | usb_irq           | USB controller interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 22                             | pcm_irq           | PCM interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 23                             | src_in_irq        | Sample rate converter input interrupt.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |



Table 19: Interrupt list

| IRQ number (inherent priority) | IRQ name      | Description                                                                                                                |
|--------------------------------|---------------|----------------------------------------------------------------------------------------------------------------------------|
| 24                             | src_out_irq   | Sample rate converter output interrupt.                                                                                    |
| 25                             | vbus_irq      | VBUS presence interrupt. This interrupt requires a clock to be generated. It will not be issued in clock-less power modes. |
| 26                             | dma_irq       | DMA interrupt.                                                                                                             |
| 27                             | rf_diag_irq   | Baseband or Radio diagnostics Interrupt.                                                                                   |
| 28                             | trng_irq      | True random number generator interrupt.                                                                                    |
| 29                             | dcdc_irq      | DCDC timeout interrupt.                                                                                                    |
| 30                             | xtal16rdy_irq | XTAL16 oscillator ready interrupt. Clock is 16MHz (<50ppm) and 60/40 % duty cycle                                          |
| 31                             | RESERVED      |                                                                                                                            |

Interrupt priorities are programmable by the ARM Cortex M0. The lower the priority number, the higher the priority level. The priority level is stored in a byte-wide register, which is set to 0x0 at reset. Interrupts with the same priority level follow a fixed priority order using the interrupt number listed in [Table 19](#) (lower interrupt number has higher priority level).

To access the Cortex-M0 NVIC registers, CMSIS functions can be used. The input parameter IRQn of the CMSIS NVIC access functions is the IRQ number. This can be the IRQ number or (more convenient) the corresponding IRQ name listed in [Table 19](#). The corresponding interrupt handler name in the vector table for IRQ#15 is e.g. UART\_Handler. For more information on the ARM Cortex M0 interrupts and the corresponding CMSIS functions, see a.o. section 4.2 Nested Vectored Interrupt Controller on pag. 4-3 in the Cortex™-M0 User Guide Reference Material.

The Watchdog interrupt is connected to the NMI input of the processor.

## 8 Cache Controller

The cache controller is used to accelerate the system performance of the ARM Cortex™ M0 executing from QSPI FLASH and to reduce the power consumption by reducing the access of the QSPI FLASH. The cache dynamically loads both program and data code into the cache Data RAM and executes from there.

The cache controller is controlled via the CACHE\_\*\_REGs. The cache administration is kept in TAG memory. This memory can be invalidated by asserting the FLUSH bit in the CACHE\_CTRL1\_REG. The ARM Cortex™ M0 is halted during this invalidation and resumes automatically. N-way associative replacement strategy is based on the value of a pseudo random LFSR.

The cache controller supports run time configuration of cache line size and associativity. The selection of the configurations depends on the code type and application and shall be determined empirically.

For debugging purposes the Data and TAG memory can be monitored on the AHB-SYS bus (See memory map).

The cache is used for dynamic code and data caching. As an alternative for fast code executions, the data-RAM can be used for static code storage. This code must be copied from QSPI FLASH.

### Features

- Cachable range up-to 32 Mbyte starting from QSPI start address, length adjustable up-to N\*64kByte
- Cache size fixed 16 kB, TAG RAM size is 4 kB
- Run time configurable cache line 8, 16 or 32 bytes
- Run time configurable 1, 2 or 4 way associativity
- Built-in TAG memory invalidation (FLUSH)
- Random number (LFSR) for 2, 4 way replacement strategy
- Cache Data and TAG monitoring
- Instruction and Data caching upon read access, no write path to cache available.
- Bypass mode
- Cache internal latency
  - zero wait cycle for cache hits same cache line
  - one wait cycle for cache hits when changing cache line
  - 4 + cache line size/4 cycles for cache misses
  - 0 cycle in transparent bypass mode

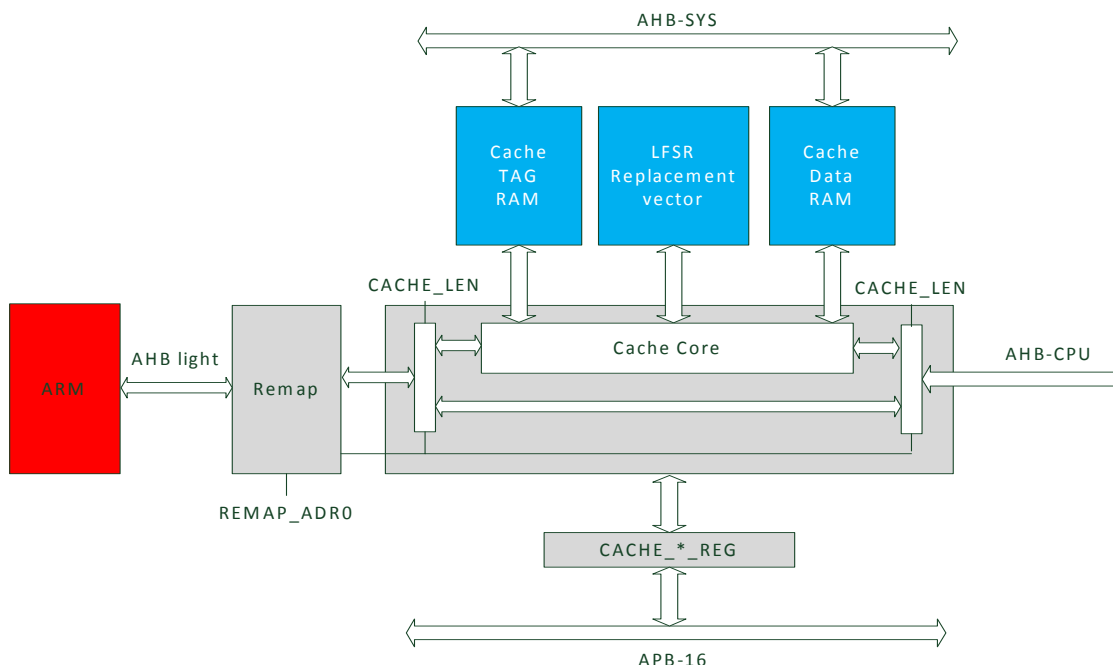


Figure 31: Cache controller block diagram

### 8.1 CACHABLE RANGE

The cache controller caches address range 0-0x1FFFFFFF (32 MB). If REMAP\_ADR0=0x1 or 0x2, all addresses from 0 to CACHE\_LEN will be cached, else the cache controller automatically asserts the

bypass mode. The bypass mode can be forced for all addresses by setting CACHERAM\_MUX = 0.

Note that the CACHE\_LEN setting is only applicable for the QSPI FLASH remap case.

## 8.2 RUNTIME RECONFIGURATION

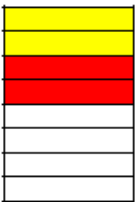
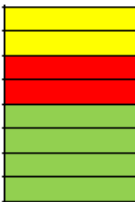
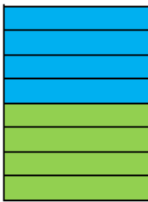
Associativity and cache line size of cache can be reconfigured at all time by writing the CACHE ASSOCCFG\_REG or CACHE LNSIZECFG\_REG registers. Reconfiguration is done without wait state and without flushing the cache. All the data available in the cache memory are kept except when the associativity is reduced (4-way -> 2-way and 2-way -> 1-way). In that case typically half of the data are inaccessible.

### 8.2.1 Cache Line reconfiguration

The dynamic configuration of the cache line size uses a physical line size of 8 bytes. When the cache line size is defined as 16 bytes or 32 bytes, 2 or 4 physical lines are involved.

Reconfiguration of the cache line occurs only when lines are replaced. Even if the cache line configuration is set to 32 bytes, cache lines of 16 bytes may remain in the cache memory as explained at the example in Table 20.

**Table 20: Cache line size reconfiguration example**

| CACHE Memory                                                                                                                                                                          | CACHE Memory                                                                                                                                                                                                                                                                                   | CACHE Memory                                                                                                                                                                                                | Main Memory                                                                          |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|                                                                                                      |                                                                                                                                                                                                               |                                                                                                                           |  |
| <b>Step1:</b><br>Cache line size = 16 bytes<br>a. CPU Reads @0x00: Miss<br>Cache Reads 16 bytes in Block 0 @0x00<br>b. CPU Reads @5010: Miss<br>Cache Reads 16 bytes in Block 5 @0x10 | <b>Step 2:</b><br>Cache line size = 32 bytes<br>a. CPU Reads @0x00: Hit<br>b. CPU Reads @5010: Hit<br>c. CPU Reads @5020: Miss<br>Cache reads 32 bytes in Block 5 @0x20<br><br>Although the configuration is 32 bytes, the lines of 16 bytes remain in the cache. Only new lines are 32 bytes. | <b>Step 3:</b><br>Cache line size = 32 bytes<br>a. CPU Reads @0x6000: Miss<br>Cache reads 32 bytes in Block 6 @0x00<br><br>16-byte cache lines already in the cache are replaced by the 32-byte cache line. |                                                                                      |

### 8.2.2 TAG memory word

The administration memory word decoding is presented in the following table:

**Table 21: TAG memory layout**

| Bit 23 | Bits 22:2   | Bits 1:0                                                                                |
|--------|-------------|-----------------------------------------------------------------------------------------|
| 0      | AHB address | 00: invalid cache line<br>01: 8 valid bytes<br>10: 16 valid bytes<br>11: 32 valid bytes |

### 8.2.3 Associativity reconfiguration

To enable associativity reconfiguration, the cache memory is organized into four banks.

Depending on the associativity, the four banks can operate as 4-way, or be concatenated resulting in either a 2-way or 1-way cache.

## 8.3 2 AND 4 WAY REPLACEMENT STRATEGY

The cache controller fills each line of the cache first starting from way0 to way3. When a line is completely full, a new way\_x victim is chosen in a pseudo random way.

When a replacement is required, the cache controller reads the value generated by a pseudo-random number generator to select which way to replace.

The pseudo-random number generator is realized using a Linear Feedback Shift Register (LFSR).

## 8.4 CACHE RESETS

The cache controller has two reset signals connected:

- The HW\_RESET. When this reset is activated, all cache logic and registers are reset to their default values, while all data in the TAG memories are cleared and all CACHE\_\*\_REG are set to their reset values. It takes around 450 AHB cycles to clear the cache TAG memory.

If a fetch request occurs during this reset period, the request will be taken into account at the end of the reset and wait-states will be inserted.

- The SW\_RESET. Upon a SW\_RESET the cache state machine and TAG memories are reset, but the CACHE\_\*\_REG are not affected and will remain as programmed.

### 8.5 CACHE MISS RATE MONITOR

The DA14680 incorporates a cache miss monitoring circuitry which is providing real time information on the number of cache misses within a certain amount of time. Upon reaching a programmable threshold, an interrupt is issued towards the CPU to take action. The CPU can dynamically change the cache line size, the associativity or start the PLL to decrease cache line fetch time and consequently power. It can even apply a combination of the aforementioned techniques to adjust system's parameters accordingly. This block only operates while the system is in active mode. Main features are:

- Up to 10 ms active time interval counter
- Registered amount of cache misses
- Registered amount of cache hits
- Programmable threshold of cache misses upon which, an interrupt is generated

CACHE\_MRM\_HITS\_REG contains the amount of cache hits and CACHE\_MRM\_MISSES\_REG the amount of misses counted within the time interval programmed at CACHE\_MRM\_TINT\_REG in CPU clock cycles.

### 8.6 CACHE MISS LATENCY AND POWER

This section describes the amount of time (in clock cycles) required from a cache miss up to the point the required code/data are fetched back to the CPU and execution continues.

The cache miss latency ( $T_{CML}$ ) can be split in the following intervals:

$T_{CM2R}$ : Time from the cache miss up to request from the QSPI Controller.

$T_{R2QA}$ : Time from request up to actual access start.

$T_{RDFL}$ : Time for reading data from the FLASH.

$T_{CLAT}$ : Time required to get data to the CPU (cache latency).

The final amount of clock cycles is calculated by the following equation:

$$T_{CML} = T_{CM2R} + T_{R2QA} + T_{RDFL} + T_{CLAT}$$

where  $T_{RDFL}$  depends on the amount of data requested and is provided by the following formula:

$$T_{RDFL} = T_{CMD} + T_{ADDR} + T_{DUM} + (N_{CACHELINE} * 2) + T_{PIPE}$$

For example, give a cache line configuration of 16 bytes, the amount of clock cycles required to read the

data from the FLASH is:

$$T_{RDFL} = 2 + 8 + 4 + (16 * 2) + 1 = 47 \text{ clock cycles.}$$

An overview of the cache miss latency calculation is shown in the following table:

**Table 22: QSPI FLASH cache miss latency**

| Time Interval | Clock Cycles                                                    | Example |
|---------------|-----------------------------------------------------------------|---------|
| $T_{CM2R}$    | 3                                                               | 3       |
| $T_{R2QA}$    | 2                                                               | 2       |
| $T_{RDFL}$    | $T_{CMD} + T_{ADDR} + T_{DUM} + (N_{CACHELINE} * 2) + T_{PIPE}$ | 47      |
| $T_{CLAT}$    | 4                                                               | 4       |
| $T_{CML}$     | for 16 bytes cache line (QPI mode)                              | 56      |

The same calculation applies for the OTP cached case:

**Table 23: OTP cache miss latency**

| Time Interval   | Clock Cycles            | Example |
|-----------------|-------------------------|---------|
| $T_{CM2R}$      | 3                       | 3       |
| $T_{OTP\_INIT}$ | 2                       | 2       |
| $T_{RDOTP}$     | $N_{CACHELINE} / 4$     | 4       |
| $T_{CLAT}$      | 4                       | 4       |
| $T_{CML}$       | for 16 bytes cache line | 13      |

The  $T_{OTP\_INIT}$  parameter represents the amount of cycles required for initiating the burst read of the memory.

Especially in the case of the QSPI FLASH, the amount of cache misses, the latency and the FLASH device might affect the power of the application since the FLASH has to be activated for long time intervals. However, it is measured that given a low cache miss rate, the average power is not dramatically increased. The following figure illustrates the current overhead as a result of the cache miss rate.

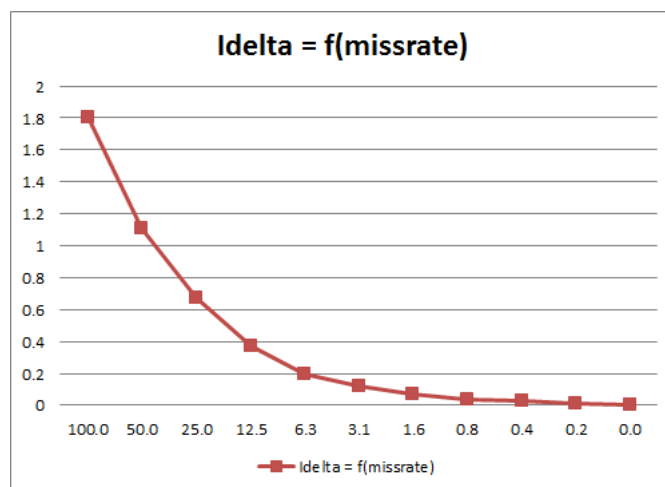


**Figure 32: Current dissipation overhead due to cache miss rate (at 3V)**

The average increase on an application using a FLASH device powered by the VDD1V8 pin (1.8V), with 5mA read current, is becoming visible (i.e. <1mA) if the miss rate exceeds 40%. For applications that are periodically repeating activities, the expected miss rate should be <5% hence the power overhead is minimum.

## 9 AMBA Bus

The DA14680 is equipped with a multi-layer AMBA bus which enables parallel data paths between different masters and slaves. The bus matrix comprises 2 main busses namely the AHB-CPU bus where the Cache, or the CPU can be masters and the AHB-DMA bus where the DMA the OTP Controller and the AES/HASH DMA can be masters.

### Features

- Enables data transfers from peripherals to memory while executing code from OTP or QSPI FLASH
- Provides programmable master priority on AHB-CPU bus
- Provides programmable ICM priority for the connected Slaves
- Enables AHB access of the CPU at the cache RAM if no cache functionality is required

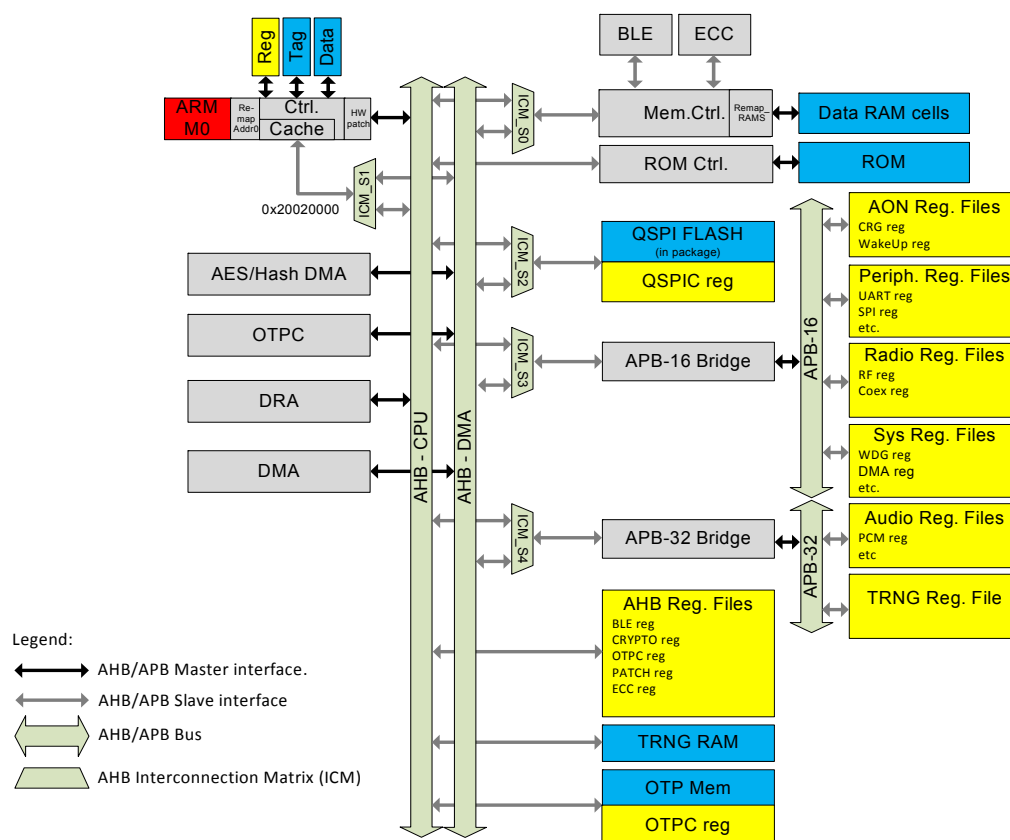


Figure 33: AMBA Bus architecture

There are six slaves which are sitting behind interconnection multiplexers (ICMs) namely:

- Memory controller which controls the DataRAM and the ROM cells
- QSPI FLASH memory controller and memory
- 16-bit APB peripheral registers
- AHB register files for the rest of the resources as well as the TRNG RAM
- 32-bit APB peripheral registers
- Cache controller which enables AHB access of the CPU at the cache RAM to be used as an extension of the DataRAM in the case of mirrored mode.

In the latter case, the ICM\_S0 provides a direct access of the CPU at the cache RAM cell. The Cache RAM cell is always mapped at address 0x20010000. The TAG RAM is not used during this mode of operation.

The priority on the AHB-DMA busses regarding the master arbitration is programmable. The default configuration provides the OTP Controller with highest priority followed by the AES/HASH and the DMA as last.

## 10 Memory Controller

The Memory controller is responsible for the interface of the memory cells with the masters of the system requesting for access. It comprises an arbiter which allows for a configurable priority level between the 3 main masters of the RAM. The memory controller also allows for the actual physical sequence of the RAM cells in a continuous memory space enabling activation of just the required amount of DataRAM thus saving on power.

### Features

- Five different RAM cells with retention capability (one 8 kB, one 24 kB and three 32 kB)
- Full flexibility of re-arranging the first 3 RAM cells in a continuous RAM space starting at 0x7FC0000
- Arbitration between the AHB masters (CPU or DMAs) the BLE core and the ECC
- Retainable configuration of the RAM cells sequence.

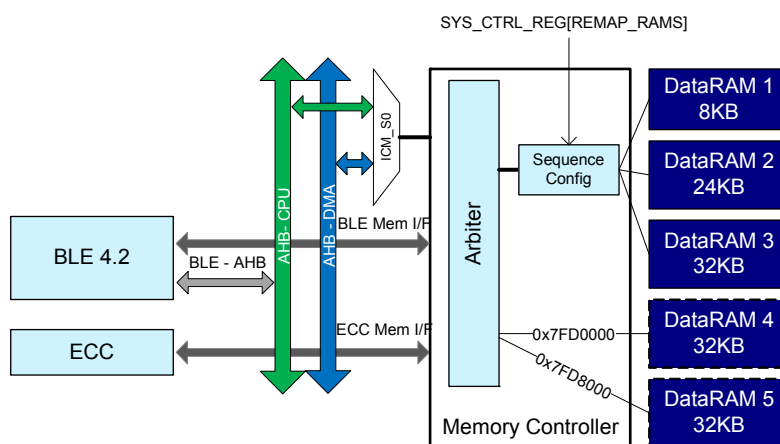


Figure 34: Memory Controller Block Diagram and environment

The Shuffle\_RAM word in the OTP header encodes the sequence of the RAM cells. This is described in the following table:

Table 24: DataRAM cells sequence

| Value | Cell     | Address   | Size (kB) |
|-------|----------|-----------|-----------|
| 0x0   | DataRAM1 | 0x7FC0000 | 8         |
|       | DataRAM2 | 0x7FC2000 | 24        |
|       | DataRAM3 | 0x7FC8000 | 32        |
| 0x1   | DataRAM2 | 0x7FC0000 | 24        |
|       | DataRAM1 | 0x7FC6000 | 8         |
|       | DataRAM3 | 0x7FC8000 | 32        |
| 0x2   | DataRAM3 | 0x7FC0000 | 32        |
|       | DataRAM1 | 0x7FC8000 | 8         |
|       | DataRAM2 | 0x7FCA000 | 24        |
| 0x3   | DataRAM3 | 0x7FC0000 | 32        |
|       | DataRAM2 | 0x7FC8000 | 24        |
|       | DataRAM1 | 0x7FCE000 | 8         |

Selecting the appropriate shuffle value, the minimum required memory space can be left retained during sleep modes hence reducing the sleep power dissipation.

The Memory Controller contains an Arbiter which con-

nects to the following busses:

- ICM which multiplexes the CPU or the DMA access. This interface is capable of operating at maximum 96MHz
- BLE Memory I/F: this is a memory interface from the BLE 4.2 Core directly accessing the RAM used as exchange memory (TX/RX descriptors etc.). This interface is always operating at 16MHz.
- ECC Memory I/F: this is a memory interface from the Elliptic Curve Crypto block directly accessing the RAM used as crypto shared memory. This interface is capable of operating at maximum 96MHz.

The Arbiter implements the priority scheme as depicted in the following figure:



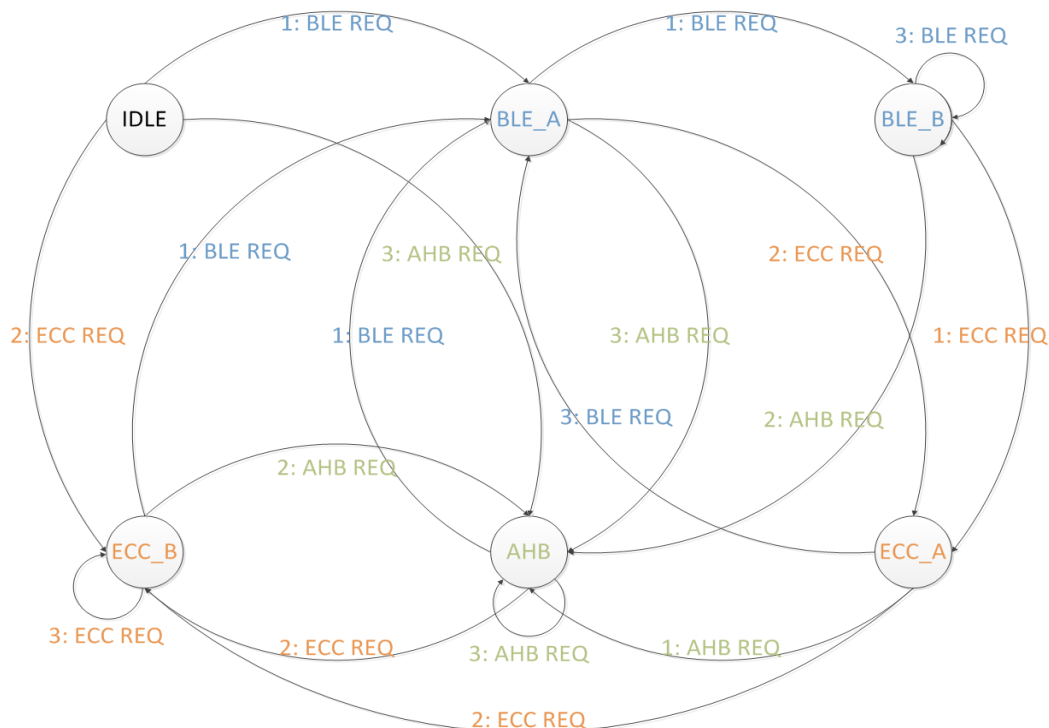


**Figure 35: Memory controller's arbitration scheme**

The overall RAM size can reach up to 144 kB using all available RAM cells of the system (except for the TAG RAM cell). Bypassing the cache controller adds the Cache RAM on the overall RAM budget at the end of the available memory space (0x7FC8000).

The Sequence Configuration block defines the sequence of the memory cells in a continuous memory space according to the REMAP\_RAMs vector. All of the cells can be retained at will. Shuffling of the memory cells will give the best configuration in terms of exchange (retainable) memory, code segment and data segment.

The Memory Controller operates at the same clock as the CPU.

## 11 OTP Controller

The OTP controller realizes all functions of the OTP macro cell in an automated and transparent way. The controller facilitates all data transfers (reading and programming), while implementing the required OTP test modes in hardware. It integrates Error Correcting Code (ECC) hardware for correcting single bit and detecting double bit errors and Built-in Self Repair (BISR) protecting the memory cell space. An embedded DMA engine enables mirroring of the OTP contents into the DataRAM via the AHB-DMA bus.

### Features

- Implements all timing constraints for any access to the physical memory cells.
- 64-bits read in a single clock cycle from the OTP cell
- Transparent random address access to the OTP memory cells via the AHB slave memory interface.
- Embedded DMA engine for fast mirroring of the OTP contents into the System RAM.
- Embedded DMA supports reading in bursts of 8 32-bit words
- Built-In Self Repair (BISR) mechanism for programming and reading
- Up to 48 MHz operation (96 MHz is not supported)
- Hardwired handshaking with the PMU to realize the mirroring procedure
- Automatic single Error Code Correction (ECC) and double error detection

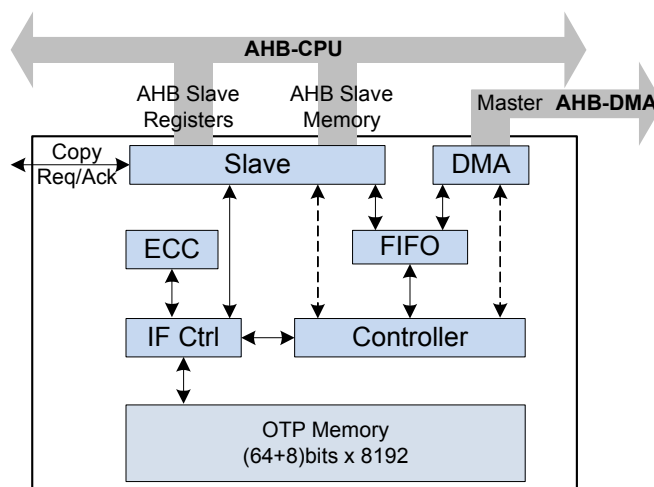


Figure 36: OTP Controller block diagram

### 11.1 OPERATING MODES

There are two different functional modes of operation for reading and programming respectively: manual (MREAD, MPROG) and automatic (AREAD, APROG). The OTP operating mode is programmable at `OTPC_MODE_REG[OTPC_MODE_MODE]`.

The **MREAD** mode enables the use of the memory slave interface. By activating this mode the contents of the macro cell are transparently mapped onto the specific AHB slave address space. The controller runs the SECDED algorithm on the fly to correct single bit errors and notify for dual bit errors by means of the status register. This mode can be used for execution of software in place (XIP).

The **AREAD** mode provides the ability for reading data from the macro cell in bursts, without the use of the slave interface. This mode is used for copying large data blocks from the macro cell, as in the case of the OTP mirroring into the DataRAM. As in the MREAD mode, SECDED is also applied on every burst. However, transfer will occur even if an error has been identified in the status register.

tified in the status register.

The **MPROG** mode provides the functionality for programming a 64 bit word. The controller expands the 64 bit word by calculating and appending an 8-bit checksum, implementing SECDED (Single Error Correction Double Error Detection). In this way a complete 72 bits word is constructed, which is stored at a selected OTP position.

Programming is performed in a single step. In the case that one or more bits have failed to be programmed correctly, software should trigger re-programming.

The 64 bit data word as well as the address are defined through configuration registers. The controller applies the corresponding control sequence for the programming and the result of the verification step is indicated in a status register.

The **APROG** mode gives the ability for programming large data blocks into the macro cell. The programming is an automated procedure, during which it is only necessary to feed the controller with the required data.

Data blocks can be fetched in two ways:

- Via the AHB master interface, i.e. the DMA.
- Via the AHB slave registers.

In the latter case, data are pumped into the OTP controller through a register, which acts as a port providing access to a FIFO. The controller expands each 64 bit word by calculating and adding automatically an 8-bit checksum, in order to provide SECDED functionality.

## 11.2 AHB MASTER INTERFACE

The AHB master interface is controlled by a DMA engine with an internal FIFO of 8 32-bit words. The DMA engine supports AHB reads and writes. The AHB address where memory access should begin, is programmed into the DMA engine at `OTPC_AHBADR_REG[OTPC_AHBADR]`. The number of 32-bit words (minus 1) of a transfer must be specified in `OTPC_NWORDS_REG[OTP_NWORDS]`.

The DMA engine internally supports the following burst types:

- Eight words incremental burst (INCR8)
- Four words incremental burst (INCR4)
- Unspecified incremental burst (INCR) with length different than 1, 4 or 8
- Single word access (SINGLE)

## 11.3 AHB SLAVE INTERFACES

The slave block combines two AHB slave interfaces. One for the registers and another for the contents of the OTP memory. The first AHB slave is read/write while the second is read only. The controller should be configured into MREAD mode prior to any access on the slave interfaces. If this is not the case, an ERROR response on the bus will occur. The same ERROR can also be triggered upon a SECDED detection.

## 11.4 ERROR CORRECTING CODE (ECC)

The error correcting code is based on the Hamming code, for a single bit error correction. The functionality of the Hamming code has been enhanced with the addition of a parity bit. The presence of the parity bit enables the detection of a double bit error.

The redundancy that is provided by the use of the two algorithms (Hamming and parity generation) is stored together with the actual data at each OTP position, consisting of a 72 bits word. The exact layout of the OTP word is presented in the following figure:

| Bit 71 | Bit 70-64  | Bit 63-0     |
|--------|------------|--------------|
| Parity | Check Bits | Payload Data |

Figure 37: OTP word layout

## 11.5 BUILD-IN SELF REPAIR (BISR)

The repair mechanism is available only during programming (APROG mode only) or reading (Both AREAD and MREAD). Only the main memory array is protected (there is no repair mechanism for the spare rows). It is also not available during blank check.

In the case of programming the OTP, the controller initially tries to write to the normal memory array. There are two cases depending on the result of the programming:

1. The programming in the main memory array succeeds (with one or zero errors). The programming ends normally.
2. The programming of the main memory array fails. If there are already 8 repair records occupied, programming fails and the device is discarded. Otherwise, a new repair record is added. The controller writes the new repair record in the spare area. In the case of a failure (two or more errors) the device is discarded. Otherwise the programming ends successfully.

Reading from the OTP cell requires the corresponding registers of the OTP controller to be loaded with the repair information. When a read action is requested, the OTP controller performs a search in the repair records.

If the address, of the read requested, is found in one of the repair records, the data are not retrieved from the normal memory array of the OTP, but from the repair records. The ECC is, in this case, bypassed.

If there is no match to a repair record, data are retrieved from the normal memory array. ECC is then activated.

## 12 Quad SPI Controller

The Quad SPI Controller (QSPIC) provides a low pin count interface to FLASH memory devices. The QSPIC supports the standard Serial Peripheral Interface (SPI) and a high performance Dual/Quad SPI Interface.

The QSPIC gives the ability to read data from a quad FLASH memory, transparently through the SPI bus. This Execute In Place (XIP) feature combined with the CPU cache, provides comparable performance to executing code from standard parallel FLASH. In this case the QSPIC generates all the control signals for the SPI bus that are needed to read data from the serial FLASH memory. Additionally, software can easily control the serial FLASH memory via a memory mapped register file which is contained in the QSPIC. All instructions supported by the FLASH memory, can be programmed using the above register file.

A special feature of the QSPIC enables for automated re-initialization of the FLASH device right after power up, without the CPU being involved, thus reducing initialization time and consequently power dissipation. A small initialization memory of 16 32-bit retainable words, contains an encoded sequence of commands which are shifted in to the FLASH memory right after waking up from power down modes.

### Features

- SPI Modes:  
Single: Data transfer via two unidirectional pins.  
Dual: Data transfer via two bidirectional pins.  
Quad: Data transfer via four bidirectional pins.
- Auto Mode: up-to 32 Mbyte transparent Code access for XIP (Execute In Place) and Data access with 3-byte and 4-byte addressing modes.
- Manual Mode: Direct register access using the QSPIC register file.
- Up-to 96 MHz QSPI clock. Clock modes 0 and 3. Master mode only.
- Vendor independent Instruction Sequencer.
- In Auto Mode the FLASH control signals are fully programmable.
- Support for single access and high performance burst mode in combination with the cache controller (in Auto Mode).
- Use of a special read instruction in the case of a specific (programmable) wrapping burst access.
- Erase suspend/resume to Support for Code and Data storage
- Hardware initialization state machine based on uCode commands.

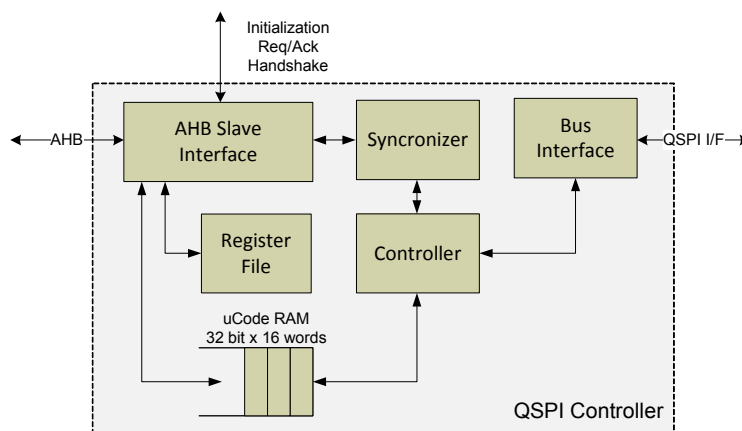


Figure 38: Quad SPI Controller architecture

### 12.1 ARCHITECTURE

The AHB slave block implements two AHB Slave interfaces which enable access to the register file and the uCode memory. The Controller implements all protocol related to the functionality of the FLASH memory. It contains a finite state machine (FSM) that generates all necessary signalling to the QSPI bus and realizes all features of the Auto mode operation. Moreover, it manages all data transfers between the two interfaces (the AHB and the QSPI).

The Bus Interface block controls the QSPI signals at the lowest level while the Synchronizer implements

"stretching" or "shortening" of the signals that cross the two clock domains.

The uCode memory is 16 words x 32 bits and contains the microcode for the initialization of the FLASH memory even before the CPU has been waken up. The signalling between the FIFO and the PMU is done through the request/acknowledge signals.

#### 12.1.1 Interface

The Quad SPI Controller uses the following signals:

- QSPI\_SCK: output serial clock

- QSPI\_CS: Active Low output Chip select.
- QSPI\_IO0:
  - DO (output) in Single SPI mode
  - IO0 (bidirectional) in Dual/Quad SPI mode.
- QSPI\_IO1:
  - DI (input) in Standard SPI mode
  - IO1 (bidirectional) in Dual/Quad SPI mode.
- QSPI\_IO2:
  - General purpose (output) (e.g. WPn Write Protect) in Standard SPI mode
  - IO2 (bidirectional) in Quad SPI mode.
- QSPI\_IO3:
  - General purpose (output) (e.g. HOLDn) in Single SPI mode
  - IO3 (bidirectional) at Quad SPI mode.
- The output drive of the pads is programmable via register bits QSPI\_GP\_REG[QSPI\_PADS\_DRV] and the slew via QSPI\_GP\_REG[QSPI\_PADS\_SLEW]
- The outputs pads have push-pull configuration and are supplied from VDDIO.

The Quad SPI Controller (QSPIC) drives all data pins constantly except for the case when a read is performed. The time for changing the direction of the pads is at least  $1.5 \times \text{QSPI\_CLK}$  (QSPI\_CLK being the clock that the FLASH operates on). In this way, data lines are always terminated thus reducing unnecessary power consumption.

The default state of the QSPI\_IOx pins is 1. This state is applied at the pins as soon as the QSPIC clock is enabled even if no access to the FLASH has yet been triggered. The value of the pins might be changed by programming the respective registers (QSPIC\_IOx\_DAT, QSPIC\_IOx\_OEN). This value will be valid only after the QSPI\_CS is pulled low, i.e. an access to the FLASH occurs.

### 12.1.2 Initialization FSM

Since the QSPIC is used in an ultra low power SoC, it is very possible that the FLASH memory will be either totally powered off, or set into deep power down mode when the system goes to any of the sleep modes. However, upon power-up or wake-up, the FLASH device requires a number of commands to get at a state where the CPU can actually execute code from. This initialization should be done prior to the CPU wakeup. The QSPIC contains a hardware state machine which decodes a number of commands in a 16 32-bit word retainable FIFO and initializes the FLASH automatically even before the CPU is waken up.

The command FIFO will be initialized upon cold boot of the system by the CPU with the commands residing in the OTP header. The start address of the RAM to be programmed with the uCode is 0x0C000040. The command encoding is presented in the following table:

**Table 25: Initialization Command Encoding**

| Bit                             | Name              | Description                                                                                                        |
|---------------------------------|-------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Byte 0</b>                   |                   |                                                                                                                    |
| 7:3                             | CMD_NBYTES        | The number of payload bytes to be sent                                                                             |
| 2:1                             | CMD_TX_MD         | QSPI bus mode when transmitting the command:<br>0x0: single SPI<br>0x1: Dual SPI<br>0x2: Quad SPI<br>0x3: Reserved |
| 0                               | CMD_VALID         | 1: the command record is valid<br>0: the command record is not valid                                               |
| <b>Byte 1</b>                   |                   |                                                                                                                    |
| 7:0                             | CMD_WT_CN<br>T_LS | Number of clock cycles to wait after applying the command (least significant byte)                                 |
| <b>Byte 2</b>                   |                   |                                                                                                                    |
| 7:0                             | CMD_WT_CN<br>T_MS | Number of clock cycles to wait after applying the command (most significant byte)                                  |
| <b>Byte 3 to (CMD_NBYTES+2)</b> |                   |                                                                                                                    |
|                                 |                   | The actual data bytes to be sent within a CS envelope                                                              |

The first byte (LSByte) in the word of the FIFO contains the flag of the command being valid or not, the bus mode of operation and the number of bytes contained in the payload to be sent.

The second and third byte define the amount of clock cycles that the QSPIC has to wait after applying the command. The clock to be used is the RC16 (~16 MHz), which results to a maximum of 4 ms waiting time. If more time is required by the FLASH, then multiple identical commands might be issued.

Example:

Considering 0xAB to be the opcode for releasing the FLASH from deep power down mode, the FIFO would be initialized with the following sequence:

**Table 26: FLASH Initialization uCode example**

| Byte | Value | Description                                               |
|------|-------|-----------------------------------------------------------|
| 0    | 0x11  | Valid command record, single SPI mode, 2 bytes of payload |
| 1    | 0x01  | 1 clock cycle wait after command is sent                  |
| 2    | 0x00  |                                                           |
| 3    | 0xAB  | Actual FLASH command opcode                               |



### 12.1.3 SPI modes

The Quad SPI Controller (QSPIC) supports the following SPI standards:

- Single: Data transfer via two unidirectional pins.

**Note 3:** The QSPIC supports communication to any single/dual or Quad SPI FLASH memory. Contrary to the Standard SPI interface, the supported Single SPI interface does **not** support the bus modes 1 and 2, does **not** support full-duplex communications and does **not** support any SPI slave mode.

- Dual: Data transfer via two bidirectional pins.
- Quad: Data transfer via four bidirectional pins.

### 12.1.4 Access modes

The access to a serial FLASH connected to the QSPI can be done in two modes:

- Auto mode
- Manual mode

These modes are **mutually exclusive**. The serial FLASH can be controlled only in one of the two modes. The registers which control the mode of operation can be used at any mode.

In auto mode, 3-bytes and 4-bytes addressing modes are supported. With QSPIC\_CTRLMODE\_REG [QSPIC\_USE\_32BA]=0, up to 16 MBytes QSPI (3-bytes addressing) can be accessed. If QSPIC\_USE\_32BA=1, the 4-bytes addressing is enabled for accessing up to 32 Mbyte QSPI FLASH.

#### Auto mode

In auto mode a read access from the serial FLASH memory is fully transparent to the CPU. A read access at the interface is translated by the QSPIC into the respective SPI bus control commands needed for the FLASH memory access.

When the Auto Mode is disabled, any access (reading or writing) will be ignored. When the Auto Mode is enabled, only read access is supported. A write access causes hard fault. A read access can be single access, incremental burst or wrapping burst. Wrapping burst is supported even when the FLASH device doesn't support any special instruction for wrapping burst. A special read instruction can be used in the case of a specific (programmable) wrapping burst access. When a FLASH supports a special instruction for wrapping burst access, this feature reduces access time (less wait states). For maximizing the utilization of the bus and minimizing the number of wait states, it is recommended to use burst accesses. However, non-sequential random accesses are supported with the cost of more wait states.

#### Manual Mode

In manual mode the FLASH memory is controlled via a register file. All instructions that are supported by a FLASH memory can be programmed using the register file. Moreover, the mode of interface (SPI, Dual SPI, Quad SPI) and the mode of operation (Auto or Manual

Mode) can be configured via this register file. The register file supports the following data sizes for reading and writing accesses: 8-bits, 16-bits and 32 bits.

### 12.1.5 Endianness

The QSPIC operates in little-endian mode. For 32-bit or 16 bit access (for read and write operations) to a serial FLASH memory, the least-significant byte comes first. For 32-bit access the byte ordering is: data [7:0], data [15:8], data [23:16], data [31:24] while for 16-bit access the byte ordering is: data [7:0], data [15:8].

### 12.1.6 Erase Suspend/Resume

The QSPI FLASH can be used for Data Storage, combining the EEPROM functionality + Program storage in one single device.

For this purpose the QSPI ERASE/SUSPEND ERASE RESUME commands are automatically executed as shown in [Figure 39](#)

To store data in QSPI FLASH, execution from QSPI must temporary be stopped by running directly from RAM or from a cached program part. The sector designated for storage must be erased first in case it contained data already.

The process is implemented in a HW FSM and consists of the following steps:

1. The controller is in Auto mode and read requests are served. The Erase procedure is initiated by setting QSPIC\_ERASE\_EN=1. The address of the sector that will be erased, is defined at QSPIC\_ERS\_ADDR. When an Erase procedure is requested, the controller jumps to state 2.
2. Read requests are still served. As soon as the Read requests stop (also possible due to late bus master change, e.g. DMA) and there is no any new Read request for a number of AHB clock cycles equal to QSPIC\_ERSRES\_HLD, then QSPIC\_WEN\_INST and QSPIC\_ERS\_INST instructions are sent to the FLASH. The QSPIC\_RESSUS\_DLY counter is started and the controller jumps to state 3.
3. Erasing is in progress in FLASH and the QSPI controller waits until one of the following events occur:
  - A status check request. This request can be forced by writing QSPIC\_CHCKERASE\_REG. The QSPI controller will then read the status of the FLASH memory and check if erasing has finished. Reading of the status is delayed by QSPIC\_RESSTS\_DLY cycles or by QSPIC\_RESSUS\_DLY cycles. The first is based on the clock of the SPI bus, while the latter on an internal 222 KHz clock. The selection between the two delays is configured by QSPIC\_STSDLY\_SEL bit. If erasing has finished, the QSPI controller returns to the normal operation (state 1) and sets QSPIC\_ERASE\_EN=0, otherwise it remains at

- state 3.
- A FLASH read data request on the AHB bus. The QSPI controller reads the status of the FLASH memory and checks if erasing is done. The reading of the status will be delayed again as in the previous case by QSPIC\_RESSUS\_DLY or QSPIC\_RESSTS\_DLY. If erasing has ended, the controller returns to normal operation (state 1) and sets QSPIC\_ERASE\_EN=0. The read request will be served as soon as the controller reaches state 1. If erasing has not ended, the controller proceeds to state 4.
- The QSPIC\_SUS\_INST is sent as soon as the QSPIC\_RESSUS\_DLY/QSPIC\_RESSTS\_DLY counter is 0. The controller jumps to state 5.
  - The controller polls the FLASH status register,
- until the FLASH device becomes ready (erasing is suspended). The controller will then proceed to state 6.
- The erasing process in the FLASH is now suspended and the controller may read the FLASH. The requested data are retrieved from the FLASH device. If the reading on the AHB stops (e.g. Cache hit) and there is no new Read requests for a number of AHB clock cycles equal to QSPIC\_ERSRES\_HLD, the controller goes to state 7.
  - The QSPIC\_RES\_INST instruction is applied and the controller jumps back to state 3. Also, the QSPIC\_RESSUS\_DLY counter is started. As result, the erase procedure is resumed.

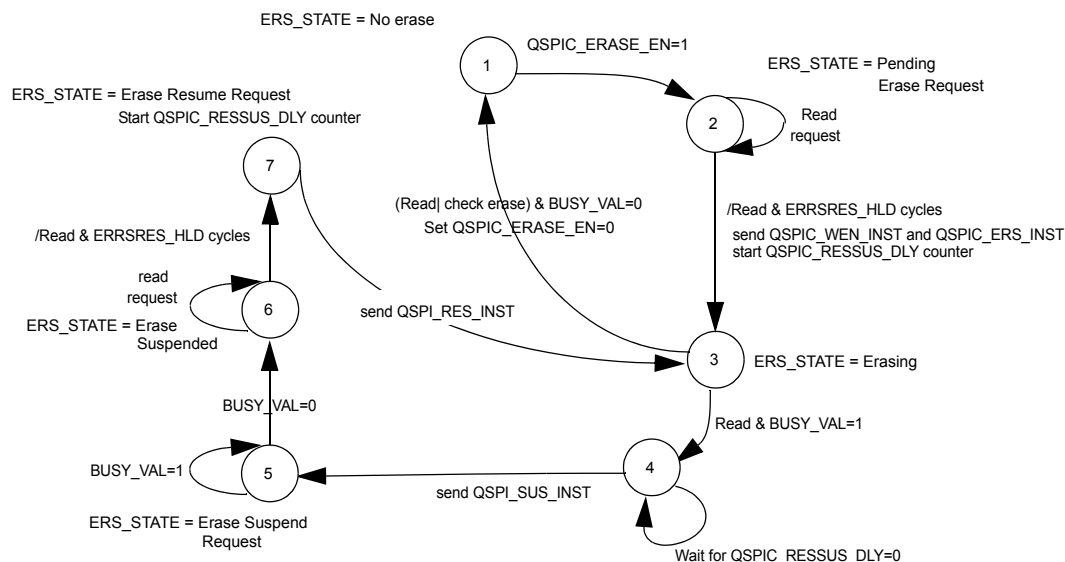


Figure 39: Erase Suspend/Resume in Auto mode

Refer to AN-D-185 for further Application information

Note that, QSPI\_RESSTS\_DLY counts QSPI\_CLK cycles, so before changing the QSPI\_CLK, make sure that QSPI\_RESSTS\_DLY is set large enough to meet the timing parameter requirements of the FLASH device used.

### 12.1.7 QSPI FLASH Programming

Programming the sectors is done in manual mode with polling status bit in the QSPI FLASH. During programming, the ARM Cortex™ M0 **must** run from cache without cache misses or from RAM. **Furthermore interrupts should be disabled while programming the FLASH.**

Byte programming is relatively short, so a polling loop could be acceptable to meet system latency require-

ments. Refer to vendor's datasheet.

## 12.2 PROGRAMMING

### 12.2.1 Auto Mode

#### Chip selection

In auto mode the QSPI executes from address 0. See ARM chapter remap function

#### Burst control phases

In the case of Auto Mode of operation the QSPIC generates a sequence of control signals in SPI BUS. This sequence of control signals is analysed to the following **phases**: instruction phase, address phase, extra byte phase, dummy clocks phase and read data phase. These phases can be programmed via registers

- QSPIC\_BURSTCMDA\_REG



- QSPIC\_BURSTCMBD\_REG.

Bits QSPIC\_INST are used to set the selected instruction for the cases of incremental burst or single read access. If bit QSPIC\_WRAP\_MD is equal to 1, bit QSPIC\_INST\_WB can be used to set the used instruction for the case of a wrapping burst read access of length and size described by the bits QSPIC\_WRAP\_LEN and QSPIC\_WRAP\_SIZE respectively. In all other cases the QSPIC\_INST is the selected instruction.

If the **instruction** must be transmitted only in the first access after the selection of Auto Mode, then the QSPIC\_INST\_MD must be equal to 1.

To enable the **extra byte phase** set QSPIC\_EXT\_BYTE\_EN=1 register. The transmitted byte during the extra byte phase is specified by the QSPIC\_EXT\_BYTE register. To disable (hi-Z) the output pads during the transmission of bits [3:0] of extra byte, set QSPIC\_EXT\_HF\_DS =1.

The number of **dummy bytes** during the dummy clocks phase is specified by register QSPIC\_DMY\_NUM and enabled by QSPIC\_DMY\_FORCE.

The SPI BUS mode during each phase can be set with register bits:

- QSPIC\_INST\_TX\_MD for the instruction phase
- QSPIC\_ADR\_TX\_MD for the address phase
- QSPIC\_EXT\_TX\_MD for the extra byte phase
- QSPIC\_DMY\_TX\_MD for the dummy byte phase
- QSPIC\_DAT\_RX\_MD for the read data phase.

If the Quad SPI mode is selected in any of the above phases, write 0 to the QSPIC\_IO3\_OEN and QSPIC\_IO2\_OEN.

If the FLASH Memory needs to be accessed with any instruction but the read instruction, then the **Manual Mode** must be used.

The final step to enable the use of Auto Mode of operation is to set the QSPIC\_AUTO\_MD equal to 1.

### 12.2.2 Manual Mode

For the Manual Mode QSPIC\_AUTO\_MD must be equal to zero.

Manual operation of the bus signals is done via QSPIC\_CTRLBUS\_REG:

- The start/end of an access can be controlled using bits QSPIC\_EN\_CS and QSPIC\_DIS\_CS respectively.
- The SPI bus mode of operation can be configured with bits QSPIC\_SET\_SINGLE, QSPIC\_SET\_DUAL and QSPIC\_SET\_QUAD.

Writing to QSPIC\_WRITEDATA register is generating a data transfer from the QSPIC to the SPI bus.

A read access at QSPIC\_READDATA register is gener-

ating a data transfer from the SPI bus.

Writing to QSPIC\_DUMMYDATA register is generating a number of dummy clock pulses to the SPI bus.

When access to the SPI bus via QSPIC\_WRITEDATA, QSPIC\_READDATA and QSPIC\_DUMMYDATA is very slow, most probably the delay in accessing the internal AHB is large. In this case, set the QSPIC\_HRDY\_MD register equal to 1 to increase priority when accessing the required registers. All masters of the SoC can access the AHB bus interface without waiting of the SPI Bus access completion. Polling of the QSPIC\_BUSY register must be done to check the end of the activity at the SPI bus, before issuing any more accesses. If a read transaction is finished, QSPIC\_RECVDATA contains the received data.

The state and the value of the QSPI\_IO[3:2] is specified with the following registers bits:

- QSPIC\_IO3\_OEN, QSPIC\_IO3\_DAT  
(Used for the WPn, Write Protect function).
- QSPIC\_IO2\_OEN, QSPIC\_IO2\_DAT respectively  
(Used for the HOLDn function).

### 12.2.3 Clock selection

The SPI clock mode as set with bit QSPIC\_CLK\_MD

The supported modes for the generated SPI clock is:

- 0 = Mode 0. The QSPI\_SCK is low, when the bus is idle (QSPI\_CS is high).
- 1 = Mode 3. The QSPI\_SCK is high, when the bus is idle (QSPI\_CS is high).

The QSPI\_CLK frequency has a programmable divider CLK\_AMBA\_REG[QSPI\_DIV] which divides either XTAL16 or PLL by 1,2,4,8.

This results in frequency ranges:

- In XTAL16 mode: between 2 MHz and 16 MHz
- In PLL mode: between 12 MHz and 96 MHz.

The QSPI\_CLK can be faster or slower than HCLK.

### 12.2.4 Received data

The standard method to sample received data is by using the positive edge of the QSPI\_SCK. However, when the output delay of the FLASH memory is high, a timing problem on the read path is very likely. For this reason the QSPIC can be programmed to sample the received data with the negative edge of the QSPI\_SCK. This is configured with the QSPIC\_RXD\_NEG register.

Furthermore the receive data can be pipelined by setting QSPI\_RPIPE\_EN=1 and the sampling clock can be delayed using QSPI\_PCLK\_MD. This enables sampling the received data later than the actual clock edge allows.

## 12.3 TIMING

This section contains timing diagrams for input and output signals

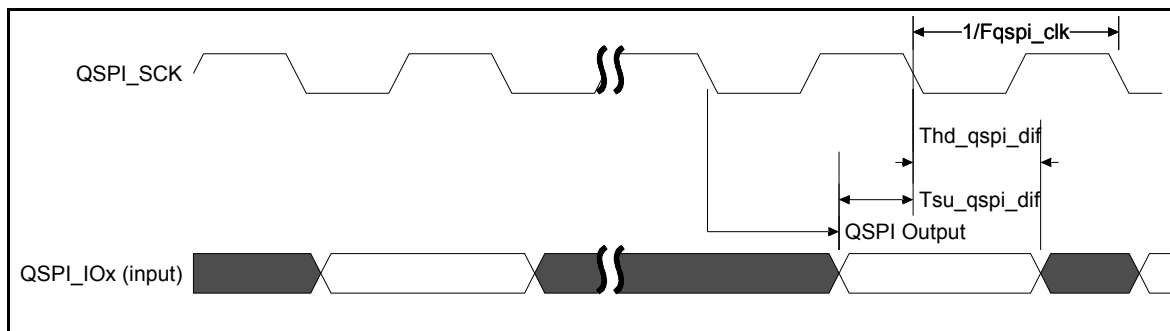


Figure 40: QSPI Input Timing

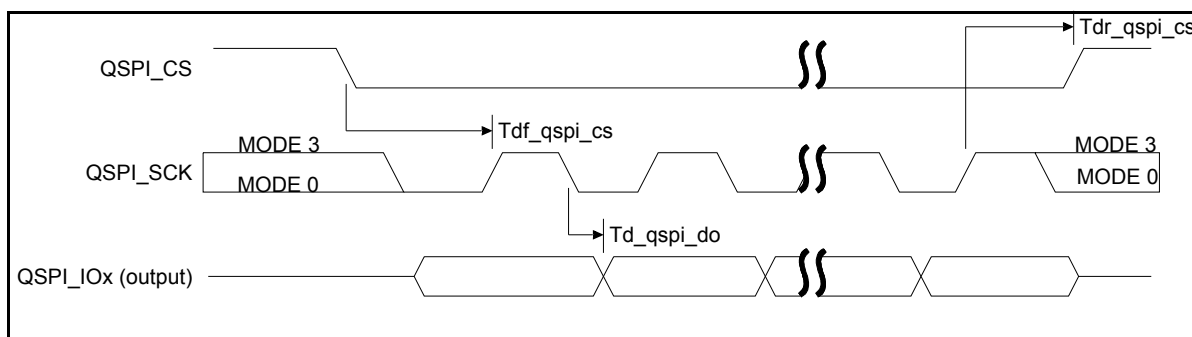


Figure 41: QSPI output Timing

Table 27: QSPI bus timing (VDDIO = 1.8V, Cload=15 pF)

| PARAMETER    | DESCRIPTION                                                                                           | CONDITIONS            | MIN               | TYP | MAX               | UNITS |
|--------------|-------------------------------------------------------------------------------------------------------|-----------------------|-------------------|-----|-------------------|-------|
| Fqspi_sck    | QSPI_SCK frequency                                                                                    |                       |                   |     | 96                | MHz   |
| Td_qspi_do   | Delay QSPI_SCK to QSPI_IOx                                                                            |                       | -1.4              |     | 3                 | ns    |
| Tdr_qspi_cs  | Delay QSPI_SCK to QSPI_CS                                                                             | Tqspi_sck=1/Fqspi_sck | Tqspi_sck<br>-2.5 |     | Tqspi_sck<br>+2.0 | ns    |
| Tdf_qspi_cs  | Delay QSPI_CS to QSPI_SCK                                                                             | Tqspi_sck=1/Fqspi_sck | Tqspi_sck<br>-2.5 |     | Tqspi_sck<br>+3.5 | ns    |
| Tsu_qspi_dif | Setup time QSPI_IO to QSPI_SCK falling edge with variable readpipe sample clock delay QSPI_RPIPE_EN=1 | QSPIC_PCLK_MD=6       | 2.6               |     |                   | ns    |
| Thd_qspi_dif | Hold time QSPI_SCK falling edge to QSPI_IO with variable readpipe sample clock delay QSPI_RPIPE_EN=1  | QSPIC_PCLK_MD=6       | -0.3              |     |                   | ns    |

**Note 4:** Total Delay QSPI FLASH output + PCB delay + Tsux\_qspi\_dif < 1/Fqspi\_sck.

E.g Winbond output + PCB delay = 6ns, -> Tsux\_qspi\_dif < 1/96MHz - 7 = 3.41 -> QSPI\_PCLK\_MD= 6, QSPI\_RPIPE\_EN=1 is recommended value for all QSPI\_SCK frequencies and shall be set before the maximum frequency is applied

### 13 DMA Controller

The DMA controller has eight Direct Memory Access (DMA) channels for fast data transfers from/to SPI, UART, I2C, USB, PDM and ECC to/from any on-chip RAM.

The DMA controller off-loads the ARM interrupt rate if an interrupt is given after a number of transfers.

More peripherals DMA requests are multiplexed on the 8 available channels, to increase utilization of the DMA service throughout the system.

#### Features

- 8 channels with optional peripheral trigger
- Full 32 bit source and destination pointers.
- Flexible interrupt generation.
- Programmable length
- Flexible peripheral request per channel
- Option to initialize memory

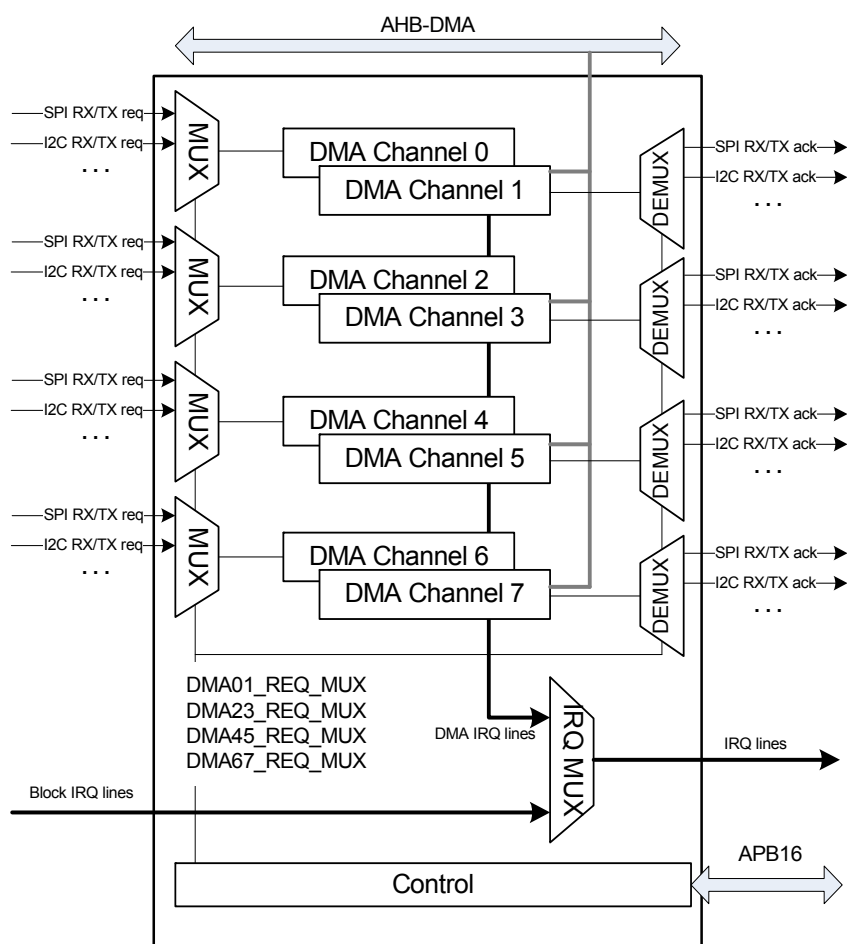


Figure 42: DMA controller block diagram

### 13.1 DMA PERIPHERALS

There is a list of peripherals that can request for a DMA service. The list is presented in the following table:

**Table 28: DMA served peripherals**

| Name   | Direction |
|--------|-----------|
| SPI    | RX        |
| SPI    | TX        |
| SPI2   | RX        |
| SPI2   | TX        |
| UART   | RX        |
| UART   | TX        |
| UART2  | RX        |
| UART2  | TX        |
| I2C    | RX        |
| I2C    | TX        |
| I2C2   | RX        |
| I2C2   | TX        |
| USB_FS | RX        |
| USB_FS | TX        |
| ADC    | Read      |
| PCM    | RX        |
| PCM    | TX        |
| SRC    | RX        |
| SRC    | TX        |

ECC can also be served by the DMA but will not be requesting for data as other peripherals.

Please note that for the TX DMA transfers to UART/UART2 and I2C/I2C2, it is required that  $pclk = hclk$  (i.e.  $CLK\_AMBA\_REG[PCLK\_DIV]=0x0$ ) and that the TX FIFO threshold level is not set to the highest value available where applicable.

### 13.2 INPUT/OUTPUT MULTIPLEXER

The multiplexing of peripheral requests is controlled by `DMA_REQ_MUX_REG`. Thus, if `DMA_REQ_MUX_REG[DMAxy_SEL]` is set to a certain (non-reserved) value, the TX/RX request from the corresponding peripheral will be routed to DMA channels y (TX request) and x (RX request) respectively.

Similarly, an acknowledging de-multiplexing mechanism is applied.

However, when two or more bit-fields (peripheral selectors) of `DMA_REQ_MUX_REG` have the same value, the lesser significant selector will be given priority (see also the register's description).

### 13.3 DMA CHANNEL OPERATION

A DMA channel is switched on with bit **DMA\_ON**. This bit is automatically reset if the dma transfer is finished. The DMA channels can either be triggered by software or by a peripheral DMA request. If **DREQ\_MODE** is 0, then a DMA channel is immediately triggered. If **DREQ\_MODE** is 1 the DMA channel can be triggered by a hardware interrupt.

If DMA starts, data is transferred from address **DMAx\_A\_START\_REG** to address **DMAx\_B\_START\_REG** for a length of **DMAx\_LEN\_REG**, which can be 8, 16 or 32 bits wide. The address increment is implemented using a 16-bit index counter (**DMAx\_IDX\_REG**), initialized to 0 when the transfer starts. This register is increased by 1 at the end of each DMA cycle and is then compared to **DMAx\_LEN\_REG**, to determine the transfer's completion. It is then automatically reset to 0 again.

Based on this register, the DMA engine forms the source/destination address at each DMA cycle, by adding it to **DMA\_A/B\_START\_ADDR**, after shifting it by 1 (when **DMAx\_CTRL\_REG[BW]=0x1**) or by 2 (when **DMAx\_CTRL\_REG[BW]=0x2**).

It also noted that, **AINC/BINC** must be set to '0' for source/destination register access.

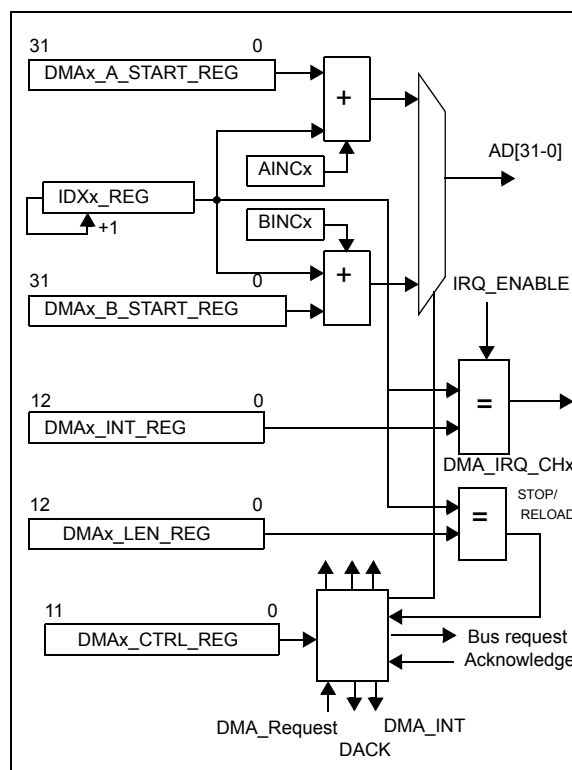


**Figure 43: DMA channel diagram**

If at the end of a DMA cycle, the DMA start condition is still true, the DMA continues. The DMA stops if **DREQ\_MODE** is low or if **DMAx\_LEN\_REG** is equal to the internal index register. This condition also clears

the **DMA\_ON** bit.

If bit **CIRCULAR** is set to 1, the DMA controller automatically resets the internal index registers and continues from its starting address without intervention of the ARM Cortex™ M0. If the DMA controller is started with **DREQ\_MODE** = 0, the DMA will always stop, regardless of the state of **CIRCULAR**.

Each DMA channel can generate an interrupt if **DMAx\_INT\_REG** is equal to **DMAx\_IDX\_REG**. After the transfer and before **DMAx\_IDX\_REG** is incremented, the interrupt is generated.

Example: if **DMA\_x\_INT\_REG**=0 and **DMA\_x\_LEN\_REG**=0, there will be one transfer and an interrupt.

fers (**DREQ\_MODE**='0') cannot be interrupted. Thus, in that case, the corresponding DMA channels will be frozen after any on-going Memory-to-Memory transfer is completed.

### 13.4 DMA ARBITRATION

The priority level of a DMA channel can be set with bits **DMA\_PRIO[2-0]**. These bits determine which DMA channel will be activated in case more than one DMA channel requests DMA. If two or more channels have the same priority, an inherent priority applies, (see register description).

With **DREQ\_MODE** = 0, a DMA can be interrupted by a channel with a higher priority if the **DMA\_IDLE** bit is set. **DMA\_IDLE** is a don't care if **DREQ\_MODE** = 1.

When **DMA\_INIT** is set, however, the DMA channel currently performing the transfer locks the bus and cannot be interrupted by any other channel, until the transfer is completed, regardless if **DMA\_IDLE** is set. The purpose of **DMA\_INIT** is to initialize a specific memory block with a certain value, fetched also from memory, without any interruption from other active DMA channels that may request the bus at the same time. Consequently, it should be used only for memory initialization, while when the DMA transfers data to/from peripherals, it should be set to '0'. Note also that, when **DMA\_INIT** is enabled, **BINC** should be set to '1', while **AINC** is don't care, as the DMA performs a single read in this mode.

It should be noted that memory initialization could also be performed without having the **DMA\_INIT** enabled and by simply setting **AINC** to '0' and **BINC** to '1', provided that the source address memory value will not change during the transfer. However, it is not guaranteed that the DMA transfer will not be interrupted by other channels of higher priority, when these request access to the bus at the same time.

### 13.5 FREEZING DMA CHANNELS

Each channel of the DMA controller can be temporarily disabled by writing a 1 to freeze all channels at **SET\_FREEZE\_REG**.

To enable the channels again, a 1 to bits at the **RESET\_FREEZE\_REG** must be written.

There is no hardware protection from erroneous programming of the DMA registers.

It is noted that the on-going Memory-to-Memory trans-

## 14 AES/Hash Engine

The Crypto engine aims to accelerate the algorithm calculations that are needed in order to implement the RFC4835. It implements AES in ECB, CBC and CTR modes. It also comprises HASH functions (SHA-1, SHA224/256/384/512, MD5). It supports AES128, AES 256 as well as HMAC-SHA-256 authentication protocol.

The AES/HASH engine uses a DMA engine for transferring encrypted/decrypted data to a shared memory in the AHB bus. The control registers of the IP are connected to the AHB bus.

The AES/HASH engine gives more flexibility to the way input data can be provided to the module. A calculation can be applied on fragmented input data and not on data residing at a specific memory space, by means of successive register programming in the internal DMA engine.

### Features

- AES (Advanced Encryption Standard) with 128, 192 or 256 bits key cryptographic algorithm.
- HASH functions: MD5, SHA-1, SHA224/256/384/512 bits
- Modes of operation
  - ECB (Electronic Code Book)
  - CBC (Cipher Block Chaining)
  - CTR (Counter)
- AHB Master DMA machine for data manipulation.
- AHB Slave register file for configuration.

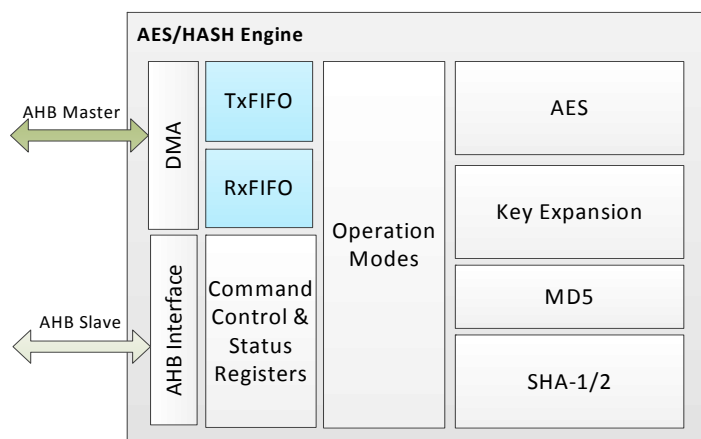


Figure 44: AES/HASH Architecture

### 14.1 ARCHITECTURE

#### 14.1.1 AES/HASH engine

The architectural view of the AES/HASH engine is the following:

The AES/HASH includes a DMA engine (AHB Master Interface) for transferring data between the IP and a sheared memory. The control registers of the AES/HASH are connected to AHB interface (AHB Slave interface).

The “Modes” controls the AES by implementing the respective mode that the selected encryption algorithms will operate each time. Also “Modes” communicates with DMA via two FIFO’s (Rx FIFO and Tx FIFO) which isolate the operation of the AES/HASH IP from the current status of the AHB-AMBA bus and also enable parallel transmission of data in bursts. By using burst transmission, the bus is utilized better because the bus access requests are reduced.

The “Ctrl FSM” block checks the FIFO’s and DMA status continuously and decides for the amount of data traffic, plus which of the FIFO’s will be used. Also decides the “switching off” of the AES/HASH after transferring all results to the memory.

The “HASH” block contains all the logic required for the realization of the hash algorithms calculations as well as circuitry for the padding of data. It also contains glue logic for the transfer of the results to the “Modes” block.

#### 14.1.2 AES

This part of the architecture implements the AES algorithm describing in the AES-FIPS PUB 197. The capabilities that offer are the encryption and decryption of 128 bits data blocks by using 128, 192 or 256 bits encryption key.



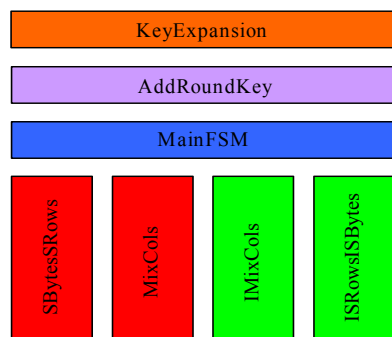


Figure 45: AES Architecture

The internal structure of the AES correlates with the logic function of the AES algorithm.

- **KeyExpansion:** The “KeyExpansion” is the process of generating the number of keys based on the initial key. More specific generates 11, 13 and 15 keys from an initial key of 128, 192 and 256 bits respectively. Each round of the algorithm uses each one of the above keys. For the encryption of each 128 bits input we need to use all generated, from this process, keys.
- **AddRoundKey:** Adds (modulo 2) the intermediate status that the input data already transformed (128 bits) with one of the generated (from “KeyExpansion”) keys. The output of this module contains the result that produced from the application of all the transformations that take place for the current round of the algorithm.
- **SBytesSRRows:** When encryption is taking place the module applies the SubBytes transformation first and then the ShiftRows transformation on the input data.
- **ISRowsISBytes:** When decryption is taking place this module applies the Inverse ShiftRows transformation and then the Inverse SubBytes on the input data.
- **MixCols:** Being used for encryption and implements the MixColumns transformation.
- **IMixCols:** Being used for decryption and implements the Inverse MixColumns transformation.
- **MainFSM:** The basic FSM that controls all previous modules. In general controls the complete AES encryption/decryption.

All parts of AES is implemented using hardware including the Key Expansion part.

#### 14.1.3 Modes

The block “Modes” uses the AES in order to implement the following modes of operations:

- ECB (Electronic Code Book)

- CBC (Cipher Block Chaining)
- CTR (Counter)

Padding requirements of the algorithms, to convert all data to multiples of 16 bytes (for AES), must be addressed by software.

By applying successive programming of AES-CBC encryptions using software, the realization of the HMAC-XCBC-AES-96 algorithm is possible.

The implementation of the AES-CCM is feasible just like the implementation of AES-CTR algorithms for encryption, and AES-CBC for authentication.

#### 14.1.4 HASH

The structure of the HASH block is presented in the following figure:



Figure 46: HASH block diagram

INPP applies padding at the input data as required by the hash algorithms. Two types of padding are implemented, due to the different algorithms supported. The purpose is to ensure that the message is a multiple of 512 bits or 1024 bits depending on the algorithm. Padding is done in a similar way in both cases. After the last data byte, one extra byte of value 0x80 is added. Next, a number of bytes (0x00) is added so that the overall size of the data block (including the extra bytes) mod 512/1024 is 448/896 depending on the algorithm. Following that, a 64/128-bits big-endian number is attached which represents the size of the data block, in bits (without the padding). While in this process, TX/RX FIFOs are switched into 8-bytes mode.

OUTPP packetizes the algorithm result (128 to 512 bits) into blocks of 64 bytes so that they can be shifted to the TX FIFO.

HASH FUNC contains the logic implementing the following hash algorithms:

1. MD5: RFC1321
2. SHA-1: FIPS PUB 180-4
3. SHA-224/256: FIPS PUB180-4. In this case only initialization changes.
4. SHA-384/512: FIPS PUB 180-4

As depicted in the following figure, HASH FUNC comprises common and specific resources for all algorithms.



All registers are contained in the common resources.  
Also, 2 32-bit adders utilized by all hash algorithms are part of the common resources.

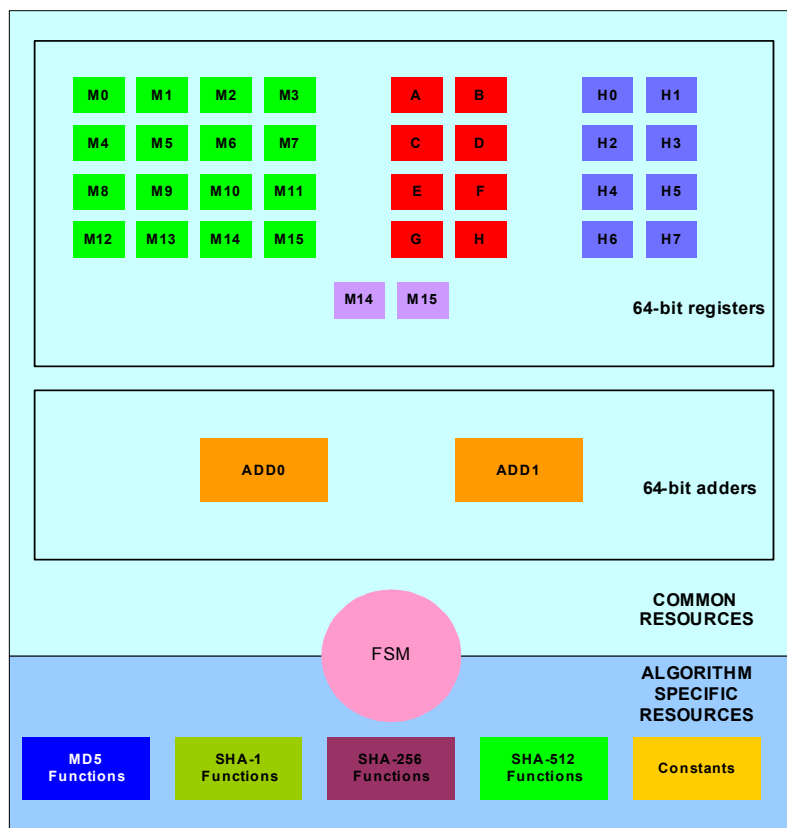


Figure 47: HASH FUNC architecture

## 14.2 PROGRAMMING

The basic register for the programming of AES/HASH engine is the CRYPTO\_CTRL\_REG. Select the cryptographic algorithm by setting the CRYPTO\_ALG register. The mode of operation can be programmed by choosing the suitable value for the CRYPTO\_ALG\_MD register. When only the final block of the resulting data must be stored at the memory, the user should set the CRYPTO\_OUT\_MD=1. The encryption/decryption function is selected by programming the CRYPTO\_ENCDEC register. If the selected algorithm is the AES, the CRYPTO\_AES\_KEY\_SZ register should be used to set the key size of the algorithm. To generate an interrupt request at the end of the operation, CRYPTO\_IRQ\_EN=1 should be set.

Proportionally with the selected cryptographic algorithm and the selected mode of operation, read the CRYPTO MREGs table and program the suitable registers with the parameters of the selected algorithms.

Flag CRYPTO\_AES\_KEXP controls key expansion. With CRYPTO\_AES\_KEXP=1, key expansion will be performed by the dedicated hardware engine. Otherwise key expansion should be performed by software

and generated keys should be stored into CRYPTO\_KEYS memory.

In case of CRYPTO\_AES\_KEXP = 1, CRYPTO\_KEYS\_START should be programmed with the cipher key.

The source address is set by writing the CRYPTO\_FETCH\_ADDR\_REG register. The AES/HASH engine reads the input data from this memory location.

The destination address is set by writing the CRYPTO\_DEST\_ADDR\_REG register. The AES/HASH engine writes the output data to this memory location.

The calculation is started by setting CRYPTO\_START\_REG=1.

The end of calculation is indicated by the CRYPTO\_INACTIVE flag when CRYPTO\_MORE\_IN=0. If not, the processing of the current data input is denoted by CRYPTO\_WAIT\_FOR\_IN. When CRYPTO\_INACTIVE=1, the calculation is finished and the resulting data are in the memory. When

CRYPTO\_WAIT\_FOR\_IN=1 more data are to be processed. In both cases, if CRYPTO\_IRQ\_EN=1, an interrupt request is generated.

To clear an interrupt request from the AES/HASH engine, CRYPTO\_CLRIRQ=1 should be programmed.

For the hash functions activation, the CRYPTO\_HASH\_SEL field in the CRYPTO\_CTRL\_REG has to be set. When this bit is set, the CRYPTO\_ALG field refers to the hash algorithms rather than the encryptions algorithms. CRYPTO\_ALG\_MD selects between HASH algorithms based on 32 or 64-bit arithmetic. Table 29 shows the programming selection of the hash algorithms. With means of the CRYPTO\_HASH\_OUT\_LEN, the number of bytes to be eventually used is defined. This number is programmable ranging from 1 to 64.

**Table 29: Hash function selection**

| CRYPTO_ALG_MD | CRYPTO_ALG | Hash algorithm |
|---------------|------------|----------------|
| 00            | 00         | MD5            |
| 00            | 01         | SHA-1          |
| 00            | 10         | SHA-256/224    |
| 00            | 11         | SHA-256        |
| 01            | 00         | SHA-384        |
| 01            | 01         | SHA-512        |
| 01            | 10         | SHA-512/224    |
| 01            | 11         | SHA-512/256    |

Note that there are some restrictions for the number of bytes to be processed (CRYPTO\_LEN value) that are related to the algorithm currently in use (CRYPTO\_HASH\_SEL and CRYPTO\_ALG), the mode of operation (CRYPTO\_ALG\_MD) and whether there are more data to be consumed (CRYPTO\_MORE\_IN) as depicted in the following table:

**Table 30: Restrictions on CRYPTO\_LEN**

| ALGORITHM | CRYPTO_HASH_SEL | CRYPTO_ALG_MD | CRYPTO_ALG | CRYPTO_LEN<br>CRYPTO_MORE_IN<br>= 0 | CRYPTO_LEN<br>CRYPTO_MORE_IN<br>= 1 |
|-----------|-----------------|---------------|------------|-------------------------------------|-------------------------------------|
| AES ECB   | 0               | 00            | 00         | multiple of 16                      | multiple of 16                      |
| AES ECB   |                 | 01            | 00         | multiple of 16                      | multiple of 16                      |
| AES CTR   |                 | 10            | 00         | no restriction                      | multiple of 16                      |
| AES CBC   |                 | 11            | 00         | no restriction                      | multiple of 16                      |

Table 30: Restrictions on CRYPTO\_LEN

| ALGORITHM   | CRYPTO_HASH_SEL | CRYPTO_ALG_MD | CRYPTO_ALG | CRYPTO_LEN<br>CRYPTO_MORE_IN<br>= 0 | CRYPTO_LEN<br>CRYPTO_MORE_IN<br>= 1 |
|-------------|-----------------|---------------|------------|-------------------------------------|-------------------------------------|
| MD5         | 1               | 00            | 00         | no restriction                      | multiple of 8                       |
| SHA-1       |                 | 00            | 01         | no restriction                      | multiple of 8                       |
| SHA-256/224 |                 | 00            | 10         | no restriction                      | multiple of 8                       |
| SHA-256     |                 | 00            | 11         | no restriction                      | multiple of 8                       |
| SHA-384     |                 | 01            | 00         | no restriction                      | multiple of 8                       |
| SHA-512     |                 | 01            | 01         | no restriction                      | multiple of 8                       |
| SHA-512/224 |                 | 01            | 10         | no restriction                      | multiple of 8                       |
| SHA-512/256 |                 | 01            | 11         | no restriction                      | multiple of 8                       |

The amount of clock cycles required to perform a HASH or an encryption/decryption/key expansion task, is presented in the following table:

Table 31: Latency of various crypto algorithms

| Algorithm   | Activity      | Clock Cycles / Block | Block Size (bits) | Clock Cycles for<br>256 kB (M) |
|-------------|---------------|----------------------|-------------------|--------------------------------|
| MD5         | Hash          | 231                  | 512               | 0.95                           |
| SHA-1       | Hash          | 281                  | 512               | 1.15                           |
| SHA-256/224 | Hash          | 423                  | 512               | 1.73                           |
| SHA-256     | Hash          | 423                  | 512               | 1.73                           |
| SHA-384     | Hash          | 565                  | 1024              | 1.16                           |
| SHA-512     | Hash          | 565                  | 1024              | 1.16                           |
| SHA-512/224 | Hash          | 565                  | 1024              | 1.16                           |
| SHA-512/256 | Hash          | 565                  | 1024              | 1.16                           |
| AES-128     | Key Expansion | 54                   |                   |                                |
| AES-128     | Encryption    | 86                   | 128               | 1.41                           |
| AES-192     | Key Expansion | 59                   |                   |                                |
| AES-192     | Encryption    | 101                  | 128               | 1.65                           |
| AES-256     | Key Expansion | 70                   |                   |                                |
| AES-256     | Encryption    | 117                  | 128               | 1.92                           |

## 15 ECC Engine

The ECC Engine is a very flexible block based on a 4x4 array of Dual-Field Processing Elements (DFPEs) that can be used to execute all operations & algorithms required for Elliptic Curve Cryptography systems such as Diffie-Hellman (ECDH) Key Exchange and Elliptic Curve Digital Signature Algorithm (ECDSA). The ECC engine is flexible enough to support any other crypto algorithm since it comprises a  $\mu$ -Code based controller which reads code from a dedicated RAM, making upgrades possible.

A powerful dedicated ALU consisting of 4 16x16 multipliers (DFPEs) combined with an effective pipelining scheme, allows for maximum throughput while preserving a small memory footprint. A Tightly Coupled Memory (TCM) assists in the intermediate results storage without creating traffic on the system's bus and memory.

The ECC engine comes with an APB interface for configuration and a DMA engine for data transactions to and from the DataRAM of the DA14680 without the interference of the CPU.

### Features

- RAM-based sequencer gives a maximum of flexibility and facilitates functional upgrades
- Supports arbitrary data/key sizes for ECC (up to 256 bits)
- Supports high-level PK Algorithms (ECDSA, ECDH, EdDSA)
- Low memory requirements:
  - 2 kBytes from system's DataRAM
  - 2 kBytes for the TCM
  - 3 kBytes for the  $\mu$ -Code RAM (retainable)
- High throughput: 90 256-bit ECC-point multiplications per second at 96 MHz

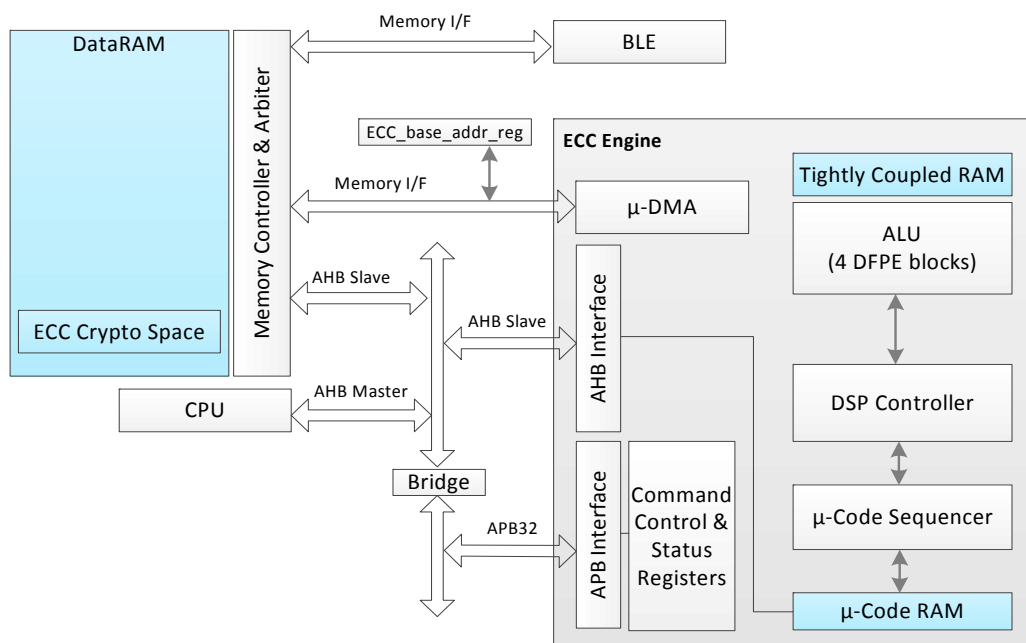


Figure 48: ECC Engine Block Diagram

### 15.1 ARCHITECTURE

The ECC controller comprises an AHB interface which is used for downloading the respective curve into the 3 kB  $\mu$ -Code RAM which can be retained during any Sleep mode (PMU\_CTRL\_REG[RETAIN\_ECCRAM]). The source of this  $\mu$ -Code might be the ECC Section of the OTP header (see Table 5) or any other RAM or FLASH location. The  $\mu$ -Code RAM base address is at 0x40030000. There is also, an APB32 interface which is used for accessing the block's registers at base

address 0x50006000. Finally, it includes an memory interface which directly connects to the memory controller of the DA14680 and uses a memory space with pre-defined base address as the buffer for data in and data out such as e.g. curve parameters and signature generation results. The pre-defined address is configured by ECC\_BASE\_ADDR\_REG which allocates memory in pages of 1 kB.

The DSP controller is responsible for the primitive calculation with help of the ALU and its dedicated 2 kB of

tightly coupled memory used to store intermediate variables.

### 15.1.1 Supported curves

Due to the flexible architecture any curve should be able to implement in  $\mu$ -Code. The currently available are:

- NIST recommended Curves:
  - Prime Field P-192, P-224, P-256
  - Binary Field K-163, K-233
  - Binary Field B-163, B-233
- Brainpool and SEC2 Curves
- Curve25519
- Average  $\mu$ -Code size for these curves is 3 kB.
- Average clock cycle amount for executing each curve is 100 k cycles

### 15.1.2 Supported high level algorithms

A number of cryptography algorithms involving other hardware resources than just the ECC are available as described in the following table, including their latency in system clock cycles:

**Table 32: ECC algorithms latency**

| Algorithm         | Routine         | ECC cycles (M) |
|-------------------|-----------------|----------------|
| JPAKE-p256        | step1_generate  | 4.5            |
|                   | step1_verify    | 4.5            |
|                   | step2_generate  | 2.3            |
|                   | step2_verify    | 2.3            |
|                   | seskey_generate | 2.3            |
| EdDSA - Ed25519   | pk_generate     | 0.7            |
|                   | sign_generate   | 0.7            |
|                   | sign_verify     | 1.4            |
| ECDH - Curve25519 | pk_generate     | 0.5            |
|                   | seskey_generate | 0.5            |
| ECDH - P256       | pk_generate     | 1.1            |
|                   | seskey_generate | 1.1            |
| ECDSA - P256      | pk_generate     | 1.1            |
|                   | sign_generate   | 1.15           |
|                   | sign_verify     | 2.3            |
| ECKCDSA - P256    | pl_generate     | 1.1            |
|                   | sign_generate   | 1.2            |
|                   | sign_verify     | 2.2            |

## 15.2 PROGRAMMING

In order to interact with the ECC several steps are necessary.

At first the ECC Crypto Module needs to get started and initialized:

- Enable clock to the ECC Crypto Module at CLK\_AMBA\_REG[ECC\_CLK\_ENABLE]
- Specify address space inside DataRAM to be used by the ECC block by programming the ECC\_BASE\_ADDR\_REG
- Load micro code into ECC  $\mu$ -Code RAM starting at address 0x40030000. The microcode itself can be located in the OTP header inside the Elliptic curve content section (ECS), or in the FLASH
- In case the interrupt is to be used, the corresponding IRQ vector needs to be provided and the IRQ needs to get enabled both in the Interrupt controller and the ECC block by programming

Before executing any arithmetic operation, all required parameters, operands and data must be written in the shared memory whose address is defined at ECC\_BASE\_ADDR\_REG.

Depending on the operations to execute, some parameters or operands (like the prime modulus N or P) must be located in pre-defined/fixed addresses while some other input/output operands and results can be passed to/from programmable addresses by using specific pointers (ECC\_OPPTRA, ECCOPPTRB and ECC\_OPPTRC) in the configuration register ECC\_CONFIG\_REG.

The programming flow is illustrated in [Figure 49](#)

Please note that the size of the individual operands must not exceed 256bit. However operand widths of 64bit as well as 128bit are also supported. In the arithmetic unit all operands are executed on blocks whose length is a multiple of 64bit. If needed input data that are written in memory must be extended with zeroes to get the right length (defined size of operands). Data is stored in the crypto memory following the little endian format.

### 15.2.1 Example: ECDSA signature generation

As explained in the previous section, some of the parameters or operands need to be available at predefined fixed addresses before starting the execution of the ECDSA signature generation. They all reside in the ECC crypto space in the DataRAM. The following table describes the actual sequence of the steps required and their respective commands.

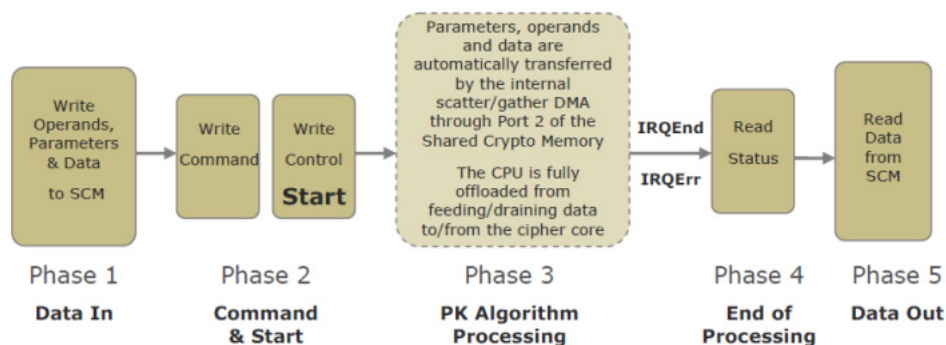


Figure 49: ECC programming flow

Table 33: ECDSA Signature Generation Process

| Step | Operation                                                                  | Command                                                                                                                                   |
|------|----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | select $k < n$                                                             | Use TRNG block                                                                                                                            |
| 2    | Compute $h = \text{SHA-1}(m)$<br>Store $h$                                 | Use HASH block                                                                                                                            |
| 3    | Compute $P_1 = k \cdot G$                                                  | $P_1(x_1, y_1) = k \cdot G(x_G, y_G) = \text{Point\_Mul}(k, G)$                                                                           |
| 4    | Compute $r = x_1 \bmod n$<br>If $r=0$ then go to Step 1                    | $r = \text{MODRED}(x_1, n)$<br>Check_r                                                                                                    |
| 5    | Compute $w = k^{-1} \bmod n$                                               | $w = \text{MODINV}(k, n)$                                                                                                                 |
| 6    | Compute $s = k^{-1} (h + d \cdot r) \bmod n$<br>If $s=0$ then go to Step 1 | $d \cdot r = \text{MODMUL}(d, r, n)$<br>$h + d \cdot r = \text{MODADD}(h, d \cdot r, n)$<br>$s = \text{MODMUL}(k^{-1}, h + d \cdot r, n)$ |
| 7    | $(r, s)$ is the signature for the message $m$                              |                                                                                                                                           |
|      |                                                                            | Check Status Register (ECC_STATUS_REG[ECC_BUSY])                                                                                          |

where:

- $G(x_G, y_G)$  is the generator point of the elliptic curve chosen,
- $n$  is the order of point  $G$
- $h$  is the 160 bits hash digest of the message  $m$
- $d$  is the private key ( $d < n$ )

The layout of the ECDSA signature generation in terms of the actual parameters and operands to be used by the u-Code is shown in the following table:

Table 34: ECC ECDSA Signature Generation memory layout

| Addr | Operand       | Note        |
|------|---------------|-------------|
| 0x0  | $p$ or $q$ or | Pre-defined |
| 0x1  | $n$           | Pre-defined |
| 0x2  | $x_G/G(x)$    | Pre-defined |
| 0x3  | $y_G/G(y)$    | Pre-defined |

Table 34: ECC ECDSA Signature Generation memory layout

| Addr | Operand                   | Note        |
|------|---------------------------|-------------|
| 0x4  | $a$                       | Pre-defined |
| 0x5  | $b$                       | Pre-defined |
| 0x6  | $d$ (private key)         | Pre-defined |
| 0x7  | $k$                       | Pre-defined |
| 0x8  | $x_Q/Q(x)$ Public Key     |             |
| 0x9  | $y_Q/Q(y)$ Public Key     |             |
| 0xA  | $r$                       | Result      |
| 0xB  | $s$                       | Result      |
| 0xC  | $h = \text{SHA-1}(m)$     | Pre-defined |
| 0xD  | $w$                       |             |
| 0xE  | $P_1(x) = (k \cdot G)(x)$ |             |
| 0xF  | $P_1(y) = (k \cdot G)(y)$ |             |

Parameters or operands noted as “Pre-defined” should be available at this specific pre-defined fixed address before generating the ECDSA Signature. The actual signature (r, s) is available at locations 0xA and 0xB.

The above memory layout represents the actual memory description of the shared ECC memory (part of the DataRAM).



## 16 True Random Number Generator (TRNG)

The TRNG is a non-deterministic Random Number generator used to provide the seed for encryption processes.

Its output can be used as entropy input for a FIPS 140-2 approved deterministic random number generation process which is handled by SW and HW accelerators of the DA14680.

The TRNG contains oscillator rings in digital logic which combined create metastability on a Flip-Flop eventually being the source of the entropy bits.

### Features

- Optional NIST SP800-90A Hash\_DRBG post processing using SHA-256 function using the on-chip HW accelerators
- Random numbers access through 32x32 bits FIFO on AHB bus
- Dedicated TRNG\_IRQ Interrupt line
- Start-up time 512 pclk cycles per 32 random bits
- Clock enable signals for optimal power saving

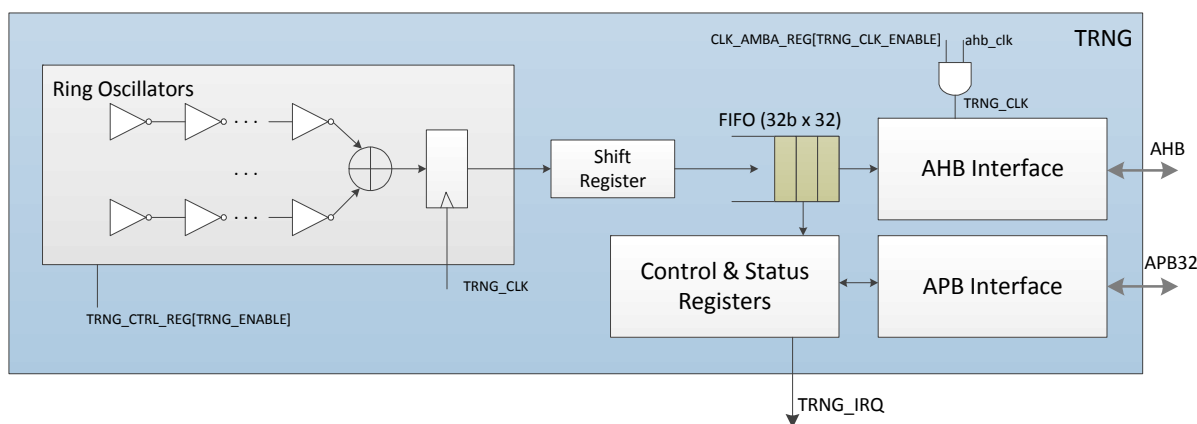


Figure 50: TRNG block diagram

### 16.1 ARCHITECTURE

Figure 50 shows the TRNG block diagram.

The TRNG comprises a number oscillator rings consisting of several inverters each. The output of the oscillator rings are accumulated in a shift register for whitening before being stored in a 32x32 bits deep FIFO. The oscillator rings are enabled when the TRNG\_CTRL\_REG[TRNG\_ENABLE] is set and as long as the FIFO is not full.

The 32-bit random numbers are accessible via an AHB interface while the registers are accessible via the APB32 interface.

### 16.2 PROGRAMMING

There is a simple sequence of steps that need to be followed to program the TRNG engine:

1. Set CLK\_AMBA\_REG[TRNG\_CLK\_EN]=1 to enable the AHB bus access
2. Set TRNG\_CTRL\_REG[TRNG\_MODE]=0 to select the ring oscillators rather than a pseudo random generator
3. Set TRNG\_CTRL\_REG[TRNG\_ENABLE]=1 to start the random number generation. This signal is ignored when the FIFO is already full

4. Poll TRNG\_FIFOLVL\_REG which provides the amount of data in the FIFO, or wait for the TRNG\_IRQ
5. Read the random number from the TRNG FIFO at address 0x40040000 (TRNG\_M)
6. To save power, set TRNG\_CLK\_EN=0 and set TRNG\_ENABLE=0. The FIFO can only be accessed if TRNG\_CLK\_EN=1

Note that, all signals are handled following the little-endian format. That means that the Least Significant Byte (LSB) is stored at the lowest address.

#### 16.2.1 Latency

After TRNG\_CTRL\_REG[TRNG\_ENABLE] is set to 1, it takes 512 pclk clock cycles/FIFO entry. So for 32 FIFO entries  $512 \times 32 = 16K$  pclk clock cycles or 1 ms if the 16 MHz clock is used.

## 17 Temperature Sensor

The DA14680 has a built-in temperature sensor which is part of the charging circuit and protects the die against too high temperature. It can however be used as a temperature sensor even if the charging circuit is not activated.

### Features

- Supply voltage 1.9 V - 3.6 V
- Temperature range -40 °C to 100 °C
- Uncalibrated (3 sigma) accuracy of +/- 17 °C
- Accuracy after 1 point calibration +/- 8.8 °C. (Assuming perfect reference temperature)
- Accuracy after 2 points calibration +/- 4 °C. (Assuming perfect reference temperature)

### 17.1 PROGRAMMING

The temperature sensor can be read out through the GPADC, even when the charger is off, by setting:

- CHARGER\_CTRL2\_REG[CHARGER\_TEST]=1
- GP\_ADC\_CTRL\_REG[GP\_ADC\_SEL]=14

The formulas which provide the relation between ADCx values (in LSBs) and the actual temperature Tx (in °C) are explained below.

For uncalibrated measurements:

- $T_x = (ADC_x - 712) / 2.44$

For one point calibration measurements:

- Measure  $T_{1p\_cal}$  with a thermometer, and read  $ADC_{1p\_cal}$
- $T_x = T_{1p\_cal} + (ADC_x - ADC_{1p\_cal}) / 2.44$

For two points calibration measurements:

- Measure  $T_{2p\_cal\_1}$  with a thermometer, and read  $ADC_{2p\_cal\_1}$
- Measure  $T_{2p\_cal\_2}$  with a thermometer, and read  $ADC_{2p\_cal\_2}$ . It is recommended that  $T_{2p\_cal\_2} - T_{2p\_cal\_1} > 40$  °C to achieve best accuracy.
- Calculate  $T_c = (ADC_{2p\_cal\_2} - ADC_{2p\_cal\_1}) / (T_{2p\_cal\_2} - T_{2p\_cal\_1})$
- $T_x = T_{2p\_cal\_1} + (ADC_x - ADC_{2p\_cal\_1}) / T_c$

Please note that, while measuring and/or calibration, the system's power dissipation should be kept the same, or else the measurement is affected by the internal thermal gradient.

## 18 Wakeup Timer

The Wakeup Timer can be programmed to wake up the DA14680 from power down mode after a pre-programmed number of GPIO events.

Each of the GPIO inputs can be selected to generate an event by programming the corresponding WKUP\_SELECT\_Px\_REG register. When all WKUP\_SELECT\_Px\_REG registers are configured to generate a wakeup interrupt, a toggle on any GPIO will wake up the system.

The input signal edge can be selected by programming the WKUP\_POL\_Px\_REG register.

The block diagram illustrating the Wake Up function is shown in Figure 51.

### Features

- Monitors any GPIO state change
- Implements debouncing time from 0 upto 63 ms
- Accumulates external events and compares the number to a programmed value
- Generates an interrupt to the CPU

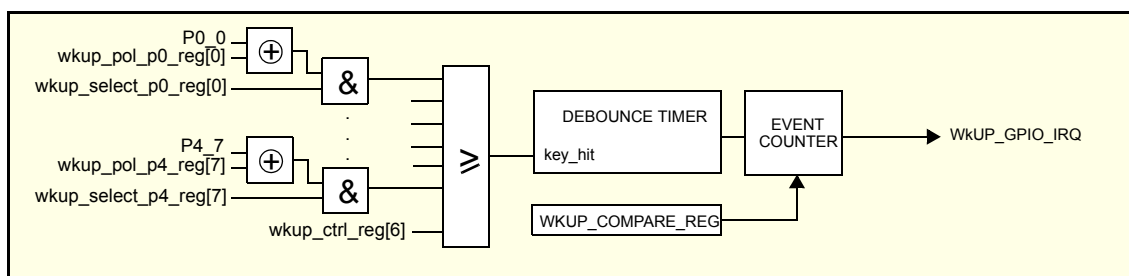


Figure 51: Wakeup Timer block diagram

A LOW to HIGH level transition on the selected input port, while WKUP\_POL\_Px\_REG[y] = 0, sets internal signal "key\_hit" to '1'. This signal triggers the event counter state machine as shown in Figure 52.

The debounce timer is loaded with value WKUP\_CTRL\_REG[WKUP\_DEB\_VALUE]. The timer counts down every 1 ms. If the timer reaches 0 and the "key\_hit" signal is still '1', the event counter will be incremented.

The event counter is edge sensitive. After detecting an active edge a reverse edge must be detected first before it goes back to the IDLE state and from there starts waiting for a new active edge.

If the event counter is equal to the value set in the WKUP\_COMPARE\_REG register, the counter will be reset and an interrupt will be generated, if it was enabled by WKUP\_CTRL\_REG[ENABLE\_IRQ].

The interrupt can be cleared by writing any value to register WKUP\_RESET\_IRQ\_REG.

The event counter can be reset by writing any value to register WKUP\_RESET\_CNTR\_REG.

The value of the event counter can be read at any time by reading register WKUP\_COUNTER\_REG.

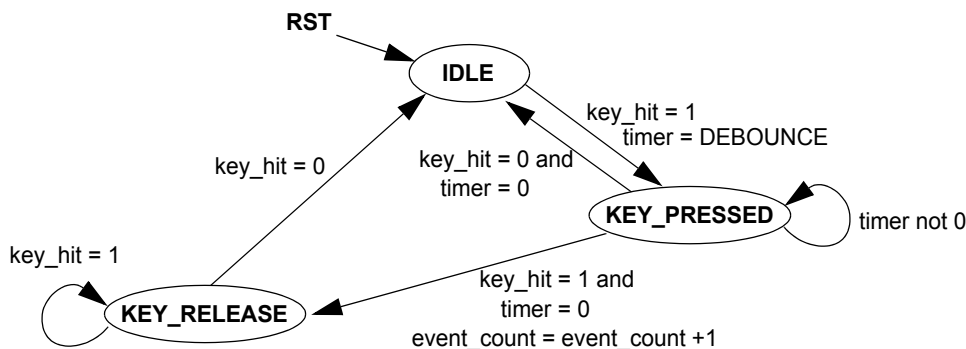


Figure 52: Event counter state machine for the Wakeup interrupt generator

The DA14680 is equipped with a high-speed ultra low power 10-bit general purpose Analog-to-Digital Converter (GPADC). It can operate in unipolar (single ended) mode as well as in bipolar (differential) mode. The ADC has its own voltage regulator (LDO) of 1.2 V, which represents the full scale reference voltage.

- 10-bit dynamic ADC
- Maximum sampling rate 4 Msample/s
- Ultra low power (5  $\mu$ A typical supply current at 100 ksamples/s)

- Single-ended as well as differential input with two input scales
- Eight single-ended or two differential external input channels
- Oversampling up to 128 steps providing effectively up to 11.2 bits precision (ENOB)
- Battery monitoring function
- Chopper function
- Offset and zero scale adjust
- Common-mode input level adjust
- DMA support

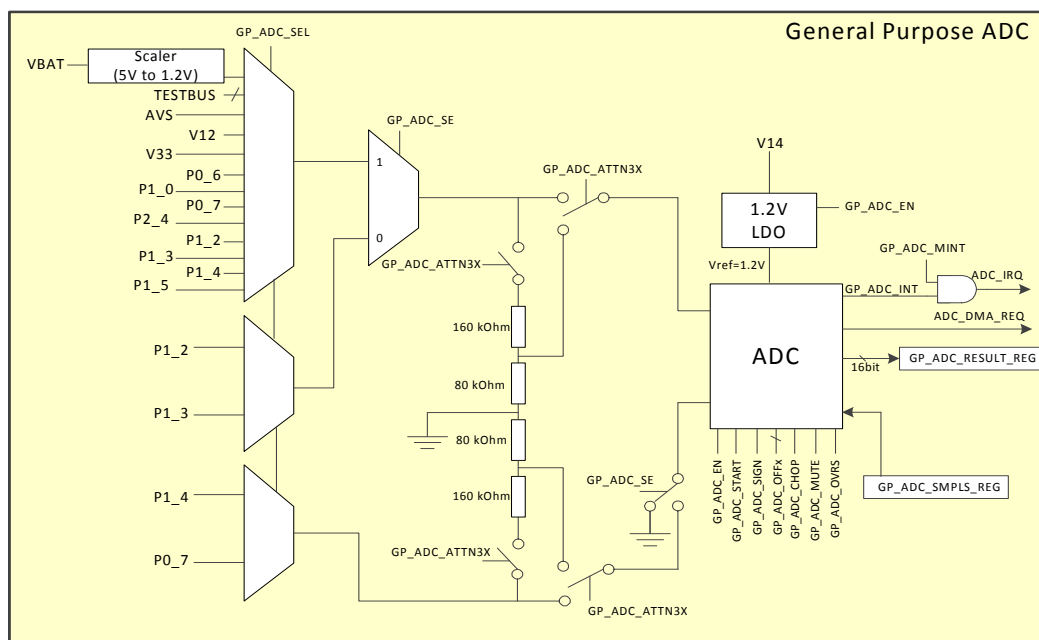


**Figure 53: Block diagram of the General Purpose ADC**

The ADC architecture shown in [Figure 53](#) has the following sub blocks:

- Analog to Digital converter (ADC)
  - ADC analog part internally clocked with 100 MHz (default) or the ADC\_CLK selectable with GP\_ADC\_CTRL\_REG[GP\_ADC\_CLK\_SEL]
  - ADC logic part clocked with the ADC\_CLK (16 MHz or 96 MHz) with CLK\_PER\_REG[ADC\_CLK\_SEL]
- 1.2V LDO for the ADC supply with a high PSRR enabled with GP\_ADC\_CTRL\_REG[GP\_ADC\_EN]
- APB Bus interface clocked with the APB clock. Control and status registers are available through registers GP\_ADC\_\*
- Maskable Interrupt (ADC\_IRQ) and DMA request

- ADC Input channels selector. Up-to eight specific GPIO ports, battery voltage (VBAT1) and the analog ground level (AVS) can be measured.

The ADC has the following modes of operation as shown in [Figure 54](#):

- Manual mode
  - Continuous mode
- In both modes the  $\Delta$  is increased by enabling:
- Oversampling mode
  - Chopper mode

In both modes the ADC performance might be increased by enabling:

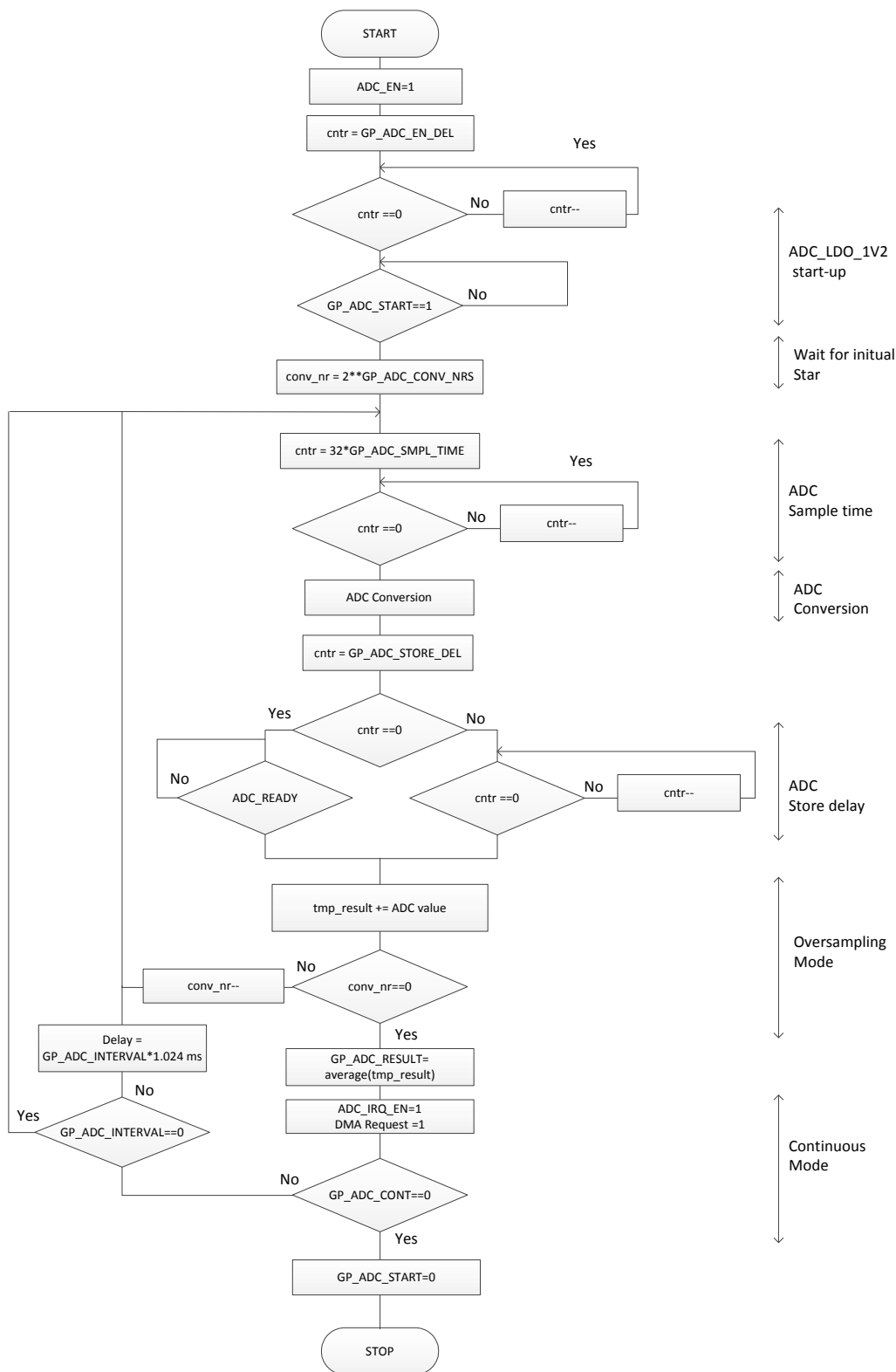


Figure 54: GPADC operation flow diagram

## 19.2 INPUT CHANNELS AND INPUT SCALE

The DA14680 has a multiplexer between the ADC and eight specific GPIO ports which can be sampled. Furthermore, the ADC can also be used to monitor the battery voltage (VBAT1) and the analog ground level (AVS).

Single-ended or differential operation for the external

channels is selected via bit GP\_ADC\_CTRL\_REG[GP\_ADC\_SE]. In differential mode the voltage difference between two GPIO input ports will be converted via bit GP\_ADC\_CTRL2\_REG[GP\_ADC\_ATT3X] the input scale can be enlarged by a factor of three, as summarized in Table 35.

**Table 35: GPADC input channels and voltage scale**

| GP_ADC_ATT3X | GP_ADC_SE | Input channels                                 | Input scale      | Input limits       |
|--------------|-----------|------------------------------------------------|------------------|--------------------|
| 0            | 1         | P0_6, P1_0, P0_7, P2_4, P1_2, P1_3, P1_4, P1_5 | 0 V to +1.2 V    | -0.1 V to +1.3 V   |
| 0            | 0         | [P1_2, P1_4], [P1_3, P0_7]                     | -1.2 V to +1.2 V | -1.3 V to +1.3 V   |
| 1            | 1         | P0_6, P1_0, P0_7, P2_4, P1_2, P1_3, P1_4, P1_5 | 0 V to +3.6 V    | -0.1 V to +3.45 V  |
| 1            | 0         | [P1_2, P1_4], [P1_3, P0_7]                     | -3.6 V to +3.6 V | -3.45 V to +3.45 V |

## 19.3 STARTING THE ADC

The GPADC is a dynamic ADC and consumes no static power, except for the LDO which consumes less than 5  $\mu$ A.

Enabling/disabling of the ADC is triggered by configuring bit GP\_ADC\_CTRL\_REG[GP\_ADC\_EN]. When set, first the LDO is enabled, then after the delay value set in GP\_ADC\_CTRL3\_REG[GP\_ADC\_EN\_DEL] (recommended value is 20 $\mu$ s to account for LDO settling time) the ADC will be enabled and an AD-conversion can be started.

See Table 36 for recommended values.

The ADC LDO consumes about 5 $\mu$ A, so GP\_ADC\_EN must be set to 0 if the ADC is not used.

**Table 36: ADC\_LDO\_1V2 start-up time**

| ADC_CLK (MHz) | GP_ADC_EN_DEL | Delay ( $\mu$ s) |
|---------------|---------------|------------------|
| 16            | 0x0B          | 22.0             |
| 96            | 0x40          | 21.3             |

## 19.4 ADC CONVERSION MODES

### 19.4.1 Manual Mode

Each conversion has two phases: the sampling phase and the conversion phase. When bit GP\_ADC\_CTRL\_REG[GP\_ADC\_EN] is set to '1', the ADC samples the selected input voltage. Writing a '1' at bit GP\_ADC\_CTRL\_REG[GP\_ADC\_START] ends the sampling phase and triggers the conversion phase. When the conversion is ready, the ADC resets bit GP\_ADC\_START to '0' and returns to the sampling phase asserting the interrupt line GP\_ADC\_INT. SW should always check that GP\_ADC\_START=0 before starting a new conversion.

The conversion itself is fast and takes approximately one clock cycle of 16 MHz, though the data handling will require several additional clock cycles, depending

on the software code style. The fastest code can handle the data in four clock cycles of 16 MHz, resulting to a highest sampling rate of 16 MHz/5 = 3.3 Msample/s.

At full speed the ADC consumes approximately 50  $\mu$ A. If the data rate is less than 100 ksample/s, the current consumption will be in the range of 5  $\mu$ A.

### 19.4.2 Continuous Mode

Setting GP\_ADC\_CTRL\_REG[GP\_ADC\_CONT] to '1' it is possible to use the ADC in a continuous mode meaning that a new conversion will be started after the previous conversion has finished without using the GP\_ADC\_START bit. Still the GP\_ADC\_START bit is needed to trigger the first conversion but following that, new converted data will be generated automatically. To correctly terminate this mode it is required to disable the GP\_ADC\_CONT bit first and afterwards wait until the GP\_ADC\_START bit is cleared so the ADC is in a defined state. By using GP\_ADC\_CTRL3\_REG[GP\_ADC\_INTERVAL] it is possible to determine the time interval between conversions. If kept zero, the conversion will be restarted immediately. With values different than zero, it is possible to program how many milliseconds it should take before restarting a new conversion. If GP\_ADC\_INTERVAL is not zero, it can take up to one millisecond before the first conversion is executed because it is synchronized to a one millisecond periodic signal.

## 19.5 NON-IDEAL EFFECTS

Besides Differential Non-Linearity (DNL) and Integral Non-Linearity (INL), each ADC has a gain error (linear) and an offset error (linear). The gain error of the GPADC slightly reduces the effective input scale (up to 50 mV). The offset error causes the effective input scale to become non-centred. The offset error of the GPADC is less than 20 mV and can be reduced by chopping or by offset calibration.

The ADC result will also include some noise. If the

input signal itself is noise free (inductive effects included), the average noise level will be  $\pm 1$  LSB. Taking more samples and calculating the average value will reduce the noise and increase the resolution.

With a 'perfect' input signal (e.g. if a filter capacitor is placed close to the input pin) most of the noise comes from the low-power voltage regulator (LDO) of the ADC. Since the DA14680 is targeted for ultra-compact applications, there is no pin available to add a capacitor at this voltage regulator output.

The dynamic current of the ADC causes extra noise at the regulator output. This noise can be reduced by setting bits GP\_ADC\_CTRL2\_REG[GP\_ADC\_I20U] and GP\_ADC\_CTRL2\_REG[GP\_ADC\_IDYN] to '1'. Bit GP\_ADC\_I20U enables a constant 20  $\mu$ A load current at the regulator output so that the current will not drop to zero. Bit GP\_ADC\_IDYN enables a 10  $\mu$ A load current during sampling phase so that the load current during sampling and conversion phase becomes approximately the same.

### 19.6 SAMPLING TIME (SMPL\_TIME)

The default ADC sampling time is sufficient in normal operation to achieve 10 bit accuracy. If ATTN3X is enabled or the VBAT channel is selected, additional (internal) resistance is routed in series with the sampling capacitor, therefore it requires more time to settle to the same accuracy. In general, enabling ATTN3X results in a time-constant  $t=60$  nsec, and sampling VBAT has a time-constant  $t=500$  nsec. The total sampling time required for a given accuracy can be calculated using:  $T_{\text{sample}} = -\ln(1.2/2^N) \cdot t$ , where N is the desired accuracy in bits and t the time-constant mentioned before. The corresponding value for the SMPL\_TIME can then easily be determined e.g. sampling VBAT with 10b accuracy and ADC\_CLK=16MHz requires SMPL\_TIME=2, or ATTN3X enabled requires SMPL\_TIME=1 for 10b accuracy when ADC\_CLK=16MHz. Obviously, when oversampling is used, the expected accuracy increases. Most of the times, setting SMPL\_TIME to "1" or "2" will be sufficient. If VBAT is selected, SMPL\_TIME=3 is the preferred value when oversampling (this gives 11b sampling accuracy).

### 19.7 OVERSAMPLING

In this mode multiple successive conversions will be executed and the results are added together to increase the effective number of bits (ENOB). The number of conversions that are executed is programmable at GP\_ADC\_CTRL2\_REG[GP\_ADC\_CONV\_NRS]. The six least significant bits inside the GP\_ADC\_RESULT\_REG can be discarded if no oversampling is used. But if, for example, four samples are programmed, two extra bits are generated and only the

four least significant bits can be discarded. T

**Table 37: Oversampling mode effective number of bits**

| Oversampling (GP_ADC_CONV_NRS) | Effective number of bits (ENOB) in GP_ADC_RESULT_REG |
|--------------------------------|------------------------------------------------------|
| 1                              | 9.05                                                 |
| 2                              | 9.45                                                 |
| 4                              | 9.83                                                 |
| 8                              | 10.21                                                |
| 16                             | 10.52                                                |
| 32                             | 10.85                                                |
| 64                             | 11.10                                                |
| 128                            | 11.27                                                |

The preferred settings for acquiring the results of Table 37 are the following:

**Table 38: Preferred settings for ENOB measurements**

| Description                                                       | Register Setting                     |
|-------------------------------------------------------------------|--------------------------------------|
| Run digital at PLL speed                                          | PLL_SYS_CTRL1_REG[PLL_EN]=1          |
| Use internal 100MHz clock as SAR clock.                           | GP_ADC_CTRL_REG[GP_ADC_CLK_SEL]=0    |
| Use auto zero and reference sampling in the LDO to suppress noise | GP_ADC_CTRL_REG[GP_ADC_LDO_ZERO]=1   |
| Disable chopping (default)                                        | GP_ADC_CTRL_REG[GP_ADC_CHOP]=0       |
| Sign is not inverted (default)                                    | GP_ADC_CTRL_REG[GP_ADC_SIGN]=0       |
| Measure single ended                                              | GP_ADC_CTRL_REG[GP_ADC_SE]=1         |
| Single-Ended: P1[2]                                               | GP_ADC_CTRL_REG[GP_ADC_SEL]=0        |
| Enable dynamic LDO current load                                   | GP_ADC_CTRL2_REG[GP_ADC_IDYN]=1      |
| Enable static LDO 20A current load                                | GP_ADC_CTRL2_REG[GP_ADC_I20U]=1      |
| 32 ADC clock cycles sampling time                                 | GP_ADC_CTRL2_REG[GP_ADC_SMPL_TIME]=1 |
| Negative Offset centred (zero offset)                             | GP_ADC_OFFN_REG=0x200                |
| Positive Offset centred (zero offset)                             | GP_ADC_OFFP_REG=0x200                |



### 19.8 CHOPPING

Chopping is a technique to cancel offset by taking two samples with opposite signal polarity. This method also smooths out other non-ideal effects and is recommended for DC and slowly changing signals.

Chopping is enabled by setting bit GP\_ADC\_CTRL\_REG[GP\_ADC\_CHOP] to '1'.

The mid-scale value of the ADC is the 'natural' zero point of the ADC (ADC result = 511.5 = 1FF or 200 Hex = 01.1111.1111 or 10.0000.0000 Bin). Ideally this corresponds to  $V_i = 1.2V/2 = 0.6 V$  in single-ended mode and  $V_i = 0.0 V$  in differential mode.

If bit GP\_ADC\_CTRL2\_REG[GP\_ADC\_ATTN3X] is set to '1', the zero point is 3 times higher (1.8 V single-ended and 0.0 V differential).

With bit GP\_ADC\_CTRL\_REG[GP\_ADC\_MUTE], the ADC input is switched to the centre scale input level, so the ADC result ideally is 511.5. If instead a value of 515 is observed, the output offset is +3.5 (adc\_off\_p = 3.5).

With bit GP\_ADC\_CTRL\_REG[GP\_ADC\_SIGN] the

sign of the ADC input and output is changed. Two sign changes have no effect on the signal path, though the sign of the ADC offset will change.

If  $\text{adc\_off\_p} = 3.5$  the ADC\_result with opposite GP\_ADC\_SIGN will be 508. The sum of these equals  $515 + 508 = 1023$ . This is the mid-scale value of an 11-bit ADC, so one extra bit due to the over-sampling by a factor of two.

The LSB of this 11-bit word should be ignored if a 10-bit word is preferred. In that case the result is 511.5, so the actual output value will be 511 or 512.

### 19.9 OFFSET CALIBRATION

A relative high offset caused by a very small dynamic comparator (up to 20 mV, so approximately 20 LSB).

This offset can be cancelled with the chopping function, but it still causes unwanted saturation effects at zero scale or full scale. With the GP\_ADC\_OFFP and GP\_ADC\_OFFN registers the offset can be compensated in the ADC network itself.

To calibrate the ADC follow the steps in [Table 39](#).

**Table 39: GPADC calibration procedure for single-ended and differential modes**

| Step                                                                                              | Single-ended mode (GP_ADC_SE = 1)                                                                | Differential mode (GP_ADC_SE = 0)                                                        |
|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| 1                                                                                                 | Set GP_ADC_OFFP = GP_ADC_OFFN = 0x200; GP_ADC_MUTE = 0x1; GP_ADC_SIGN = 0x0                      | Set GP_ADC_OFFP = GP_ADC_OFFN = 0x200; GP_ADC_MUTE = 0x1; GP_ADC_SIGN = 0x0              |
| 2                                                                                                 | Start conversion                                                                                 | Start conversion                                                                         |
| 3                                                                                                 | $\text{adc\_off\_p} = \text{GP\_ADC\_RESULT} - 0x200$                                            | $\text{adc\_off\_p} = \text{GP\_ADC\_RESULT} - 0x200$                                    |
| 4                                                                                                 | Set GP_ADC_SIGN = 0x1                                                                            | Set GP_ADC_SIGN = 0x1                                                                    |
| 5                                                                                                 | Start conversion                                                                                 | Start conversion                                                                         |
| 6                                                                                                 | $\text{adc\_off\_n} = \text{GP\_ADC\_RESULT} - 0x200$                                            | $\text{adc\_off\_n} = \text{GP\_ADC\_RESULT} - 0x200$                                    |
| 7                                                                                                 | GP_ADC_OFFP = $0x200 - 2 * \text{adc\_off\_p}$<br>GP_ADC_OFFN = $0x200 - 2 * \text{adc\_off\_n}$ | GP_ADC_OFFP = $0x200 - \text{adc\_off\_p}$<br>GP_ADC_OFFN = $0x200 - \text{adc\_off\_n}$ |
| <b>Note:</b> The average of GP_ADC_OFFP and GP_ADC_OFFN should be 0x200 (with a margin of 20 LSB) |                                                                                                  |                                                                                          |

It is recommended to implement the above calibration routine during the initialization phase of the DA14680. To verify the calibration results, check whether the GP\_ADC\_RESULT value is close to 0x200 while bit GP\_ADC\_MUTE = 1.

### 19.10 ZERO-SCALE ADJUSTMENT

The GP\_ADC\_OFFP and GP\_ADC\_OFFN registers can also be used to set the zero-scale or full-scale input level at a certain target value. For instance, they can be used to calibrate GP\_ADC\_RESULT to 0x000 at an input voltage of exactly 0.0 V, or to calibrate the zero scale of a sensor.

### 19.11 COMMON MODE ADJUSTMENT

The common mode level of the differential signal must be 0.6 V (or 1.8 V with GP\_ADC\_ATTN3X = 1). If the common mode input level of 0.6 V cannot be achieved, the common mode level of the GP\_ADC can be adjusted (the GP\_ADC can tolerate a common mode

margin up to 50 mV) according to [Table 40](#).

**Table 40: Common Mode adjustment**

| CM Voltage (V <sub>ccm</sub> ) | GP_ADC_OFFP = GP_ADC_OFFN |
|--------------------------------|---------------------------|
| 0.3 V                          | 0x300                     |
| 0.6 V                          | 0x200                     |
| 0.9 V                          | 0x100                     |

Any other common mode level between 0.0 V and 1.2 V can be calculated from the table above. Offset calibration can be combined with common mode adjustment by replacing the "0x200" value in the offset calibration routine by the value required to get the appropriate common mode level.

**Note:** The input voltage limits for the ADC in differential mode are: -1.3 V to +1.3 V (for GP\_ADC\_ATTN3X = 0,

see Table 35). The differential input range of the ADC is:  $-1.2\text{ V} < V[P0\_0, P0\_1] < +1.2\text{ V}$ . Therefore, if  $V_{\text{cm}} < 0.5\text{ V}$  or  $V_{\text{cm}} > 0.7\text{ V}$ , the input can no longer cover the whole ADC range.

#### 19.12 INPUT IMPEDANCE, INDUCTANCE AND INPUT SETTLING

The GPADC has no input buffer stage. During sampling phase a capacitor of 1 pF (0.5 pF in differential) is switched to the input line. The pre-charge of this capacitor is at mid-scale level so the input impedance is infinite.

At 100 ksample/s, zero or full-scale single-ended input signal, this sampling capacitor will load the input with:  
 $I_{\text{LOAD}} = V \cdot C \cdot f_S = \pm 0.6\text{ V} \cdot 1\text{ pF} \cdot 100\text{ kHz} = \pm 60\text{ nA}$   
 (differential:  $\pm 1.2\text{ V} \cdot 0.5\text{ pF} \cdot 100\text{ kHz} = \pm 60\text{ nA}$  at both pins).

During sampling phase a certain settling time is required. A 10-bit accuracy requires at least 7 time constants of the output impedance of the input signal source and the 1 pF sampling capacitor. The conversion time is approximately one clock cycle of 16 MHz (62.5 ns).

$$7 \cdot R_{\text{OUT}} \cdot 1\text{ pF} - 62.5\text{ ns} < 1/f_S$$

$$\Rightarrow R_{\text{OUT}} < (1 + 62.5\text{ ns} \cdot f_S) / (7 \cdot 1\text{ pF} \cdot f_S)$$

Examples:

$$R_{\text{OUT}} < 8.9\text{ M}\Omega \text{ at } f_S = 100\text{ kHz}$$

$$R_{\text{OUT}} < 890\text{ k}\Omega \text{ at } f_S = 1\text{ MHz}$$

The inductance from the signal source to the ADC input pin must be very small. Otherwise, filter capacitors are required from the input pins to ground (differential mode: from pin to pin).

To observe the noise level of the ADC and the voltage regulator, bit GP\_ADC\_CTRL\_REG[GP\_ADC\_MUTE] must be set to '1'. The noise should be less than  $\pm 1$  LSB on average, with occasionally a  $\pm 2$  LSB peak value. If a higher noise level is observed on the input channel(s), applying filter capacitor(s) will reduce the noise.

The 3x input attenuator is realized with a resistor divider network. When bit GP\_ADC\_CTRL\_REG2[GP\_ADC\_ATTN3X] is set to '1', the input impedance of the selected ADC input channel becomes 300 k $\Omega$  (typical) instead of infinite.

Bluetooth Low Energy

## 20 Sample rate Converter (SRC)

The SRC is a HW accelerator used to convert the sample rate of audio samples between various interfaces. Its primary purpose is to directly connect PCM and PDM channels while converting the rate accordingly. It can provide up or down sampled streams to other interfaces like USB by means of the AMBA bus.

### Features

- Supported conversions:
  - SRC\_IN (24 bits) to SRC\_OUT (24 bits)
  - PDM\_IN (1bit) to SRC\_OUT (24 bits)
  - SRC\_IN (24 bits) to PDM\_OUT (1 bit)
- SRC\_IN, SRC\_OUT Sample rates 62.5 kHz to 16 MHz
- SNR > 100 dB
- Single Buffer I/O with DMA support
- Automatic mode to adjust sample rate to the applied frame sync (e.g. PCM\_FSC)
- Manual mode to generate interrupts at the programmed sample rate. Adjustment is done by SW based on buffer pointers drift (e.g for USB)
- SRC runs at 16 MHz

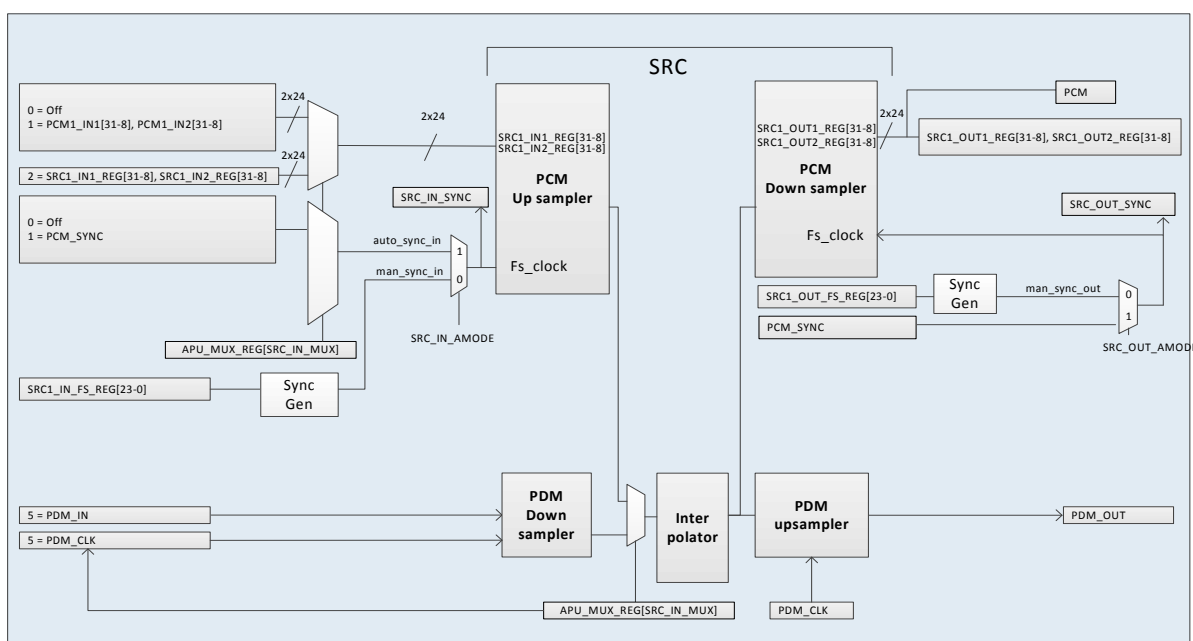


Figure 55: Sample rate Converter block diagram

### 20.1 ARCHITECTURE

#### 20.1.1 I/O channels

The SRC block converts two 24 bits channels either as a stereo pair or as two mono channels. The PCM linear data pairs are received on SRC\_IN and the output is 2x24 bits left aligned on SRC\_OUT. The two 1 bit PDM data inputs are received on PDM\_IN and are converted to 2x24 bits, left aligned to SRC\_OUT or PDM\_OUT.

#### 20.1.2 I/O multiplexers

The SRCx\_IN input multiplexer (see Figure 55) is controlled by APU\_MUX\_REG. The input of these multiplexers either come from the audio interfaces or from registers SRC\_IN1\_REG and SRC\_IN2\_REG. The data to these register is left aligned, bits 31-8 are mapped on bits 23-0 of the SRC.

The 24 bits SRCs outputs can be read in SRC\_OUT1\_REG and SRC\_OUT2\_REG and is also routed to the PCM. This input selection of these multiplexers is also controlled by APU\_MUX\_REG.

### 20.1.3 Input and Output Sample rate conversion

The SRC has a sample rate converter on the input and one at the output. Depending on the use case the converters operate in either manual or automatic conversion mode. This mode can be set in the SRC\_CTRL\_REG bits SRC\_IN\_AMODE and bit SRC\_OUT\_AMODE.

### 20.1.4 SRC conversion modes of operation

The SRC can operate in two mode of operation:

- Manual mode
- Automatic mode

In **manual mode** the sample rate to convert to is determined by the values in the SRC\_IN\_FS\_REG, SRC\_OUT\_FS\_REG.

Manual mode is used for sample rate conversion to/from the e.g. USB. The CPU compares the buffer pointers of the USB transmit buffers with the e.g. PDM receive buffer pointers. If the pointers drift, the PDM sample rate is adjusted in the SRC\_OUT\_FS\_REG. Hence, in the SRC\_OUT\_SYNC interrupt service routine, the PDM samples are read.

In **automatic** mode, the sample rate is adjusted according to the rate at which the samples are read from, or written to the SRC. For instance, if PCM slave data is transmitted to USB, PCM\_IN is selected in the SRC\_IN\_MUX. The SRC\_IN is set to automatic mode to convert sample receive at PCM\_FSC rate. The SRC\_OUT is also set to automatic mode. In the SRC\_OUT\_SYNC interrupt service routine, the PCM samples are read and the rate is determined by the main counter MAIN\_CNT (8-192kHz)

Typical use cases of the SRC are given in Table 41.

### 20.1.5 DMA operation

If more than one sample must be transfer to/from the CPU or the sample rate is so high that it interrupt the CPU too often, the DMA controller must be engaged to perform the transactions.

### 20.1.6 Interrupts

After a Sample Rate conversion the input upsampler and output down sampler generate edge triggered interrupts on SRC\_IN\_SYNC and SRC\_OUT\_SYNC to the CPU which do not have to be cleared. Note that only one sample shall be read from or written to a single register at a time (i.e. there are no FIFOs included).

### 20.1.7 SRC use cases

Table 41 shows typical use cases of the Sample Rate Converter.

**Table 41: Typical SRC use cases**

| Use case                                                         | SRCx_IN_AMODE<br>DATA path<br>SRCx_IN_SYNC  | SRCx_IN_SYNC<br>(out) | SRCx_OUT_AMODE<br>DATA path<br>SRCx_OUT_SYNC   | SRCx_OUT_SYNC<br>(out)          |
|------------------------------------------------------------------|---------------------------------------------|-----------------------|------------------------------------------------|---------------------------------|
| PCM_IN to USB                                                    | Automatic<br>PCMx_IN_DATA<br>PCMx_SYNC      | -                     | Manual<br>SRCx_OUT_REG                         | up-to 192kHz                    |
| PDM_IN to USB                                                    | Automatic<br>PDMx_IN_DATA<br>PCMx_SYNC      | -                     | Manual<br>SRCx_OUT_REG                         | up-to 192kHz                    |
| SPDIF_IN to USB                                                  | Automatic<br>SPDIF_IN_DATA<br>SPDIF_IN_SYNC | -                     | Manual<br>SRCx_OUT_REG                         | 48kHz (TYP)<br>(ISR adjusts FS) |
| USB to PCM_OUT                                                   | Manual<br>SRCx_IN_REG                       | up-to 192kHz          | Automatic<br>SRCx_OUT_REG<br>PCMx_FSC (output) | up-to 192kHz                    |
| <b>PCM to PCM resampler (output must be a multiple of 8 kHz)</b> |                                             |                       |                                                |                                 |
| PCM1_IN to<br>PCM2_OUT                                           | Automatic<br>PCM1_IN<br>PCM1_FSC (input)    |                       | Automatic<br>PCM2_OUT<br>PCM2_FSC (output)     | -                               |
| PCM2_IN to<br>PCM1_OUT                                           | Automatic<br>PCM2_IN<br>PCM2_FSC (input)    |                       | Automatic<br>PCM1_IN<br>PCM1_FSC (output)      | -                               |

## 21 PDM interface

The Pulse Density Modulation (PDM) interface provides a serial connection for up-to 2 input devices (e.g MEMS microphones) or output devices. The interfaces have a common clock PDM\_CLK and one input PDM\_DI which is capable of carrying two channels. [Figure 56](#) shows a typical connection of two microphone sharing one data line.

The PDM input data is a 1-bit data and is encoded so that the left channel is clocked in on the falling edge of PDM\_CLK and the right channel is clocked in on the rising edge of PDM\_CLK as shown in [Figure 57](#).

The 1 bits data stream is downsampled to 24 bits PCM samples in the HW Sample Rate converter (SRC) for further processing in the DSP.

The interface supports MEMS microphone sleep mode by disabling the PDM\_CLK.

The PDM interface signals are available through the PPA multiplexer. The interface levels are determined by the IO group on which the PDM signals are mapped which can be hard wired to 1.8 V and 3.3 V.

### Features

- PDM\_CLK output frequency  
62.5 kHz - 4 MHz
- Downsampling to 24 bits in SRC
- PDM\_CLK on/off to support Sleep mode
- PDM\_IN: 1 Channel in stereo format
- PDM\_OUT: 2 Channels in mono format,  
1 Channels in stereo format
- Programmable Left/Right channel selection

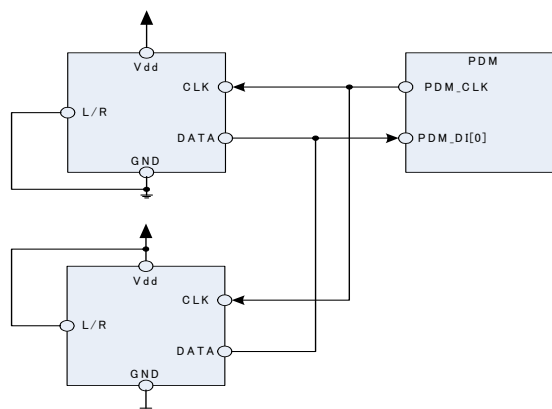


Figure 56: PDM with dual mic interface

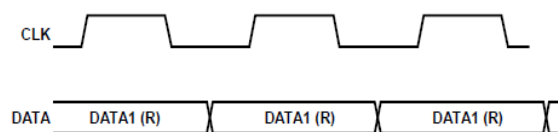


Figure 9. Mono PDM Format

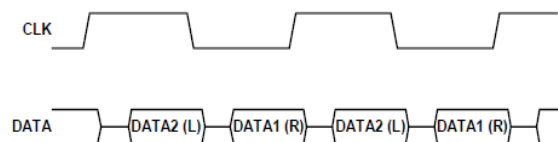


Figure 10. Stereo PDM Format

Figure 57: PDM formats

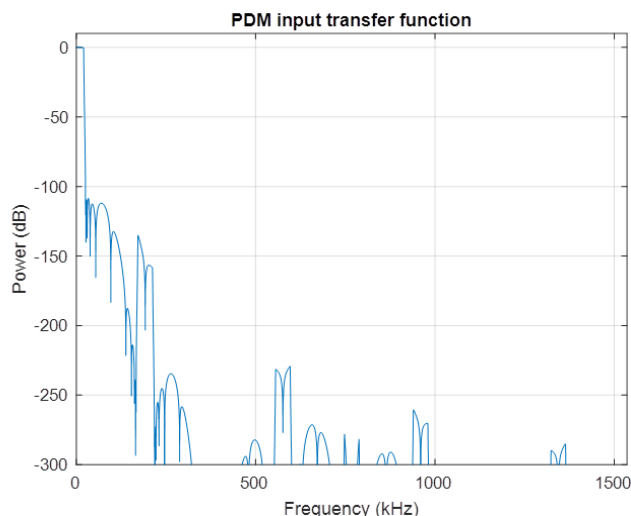


Figure 58: SRC PDM input transfer function

## 22 PCM Controller

The PCM controller is implementing an up-to 192kHz synchronous interface to external audio devices, ISDN circuits and serial data interfaces.

It is accessed through the APB32 interface. PCM can individually operate in master or slave mode. In slave mode, the phase between the external and internal frame sync can be measured and used to compensate for drift.

The data IO registers have DMA support in order to reduce the interrupt overhead to the CPU. Up-to 8 channels of 8 bits with a programmable delay are supported in received and transmit direction.

The controller supports PCM, I2S, TDM and IOM2 formats.

### Features

- PCM\_CLK Master/slave
- PCM\_FSC
- Master/slave 4 kHz to 96 kHz
- Strobe Length 1, 8, 16, 24, 32, 40, 48 and 64 bits
- PCM\_FSC before or on the first bit. (In Master mode)
- 2x32 channels
- Programmable slot delay up-to 31\*8bits
- Formats
  - PCM mode
  - I2S mode (Left/Right channel selection) with N\*8 for Left and N\*8 for Right
  - IOM2 mode (double clock per bit)
- Programmable clock and frame sync inversion
- Direct connection to Sample Rate Converter (SRC)
- Interrupt line to the CPU
- DMA support

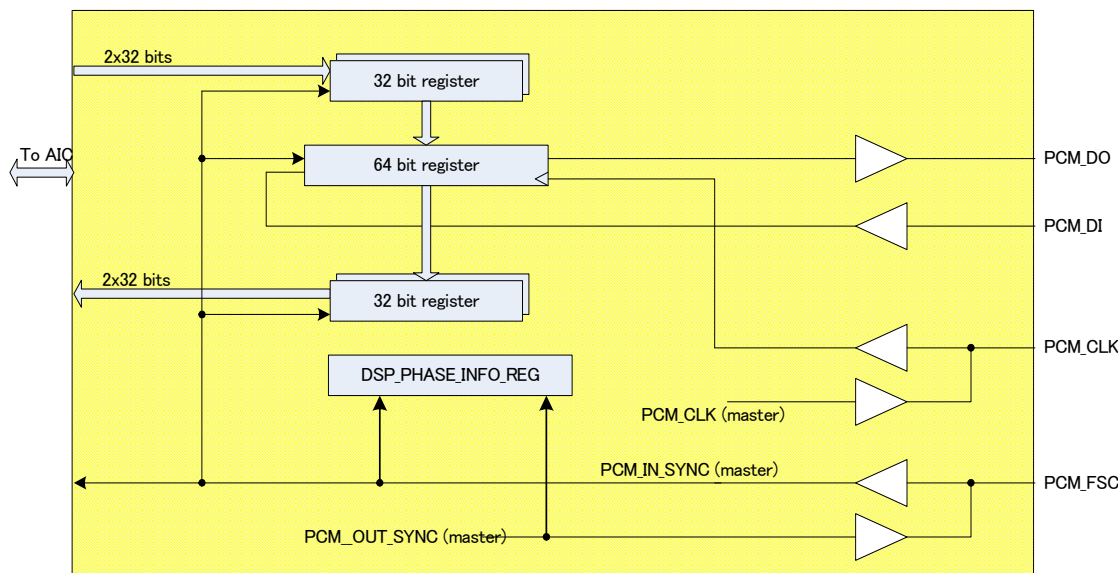


Figure 59: PCM Controller

### 22.1 ARCHITECTURE

#### 22.1.1 Interface Signals.

- PCM\_FSC, strobe signal input, output. Supports 8/16/32/48/96/128/192kHz. Can generate an interrupt to the CPU.
- PCM\_CLK, PCM clock input, output.
- PCM\_DO, PCM Data output, push pull or open drain with external pull-up resistor.
- PCM\_DI, PCM Data input.

PCM interface can be powered down by the `PCM_CTRL_REG[PCM_EN] = 0`.

#### 22.1.2 Channel ACCESS

The PCM interface has two 32-bit channels for TX and RX. Channels are accessed through 32 bits registers: `PCM1/2_OUT1_REG`, `PCM1/2_OUT2_REG`, `PCM1/2_IN1_REG` and `PCM1/2_IN2_REG`.

The registers are only word-wise (32 bits) accessible by the CPU or the DMA via the APB-32 bridge. The 32 bits registers are arranged as 8 channels of 8 bits, named channel 1 to channel 8.

By a flexible clock inversion, channel delay and strobe length adjustment various format like PCM, I2S, TDM and IOM2 can be made

### 22.1.3 Channel delay

The 8 PCM channels can be delayed with a maximum delay of 31x8bits using the bit field `PCMx_CTRL_REG[PCM_CH_DEL]`. Note that a high delay count in combination with a slow clock, can lead to the `PCMx_FSC` sync occurring before all channels are shifted in or out. The received bits of the current channel may not be properly aligned in that case.

### 22.1.4 Clock generation

Figure 60 shows the PCM clock generation block and Figure 61 the `PCM_CLK_DIV` value for given `PCM_FSC` and `PCM_CLK` in master mode. `PCM_CLK_DIV` is only 10 bits. For a higher division value, use either `DIV1` or `DIVN` must be selected.

The `PCM_CLK_DIV` calculation shows the following

use cases, with 16 bits, 32 bits, 48 bits, 64 bits.

The `PCM_FSC` will be synchronised to the main counter if `PCM_MAIN_SYNC` is '1'.

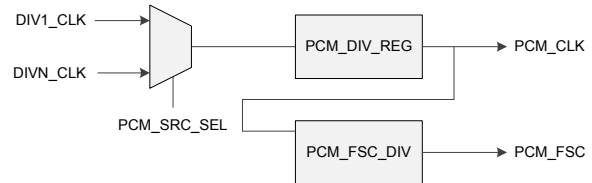


Figure 60: PCM clock generation

|             |      |                      | XTAL            |                |                 | PLL             |                |                 |
|-------------|------|----------------------|-----------------|----------------|-----------------|-----------------|----------------|-----------------|
|             |      |                      | 16000           |                |                 | 96000           |                |                 |
| sample rate | bits | desired bitclock khz | desired divider | actual divider | actual wordsize | desired divider | actual divider | actual wordsize |
| 8           | 1*8  | 64                   | 250             | 250            | 8               | 1500            | 1500           | 8               |
| 8           | 1*16 | 128                  | 125             | 125            | 16              | 750             | 750            | 16              |
| 8           | 1*24 | 192                  | 83.33333        | 80             | 25              | 500             | 500            | 24              |
| 8           | 1*32 | 256                  | 62.5            | 50             | 40              | 375             | 375            | 32              |
| 8           | 2*8  | 128                  | 125             | 125            | 8               | 750             | 750            | 8               |
| 8           | 2*16 | 256                  | 62.5            | 50             | 20              | 375             | 375            | 16              |
| 8           | 2*24 | 384                  | 41.66667        | 40             | 25              | 250             | 250            | 24              |
| 8           | 2*32 | 512                  | 31.25           | 25             | 40              | 187.5           | 150            | 40              |
| 16          | 1*8  | 128                  | 125             | 125            | 8               | 750             | 750            | 8               |
| 16          | 1*16 | 256                  | 62.5            | 50             | 20              | 375             | 375            | 16              |
| 16          | 1*24 | 384                  | 41.66667        | 40             | 25              | 250             | 250            | 24              |
| 16          | 1*32 | 512                  | 31.25           | 25             | 40              | 187.5           | 150            | 40              |
| 16          | 2*8  | 256                  | 62.5            | 50             | 10              | 375             | 375            | 8               |
| 16          | 2*16 | 512                  | 31.25           | 25             | 20              | 187.5           | 150            | 20              |
| 16          | 2*24 | 768                  | 20.83333        | 20             | 25              | 125             | 125            | 24              |
| 16          | 2*32 | 1024                 | 15.625          | 10             | 50              | 93.75           | 75             | 40              |
| 32          | 1*8  | 256                  | 62.5            | 50             | 10              | 375             | 375            | 8               |
| 32          | 1*16 | 512                  | 31.25           | 25             | 20              | 187.5           | 150            | 20              |
| 32          | 1*24 | 768                  | 20.83333        | 20             | 25              | 125             | 125            | 24              |
| 32          | 1*32 | 1024                 | 15.625          | 10             | 50              | 93.75           | 75             | 40              |
| 32          | 2*8  | 512                  | 31.25           | 25             | 10              | 187.5           | 150            | 10              |
| 32          | 2*16 | 1024                 | 15.625          | 10             | 25              | 93.75           | 75             | 20              |
| 32          | 2*24 | 1536                 | 10.41667        | 10             | 25              | 62.5            | 50             | 30              |
| 32          | 2*32 | 2048                 | 7.8125          | 5              | 50              | 46.875          | 25             | 60              |
| 48          | 1*8  | 384                  | 41.66667        | N/A            | N/A             | 250             | 250            | 8               |
| 48          | 1*16 | 768                  | 20.83333        | N/A            | N/A             | 125             | 125            | 16              |
| 48          | 1*24 | 1152                 | 13.88889        | N/A            | N/A             | 83.33333        | 80             | 25              |
| 48          | 1*32 | 1536                 | 10.41667        | N/A            | N/A             | 62.5            | 50             | 40              |
| 48          | 2*8  | 768                  | 20.83333        | N/A            | N/A             | 125             | 125            | 8               |
| 48          | 2*16 | 1536                 | 10.41667        | N/A            | N/A             | 62.5            | 50             | 20              |
| 48          | 2*24 | 2304                 | 6.944444        | N/A            | N/A             | 41.66667        | 40             | 25              |
| 48          | 2*32 | 3072                 | 5.208333        | N/A            | N/A             | 31.25           | 25             | 40              |

Figure 61: Integer `PCM_CLK_DIV` values for given `PLL_MAIN` frequencies and sample rates

Note, that in the yellow colored cases, `PCM_FSC` can not be 50% duty cycle while using the integer option for generating the `PCM_CLK`. However, this is feasible if

the fractional option is used (see `PCM_DIV_REG` and `PCM_FDIV_REG`).



## 22.1.5 DATA FORMATS

### 22.1.5.1 PCM master mode

Master mode is selected if `PCM_CTRL_REG[MAS-TER]` = 1.

In master mode `PCM_FSC` is output and falls always over Channel 0. The duration of `PCM_FSC` is programmable with `PCMx_CTRL_REG[PCM_FSCLEN]` = 1 or 8, 16, 24, 32 clock pulses high. The start position is programmable with `PCM_CTRL_REG[PCM_FSCDEL]` and can be placed before or on the first bit of channel 0. The repetition frequency of `PCM_FSC` is programmable in `PCMx_CTRL_REG[PCM_FSC_DIV]` to from 8-192kHz.

If master mode selected, `PCM_CLK` is output and provides one or two clocks per data bit programmable in `PCM_CTRL_REG[PCM_CLK_BIT]`.

The polarity of the signal can be inverted with bit

`PCM_CTRL_REG[PCM_CLKINV]`.

The `PCM_CLK` frequency selection is described in [section 22.1.4](#).

### 22.1.5.2 PCM Slave mode

In slave mode (bit `MASTER` = 0) `PCM_FSC` is input and determines the starting point of channel 0. The repetition rate of `PCM_FSC` must be equal to `PCM_SYNC` and must be high for at least one `PCM_CLK` cycle. Within one frame, `PCM_FSC` must be low for at least `PCM_CLK` cycle.

Bit `PCM_FSCDEL` sets the start position of `PCM_FSC` before or on the first bit (MSB).

In slave mode `PCM_CLK` is input. The minimum received frequency is 256 kHz, the maximum is 12.288 MHz.

In slave mode the main counter can be stopped and resumed on a `PCM1_FSC` or `PCM2_FSC` rising edge.



Figure 62: PCM interface formats

### 22.1.5.3 I2S formats

The digital audio interface supports I2S mode, Left Justified mode, Right Justified mode and TDM mode.

#### I2S mode

To support I2S mode, the MSB of the right channel is valid on the second rising edge of the bit clock after the rising edge of the PCM\_FSC, and the MSB of the left channel is valid on the second rising edge of the bit clock after the falling edge of the PCM\_FSC.

Settings for I2S mode:

- PCM\_FSC\_EDGE: 1 (all after PCM\_FSC)
- PCM\_FSCLEN: 4 (4x8 High, 4x8 Low)
- PCM\_FSC\_DEL: 0 (one bit delayed)

- PCM\_CLK\_INV: 1 (output on falling edge)
- PCM\_CH0\_DEL: 0 (no channel delay)

#### TDM mode

A time is specified from the normal 'start of frame' condition using register bits PCM\_CH0\_DEL. In the left justified TDM example illustrated in Figure 64, the left channel data is valid PCM\_CH0\_DEL clock cycles after the rising edge of the PCM\_FSC, and the right channel data is valid the same PCM\_CH0\_DEL number of clock cycles after the falling edge of the PCM\_FSC.

By delaying the channels, also left and right alignment can be achieved.

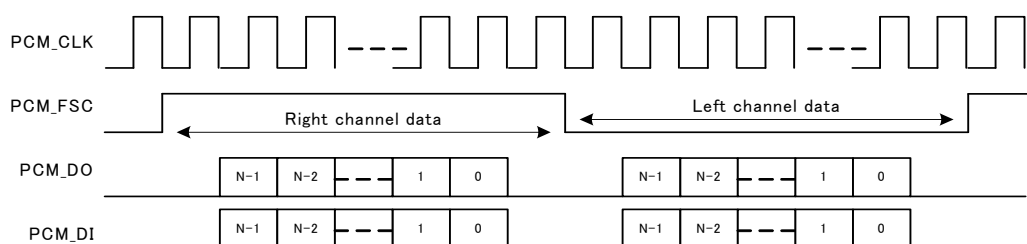


Figure 63: I2S Mode

Settings for TDM mode:

- PCM\_FSC\_EDGE: 1 (rising and falling PCM\_FSC)
- PCM\_FSCLEN: Master 1 to 4  
Slave waiting for edge.
- PCM\_FSC\_DEL: 1 (no bit delay)

- PCM\_CLK\_INV: 1 (output on falling edge)
- PCM\_CH0\_DEL: Slave 0-31 (channel delay)  
Master 1-3

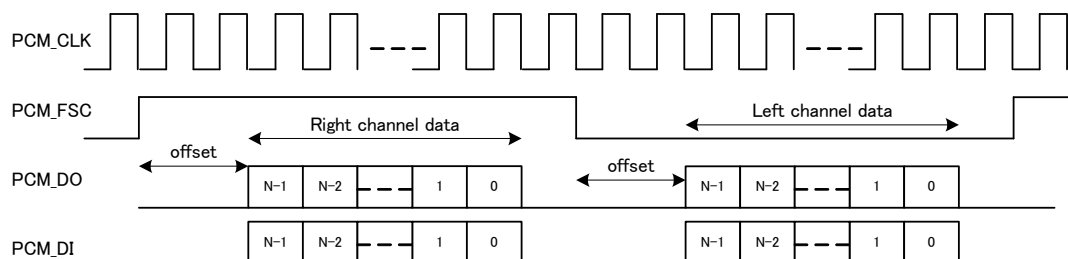


Figure 64: I2S TDM mode (left justified mode)

### 22.1.6 IOM mode

In the IOM format, the PCM\_CLK frequency is twice the data bit cell duration. In slave mode synchronization is on the first rising edge of PCM\_FSC while data is clock in on the second falling edge.

Settings for IOM mode:

- PCM\_FSC\_EDGE: 0 (rising edge PCM\_FSC)
- PCM\_FSCLEN: 0 (one cycle)

- PCM\_FSC\_DEL: 0 (no bit delay)
- PCM\_CLK\_INV: 0 (output on rising edge)
- PCM\_CH0\_DEL: 0 (no delay)
- PCM\_CLK\_BIT: 1

### 22.1.7 External synchronisation

With the PCM interface in slave mode, the PCM interface supports direct routing through the sample rate convertor (SRC). Any drift in PCM\_FSC or other frame

sync frequencies like 44.1 kHz can be directly resampled to e.g 48kHz internal sample rate.

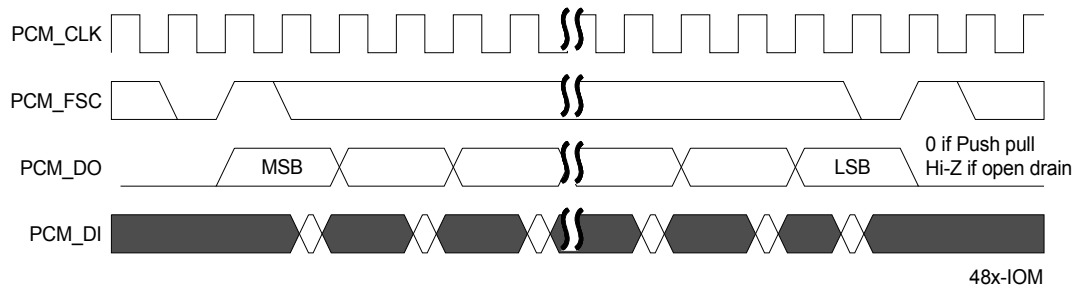


Figure 65: IOM format

## 23 UART

The DA14680 contains two instances of this block, i.e. UART and UART2.

The UART is compliant to the industry-standard 16550 and is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

There is DMA support on both UARTs. UART2 only supports hardware flow control signals (RTS, CTS) and includes a 16-byte FIFO while UART is not.

### Features

- 16 bytes Transmit and receive FIFOs. (UART2 only)
- Hardware flow control support (CTS/RTS) (UART2 only)

- Shadow registers to reduce software overhead and also include a software programmable reset
- Transmitter Holding Register Empty (THRE) interrupt mode
- IrDA 1.0 SIR mode supporting low power mode.
- Functionality based on the 16550 industry standard:
- Programmable character properties, such as number of data bits per character (5-8), optional
- parity bit (with odd or even select) and number of stop bits (1, 1.5 or 2)
- Line break generation and detection
- Prioritized interrupt identification
- Programmable serial data baud rate as calculated by the following:  $\text{baud rate} = (\text{serial clock frequency}) / (\text{divisor})$ .

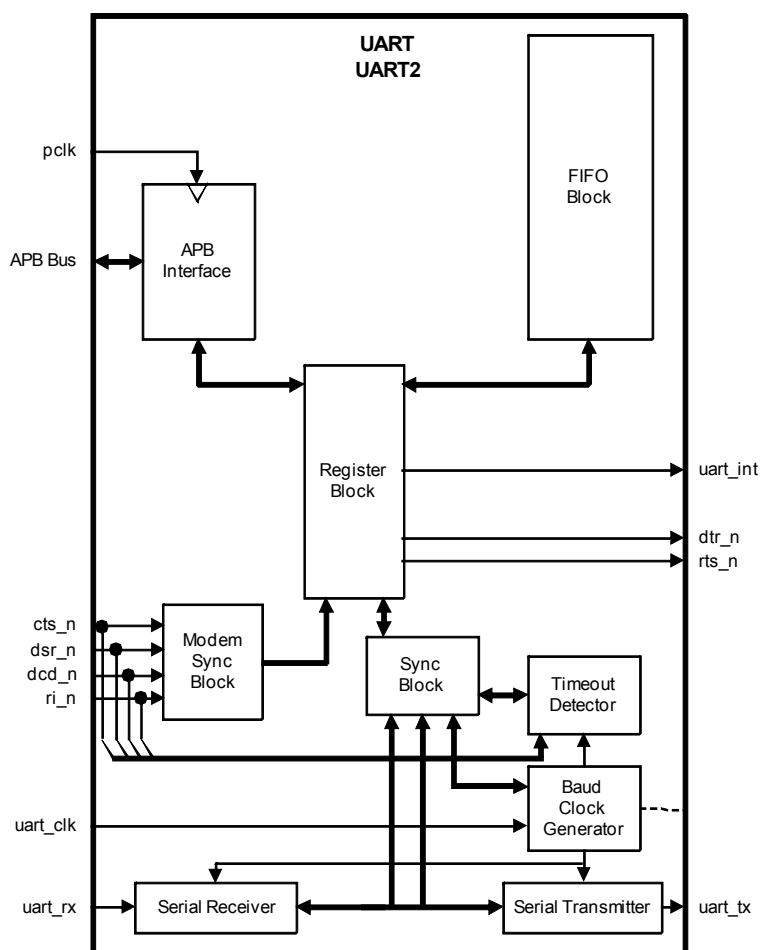


Figure 66: UART Blockdiagram

### 23.1 UART (RS232) SERIAL PROTOCOL

Because the serial communication between the UART and the selected device is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. Utilizing these bits allows

two devices to be synchronized. This structure of serial data accompanied by start and stop bits is referred to as a character, as shown in Figure 67:

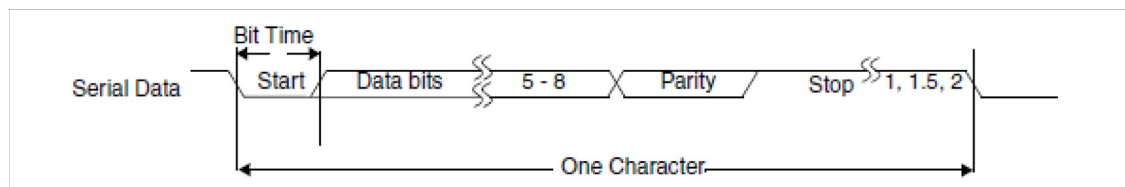


Figure 67: Serial Data Format

An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to provide the UART with the ability to perform simple error checking on the received data.

The UART Line Control Register (UART\_LCR\_REG) is used to control the serial character characteristics. The individual bits of the data word are sent after the start bit, starting with the least-significant bit (LSB). These are followed by the optional parity bit, followed by the stop bit(s), which can be 1, 1.5 or 2.

All the bits in the transmission (with exception to the half stop bit when 1.5 stop bits are used) are transmit-

ted for exactly the same time duration. This is referred to as a Bit Period or Bit Time. One BitTime equals 16 baud clocks. To ensure stability on the line the receiver samples the serial input data at approximately the mid point of the Bit Time once the start bit has been detected. As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid point for sampling is not difficult, that is every 16 baud clocks after the mid point sample of the start bit. Figure 68: shows the sampling points of the first couple of bits in a serial character.

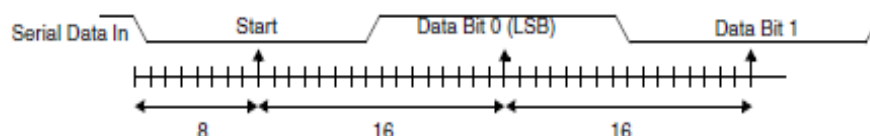


Figure 68: Receiver Serial Data Sample Points

As part of the 16550 standard an optional baud clock reference output signal (baudout\_n) is supplied to provide timing information to receiving devices that require it. The baud rate of the UART is controlled by the serial clock (sclk or pclk in a single clock implementation) and

the Divisor Latch Register (DLH and DLL). The available baud rates are presented in the following table:

Table 42: Baud rate generation

| Baud Rate | Divider | DLH/DLL Reg | UART_DLF Reg | Error % |
|-----------|---------|-------------|--------------|---------|
| 9600      | 104.166 | 104         | 3            | 0.01    |
| 19200     | 52.083  | 52          | 1            | 0.04    |
| 38400     | 26.041  | 26          | 1            | 0.07    |
| 57600     | 17.361  | 17          | 6            | 0.07    |
| 115200    | 8.680   | 8           | 11           | 0.07    |
| 230400    | 4.340   | 4           | 5            | 0.64    |
| 812500    | 1.230   | 1           | 4            | 1.53    |
| 1000000   | 1       | 1           | 0            | 0       |

### 23.2 IRDA 1.0 SIR PROTOCOL

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 kBaud.

**Note 5:** Attention. Information provided on IrDA SIR mode in this section assumes that the reader is fully familiar with the IrDA Serial Infrared Physical Layer Specifications. This specification can be obtained from the following website:  
<http://www.irda.org>

The data format is similar to the standard serial (sout and sin) data format. Each data character is sent serially, beginning with a start bit, followed by 8 data bits, and ending with at least one stop bit. Thus, the number of data bits that can be sent is fixed. No parity information can be supplied and only one stop bit is used while in this mode.

Trying to adjust the number of data bits sent or enable parity with the Line Control Register (LCR) has no

effect. When the UART is configured to support IrDA 1.0 SIR it can be enabled with Mode Control Register (MCR) bit 6. When the UART is not configured to support IrDA SIR mode, none of the logic is implemented and the mode cannot be activated, reducing total gate counts. When SIR mode is enabled and active, serial data is transmitted and received on the sir\_out\_n and sir\_in ports, respectively.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Thus, each new character begins with an infrared pulse for the start bit. However, received data is inverted from transmitted data due to the infrared pulses energizing the photo transistor base of the IrDA receiver, pulling its output low. This inverted transistor output is then fed to the UART sir\_in port, which then has correct UART polarity. Figure 69: shows the timing diagram for the IrDA SIR data format in comparison to the standard serial format.

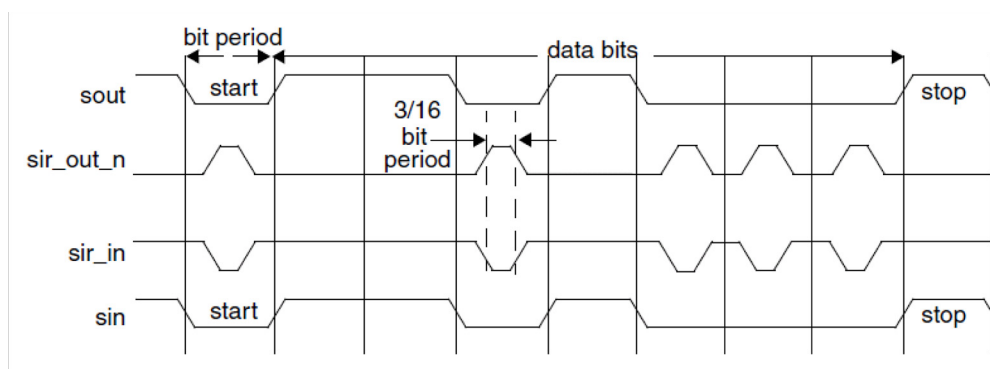


Figure 69: IrDA SIR Data Format

As detailed in the IrDA 1.0 SIR, the UART can be configured to support a low-power reception mode. When the UART is configured in this mode, the reception of SIR pulses of 1.41 microseconds (minimum pulse duration) is possible, as well as nominal 3/16 of a normal serial bit time. Using this low-power reception mode requires programming the Low Power Divisor Latch (LPDLL/LPDLH) registers. It should be noted that for all sclk frequencies greater than or equal to 7.37MHz (and obey the requirements of the Low Power Divisor Latch registers), pulses of 1.41uS are detectable. However there are several values of sclk that do not allow the detection of such a narrow pulse and these are as follows:

Table 43: Low power Divisor Latch register values

| SCLK     | Low power Divisor Latch register value | Min Pulse width for detection |
|----------|----------------------------------------|-------------------------------|
| 1.84 MHz | 1                                      | 3.77 $\mu$ S                  |
| 3.69 MHz | 2                                      | 2.086 $\mu$ S                 |

Table 43: Low power Divisor Latch register values

| SCLK     | Low power Divisor Latch register value | Min Pulse width for detection |
|----------|----------------------------------------|-------------------------------|
| 5.33 MHz | 3                                      | 1.584 $\mu$ S                 |

When IrDA SIR mode is enabled, the UART operation is similar to when the mode is disabled, with one exception; data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled. This is because the IrDA SIR physical layer specifies a minimum of 10ms delay between transmission and reception. This 10ms delay must be generated by software.

### 23.3 CLOCK SUPPORT

The UART has two system clocks (pclk and sclk). Having the second asynchronous serial clock (sclk) implemented accommodates accurate serial baud rate settings, as well as APB bus interface requirements.

With the two clock design a synchronization module is implemented for synchronization of all control and data

across the two system clock boundaries.

A serial clock faster than four-times the PCLK does not leave enough time for a complete incoming character to be received and pushed into the receiver FIFO. However, in most cases, the PCLK signal is faster than the serial clock and this should never be an issue.

The serial clock modules must have time to see new register values and reset their respective state machines. This total time is guaranteed to be no more than eight clock cycles of the slower of the two system clocks. Therefore, no data should be transmitted or received before this maximum time expires, after initial configuration.

### 23.4 INTERRUPTS

**Table 44: UART Interrupt priorities**

| Interrupt Id | Interrupt Set and Reset Functions |                                    |                                                                                                                                      |                                                                                                                                                                                                |
|--------------|-----------------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bits [3-0]   | Priority Level                    | Interrupt Type                     | Interrupt Source                                                                                                                     | Interrupt Reset Control                                                                                                                                                                        |
| 0001         | -                                 | None                               |                                                                                                                                      |                                                                                                                                                                                                |
| 0110         | Highest                           | Receiver Line status               | Overrun/parity/ framing errors or break interrupt                                                                                    | Reading the line status register                                                                                                                                                               |
| 0100         | 1                                 | Receiver Data Available            | Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)           | Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)                                                 |
| 1100         | 2                                 | Character timeout indication       | No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1 character in it during this time. | Reading the receiver buffer register                                                                                                                                                           |
| 0010         | 3                                 | Transmitter holding register empty | Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO at or below threshold (Prog. THRE Mode enabled).          | Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled). |
| 0000         | 4                                 | Reserved                           |                                                                                                                                      |                                                                                                                                                                                                |
| 0111         | Lowest                            | Reserved                           | -                                                                                                                                    | -                                                                                                                                                                                              |

### 23.5 PROGRAMMABLE THRE INTERRUPT

The UART can be configured to have a Programmable THRE Interrupt mode available to increase system performance.

When Programmable THRE Interrupt mode is selected it can be enabled via the Interrupt Enable Register (IER[7]). When FIFOs and the THRE Mode are implemented and enabled, THRE Interrupts are active at, and below, a programmed transmitter FIFO empty threshold level, as opposed to empty, as shown in the flowchart in [Figure 70](#).

The assertion of the UART interrupt (UART\_INT) occurs whenever one of the several prioritized interrupt types are enabled and active. The following interrupt types can be enabled with the IER register:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode)

When an interrupt occurs the master accesses the UART\_IIR\_REG to determine the source of the interrupt before dealing with it accordingly. These interrupt types are described in more detail in [Table 44](#)



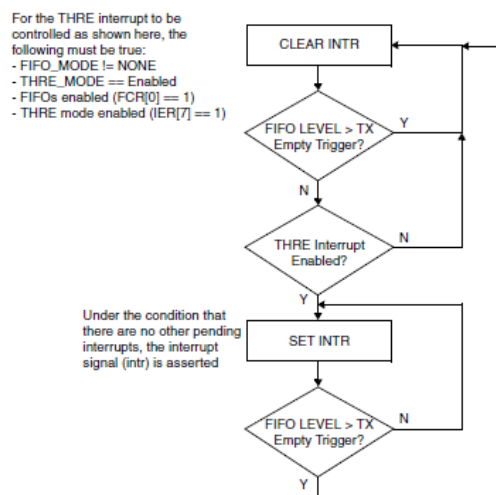


**Figure 70: Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode**

This threshold level is programmed into FCR[5:4]. The available empty thresholds are: empty, 2,  $\frac{1}{4}$  and  $\frac{1}{2}$ . See UART\_FCR\_REG for threshold setting details. Selection of the best threshold value depends on the system's ability to begin a new transmission sequence in a timely manner. However, one of these thresholds should prove optimum in increasing system performance by preventing the transmitter FIFO from running empty.

In addition to the interrupt change, Line Status Register (LSR[5]) also switches function from indicating transmitter FIFO empty, to FIFO full. This allows software to fill the FIFO each transmit sequence by polling LSR[5] before writing another character. The flow then becomes, "fill transmitter FIFO whenever an interrupt

occurs and there is data to transmit", instead of waiting until the FIFO is completely empty. Waiting until the FIFO is empty causes a performance hit whenever the system is too busy to respond immediately.

Even if everything else is selected and enabled, if the FIFOs are disabled via FCR[0], the Programmable THRE Interrupt mode is also disabled. When not selected or disabled, THRE interrupts and LSR[5] function normally (both reflecting an empty THR or FIFO). The flowchart of THRE interrupt generation when not in programmable THRE interrupt mode is shown in [Figure 71](#).

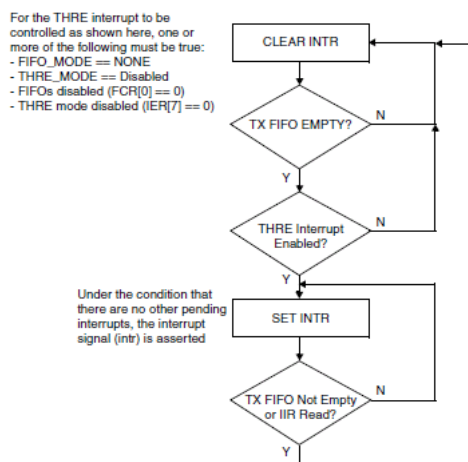


**Figure 71: Flowchart of Interrupt generation when not in Programmable THRE Interrupt Mode**

### 23.6 SHADOW REGISTERS

The shadow registers shadow some of the existing register bits that are regularly modified by software. These can be used to reduce the software overhead that is introduced by having to perform read-modify writes.

- UART\_SRBR\_REG support a host burst mode where the host increments its address but still accesses the same Receive buffer register
- UART\_STHR support a host burst mode where the host increments its address but still accesses the same transmit holding register.
- UART\_SFE\_REG accesses the FCR[0] register without accessing the other UART\_FCR\_REG bits.
- UART\_SRT\_REG accesses the FCR[7-6] register without accessing the other UART\_FCR\_REG bits.
- UART\_STER\_REG accesses the FCR[5-4] register without accessing the other UART\_FCR\_REG bits.

### 23.7 DIRECT TEST MODE

The on-chip UARTS can be used for the Direct Test Mode required for the final product PHY layer testing. It can be done either over the HCI layer, which engages a full CTS/RTS UART or using a 2-wire UART directly as described in the Bluetooth Low Energy Specification (Volume 6, Part F).

## 24 SPI+ Interface

This interface supports a subset of the Serial Peripheral Interface SPI™. The serial interface can transmit and receive 8, 16 or 32 bits in master/slave mode and transmit 9 bits in master mode. The SPI+ interface has enhanced functionality with bidirectional 2x16-bit word FIFOs. Two SPI+ controllers are instantiated in the system i.e. SPI and SPI2.

SPI™ is a trademark of Motorola, Inc.

### Features

- Slave and Master mode
- 8-bit, 9-bit, 16-bit or 32-bit operation

- Clock speeds upto 48 MHz for the SPI controller. Programmable output frequencies of SPI interface clock divided by 2, 4, 8 and 14
- SPI mode 0, 1, 2, 3 support. (clock edge and phase)
- Programmable SPI\_DO idle level
- Maskable Interrupt generation
- Bus load reduction by unidirectional writes-only and reads-only modes.
- Built-in RX/TX FIFOs for continuous SPI bursts.
- DMA support

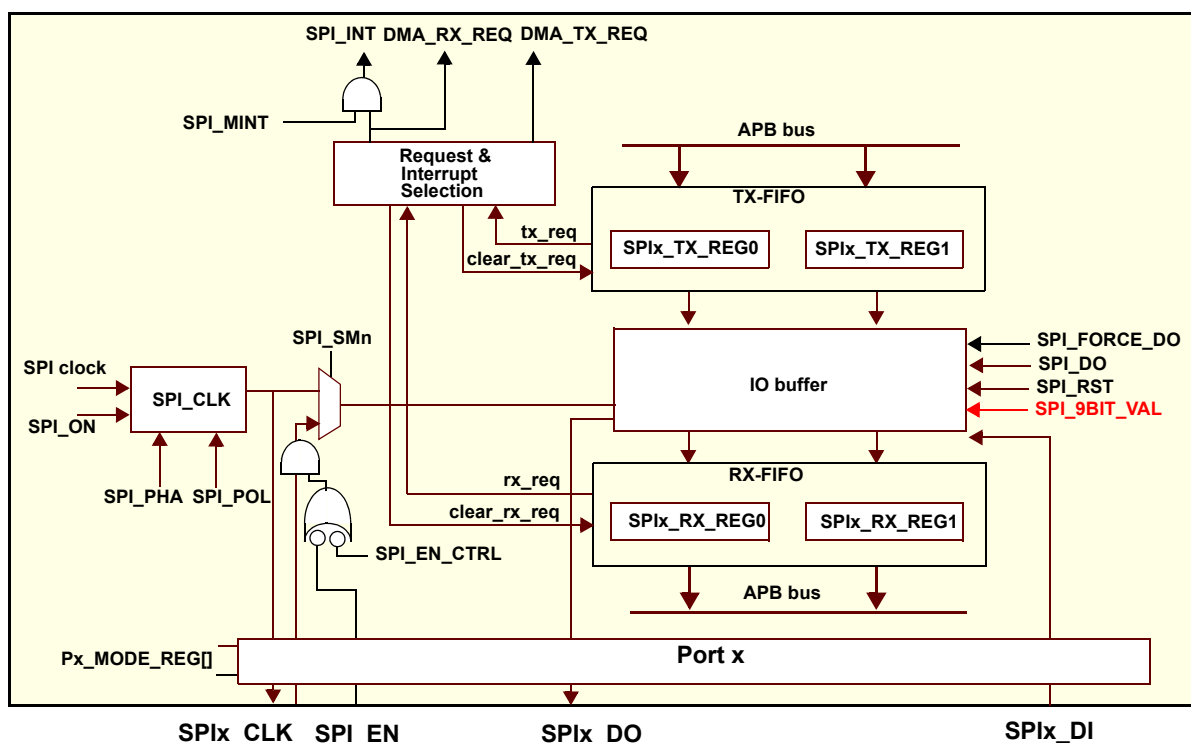


Figure 72: SPI block diagram

### 24.1 OPERATION WITHOUT FIFOS

This mode is the default mode.

#### Master mode

To enable SPI™ operation, first the individual port signal must be enabled. Next the SPI must be configured in SPI\_CTRL\_REG, for the desired mode. Finally bit SPI\_ON must be set to 1.

A SPI transfer cycle starts after writing to the SPIx\_RX\_TX\_REG0. In case of 32 bits mode, the SPIx\_RX\_TX\_REG1 must be written first. Writing to SPIx\_RX\_TX\_REG0 also sets the SPI\_TXH. As soon as the holding register is copied to the IO buffer, the SPI\_TXH is reset and a serial transfer cycle of 8/9/16/32 clock-cycles is started which causes 8/9/16/32 bits

to be transmitted on SPIx\_DO. Simultaneously, data is received on SPIx\_DI and shifted into the IO buffer. The transfer cycle finishes after the 8th/9th/16th/32nd clock cycle and SPI\_INT\_BIT bit is set in the SPIx\_CTRL\_REG and SPI\_INT\_PENDING bit in (RE)SET\_INT\_PENDING\_REG is set. The received bits in the IO buffer are copied to the SPIx\_RX\_TX\_REG0 (and SPIx\_RX\_TX\_REG1 in case of 32 bits mode) where they can be read by the CPU.

Interrupts to the CPU can be disabled using the SPI\_MINT bit. To clear the SPI interrupt source, any value to SPIx\_CLEAR\_INT\_REG must be written. Note however that SPI\_INT will be set as long as the RX-FIFO contains unread data.

### Slave mode

The slave mode is selected with SPI\_SMn set to 1 and the Px\_MODE\_REG must also select SPIx\_CLK as input. The functionality of the IO buffer in slave and master mode is identical. The SPI module clocks data in on SPIx\_DI and out on SPIx\_DO on every active edge of SPIx\_CLK. As shown in Figure 73 to Figure 76. SPI1 has an active low clock enable SPI1\_EN which can be enabled with bit SPI\_EN\_CTRL=1.

In slave mode the internal SPI clock must be more than four times the SPIx\_CLK

In slave mode the SPI\_EN serves as a clock enable and bit synchronization. If enabled with bit SPI\_EN\_CTRL. As soon as SPI\_EN is deactivated between the MSB and LSB bits, the I/O buffer is reset.

### SPI\_POL and SPI\_PHA

The phase and polarity of the serial clock can be changed with bits SPI\_POL and SPI\_PHA in the SPIx\_CTRL\_REG.

### SPI\_DO idle levels

The idle level of signal SPI\_DO depends on the master or slave mode and polarity and phase mode of the clock.

In master mode pin SPIx\_DO gets the value of bit SPI\_DO if the SPI is idle in all modes. Also if slave in SPI modes 0 and 2, SPI\_DO is the initial and final idle level.

In SPI modes 1 and 3 however there is no clock edge after the sampled lsb and pin SPIx\_DO gets the lsb value of the IO buffer. If required, the SPIx\_DO can be forced to the SPI\_DO bit level by resetting the SPI to the idle state by shortly setting bit **SPI\_RST** to 1. (Optionally **SPI\_FORCE\_DO** can be set, but this does not reset the IO buffer). The following diagrams show the timing of the SPI<sup>TM</sup> interface.

### Writes only mode

In “writes only” mode (SPI\_FIFO\_MODE = “10”) only the TX-FIFO is used. Received data will be copied to the SPIx\_RX\_TX\_REGx, but if a new SPI transfer is finished before the old data is read from the memory, this register will be overwritten.

SPI\_INT acts as a tx\_request signal, indicating that there is still place in the FIFO. It will be ‘0’ when the FIFO is full or else ‘1’ when it’s not full. This is also indicated in the SPIx\_CTRL\_REG[SPI\_TXH], which is ‘1’ if the TX-FIFO is full. Writing to the FIFO if this bit is still 1, will result in transmission of undefined data. If all data has been transferred, SPIx\_CTRL\_REG1[SPI\_BUSY] will become ‘0’.

For DMA operation only DMA1 must be configured. Starting transfers by manually writing to the SPIx\_TX\_REGx shall not be done because DMA\_tx\_req is already ‘1’ when this mode is activated.

### Reads only mode

In “reads only” mode (SPI\_FIFO\_MODE = “01”) only the RX-FIFO is used. Transfers will start immediately when the SPI is turned on in this mode. In transmit direction the SPI\_DO pin will transmit the IO buffer contents being the actual value of the SPIx\_TX\_REGx (all 0’s after reset). This means that no dummy writes are needed for reads only transfers.

In **slave mode** transfers only take place if the external master initiates them, but in master mode this means that transfers will continue until the RX-FIFO is full. If this happens SPIx\_CTRL\_REG1[SPI\_BUSY] will become ‘0’. If exactly N words need to be read from SPI device, first read (N - *fifosize*+1) words. Then wait until the SPI\_BUSY becomes ‘0’, set SPI\_FIFO\_MODE to “00” and finally read the remaining (*fifosize* +1) words. Here *fifosize* is 4/2/1 words for 8/16/32 bits mode respectively.

If this is not done, more data will be read from the SPI device until the FIFO is completely filled, or the SPI is turned off.

For DMA operation only DMA0 must be configured. Manual transfers are not needed, as the SPI will start transferring immediately when turning on this mode.

### Bidirectional transfers with FIFO

If SPI\_FIFO\_MODE is “00”, both registers are used as a FIFO. SPI\_TXH indicates that TX-FIFO is full, SPI\_INT indicates that there is data in the RX-FIFO.

DMA operation is recommended using both DMA0 and DMA1. No manual transfers are required because the requests will trigger the DMA automatically.

## 24.2 9 BITS MODE

The 9 bits mode can be used to support 9 bits displays and is selected with SPIx\_CTRL\_REG[SPI\_WORD] set to ‘11’. The value of the 9th bit, set in the SPIx\_CTRL\_REG1[SPI\_9BIT\_VAL] and is used to determine if the next 8 bits form a command word or data word. Because the 9th bit is not part of the data, the FIFO’s are still used in the 8 bits mode. The 9th bit is received but not saved because it is shifted out of the 8 bits shift register upon reception.

The 9 bits command should be entered by writing to the SPIx\_RX\_TX\_REG0, while the larger amount of data words can best be handled by the DMA controller. SPI\_9BIT\_VAL is set to “data mode”. To send a new command word at the end, the DMA (and SPI) must be stopped and the SPI\_9BIT\_VAL shall be set to “command mode” again.

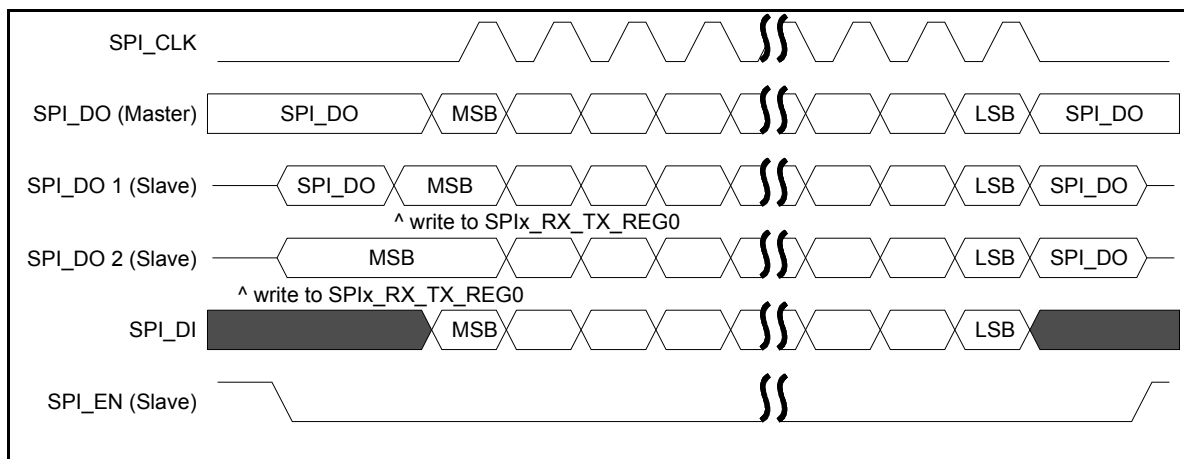


Figure 73: SPI Master/slave, mode 0: SPI\_POL=0 and SPI\_PHA=0

**Note 6:** If 9 bits SPI mode, the MSB bit in transmit direction is determined by bit SPIx\_CTRL\_REG[SPI\_9BIT\_VAL]. In receive direction, the MSB is received but not stored.

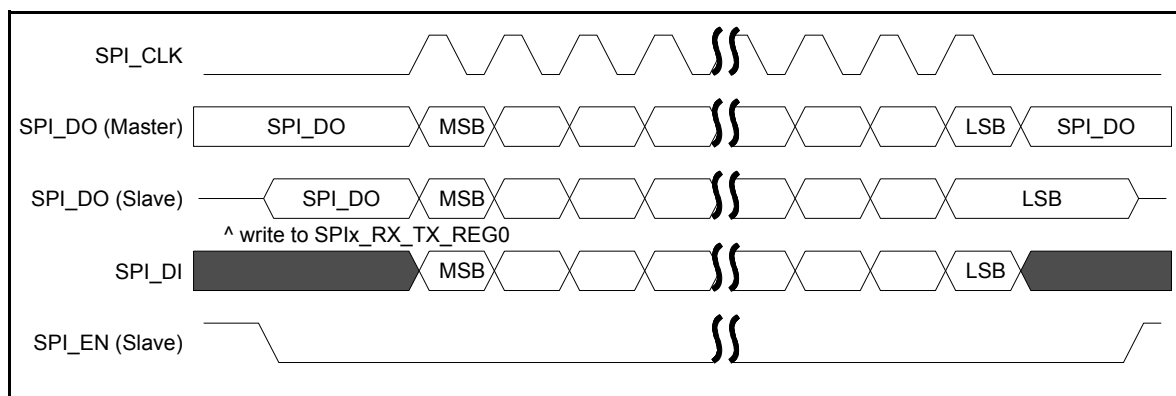


Figure 74: SPI Master/Slave, mode 1: SPI\_POL=0 and SPI\_PHA=1

For the MSB bit refer to Note 6.

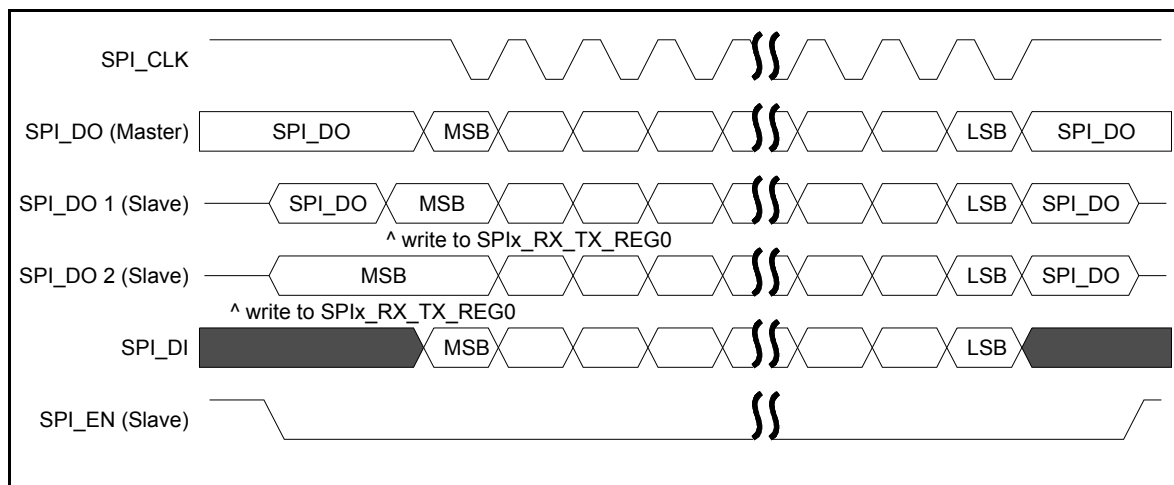


Figure 75: SPI Master/Slave, mode 2: SPI\_POL=1 and SPI\_PHA=0

For the MSB bit refer to Note 6.

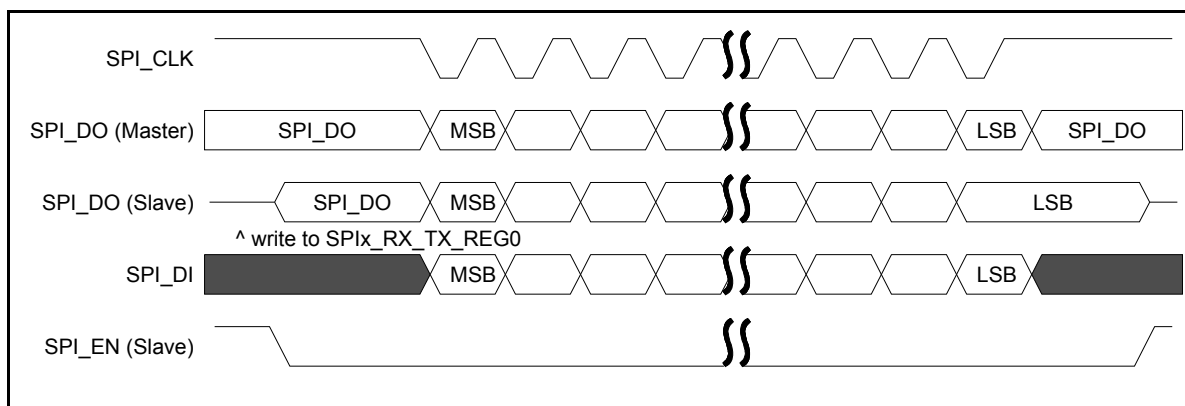


Figure 76: SPI Master/slave, mode 3: SPI\_POL=1 and SPI\_PHA=1

For the MSB bit refer to Note 6.

### 24.3 TIMING

The timing of the SPI interface when SPI controller is in slave mode is presented in the following figure:

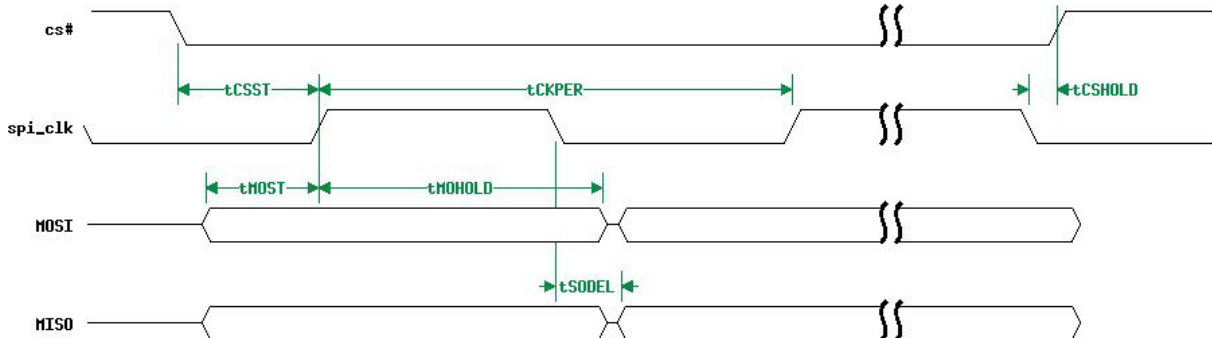


Figure 77: SPI slave mode timing (CPOL=0, CPHA=0)

Note that  $T_{int}$  represents the internal SPI clock period and is equal to  $1.5 \times spi\_clk$  period.

Table 45: SPI timing parameters

| PARAMETER | DESCRIPTION                                   | CONDITIONS          | MIN                    | TYP                    | MAX                    | UNITS |
|-----------|-----------------------------------------------|---------------------|------------------------|------------------------|------------------------|-------|
| tCLKPER   | spi_clk clock period                          | VBAT=3V,<br>DCDC=On | $0.25 \times spi\_clk$ | $0.25 \times spi\_clk$ | $0.25 \times spi\_clk$ | MHz   |
| tCSST     | CS active before spi_clk rising edge          |                     | $8.9 + T_{int}$        | $4.5 + T_{int}$        | $2.8 + T_{int}$        | ns    |
| tCSHOLD   | CS stays active after falling edge of spi_clk |                     | 0                      | 0                      | 0                      | ns    |
| tMOST     | Input data latching setup time                |                     | 9.3                    | 4.6                    | 2.8                    | ns    |
| tMOHOLD   | input data hold time                          |                     | 0                      | 0                      | 0                      | ns    |
| tSODEL    | Output data hold time                         |                     | 30                     | 14.3                   | 8.7                    | ns    |



## 25 I2C

The I2C is a programmable control bus that provides support for the communications link between Integrated Circuits in a system. It is a simple two-wire bus with a software-defined protocol for system control, which is used in temperature sensors and voltage level translators to EEPROMs, general-purpose I/O, A/D and D/A converters.

Two I2C controllers are instantiated in the system, namely I2C and I2C2.

### Features

- Two-wire I2C serial interface consists of a serial data line (SDA) and a serial clock (SCL)
- Standard and Fast mode support (up to 400 kb/s)
- Clock synchronization
- 4 deep transmit/receive 9-bit wide FIFOs
- Master transmit, Master receive operation
- 7 or 10-bit addressing
- 7 or 10-bit combined format transfers
- Bulk transmit mode
- Default slave address of 0x055
- Interrupt or polled-mode operation
- Handles Bit and Byte waiting at both bus speeds
- Programmable SDA hold time
- DMA request signals

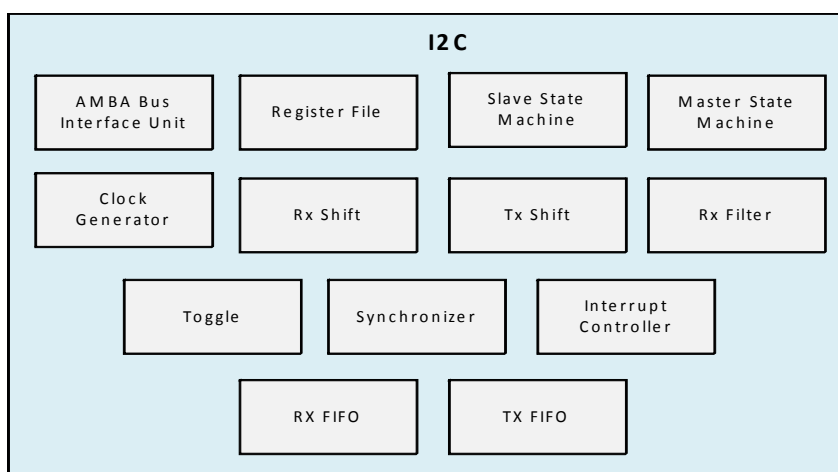


**Figure 78: I2C Block diagram**

The I2C controller block diagram is shown in [Figure 78](#). It contains the following sub blocks:

- **AMBA Bus Interface Unit.** Interfacing the APB interface to access the register file
- **Register File.** Contains configuration registers and is the interface with software.
- **Master State Machine.** Generates the I2C protocol for the master transfers.
- **Clock Generator.** Calculates the required timing to do the following:
  - Generate the SCL clock when configured as a master
  - Check for bus idle
  - Generate a START and a STOP
  - Setup the data and hold the data

**Rx Shift.** Takes data into the design and extracts it in byte format.

**Tx Shift.** Presents data supplied by CPU for transfer on the I2C bus.

**Rx Filter.** Detects the events in the bus; for example, start, stop and arbitration lost.

**Toggle.** Generates pulses on both sides and toggles to transfer signals across clock domains.

**Synchronizer.** Transfers signals from one clock domain to another.

**Interrupt Controller.** Generates the raw interrupt and interrupt flags, allowing them to be set and cleared.

**RX FIFO/TX.** Holds the RX FIFO and TX FIFO register banks and controllers, along with their status levels.

### 25.1 I2C BUS TERMS

The following terms relate to how the role of the I2C device and how it interacts with other I2C devices on the bus.

- **Transmitter.** the device that sends data to the bus. A transmitter can either be a device that initiates the data transmission to the bus (a master-transmitter)

or responds to a request from the master to send data to the bus (a slave-transmitter).

- **Receiver.** The device that receives data from the bus. A receiver can either be a device that receives data on its own request (a master-receiver) or in response to a request from the master (a slave-receiver).
- **Master.** The component that initializes a transfer (START command), generates the clock (SCL) signal and terminates the transfer (STOP command). A master can be either a transmitter or a receiver.
- **Slave.** The device addressed by the master. A slave can be either receiver or transmitter.

These concepts are illustrated in Figure 79:

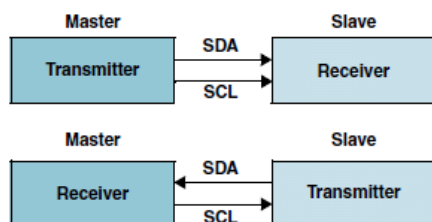


**Figure 79: Master/Slave and Transmitter/Receiver Relationships**

- **Multi-master.** The ability for more than one master to co-exist on the bus at the same time without collision or data loss.
- **Arbitration.** The predefined procedure that authorizes only one master at a time to take control of the bus. For more information about this behaviour, refer to Multiple Master Arbitration chapter
- **Synchronization.** The predefined procedure that synchronizes the clock signals provided by two or more masters. For more information about this feature, refer to Clock Synchronization chapter
- **SDA.** Data signal line (Serial Data)
- **SCL.** Clock signal line (Serial CLock)

### 25.1.1 Bus Transfer Terms

The following terms are specific to data transfers that occur to/from the I2C bus.

- **START (RESTART).** data transfer begins with a START or RESTART condition. The level of the SDA data line changes from high to low, while the SCL clock line remains high. When this occurs, the bus becomes busy.
- **STOP.** data transfer is terminated by a STOP condition. This occurs when the level on the SDA data line passes from the low state to the high state, while the SCL clock line remains high. When the data transfer has been terminated, the bus is free or idle once again. The bus stays busy if a RESTART is generated instead of a STOP condition.

**Note 7:** START and RESTART conditions are functionally identical.

### 25.2 I2C BEHAVIOUR

The I2C can be only be controlled via software to be an I2C master only, communicating with other I2C slaves;

The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The acknowledgement of data is sent by the device that is receiving data, which can be either a master or a slave. As mentioned previously, the I2C protocol also allows multiple masters to reside on the I2C bus and uses an arbitration procedure to determine bus ownership.

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledge (ACK) pulse after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behaviour is illustrated in Figure 80:

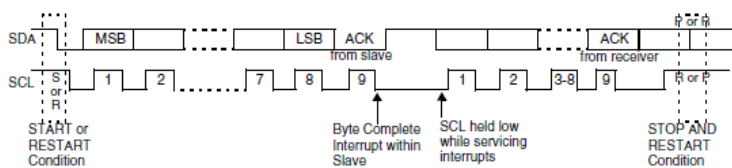


**Figure 80: Data transfer on the I2C Bus**

The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only while the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wire-AND functions on the bus. The maximum number of devices on the bus is limited by only the maximum capacitance specification of 400 pF. Data is transmitted in byte packages.

### 25.2.1 START and STOP Generation

When operating as an I2C master, putting data into the transmit FIFO causes the I2C controller to generate a START condition on the I2C bus. Allowing the transmit FIFO to empty causes the I2C controller to generate a STOP condition on the I2C bus.

When operating as a slave, the I2C controller does not generate START and STOP conditions, as per the protocol. However, if a read request is made to the I2C controller, it holds the SCL line low until read data has been supplied to it. This stalls the I2C bus until read data is provided to the slave I2C controller, or the I2C controller slave is disabled by writing a 0 to I2C\_ENABLE.

### 25.2.2 Combined Formats

The I2C controller supports mixed read and write combined format transactions in both 7-bit and 10-bit addressing modes.

The I2C controller does not support mixed address and mixed address format. That is, a 7-bit address transaction followed by a 10-bit address transaction or vice versa. Combined format transactions.

To initiate combined format transfers, I2C\_CON.I2C\_RESTART\_EN should be set to 1. With this value set and operating as a master, when the I2C controller completes an I2C transfer, it checks the transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the transmit FIFO is empty when the current I2C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

### 25.3 I2C PROTOCOLS

The I2C controller has the following protocols:

- START and STOP Conditions
- Addressing Slave Protocol
- Transmitting and Receiving Protocol
- START BYTE Transfer Protocol

#### 25.3.1 START and STOP Conditions

When the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master wants to terminate the transmission, the master issues a STOP condition. This is defined to be a low-to-high transition of the SDA line while SCL is 1. [Figure 81](#): shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when SCL is 1.

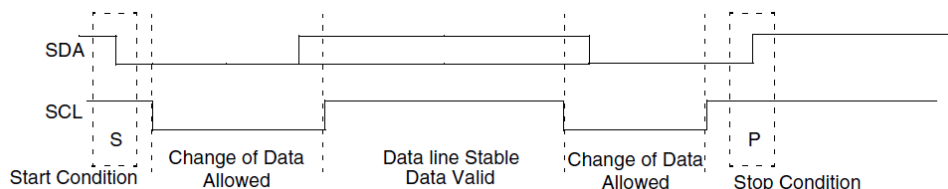


**Figure 81: START and STOP Conditions**

**Note 8:** The signal transitions for the START/STOP conditions, as depicted in [Figure 80](#), reflect those observed at the output signals of the Master driving the I2C bus. Care should be taken when observing the SDA/SCL signals at the input signals of the Slave(s), because unequal line delays may result in an incorrect SDA/SCL timing relationship.

### 25.3.2 Addressing Slave Protocol

There are two address formats: the 7-bit address format and the 10-bit address format.

#### 7-bit Address Format

During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit as shown in [Figure 82](#). When bit 0 (R/W) is set to 0, the master writes to the slave. When bit 0 (R/W) is set to 1, the master reads

from the slave.

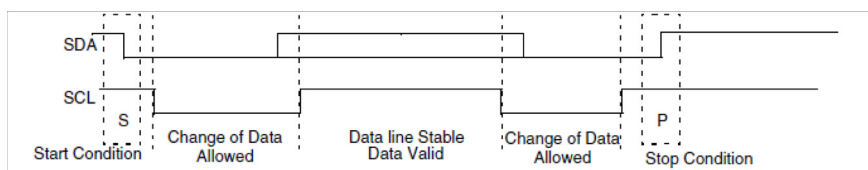


Figure 82: 7-bit Address Format

**10-bit Address Format**

During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the

slaves address bits 9:8, and the LSB bit (bit 0) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address. Figure 83: shows the 10-bit address format, and Table 46 defines the special purpose and reserved first byte addresses.

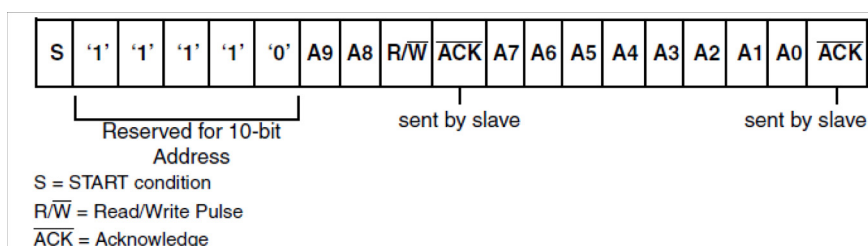


Figure 83: 10-bit Address Format

**Table 46: I2C Definition of Bits in First Byte**

| Slave address | R/W Bits | Description                                                                                                     |
|---------------|----------|-----------------------------------------------------------------------------------------------------------------|
| 0000 000      | 0        | General Call Address. I2C controller places the data in the receive buffer and issues a General Call interrupt. |
| 0000 000      | 1        | START byte. For more details, refer to "START BYTE Transfer Protocol" 0000                                      |
| 0000 001      | X        | CBUS address. I2C controller ignores these accesses                                                             |
| 0000 010      | X        | Reserved                                                                                                        |
| 0000 011      | X        | Reserved                                                                                                        |
| 0000 1XX      | X        | High-speed master code (for more information, refer to "Multiple Master Arbitration")                           |
| 1111 1XX      | X        | Reserved                                                                                                        |
| 1111 0XX      | X        | 10-bit slave addressing                                                                                         |

I2C controller does not restrict you from using these reserved addresses. However, if you use these reserved addresses, you may run into incompatibilities with other I2C components.

**25.3.3 Transmitting and Receiving Protocol**

The master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests

from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively.

**Master-Transmitter and Slave-Receiver**

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal

(ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

If the master-transmitter is transmitting data as shown in Figure 84, then the slave-receiver responds to the master-transmitter with an acknowledge pulse after every byte of data is received.

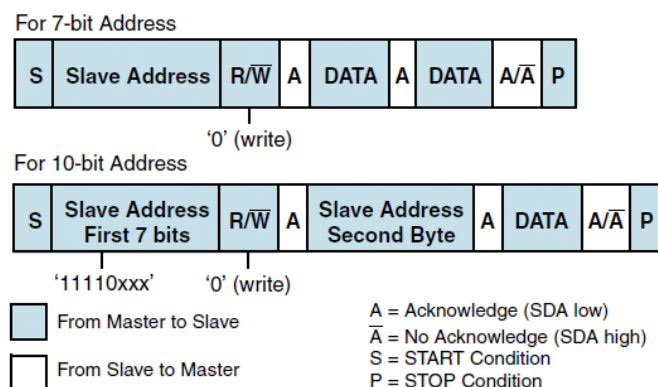


Figure 84: Master-Transmitter Protocol

#### Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 85, then the master responds to the slave-transmitter with an acknowledge pulse after a byte of data has been received, except for the last byte. This is the way the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

When a master does not want to relinquish the bus with a STOP condition, the master can issue a RESTART condition. This is identical to a START condition except it occurs after the ACK pulse. The master can then communicate with the same slave or a different slave.

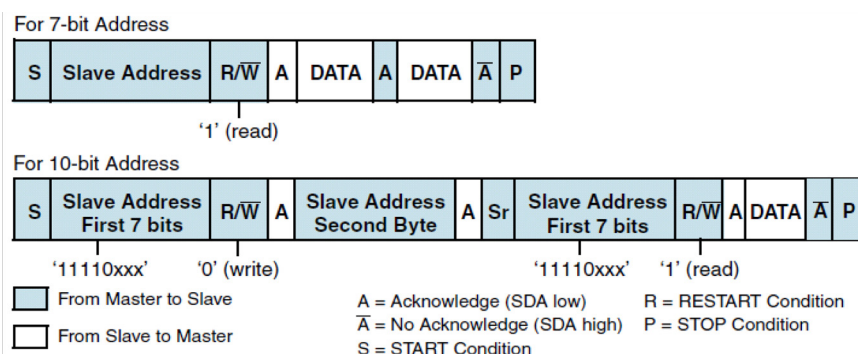


Figure 85: Master-Receiver Protocol

### START BYTE Transfer Protocol

The START BYTE transfer protocol is set up for systems that do not have an on-board dedicated I2C hardware module. When the I2C controller is addressed as a slave, it always samples the I2C bus at the highest speed supported so that it never requires a START BYTE transfer. However, when I2C controller is a master, it supports the generation of START BYTE transfers at the beginning of every transfer in case a slave

device requires it. This protocol consists of seven zeros being transmitted followed by a 1, as illustrated in Figure 86. This allows the processor that is polling the bus to under-sample the address phase until 0 is detected. Once the microcontroller detects a 0, it switches from the under sampling rate to the correct rate of the master.

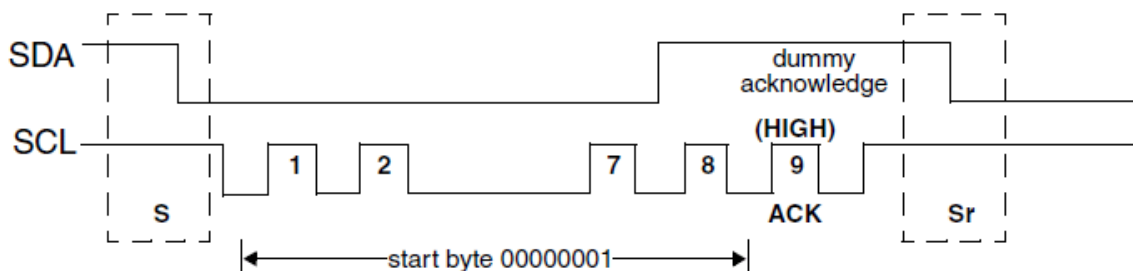


Figure 86: START BYTE Transfer

The START BYTE procedure is as follows:

1. Master generates a START condition.
2. Master transmits the START byte (0000 0001).
3. Master transmits the ACK clock pulse. (Present only to conform with the byte handling format used on the bus)
4. No slave sets the ACK signal to 0.
5. Master generates a RESTART (R) condition.

A hardware receiver does not respond to the START BYTE because it is a reserved address and resets after the RESTART condition is generated.

### 25.4 MULTIPLE MASTER ARBITRATION

The I2C controller bus protocol allows multiple masters to reside on the same bus. If there are two masters on the same I2C-bus, there is an arbitration procedure if both try to take control of the bus at the same time by generating a START condition at the same time. Once a master (for example, a microcontroller) has control of the bus, no other master can take control until the first master sends a STOP condition and places the bus in an idle state.

Arbitration takes place on the SDA line, while the SCL line is 1. The master, which transmits a 1 while the other master transmits 0, loses arbitration and turns off its data output stage. The master that lost arbitration can continue to generate clocks until the end of the byte transfer. If both masters are addressing the same slave device, the arbitration could go into the data phase. Figure 87 illustrates the timing of when two masters are arbitrating on the bus.

For high-speed mode, the arbitration cannot go into the data phase because each master is programmed with a unique high-speed master code. This 8-bitcode is

defined by the system designer and is set by writing to the High Speed Master Mode Code Address Register, I2C\_HS\_MADDR. Because the codes are unique, only one master can win arbitration, which occurs by the end of the transmission of the high-speed master code.

Control of the bus is determined by address or master code and data sent by competing masters, so there is no central master nor any order of priority on the bus.

Arbitration is not allowed between the following conditions:

- A RESTART condition and a data bit
- A STOP condition and a data bit
- A RESTART condition and a STOP condition

Slaves are not involved in the arbitration process.



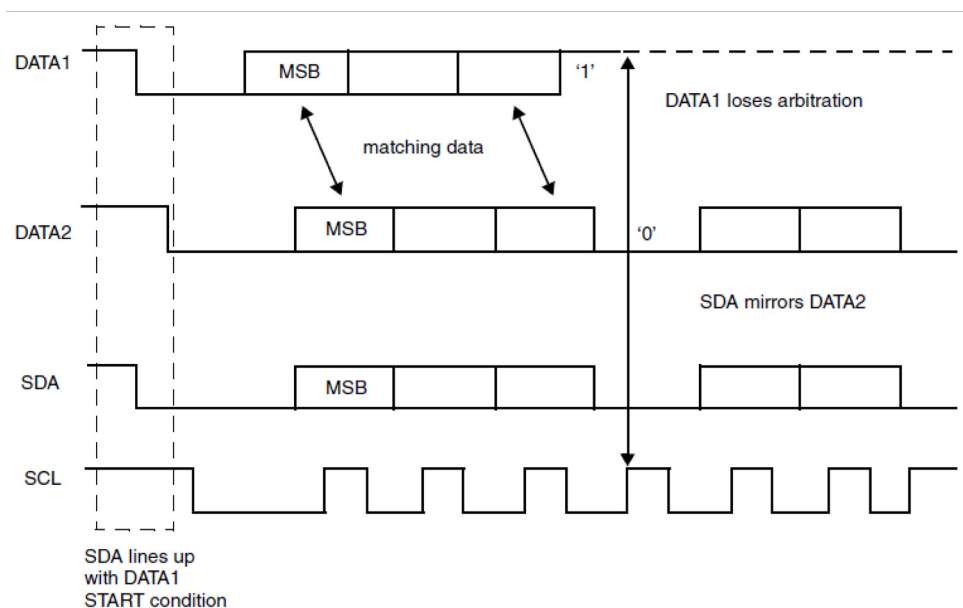


Figure 87: Multiple Master Arbitration

### 25.5 CLOCK SYNCHRONIZATION

When two or more masters try to transfer information on the bus at the same time, they must arbitrate and synchronize the SCL clock. All masters generate their own clock to transfer messages. Data is valid only during the high period of SCL clock. Clock synchronization is performed using the wired-AND connection to the SCL signal. When the master transitions the SCL clock to 0, the master starts counting the low time of the SCL clock and transitions the SCL clock signal to 1 at the beginning of the next clock period. However, if another master is holding the SCL line to 0, then the master

goes into a HIGH wait state until the SCL clock line transitions to 1.

All masters then count off their high time, and the master with the shortest high time transitions the SCL line to 0. The masters then count out their low time and the one with the longest low time forces the other master into a HIGH wait state. Therefore, a synchronized SCL clock is generated, which is illustrated in [Figure 88](#). Optionally, slaves may hold the SCL line low to slow down the timing on the I2C bus.

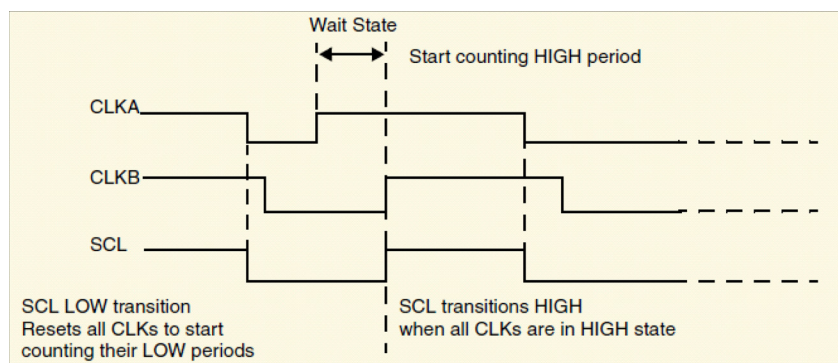


Figure 88: Multiple Master Clock synchronization.



## 25.6 OPERATION MODES

This section provides information on the following topics:

- Slave Mode Operation
- Master Mode Operation
- Disabling I2C controller

**Note 9:** It is important to note that the I2C controller should only be set to operate as an I2C Master, or I2C Slave, but not both simultaneously. This is achieved by ensuring that bit 6 (I2C\_SLAVE\_DISABLE) and 0 (I2C\_MASTER\_MODE) of the I2C\_CON register are never set to 0 and 1, respectively.

### 25.6.1 Slave Mode Operation

This section includes the following procedures:

- Initial Configuration
- Slave-Transmitter Operation for a Single Byte
- Slave-Receiver Operation for a Single Byte
- Slave-Transfer Operation For Bulk Transfers

#### Initial Configuration

To use the I2C controller as a slave, perform the following steps:

1. Disable the I2C controller by writing a '0' to bit 0 of the I2C\_ENABLE register.
2. Write to the I2C\_SAR register (bits 9:0) to set the slave address. This is the address to which the I2C controller responds.
3. Write to the I2C\_CON register to specify which type of addressing is supported (7- or 10-bit by setting bit 3). Enable the I2C controller in slave-only mode by writing a '0' into bit 6 (I2C\_SLAVE\_DISABLE) and a '0' to bit 0 (MASTER\_MODE).

**Note 10:** Slaves and masters do not have to be programmed with the same type of addressing 7- or 10-bit address. For instance, a slave can be programmed with 7-bit addressing and a master with 10-bit addressing, and vice versa.

4. Enable the I2C controller by writing a '1' in bit 0 of the I2C\_ENABLE register.

**Note 11:** Depending on the reset values chosen, steps 2 and 3 may not be necessary because the reset values can be configured. For instance, if the device is only going to be a master, there would be no need to set the slave address because you can configure I2C controller to have the slave disabled after reset and to enable the master after reset. The values stored are static and do not need to be reprogrammed if the I2C controller is disabled.

#### Slave-Transmitter Operation for a Single Byte

When another I2C master device on the bus addresses the I2C controller and requests data, the I2C controller acts as a slave-transmitter and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the slave address in the I2C\_SAR register of the I2C controller.

2. The I2C controller acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a slave-transmitter.

3. The I2C controller asserts the RD\_REQ interrupt (bit 5 of the I2C\_RAW\_INTR\_STAT register) and holds the SCL line low. It is in a wait state until software responds. If the RD\_REQ interrupt has been masked, due to I2C\_INTR\_MASK[5] register (M\_RD\_REQ bit field) being set to 0, then it is recommended that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the I2C\_RAW\_INTR\_STAT register.

- a. Reads that indicate I2C\_RAW\_INTR\_STAT[5] (R\_RD\_REQ bit field) being set to 1 must be treated as the equivalent of the RD\_REQ interrupt being asserted.

- b. Software must then act to satisfy the I2C transfer.

- c. The timing interval used should be in the order of 10 times the fastest SCL clock period the I2C controller can handle. For example, for 400 kb/s, the timing interval is 25us.

**Note 12:** The value of 10 is recommended here because this is approximately the amount of time required for a single byte of data transferred on the I2C bus.

4. If there is any data remaining in the TX FIFO before receiving the read request, then the I2C controller asserts a TX\_ABORT interrupt (bit 6 of the I2C\_RAW\_INTR\_STAT register) to flush the old data from the TX FIFO.

**Note 13:** Because the I2C controller's TX FIFO is forced into a flushed/reset state whenever a TX\_ABORT event occurs, it is necessary for software to release the I2C controller from this state by reading the I2C\_CLR\_TX\_ABORT register before attempting to write into the TX FIFO. See register I2C\_RAW\_INTR\_STAT for more details.

If the TX\_ABORT interrupt has been masked, due to of I2C\_INTR\_MASK[6] register (M\_TX\_ABORT bit field) being set to 0, then it is recommended that re-using the timing routine (described in the previous step), or a similar one, be used to read the I2C\_RAW\_INTR\_STAT register.

- a. Reads that indicate bit 6 (R\_TX\_ABORT) being set to 1 must be treated as the equivalent of the TX\_ABORT interrupt being asserted.

- b. There is no further action required from software.

- c. The timing interval used should be similar to that described in the previous step for the I2C\_RAW\_INTR\_STAT[5] register.

5. Software writes to the I2C\_DATA\_CMD register with the data to be written (by writing a '0' in bit 8).

6. Software must clear the RD\_REQ and TX\_ABORT interrupts (bits 5 and 6, respectively) of the I2C\_RAW\_INTR\_STAT register before proceeding.

If the RD\_REQ and/or TX\_ABORT interrupts have been masked, then clearing of the I2C\_RAW\_INTR\_STAT register will have already been performed when either the R\_RD\_REQ or R\_TX\_ABORT bit has been read as 1.

7. The I2C controller releases the SCL and transmits the byte.

8. The master may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

### Slave-Receiver Operation for a Single Byte

When another I2C master device on the bus addresses the I2C controller and is sending data, the I2C controller acts as a slave-receiver and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the I2C controller's slave address in the I2C\_SAR register.
2. The I2C controller acknowledges the sent address and recognizes the direction of the transfer to indicate that the I2C controller is acting as a slave-receiver.
3. I2C controller receives the transmitted byte and places it in the receive buffer.

**Note 14:** If the RX FIFO is completely filled with data when a byte is pushed, then an overflow occurs and the I2C controller continues with subsequent I2C transfers. Because a NACK is not generated, software must recognize the overflow when indicated by the I2C controller (by the R\_RX\_OVER bit in the I2C\_INTR\_STAT register) and take appropriate actions to recover from lost data. Hence, there is a real time constraint on software to service the RX FIFO before the latter overflow as there is no way to re-apply pressure to the remote transmitting master. You must select a deep enough RX FIFO depth to satisfy the interrupt service interval of their system.

4. I2C controller asserts the RX\_FULL interrupt (I2C\_RAW\_INTR\_STAT[2] register).

If the RX\_FULL interrupt has been masked, due to setting I2C\_INTR\_MASK[2] register to 0 or setting I2C\_TX\_TL to a value larger than 0, then it is recommended that a timing routine (described in "Slave-Transmitter Operation for a Single Byte") be implemented for periodic reads of the "I2C\_STATUS" on page 138 register. Reads of the I2C\_STATUS register, with bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX\_FULL interrupt being asserted.

5. Software may read the byte from the I2C\_DATA\_CMD register (bits 7:0).

6. The other master device may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

### Slave-Transfer Operation For Bulk Transfers

In the standard I2C protocol, all transactions are single byte transactions and the programmer responds to a remote master read request by writing one byte into the slave's TX FIFO. When a slave (slave-transmitter) is issued with a read request (RD\_REQ) from the remote master (master-receiver), at a minimum there should be at least one entry placed into the slave-transmitter's TX FIFO.

I2C controller is designed to handle more data in the TX FIFO so that subsequent read requests can take that data without raising an interrupt to get more data. Ultimately, this eliminates the possibility of significant latencies being incurred between raising the interrupt for data each time had there been a restriction of having only one entry placed in the TX FIFO.

This mode only occurs when I2C controller is acting as a slave-transmitter. If the remote master acknowledges the data sent by the slave-transmitter and there is no data in the slave's TX FIFO, the I2C controller holds the I2C SCL line low while it raises the read request interrupt (RD\_REQ) and waits for data to be written into the TX FIFO before it can be sent to the remote master.

If the RD\_REQ interrupt is masked, due to bit 5 (M\_RD\_REQ) of the I2C\_INTR\_STAT register being set to 0, then it is recommended that a timing routine be used to activate periodic reads of the I2C\_RAW\_INTR\_STAT register. Reads of I2C\_RAW\_INTR\_STAT that return bit 5 (R\_RD\_REQ) set to 1 must be treated as the equivalent of the RD\_REQ interrupt referred to in this section. This timing routine is similar to that described in "Slave-Transmitter Operation for a Single Byte"

The RD\_REQ interrupt is raised upon a read request, and like interrupts, must be cleared when exiting the interrupt service handling routine (ISR). The ISR allows you to either write 1 byte or more than 1 byte into the TX FIFO. During the transmission of these bytes to the master, if the master acknowledges the last byte, then the slave must raise the RD\_REQ again because the master is requesting for more data.

If the programmer knows in advance that the remote master is requesting a packet of n bytes, then when another master addresses I2C controller and requests data, the TX FIFO could be written with n number bytes and the remote master receives it as a continuous stream of data. For example, the I2C controller slave continues to send data to the remote master as long as the remote master is acknowledging the data sent and there is data available in the TX FIFO. There is no need to hold the SCL line low or to issue RD\_REQ again.

If the remote master is to receive n bytes from the I2C controller but the programmer wrote a number of bytes larger than n to the TX FIFO, then when the slave finishes sending the requested n bytes, it clears the TX FIFO and ignores any excess bytes.

The the I2C controller generates a transmit abort (TX\_ABRT) event to indicate the clearing of the TX FIFO in this example. At the time an ACK/NACK is expected, if a NACK is received, then the remote master has all the data it wants. At this time, a flag is raised within the slave's state machine to clear the leftover data in the TX FIFO. This flag is transferred to the processor bus clock domain where the FIFO exists and the contents of the TX FIFO is cleared at that time.

### 25.6.2 Master Mode Operation

This section includes the following topics:

- Initial Configuration
- Dynamic I2C\_TAR or I2C\_10BITADDR\_MASTER Update
- Master Transmit and Master Receive

#### Initial Configuration

The procedures are very similar and are only different with regard to where the I2C\_10BITADDR\_MASTER bit is set (either bit 4 of I2C\_CON register or bit 12 of I2C\_TAR register).

To use the I2C controller as a master perform the following steps:

1. Disable the I2C controller by writing 0 to the I2C\_ENABLE register.
2. Write to the I2C\_CON register to set the maximum speed mode supported (bits 2:1) and the desired speed of the I2C controller master-initiated transfers, either 7-bit or 10-bit addressing (bit 4).

Ensure that bit 6 I2C\_SLAVE\_DISABLE = 1 and bit 0 MASTER\_MODE = 1

**Note 15:** Slaves and masters do not have to be programmed with the same type of addressing 7- or 10-bit address. For instance, a slave can be programmed with 7-bit addressing and a master with 10-bit addressing, and vice versa.

3. Write to the I2C\_TAR register the address of the I2C device to be addressed (bits 9:0). This register also indicates whether a General Call or a START BYTE command is going to be performed by I2C.

4. Only applicable for high-speed mode transfers. Write to the I2C\_HS\_MADDR register the desired master code for the I2C controller. The master code is programmer-defined.

5. Enable the I2C controller by writing a 1 in bit 0 of the I2C\_ENABLE register.

6. Now write transfer direction and data to be sent to the I2C\_DATA\_CMD register. If the I2C\_DATA\_CMD register is written before the I2C controller is enabled, the data and commands are lost as the buffers are kept cleared when I2C controller is disabled.

This step generates the START condition and the address byte on the I2C controller. Once I2C controller is enabled and there is data in the TX FIFO, I2C controller starts reading the data.

**Note 16:** Depending on the reset values chosen, steps 2, 3, 4, and 5 may not be necessary because the reset values can be configured. The values stored are static and do not need to be reprogrammed if the I2C controller is disabled, with the exception of the transfer direction and data.

#### Master Transmit and Master Receive

The I2C Controller supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the I2C\_DATA\_CMD\_REG register. The CMD bit [8] should be written to 0 for I2C write operations.

Subsequently, a read command may be issued by writing "don't cares" to the lower byte of the I2C\_DATA\_CMD\_REG register, and a 1 should be written to the CMD bit.

The I2C Controller master continues to initiate transfers as long as there are commands present in the transmit FIFO. If the transmit FIFO becomes empty, the master inserts a STOP condition after completing the current transfer.

#### Disabling I2C controller

The register I2C\_ENABLE\_STATUS is added to allow software to unambiguously determine when the hardware has completely shutdown in response to the I2C\_ENABLE register being set from 1 to 0. Only one register is required to be monitored.

#### Procedure:

1. Define a timer interval (ti2c\_poll) equal to the 10 times the signalling period for the highest I2C transfer speed used in the system and supported by I2C controller. For example, if the highest I2C transfer mode is 400 kb/s, then this ti2c\_poll is 25us.
2. Define a maximum time-out parameter, MAX\_T\_POLL\_COUNT, such that if any repeated polling operation exceeds this maximum value, an error is reported.
3. Execute a blocking thread/process/function that prevents any further I2C master transactions to be started by software, but allows any pending transfers to be completed.

**Note 17:** This step can be ignored if I2C controller is programmed to operate as an I2C slave only.

4. The variable POLL\_COUNT is initialized to zero.
5. Set I2C\_ENABLE to 0.
6. Read the I2C\_ENABLE\_STATUS register and test the I2C\_EN bit (bit 0). Increment POLL\_COUNT by one. If POLL\_COUNT >= MAX\_T\_POLL\_COUNT, exit with the relevant error code.
7. If I2C\_ENABLE\_STATUS[0] is 1, then sleep for ti2c\_poll and proceed to the previous step.

Otherwise, exit with a relevant success code.

## 26 InfraRed Generator

The InfraRed generator provides a flexible way of transmitting any IR code used in remote controls. It has an efficient message queue where users can describe the waveform of a specific IR command in just a few bytes independently from the protocol. It sits on the 16-bit APB bus and receives a separate, up to 16MHz clock used for the carrier generation.

### Features

- Carrier frequencies from 30 to 60 KHz
- Supports Pulse width and Pulse Distance encoding
- Supports Manchester encoding
- Supports Time mode (no carrier)
- Any combination of Mark and Space symbols
- Code FIFO of 32 16-bit words for encoding commands
- Automatic Repeat function, transparent to SW
- Interrupt generation upon transmit completion

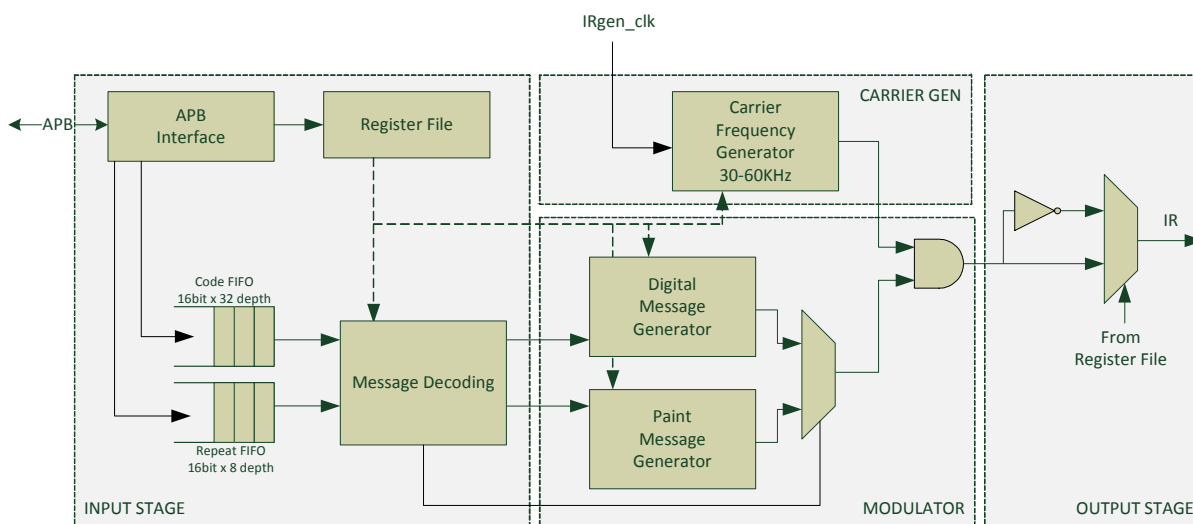


Figure 89: InfraRed Generator Block Diagram

### 26.1 ARCHITECTURE

The IR generator, is based upon the concept of using two different ways of describing data, being able to support any IR protocol:

- **Digital Message:** This message represents a logic 1 or a logic 0 and is clearly described in form of Mark and Space duration as well as sequence.
- **Paint Message:** This message represents a totally custom "painted" waveform. The way of efficiently describing this, is to encode the symbol type (Mark/

Space) and the symbol duration within a code word for HW to be able to understand and proceed accordingly to the correct modulation.

The composition of a command consists of a number of control words which contain information about digital or paint messages in an efficient way. A breakdown of the well-known IR code NEC command is presented in the following figure as an example:

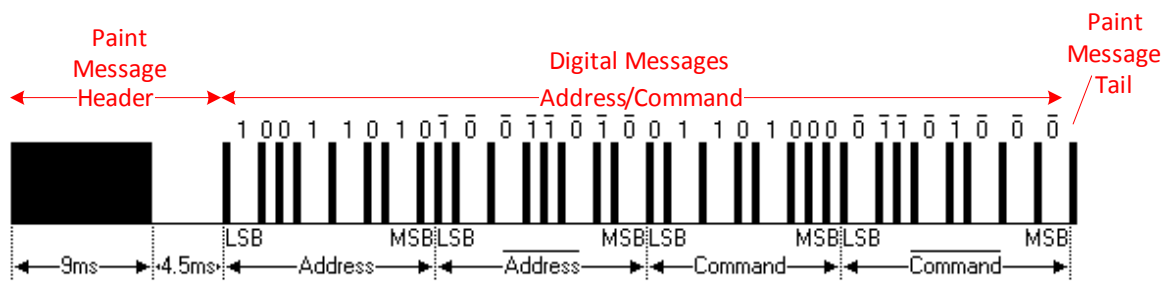


Figure 90: Paint/Digital messages on a NEC based example

The encoding of the Paint/Digital messages can be implemented according to the following tables:

**Table 47: IR Digital Message Encoding**

| Bit   | Name            | Description                                                                                    |
|-------|-----------------|------------------------------------------------------------------------------------------------|
| 15    | Message Type    | 1 = Digital message. That means that the message information itself defines logic 1 or logic 0 |
| 14:11 | Message Length  | Number of valid bits minus 1. Range 0 to 10                                                    |
| 10:0  | Message Payload | Digital Message consisting of logic 1s and logic 0s                                            |

**Table 48: IR Paint Message Encoding**

| Bit  | Name         | Description                                 |
|------|--------------|---------------------------------------------|
| 15   | Message Type | 0 = Paint message                           |
| 14   | Symbol Type  | 1 = Mark<br>0 = Space                       |
| 13:0 | Duration     | Mark/Space duration in carrier clock cycles |

So, basically, the MSbit of the word defines if the message is a digital or a paint one.

The block consists of four sub-blocks:

#### Input Stage

it consists of the APB interface the Register File, 2 FIFOs and the Message Decoding engine. This sub-block is responsible for the configuration of the system and the storage and decoding of the encoded words. These words will reside in the Code FIFO. The Repeat FIFO can be used to load special commands for repeating a key press i.e. this is only protocol dependent. The output of the Code FIFO is decoded by the Message Decoding engine and pushed in forms of commands into the next sub-block, the modulator. This sub-block is also responsible for firing up the repeat timer in case that a key is constantly pressed. The repetition time as well the message to be sent varies according to the protocol.

#### Carrier Generator

This sub-block is responsible for the carrier frequency generation. It has its own gated clock which can be up to 16MHz and can generate frequencies in the range of 30 - 60 KHz.

#### Modulator

This sub-block is responsible for the generation of the modulation signal which gates the carrier clock pulse train. The Modulator state machine select between digital or pain message and controls the gate accordingly.

#### Output Stage

This sub-block can be programmed to invert the output optionally

#### 26.2 PROGRAMMING

Initially the clock of the block has to be enabled by asserting the CLK\_PER\_REG[IR\_CLK\_ENABLE] bit. The carrier ON and OFF time in clock cycles have to be programmed at IR\_FREQ\_CARRIER\_Oxx\_REGS. Following that, the logic one and zero are defined in terms of clock cycles high (mark) and clock cycles low (zero) with help of the IR\_LOGIC\_ONE/ZERO\_TIME\_REGS

Another important feature that needs to be initialized is whether the logic one/zero start with a mark and is followed by a zero, or the other way around. This is defined in IR\_CTRL\_REG[IR\_LOGIC\_ONE\_FORMAT] and IR\_CTRL\_REG[IR\_LOGIC\_ZERO\_FORMAT] respectively. Finally, the time required for an automated retransmit is defined at IR\_REPEAT\_TIME\_REG.

Sending commands using a specific protocol is simply writing the correct words into the Code FIFO and then set IR\_CTRL\_REG[IR\_TX\_START] to trigger the transmission. The actual word values have to comply to the encoding schemes as presented in [Table 47](#) and [Table 48](#).



## 27 Quadrature Decoder

The DA14680 has a integrated quadrature decoder that can automatically decode the signals for the X, Y and Z axes of a HID input device, reporting step count and direction. This block can be used for waking up the chip as soon as there is any kind of movement from the external device connected to it. The block diagram of the quadrature decoder is presented in Figure 91.

### Features

- Three 16-bit signed counters that provide the step count and direction on each of the axes (X, Y and Z)
- Programmable system clock sampling at maximum 16 MHz.
- APB interface for control and programming
- Programmable source from P0, P1 and P2 ports
- Digital filter on the channel inputs to avoid spikes

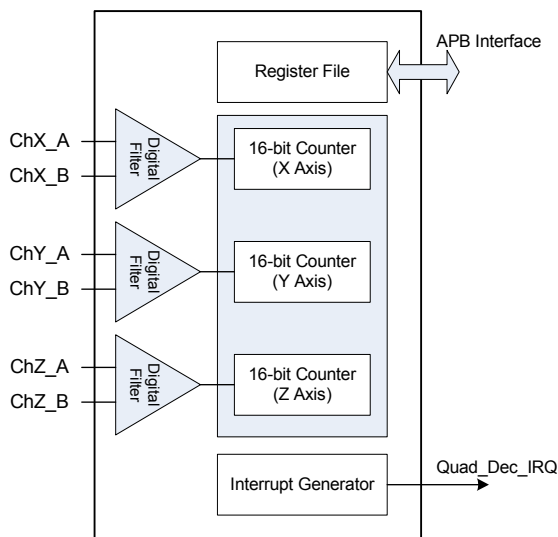


Figure 91: Block diagram of the Quadrature Decoder

### 27.1 ARCHITECTURE

Channels are expected to provide a pulse train with 90 degrees rotation as displayed in Figure 92 and Figure 93.

Depending on whether channel A or channel B is leading in phase, the quadrature decoding block calculates the direction on the related axis. Furthermore, the signed counter value represents the number of steps moved.

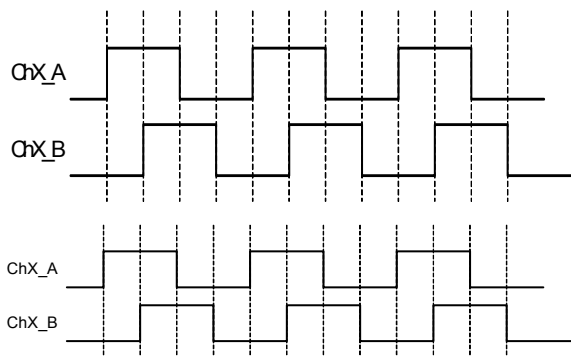


Figure 92: Moving forward on axis X

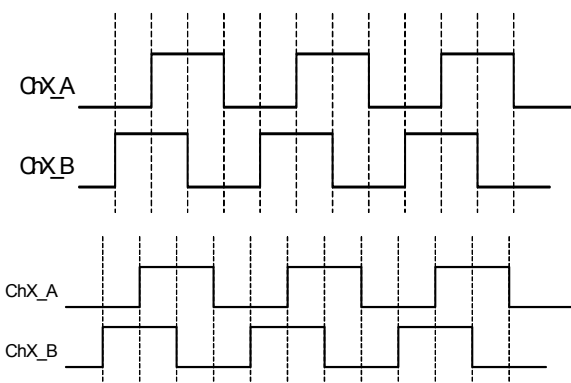


Figure 93: Moving backwards on axis X

### 27.2 PROGRAMMING

Since six channels are required (two for each axis), any GPIO can be mapped onto this block. The user can choose which GPIOs to use for the channels by programming the Pxy\_MODE\_REG[PID] with values from 30 up to 44.

The digital filter eliminates any spike shorter than two clock periods. The counter holds the movement events of the channel. When a channel is disabled the counter is reset. The counters are accessible via the APB bus.

The quadrature decoder operates on the system clock. The QDEC\_CLOCKDIV register defines the number of clock cycles of the period at which the decoding logic samples the data on the channel inputs.

The interrupt block monitors the movement events and generates an interrupt every N events. The value for N is defined in the QDEC\_CTRL register.

**Note:** if there are events from multiple channels in the same cycle, the interrupt event counter is only increased by one.

## 28 Keyboard Scanner

The Keyboard Scanner is a programmable hardware state machine which takes care of scanning the keyboard matrix and providing a clear view of key presses/releases to SW. Once enabled, it takes care of any key press or release without interfering with the CPU up to the point that a de-bounced key press is identified. It operates at a programmable frequency clock, allowing for a very short scan cycle time. Debouncing of up to 12 simultaneous keys is allowed with dedicated counters. The resulting events are reported in a FIFO which triggers an interrupt to the CPU at the end of the scan cycle.

The block diagram of the Keyboard Controller is presented in [Figure 94](#).

### Features

- Autonomous keyboard matrix scan without CPU interference
- 250 kHz clock for low power
- Maximum 16 Rows / 32 Columns support for any GPIO
- Up to 12, 7-bit de-bounce counters, counting matrix scan cycles
- Different press and release debounce times support
- Configurable row activation time
- Short scan cycle (1.3  $\mu$ s for a 10x10 matrix using 16 MHz clock)
- Row/Column report for a key press/release

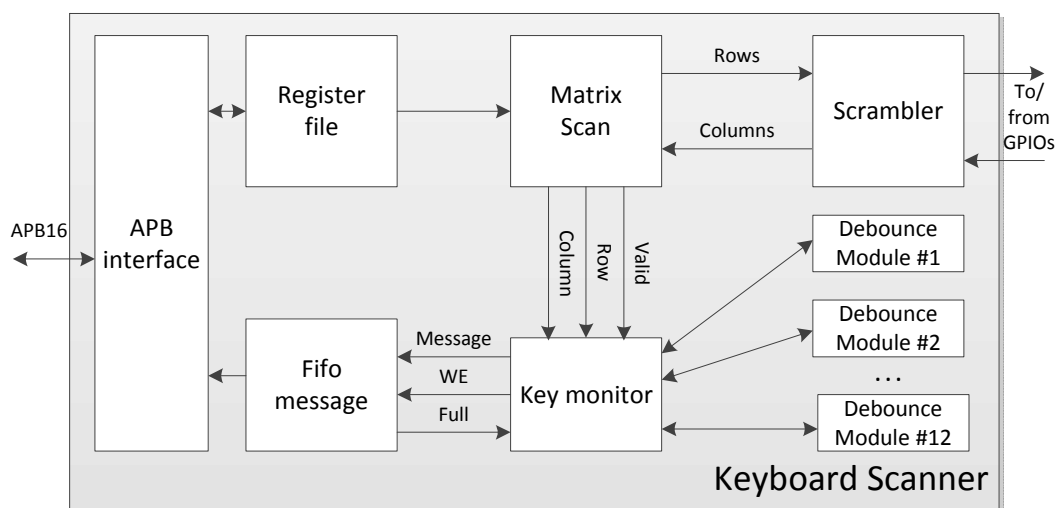


Figure 94: Keyboard Scanner block diagram

### 28.1 ARCHITECTURE

The Keyboard Scanner comprises a Key Monitoring state machine which controls the 12 Debouncing Modules, a Matrix Scan state machine which is responsible for a totally automated scan of the preferred key matrix, a FIFO which reports to the CPU what has been pressed/released and an APB based register file which is used to configure the block. A Scrambler is also utilized to allow for flexibility in selecting GPIOs used as rows or columns in the key matrix. Finally, the Key Monitor gathers the column/row identities for an event (key press or release), controls the 12 debouncing modules and writes the message into the FIFO triggering an interrupt to the CPU.

The Scrambler is implementing the assignment of specific GPIOs as rows (outputs) or columns (inputs) to form a matrix as shown in the example of [Figure 95](#).

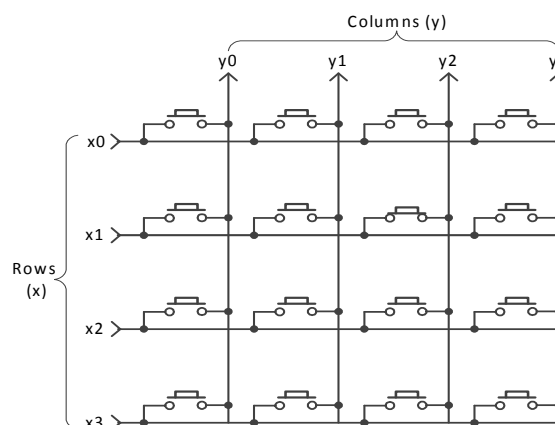


Figure 95: 4x4 Keyboard Matrix example



The configuration of the GPIOs can be realized with use of the KBSCN\_GPIOx\_MODE\_REG. The actual size of the matrix should be programmed in KBSCN\_MATRIX\_SIZE\_REG.

The Matrix Scan block implements a finite state

machine which takes care of the full scanning of the key matrix without the CPU intervention. The scan cycle is presented in the following figure:

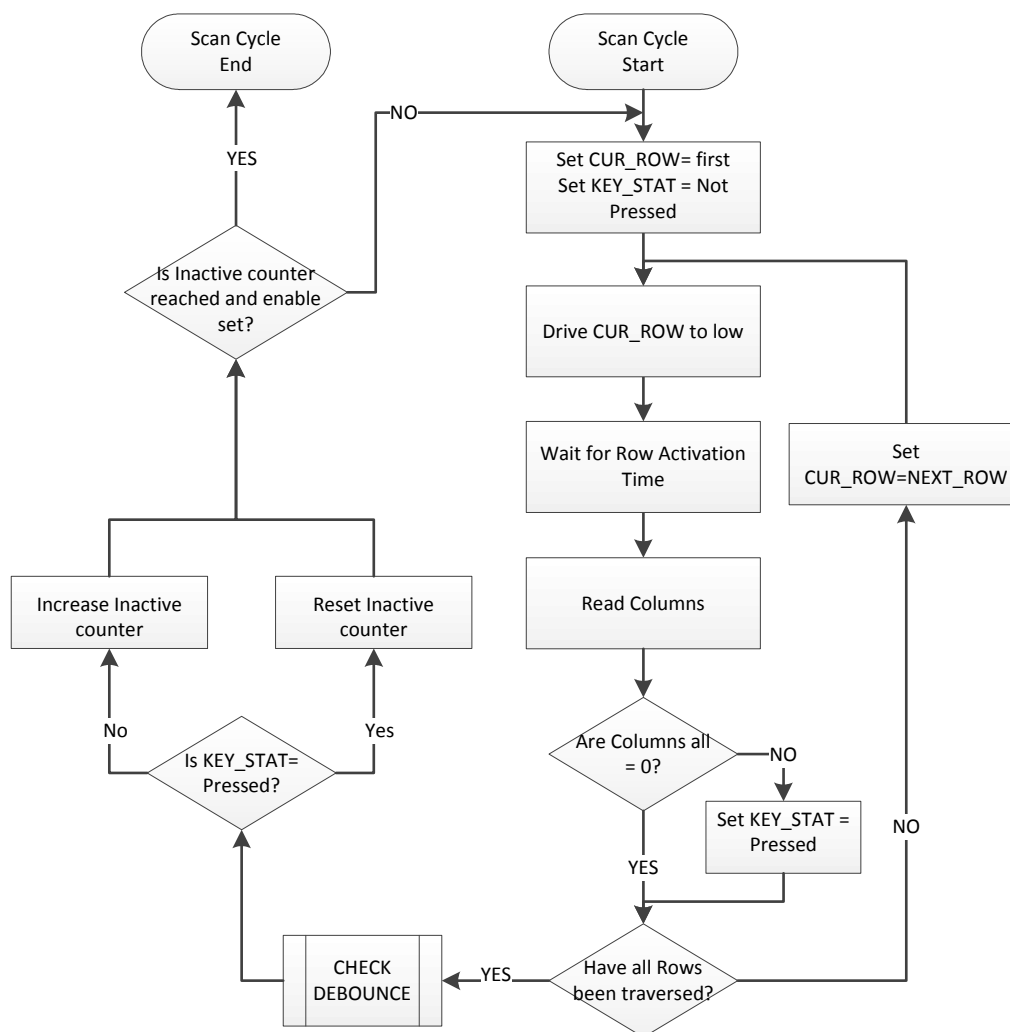


**Figure 96: Scan cycle state machine**

The FSM is setting each row to '0', waits for a pre-configured amount of time (activation time, programmed in KBSCN\_CTRL2\_REG) and then it reads the columns vector. If no zeros are sensed in the columns word, it continues scanning until all rows have been exercised.

If there is a key press while scanning, a flag (KEY\_STAT) changes value. Flag KEY\_STAT is also set when debouncing (for a key press or release) is in progress. As soon as the whole matrix has been scanned, the debouncer modules are started. These are incremented per scan cycle until a programmed value is reached (KBSCN\_DEBOUNCE\_REG). Up to 12 concurrent debouncing operations are supported.

Next, the FSM checks on the inactivity of the key

matrix. Setting bit KBSCN\_CTRL\_REG[KBSCN\_INACTIVE\_EN]=1 will automatically stop the scanning activity if KBSCN\_CTRL\_REG[KBSCN\_INACTIVE\_TIME] scan cycles have elapsed without any key press or debounce activity. If bit KBSCN\_CTRL\_REG[KBSCN\_INACTIVE\_EN]=0, the scanning activity will continue.

A key press/release message is prepared and stored in the FIFO for the CPU after every scan cycle provided that an event has been sensed. Each message is 11 bit wide and can be read at KBSCN\_MESSAGE\_KEY\_REG. Multiple messages are possible in the case of multiple events (presses or

releases). Its structure is presented in the table below:

**Table 49: Keyboard Scanner Message Structure**

| Bit | Description                                                                                                                                                  |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10  | Designates if this is the last message of this scan cycle. If it is high, then the value if the register is 0xFF (the rest of the bits are all high as well) |
| 9   | Key state. 0: key is pressed, 1: key is released                                                                                                             |
| 8:4 | Defines the number of the column of the key event                                                                                                            |
| 3:0 | Defines the number of the row of the key event                                                                                                               |

An interrupt is triggered towards the CPU in three cases:

1. A message has been placed in the message FIFO
2. There is an underflow/overflow of the message FIFO
3. The inactive time (in scan cycles) has elapsed

This information as well as the actual amount of messages residing in the FIFO can be accessed at KBSCN\_STATUS\_REG. Note that, the FIFO can be erased by SW using KBSCN\_CTRL\_REG[KBSCN\_RESET\_FIFO].

The number of KEYB\_CLK cycles required for a full scan cycle is given by the formula below:

$C_{scan} = N_{ROWS} * (C_{ACT} + 2)$ , where  $C_{ACT}$  is the amount of clocks for the activation time as programmed in KBSCN\_CTRL2\_REG[KBSCN\_ROW\_ACTIVE\_TIME] and  $N_{ROWS}$  the number of rows.

The clock can be 250 kHz up to 96 MHz if the PLL is enabled.

## 28.2 PROGRAMMING

To initialize the Keyboard Scanner, the clock has to be first enabled by setting CLK\_PER\_REG[KBSCAN\_ENABLE]=1 and KBSCN\_CTRL\_REG[KBSCN\_CLKDIV] to the preferred value. The latter defines the clock frequency of the block, when the clock divider is enabled by setting CLK\_PER\_REG[KBSCAN\_CLK\_SEL]=0.

Following that, the rows have to be programmed in output mode via the Pxy\_MODE\_REG registers. There is a dedicated PID for this namely number 47 (KB\_ROW). The rows have to be explicitly defined as pulled-up inputs with use of the respective Pxy\_MODE\_REG registers.

Both rows and columns need to be defined in the KBSCN\_GPIOz\_MODE\_REG so that the Keyboard Scanner understands which I/O is what.

The row active time, i.e. the time needed for the key matrix to settle after applying a high or low level, has to be programmed in the register field KBSCN\_CTRL2\_REG[KBSCN\_ROW\_ACTIVE\_TIME]. This value represents keyboard scanner clock cycles.

The Matrix size has to be configured in KBSCN\_MATRIX\_SIZE\_REG and the debounce times for press and release (can be different) should be programmed at KBSCN\_DEBOUNCE\_REG. Reset value is 0x1 which means no debounce at all.

A special feature enables the scanning to automatically stop after KBSCN\_CTRL\_REG[KBSCN\_INACTIVE\_TIME] clock cycles have elapsed without activity. Last thing to do for the initialization is enable the interrupt of the Keyboard Scanner by KBSCN\_CTRL\_REG[KBSCN\_IRQ\_FIFO\_MASK] = 1 and KBSCN\_CTRL\_REG[KBSCN\_IRQ\_MESSAGE\_MASK] = 1 and enable the block by setting KBSCN\_CTRL\_REG[KBSCN\_EN]=1.

The interrupt triggers the CPU to read a message residing in the 12-words FIFO by reading the KBSCN\_MESSAGE\_KEY\_REG. This message contains the key id that was recently pressed or released as well as the indication of being or not, the only message in the FIFO (KBSCN\_LAST\_ENTRY).

## 29 Software Timers

The Software Timer block contains 3 timer modules that are software controlled, programmable and can be used for various tasks. Timer0 is a 16-bit general purpose timer with a PWM output capability. Timer2 is a 14-bit counter that generates three identical PWM signals in a quite flexible manner.

### 29.1 TIMER0

Timer0 is a 16-bit general purpose software programmable timer, which has the ability of generating Pulse Width Modulated signals, namely PWM0 and PWM1. It also generates the SWTIM\_IRQ interrupt to the ARM Cortex M0. It can be configured in various modes regarding output frequency, duty cycle and the modulation of the PWM signals.

#### Features

- 16-bit general purpose timer
- Ability to generate 2 Pulse Width Modulated signals (i.e. PWM0 and PWM1)
- Programmable output frequency:  

$$f = (16, 8, 4, 2 \text{ MHz or } 32 \text{ kHz}) / (M+1) \cdot (N+1)$$
 with  $N = 0$  to  $(2^{**}16)-1$ ,  $M = 0$  to  $(2^{**}16)-1$
- Programmable duty cycle:  

$$\delta = (M+1) / ((M+1) + (N+1)) \cdot 100\%$$
- Separately programmable interrupt timer:  

$$T = (16, 8, 4, 2 \text{ MHz or } 32 \text{ kHz}) / (ON+1)$$

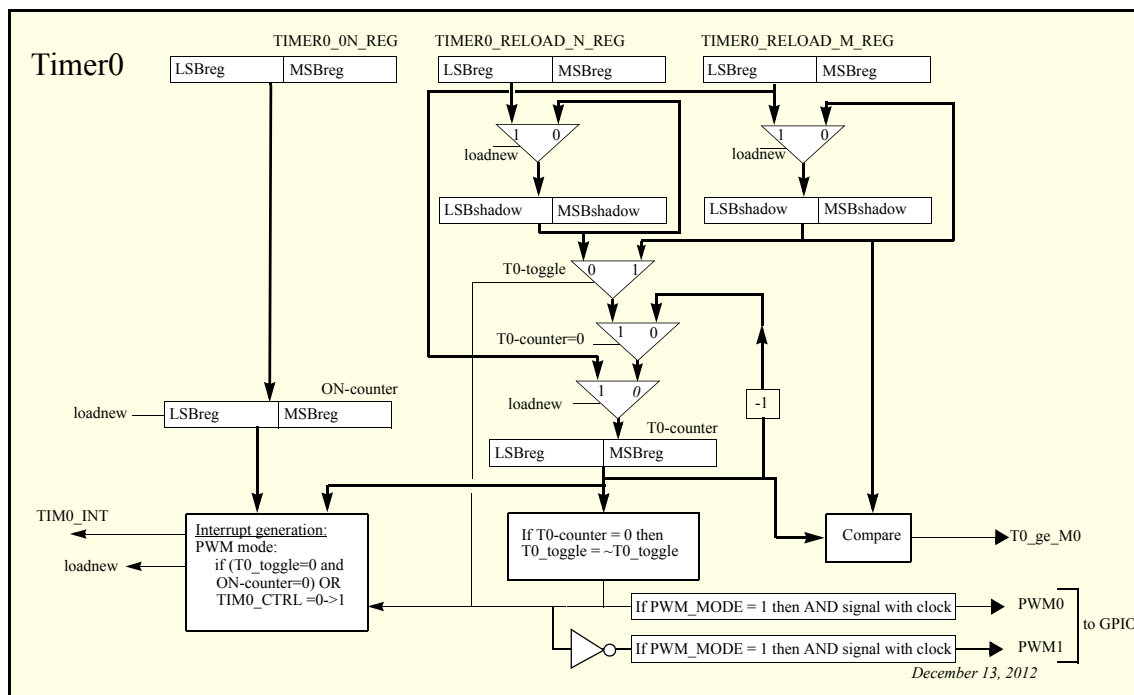


Figure 97: Timer0 block diagram

Figure 97 shows the block diagram of Timer0. The 16 bits timer consists of two counters: T0-counter and ON-counter, and three registers: TIMER0\_RELOAD\_M\_REG, TIMER0\_RELOAD\_N\_REG and TIMER0\_ON\_REG. Upon reset, the counter and register values are 0x0000. Timer0 will generate a Pulse Width Modulated signal PWM0. The frequency and duty cycle of PWM0 are determined by the contents of the TIMER0\_RELOAD\_N\_REG and the TIMER0\_RELOAD\_M\_REG registers.

The timer can run at five different clocks: 16 MHz, 8 MHz, 4 MHz, 2 MHz or 32 kHz. The 32 kHz clock is selected by default with bit TIM0\_CLK\_SEL in the TIMER0\_CTRL\_REG register. This 'slow' clock has no

enabling bit. The other four options can be selected by setting the TIM0\_CLK\_SEL bit and the TMR\_ENABLE bit in the CLK\_PER\_REG (default disabled). This register also controls the frequency via the TMR\_DIV bits. An extra clock divider is available that can be activated via bit TIM0\_CLK\_DIV of the timer control register TIMER0\_CTRL\_REG. This clock divider is only used for the ON-counter and always divides by 10.

Timer0 operates in PWM mode. The signals PWM0 and PWM1 can be mapped to any GPIOs.

#### Timer0 PWM mode

If bit TIM0\_CTRL in the TIMER0\_CTRL\_REG is set, Timer0 will start running. SWTIM\_IRQ will be generated and the T0-counter will load its start value from

the `TIMER0_RELOAD_M_REG` register, and will decrement on each clock. The ON-counter also loads its start value from the `TIMER0_ON_REG` register and decrements with the selected clock.

When the T0-counter reaches zero, the internal signal T0-toggle will be toggled to select the `TIMER0_RELOAD_N_REG` whose value will be loaded in the T0-counter. Each time the T0-counter reaches zero it will alternately be reloaded with the values of the M0- and N0-shadow registers respectively. PWM0 will be high when the M0-value decrements and low when the N0-value decrements. For PWM1 the opposite is applicable since it is inverted. If bit `PWM_MODE` in the `TIMER0_CTRL_REG` register is set, the PWM signals are not HIGH during the 'high time' but output a clock in that stage. The frequency is based on the clock settings defined in the `CLK_PER_REG` register (also in 32 kHz mode), but the selected clock frequency is divided by two to get a 50 % duty cycle.

If the ON-counter reaches zero it will remain zero until the T0-counter also reaches zero, while decrementing the value loaded from the `TIMER0_RELOAD_N_REG` register (PWM0 is low). The counter will then generate an interrupt

(`SWTIM_IRQ`). The ON-counter will be reloaded with the value of the `TIMER0_ON_REG` register. The T0-counter as well as the M0-shadow register will be loaded with the value of the `TIMER0_RELOAD_M_REG` register. At the same time, the N0-shadow register will be loaded by the `TIMER0_RELOAD_N_REG` register. Both counters will be decremented on the next clock again and the sequence will be repeated.

Note that it is possible to generate interrupts at a high rate, when selecting a high clock frequency in combination with low counter values. This could result in missed interrupt events.

During the time that the ON-counter is non-zero, new values for the ON-register, M0-register and N0-register can be written, but they are not used by the T0-counter until a full cycle is finished. More specifically, the newly written values in the `TIMER0_RELOAD_M_REG` and `TIMER0_RELOAD_N_REG` registers are only stored into the shadow registers when the ON-counter and the T0-counter have both reached zero and the T0-counter was decrementing the value loaded from the `TIMER0_RELOAD_N_REG` register (see [Figure 98](#)).

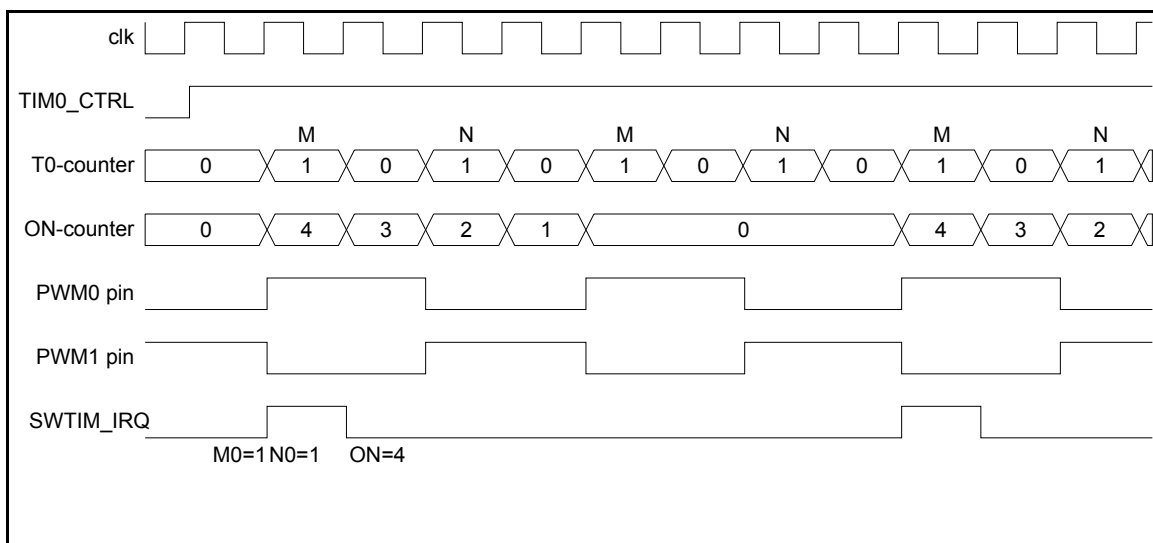


Figure 98: Timer 0 Pulse PWM mode

At start-up both counters and the PWM0 signal are LOW so also at start-up an interrupt is generated. If Timer0 is disabled all flip-flops, counters and outputs are in reset state except for the ON-register, the `TIMER0_RELOAD_N_REG` register and the `TIMER0_RELOAD_M_REG` register.

The timer input registers ON-register, `TIMER0_RELOAD_N_REG` and `TIMER0_RELOAD_M_REG` can be written and the counter registers ON-counter and T0-counter can be read. When reading from the address of the ON-register, the value of the ON-counter is returned. Reading

from the address of either the `TIMER0_RELOAD_N_REG` or the `TIMER0_RELOAD_M_REG` register, returns the value of the T0-counter.

It is possible to freeze Timer0 with bit `FRZ_SWTIM` of the register `SET_FREEZE_REG`. When the timer is frozen the timer counters are not decremented. This will freeze all the timer registers at their last value. The timer will continue its operation again when bit `FRZ_SWTIM` is cleared via register `RESET_FREEZE_REG`.

## 29.2 TIMER1

Timer1 is a 16-bit up/down timer with capture input (2 channels) and one shot pulse capability. This timer is also able of generating a PWM signal with programmable output frequency. Timer1 is the only timer that stays alive when the DA14680 is in sleep mode. It counts using the sleep clock and can still control one output, namely the P0\_6 while the rest of the IOs are frozen to reduce power dissipation.

### Features

- 16-bit general purpose timer
- Generates a Pulse Width Modulated signal (PWM5)
- 2 channels for capture input triggering
- One shot pulse with programmable pulse width
- 16-bit clock pre-scaler
- Up/down counting capability with free running mode
- Operating at sleep clock when system is in sleep mode
- Counts while in Extended Sleep mode.
- PWM5 controls the P0\_6 output while in Extended Sleep mode
- Programmable output frequency  
 $F = 16 \text{ MHz}/2 \text{ to } 16 \text{ MHz}/((2^{14})-1)$
- Dedicated interrupt line

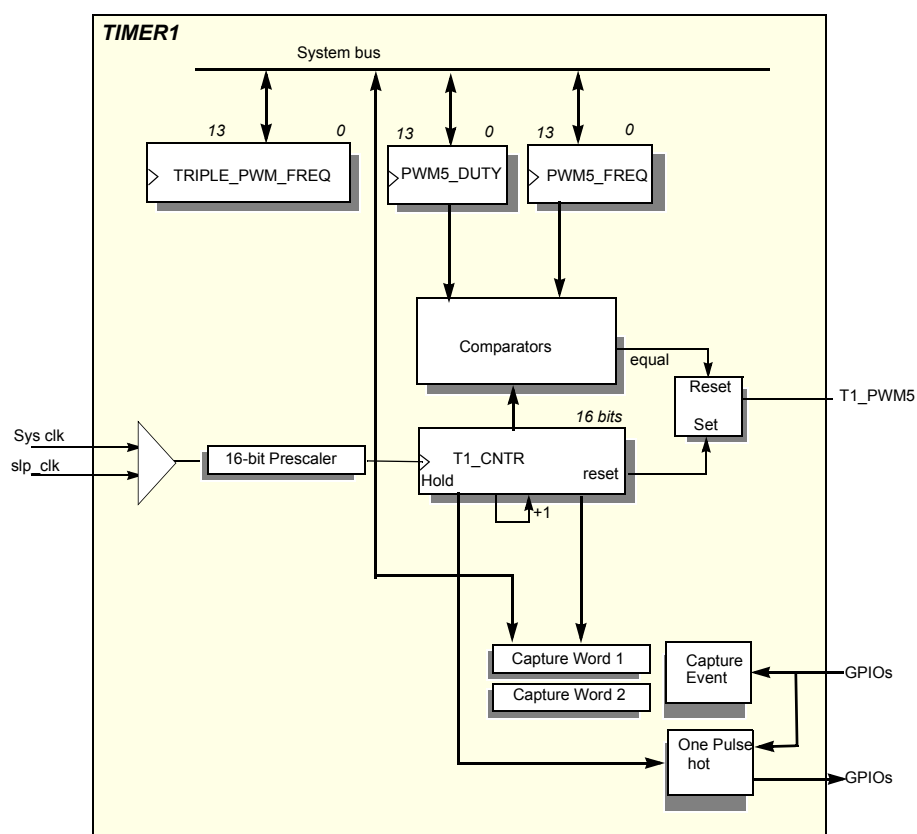


Figure 99: PWM Timer 1 block diagram

Timer1 is enabled by CAPTIM\_CTRL\_REG[CAPTIM\_EN] bit. This is the only timer that supports up/down counting with means of programming the CAPTIM\_CTRL\_REG[CAPTIM\_COUNT\_DOWN\_EN] bit.

Timer1 supports capturing 2 externally triggered events (positive or negative edges) on GPIOs that can be selected. When the first event is captured, the current value of the 16-bit free-running timer is stored into

CAPTIM\_CAPTURE\_GPIO1\_REG while upon triggering of the second event, the current timer value is stored into CAPTIM\_CAPTURE\_GPIO2\_REG. In this way, the timing interval between 2 successive events can be measured with precision.

The Timer1 support one shot pulse (CAPTIM\_CTRL\_REG[CAPTIM\_ONESHOT\_MODE\_EN]). Whenever either of the two externally selected GPIOs triggers an event, a programmable width pulse

(CAPTIM\_SHOTWIDTH\_REG) will be output on another GPIO programmed by Pxx\_MODE\_REG[PID]=53. Furthermore, the time up to the start of the pulse is again configurable by CAPTIM\_RELOAD\_REG.

Timer1 is kept alive while the system is in Extended Sleep mode. The PWM5 signal which can be configured by CAPTIM\_PWM\_FREQ\_REG and CAPTIM\_PWM\_DC\_REG with respect to the frequency and duty cycle is automatically connected to the P0\_6 pin output buffer. Hence, if enabled (CLK\_TMR\_REG[P06\_TMR1\_PWM\_MODE]=1), P0\_6

can be driving external devices even during extended sleep. However, Timer1 will not work in sleep mode if CLK\_TMR\_REG[TMR1\_DIV] has a value different than 0.

Note that, the PWM functionality is decoupled from the timer1 counting which can be read at CAPTIM\_TIMER\_VAL\_REG at any given time while the chip is in active mode.

An overview of the timer modes, reset values and interrupt generation is presented in the following table:

**Table 50: Timer1 interrupt generation**

| Mode                   | Reset Value       | Interrupt Generated       |
|------------------------|-------------------|---------------------------|
| Count up with Reload   | 0                 | When reload value reached |
| Count down with Reload | CAPTIM_RELOAD_REG | When 0 reached            |
| Count up free running  | 0                 | When reload value reached |

### 29.3 TIMER2

Timer2 has three Pulse Width Modulated (PWM) outputs. The block diagram is shown in [Figure 100](#).

#### Features

- 14-bit general purpose timer
- Generates 3 Pulse Width Modulated signals (i.e. PWM2, PWM3 and PWM4)
- Input Clock frequency 16 MHz
- Programmable output frequency  
 $F = 16 \text{ MHz} / 2 \text{ to } 16 \text{ MHz} / ((2^{**14}) - 1)$
- Three outputs with Programmable duty cycle from 0% to 100%
- Used for white LED intensity (on/off) control

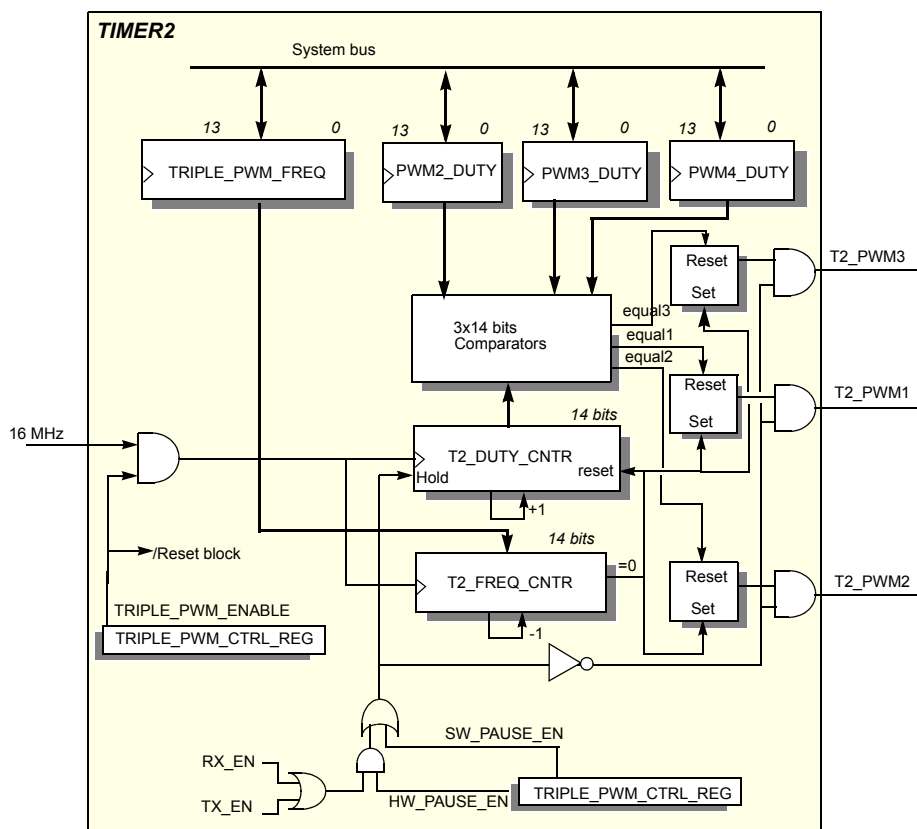


Figure 100: PWM Timer 2 block diagram

The Timer2 is clocked with the system clock (16 MHz) and can be enabled with `TRIPLE_PWM_CTRL_REG[TRIPLE_PWM_ENABLE]`.

`T2_FREQ_CNTR` determines the output frequency of the `T2_PWMn` output. This counter counts down from the value stored in register `TRIPLE_PWM_FREQUENCY`. At counter value 0, `T2_FREQ_CNTR` sets the `T2_PWMn` output to '1' and the counter is reloaded again.

`T2_DUTY_CNTR` is an up-counter that determines the duty cycle of the `T2_PWMn` output signal. After the block is enabled, the counter starts from 0. If `T2_DUTY_CNTR` is equal to the value stored in the respective `PWMn_DUTY_CYCLE` register, this resets the `T2_PWMn` output to 0. `T2_DUTY_CNTR` is reset when `TRIPLE_PWM_FREQUENCY` is 0.

Note that the value of `PWMn_DUTY_CYCLE` must be less or equal than `TRIPLE_PWM_FREQUENCY`.

Another feature of Timer2 PWM signals is the fact that the start and end cycle of the PWM wave can be programmed separately for the three signals using the `PWMx_START_CYCLE` and `PWMx_END_CYCLE`. In this way, the PWM signals can be configured to imple-

ment a certain phase shift. These signals also control the LED pins with means of `LED_CONTROL_REG[LEDx_SRC_SEL]`.

The Timer2 is enabled/disabled by programming the `TRIPLE_PWM_CTRL_REG[TRIPLE_PWM_EN]` bit.

The timing diagram of Timer2 is shown in Figure 101.

#### Freeze function

During RF activity it may be desirable to temporarily suppress the PWM switching noise. This can be done by setting `TRIPLE_PWM_CTRL_REG[HW_PAUSE_EN] = 1`. The effect is that whenever there is a transmission or a reception process from the Radio, `T2_DUTY_CNTR` is frozen and `T2_PWMx` output is switched to '0' to disable the selected `T2_PWM1`, `T2_PWM2`, `T2_PWM3`. As soon as the Radio is idle (i.e. `RX_EN` or `TX_EN` signals are zero), `T2_DUTY_CNTR` resumes counting and finalizes the remaining part of the PWM duty cycle.

`TRIPLE_PWM_CTRL_REG[SW_PAUSE_EN]` can be set to '0' to disable the automatic, hardware driven freeze function of the duty counter and keep the duty cycle constant.



Note that the RX\_EN and TX\_EN signals are not software driven but controlled by the BLE core hardware.

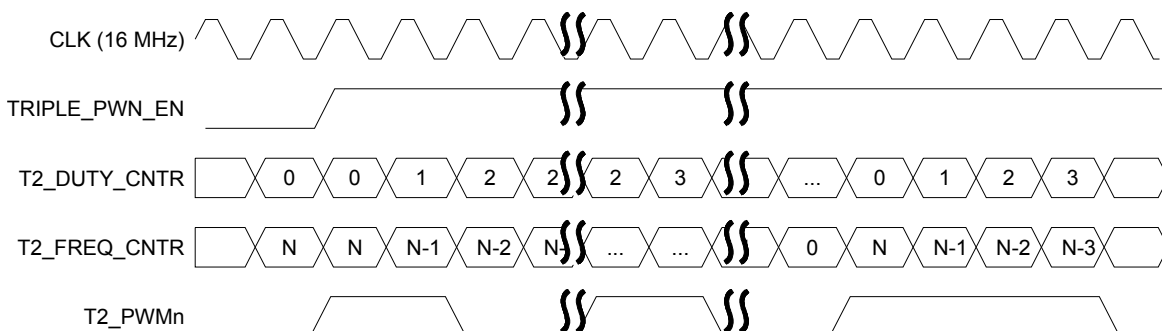


**Figure 101: Timer 2 PWM timing diagram**

#### 29.4 BRTH\_TIMER

This timer implements an automated breathing function for external LEDs without software interference. The resulting PWM signal can be mapped on any of the GPIOs of the system. It uses the system clock as input.

##### Features

- Maximum and minimum duty cycle configuration
- PWM duty cycle step granularity up to 256
- Input clock frequency 16 MHz
- Programmable output frequency  
 $f = f_{\text{system}} / (1 \text{ to } 256) \text{ MHz}$

### 30 Watchdog Timer

The watchdog timer is an 8-bit timer with sign bit that can be used to detect an unexpected execution sequence caused by a software run-away and can generate a full system reset or a Non-Maskable Interrupt (NMI). Upon expiration, a HW reset is triggered.

#### Features

- 8 bits down counter with sign bit, clocked with a 10.24 ms clock for a maximum 2.6 s time-out.

- Non-Maskable Interrupt (NMI) or WDOG reset.
- Optional automatic WDOG reset if NMI handler fails to update the Watchdog register.
- Non maskable Watchdog freeze of the Cortex-M0 Debug module when the Cortex-M0 is halted in Debug state.
- Maskable Watchdog freeze by user program.

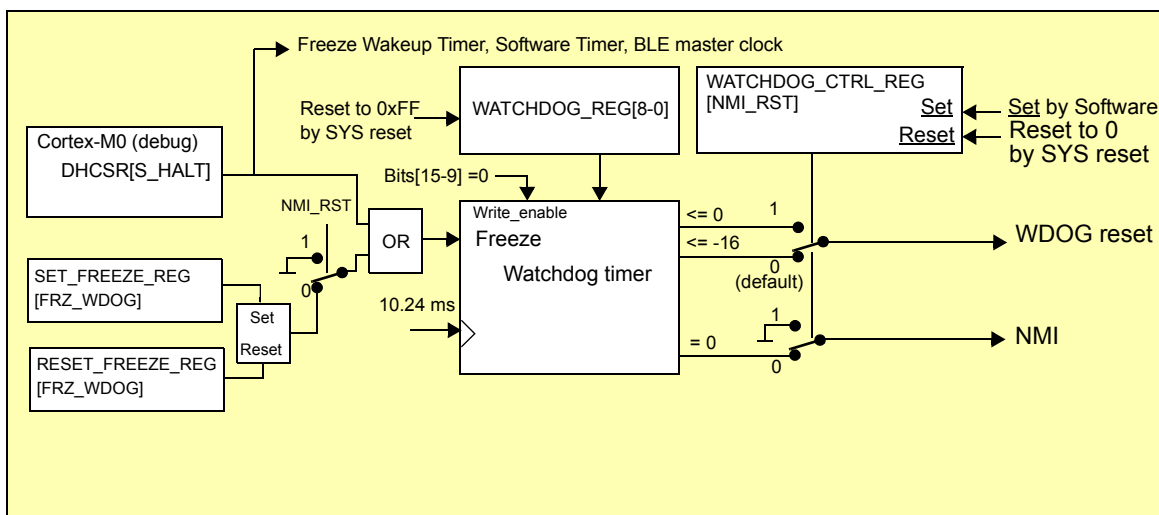


Figure 102: Watchdog Timer block diagram

The 8 bits watchdog timer is decremented by 1 every 10.24 ms. The timer value can be accessed through the WATCHDOG\_REG register which is set to 255 ( $FF_{16}$ ) at reset. This results in a maximum watchdog time-out of  $\sim 2.6$  s. During write access the WATCHDOG\_REG[WDOG\_WEN] bits must be 0. This provides extra filtering for a software run-away writing all ones to the WATCHDOG\_REG. If the watchdog counter reaches 0, the counter value will get a negative value by setting bit 8. The counter sequence becomes 1, 0,  $1FF_{16}$  (-1),  $1FE_{16}$  (-2), ...,  $1F0_{16}$  (-16).

If WATCHDOG\_CTRL\_REG[NMI\_RST] = 0, the watchdog timer will generate an NMI if the watchdog timer reaches 0 and a WDOG reset if the counter becomes less or equal to -16 ( $1F0_{16}$ ). The NMI handler must write any value  $> -16$  to the WATCHDOG\_REG to prevent the generation of a WDOG reset at counter value -16 after  $16 \times 10.24 = 163.8$  ms.

If WATCHDOG\_CTRL\_REG[NMI\_RST] = 1, the watchdog timer generates a WDOG reset if the timer becomes less or equal than 0.

The WDOG reset is one of the SYS (system) reset sources and resets the whole device, including setting the WATCHDOG\_REG register to 255, except for the RST pin, the Power On reset, the HW reset and the DBG (debug module) reset. Since the HW reset is not

triggered, the SYS\_CTRL\_REG[REMAP\_ADR0] bits will retain their value and the Cortex-M0 will start executing again from the current selected memory at address zero. Refer to the "Reset" chapter for an overview of the complete reset circuit and conditions.

For debugging purposes, the Cortex-M0 Debug module can always freeze the watchdog by setting the DHCSR[DBGKEY | C\_HALT | C\_DEBUGEN] control bits (reflected by the status bit S\_HALT). This is automatically done by the debug tool, e.g. during step-by-step debugging. Note that this bit also freezes the Wakeup Timer, the Software Timer and the BLE master clock. For additional information also see the DEBUG\_REG[DEBUGS\_FREEZE\_EN] mask register. The C\_DEBUGEN bit is not accessible by the user software to prevent freezing the watchdog.

In addition to the S\_HALT bit, the watchdog timer can also be frozen if NMI\_RST=0 and SET\_FREEZE\_REG[FRZ\_WDOG] is set to '1'. The watchdog timer resumes counting when RESET\_FREEZE\_REG[FRZ\_WDOG] is set to '1'. The WATCHDOG\_CTRL\_REG[NMI\_RST] bit can only be set by software and will only be reset on a SYS reset.

### 31 USB Interface

The USB interface is an integrated USB Node controller compatible with the full and low speed USB specification version 1.1.

It integrates a Serial Interface Engine (SIE) and USB endpoint (EP) FIFOs. Seven endpoint pipes are supported: one for the mandatory control endpoint and six to support interrupt endpoints. Each endpoint pipe has a dedicated FIFO, 8 bytes for the control endpoint and 64 bytes for the other endpoints.

The USB transceiver module is accessed through USB\_Dp and USB\_Dm pins.

#### Features

- Full Speed USB node
- Interfaces to USB V1.1 transceiver with programmable rise and fall times and integrated D+/D- pull-up resistors.
- Serial Interface Engine (SIE) consisting of a Media Access Controller (MAC), USB Specification 1.0 and 1.1 compliant
- USB Function Controller with seven FIFO-based Endpoints:
  - One bidirectional Control Endpoint 0 (8 bytes)
  - Three Transmit Endpoints (64 bytes each)
  - Three Receive Endpoints (64 bytes each)
- Automatic Data PID toggling/checking and NAK packet recovery (maximum 256 x32 bytes of data = 8 kB)

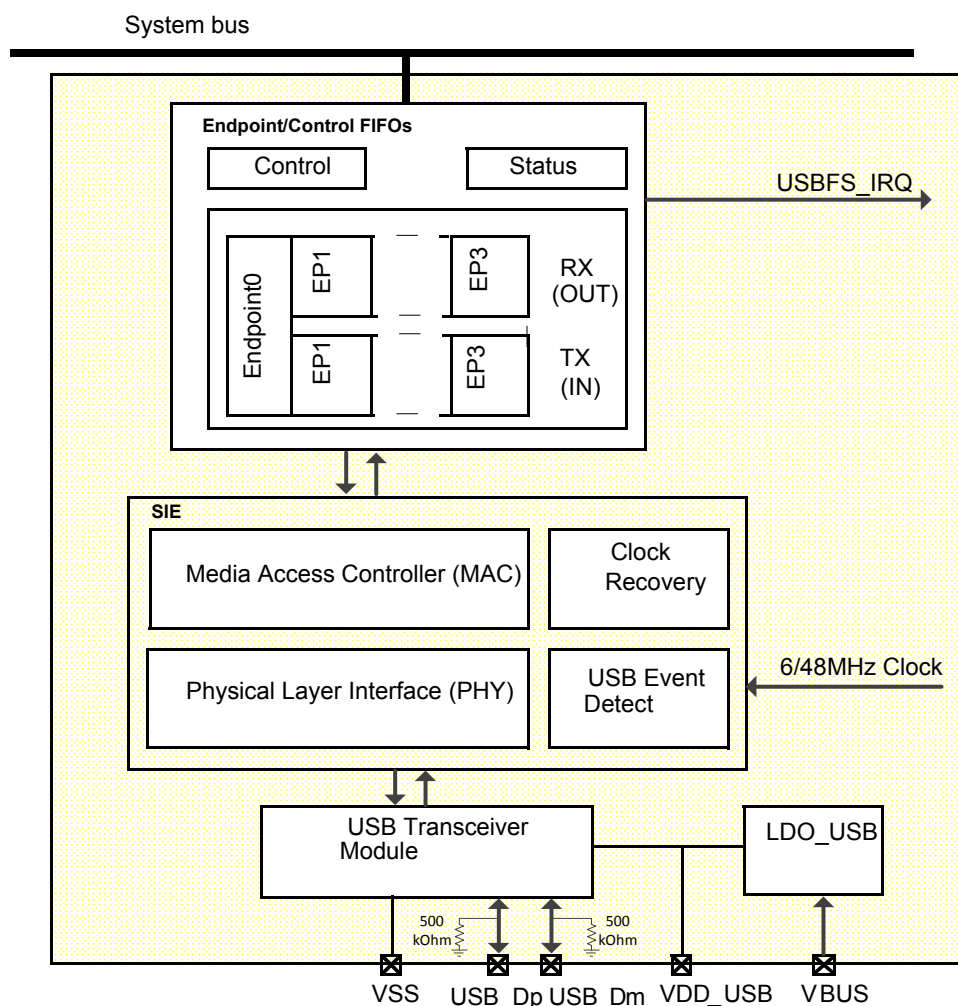


Figure 103: USB Node block diagram

### 31.1 SERIAL INTERFACE ENGINE

The SIE is comprised of physical (PHY) and Media Access Controller (MAC) modules. The PHY module includes the digital-clock recovery circuit, a digital glitch filter, End Of Packet (EOP) detection circuitry, and bit stuffing and unstuffing logic. The MAC module includes packet formatting, CRC generation and checking, and endpoint address detection. It provides the necessary control to give the NAK, ACK and STALL responses as determined by the Endpoint Pipe Controller (EPC) for the specified endpoint pipe. The SIE is also responsible for detecting and reporting USB-specific events, such as NodeReset, NodeSuspend and NodeResume. The module output signals to the transceiver are well matched (under 1 ns) to minimize skew on the USB signals.

The USB specifications assign bit stuffing and unstuffing as the method to ensure adequate electrical transitions on the line to enable clock recovery at the receiving end. The bit stuffing block ensures that whenever a string of consecutive 1's is encountered, a 0 is inserted after every sixth 1 in the data stream. The bit unstuffing logic reverses this process.

The clock recovery block uses the incoming NRZI data to extract a data clock (12 MHz FS, 1.5 MHz LS) from a 48 (FS) / 6(LS) MHz input clock. This clock is used in the data recovery circuit. The output of this block is binary data (decoded from the NRZI stream) which can be appropriately sampled using the extracted 12(6) MHz clock. The jitter performance and timing characteristics meet the requirements set forth in Chapter 7 of the USB Specification.

### 31.2 ENDPOINT PIPE CONTROLLER (EPC)

The EPC provides the interface for USB function endpoints. An endpoint is the ultimate source or sink of data. An endpoint pipe facilitates the movement of data between USB and memory, and completes the path between the USB host and the function endpoint. According to the USB specification, up to 31 such endpoints are supported at any given time. USB allows a total of 16 unidirectional endpoints for receive and 16 for transmit. As the control endpoint 0 is always bidirectional, the total number is 31. The Full/Low Speed USB node supports a maximum of seven endpoint pipes with the same function address. See [Figure 104](#): for a schematic diagram of EPC operation.

A USB function is a USB device that is able to transmit and receive information on the bus. A function may have one or more configurations, each of which defines the interfaces that make up the device. Each interface, in turn, is composed of one or more endpoints.

Each endpoint is an addressable entity on USB and is required to respond to IN and OUT tokens from the USB host (typically a PC). IN tokens indicate that the host has requested to receive information from an endpoint, and OUT tokens indicate that it is about to send information to an endpoint.

On detection of an IN token addressed to an endpoint, the endpoint pipe should respond with a data packet. If the endpoint pipe is currently stalled, a STALL handshake packet is sent under software control. If the endpoint pipe is enabled but no data is present, a NAK (Negative Acknowledgment) handshake packet is sent automatically. If the endpoint pipe is isochronous and enabled but no data is present, a bit stuff error followed by an end of packet is sent on the bus.

Similarly, on detection of an OUT token addressed to an endpoint, the endpoint pipe should receive a data packet sent by the host and load it into the appropriate FIFO. If the endpoint pipe is stalled, a STALL handshake packet is sent. If the endpoint pipe is enabled but no buffer is present for data storage, a NAK handshake packet is sent.

A disabled endpoint does not respond to IN, OUT, or SETUP tokens.

The EPC maintains separate status and control information for each endpoint pipe.

For IN tokens, the EPC transfers data from the associated FIFO to the host. For OUT tokens, the EPC transfers data in the opposite direction.

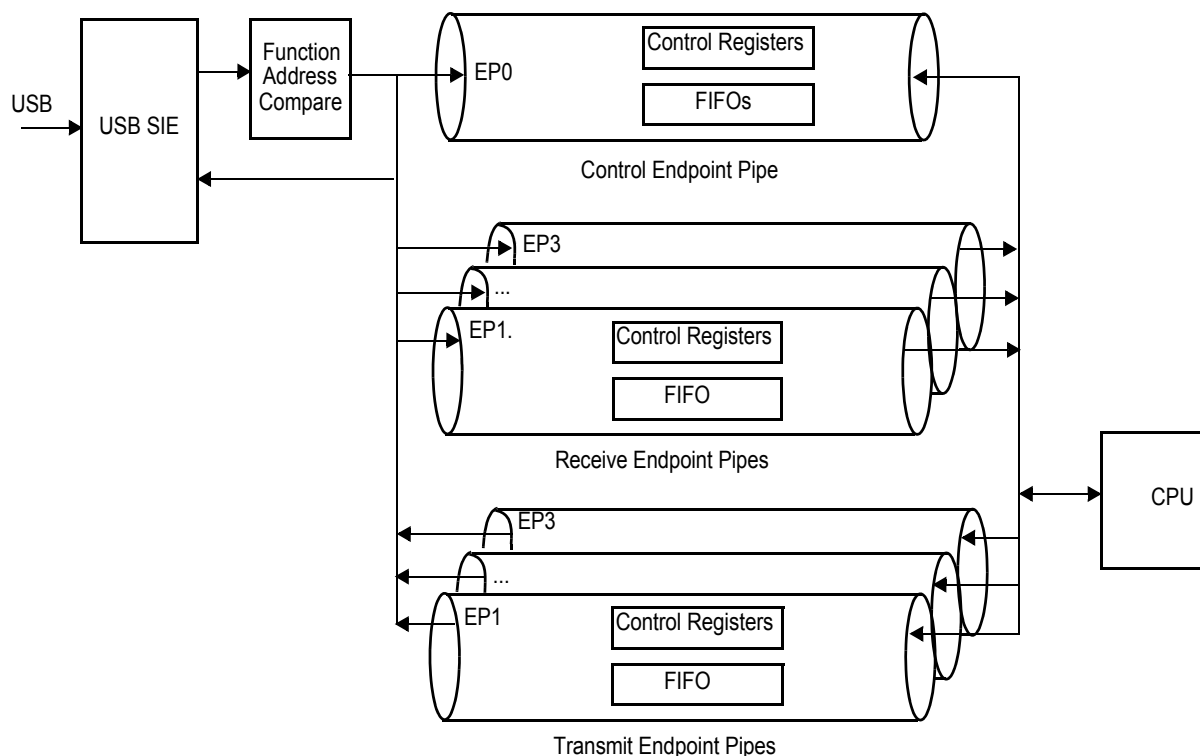


Figure 104: Endpoint Operation

### 31.3 FUNCTIONAL STATES

#### 31.3.1 Line Condition Detection

At any given time, the USB node is in one of the following states (see "Functional states" on page 134 for the functional state transitions):

- **NodeOperational:** Normal operation
- **NodeSuspend:** Device operation suspended due to USB inactivity
- **NodeResume:** Device wake-up from suspended state
- **NodeReset:** Device reset

The NodeSuspend, NodeResume, or NodeReset line condition causes a transition from one operating state to another. These conditions are detected by specialized hardware and reported via the Alternate Event (ALTEV) register. If interrupts are enabled, an interrupt is generated upon the occurrence of any of the specified conditions.

**NodeOperational State:** This is the normal operating state of the node. In this state, the node is configured for operation on the USB.

**NodeSuspend State:** A USB node is expected to enter NodeSuspend state when 3 ms have elapsed without any detectable bus activity. The USB node looks for

this event and signals it by setting the SD3 bit in the USB\_ALTEV register, which causes an USB\_INT, if enabled, to be generated. The firmware should respond by putting the USB node in NodeSuspend state.

Refer to "" on page 143 on how to obtain to lowest power consumption in the NodeSuspend state.

The USB node can resume normal operation in two ways:

- **Host initiated.** By detecting a resume signalling followed by Low speed EOP on the USB bus, an ALTEV[RESUME] interrupt is generated. The firmware responds by setting the **NodeOperational** in the USB\_NFRS\_REG (See also section on page 143)
- **Device initiated.** By detection of a local event, e.g GPIO key is pressed, a KEYB\_INT is generated. The firmware releases the USB node from NodeSuspend state by initiating a **NodeResume** on the USB using the NFRS register. The node firmware must ensure at least 5 ms of Idle on the USB, by checking the SD5 in the USB\_ALTEV before going to the NodeResume state.

**NodeResume State:** In NodeResume state, a constant "K" is signalled on the USB. This should last for at least 1 ms and no more than 15 ms, after which the USB host should continue sending the NodeResume signal

for at least an additional 20 ms, and then completes the NodeResume operation by issuing the End Of Packet (EOP) sequence.

To successfully detect the EOP, the firmware must respond by setting **NodeOperational** in the USB\_NFRS\_REG. Upon detection on the EOP, the USB\_ALTEV\_REG[EOP] is set.

If no EOP is received from the host within 100 ms, the software must re-initiate NodeResume.

*NodeReset:* When detecting a NodeResume or NodeReset signal while in NodeSuspend state, the USB node can signal this to the CPU by generating an interrupt.

USB specifications require that a device must be ready to respond to USB tokens within 10 ms after wake-up or reset.

### 31.4 FUNCTIONAL STATE DIAGRAM

Figure 105 shows the device states and transitions, as well as the conditions that trigger each transition. All Full/Low Speed USB node state transitions are initiated by the firmware.

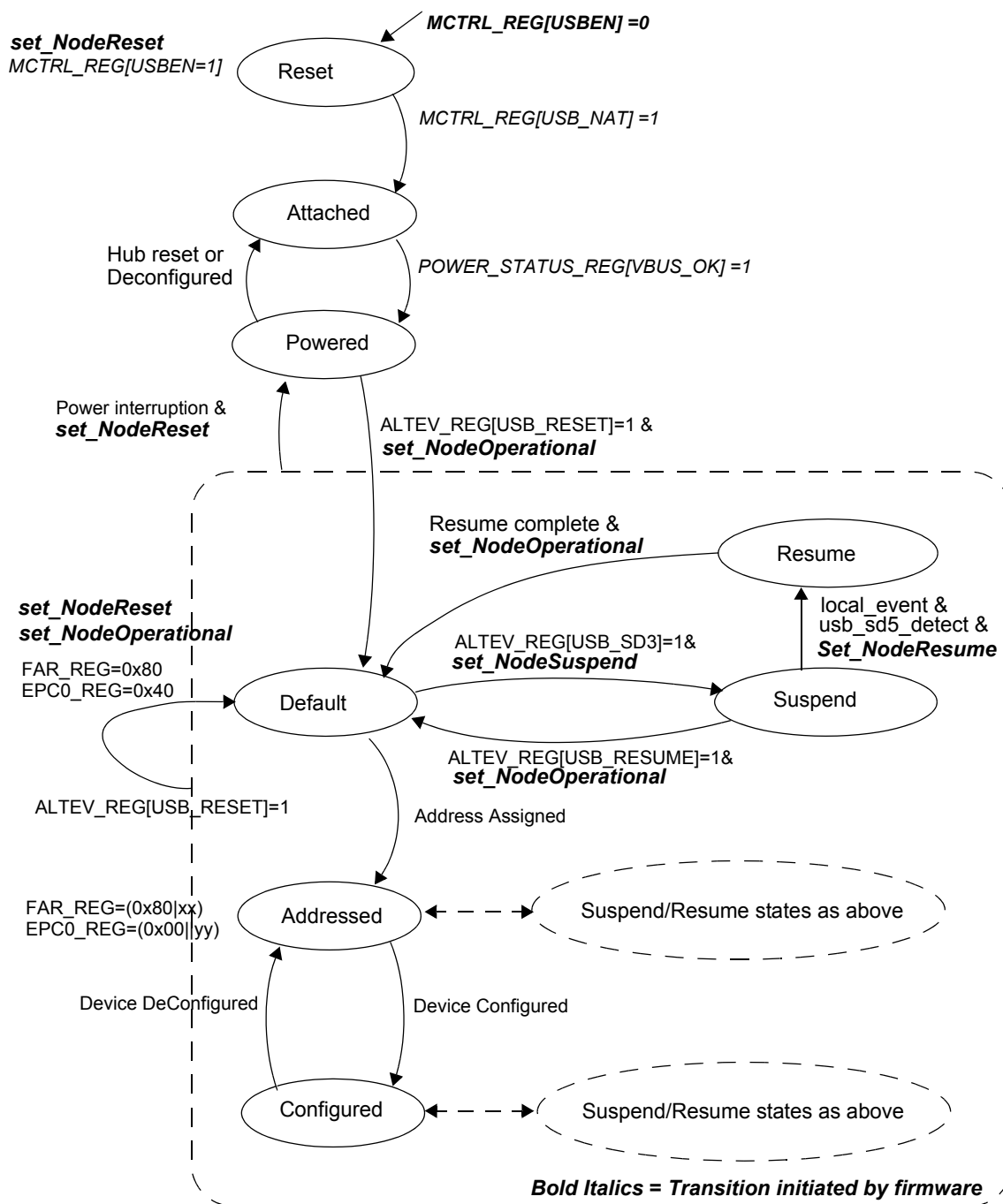


Figure 105: Node Functional State Diagram

**Note 18:** When the node is not in NodeOperational state, all registers are frozen with the exception of the endpoint controller state machines, and the TX\_EN, LAST and RX\_EN bits which are reset.

**Note 19:** In NodeResume state, resume signalling is propagated upstream.

**Note 20:** In NodeSuspend state, the node may enter a low power state and is able to detect resume signalling.



Table 51: Functional states

| State Transition   | Condition Asserted                                                                                                                                                                                                                                 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| set_NodeReset      | Node Functional State register NFS[1:0] bits are written with 00 <sub>b</sub><br>(The firmware should only initiate set_NodeReset if RESET in the ALTEV register is set.)                                                                          |
| set_NodeSuspend    | Node Functional State register NFS[1:0] bits are written with 11 <sub>b</sub><br>The firmware should only initiate set_suspend if SD3 in the ALTEV register is set.                                                                                |
| set_NodeOperation  | Node Functional State register NFS[1:0] bits are written with 10 <sub>b</sub>                                                                                                                                                                      |
| set_NodeResume     | Node Functional State register NFS[1:0] bits are written with 01 <sub>b</sub><br>The firmware should only initiate clear_suspend if SD5 in the ALTEV register is set.                                                                              |
| usb_reset_detect   | USB_RESET in the ALTEV register is set to 1                                                                                                                                                                                                        |
| local_event        | A local event that should wake up the USB.                                                                                                                                                                                                         |
| usb_sd5_detect     | USB_SD5 in the ALTEV register is set to 1.                                                                                                                                                                                                         |
| usb_suspend_detect | USB_SD3 in the ALTEV register is set to 1.                                                                                                                                                                                                         |
| usb_resume_detect  | RESUME in the ALTEV register is set to 1.                                                                                                                                                                                                          |
| resume_complete    | The node should stay in NodeResume state for at least 10 ms and then must enter USB Operational state to detect the EOP from the host, which terminates this Remote Resume operation. EOP is signalled when EOP in the ALTEV register is set to 1. |

### 31.5 ADDRESS DETECTION

Packets are broadcast from the host controller to all the nodes on the USB network. Address detection is implemented in hardware to allow selective reception of packets and to permit optimal use of microcontroller bandwidth. One function address with seven different endpoint combinations is decoded in parallel. If a match is found, then that particular packet is received into the FIFO; otherwise it is ignored.

Figure 106 shows a block diagram of the function address and endpoint decoding. The incoming USB Token, Packet Address field and four bits Endpoint field are extracted from the incoming bit stream. The address field is compared to the Function Address register (FADR) and if a match is detected, the USB Endpoint field is compared to all EP bit fields in the Endpoint Control registers (EPCx). With IN tokens, the transmit Endpoint Control registers are compared and with OUT tokens the receive Endpoint Control registers are compared. A match then enables the respective endpoint FIFO and transfers the payload data to/from the FIFO. Note that EPC0 is bidirectional and is enabled for IN, OUT and SETUP tokens.

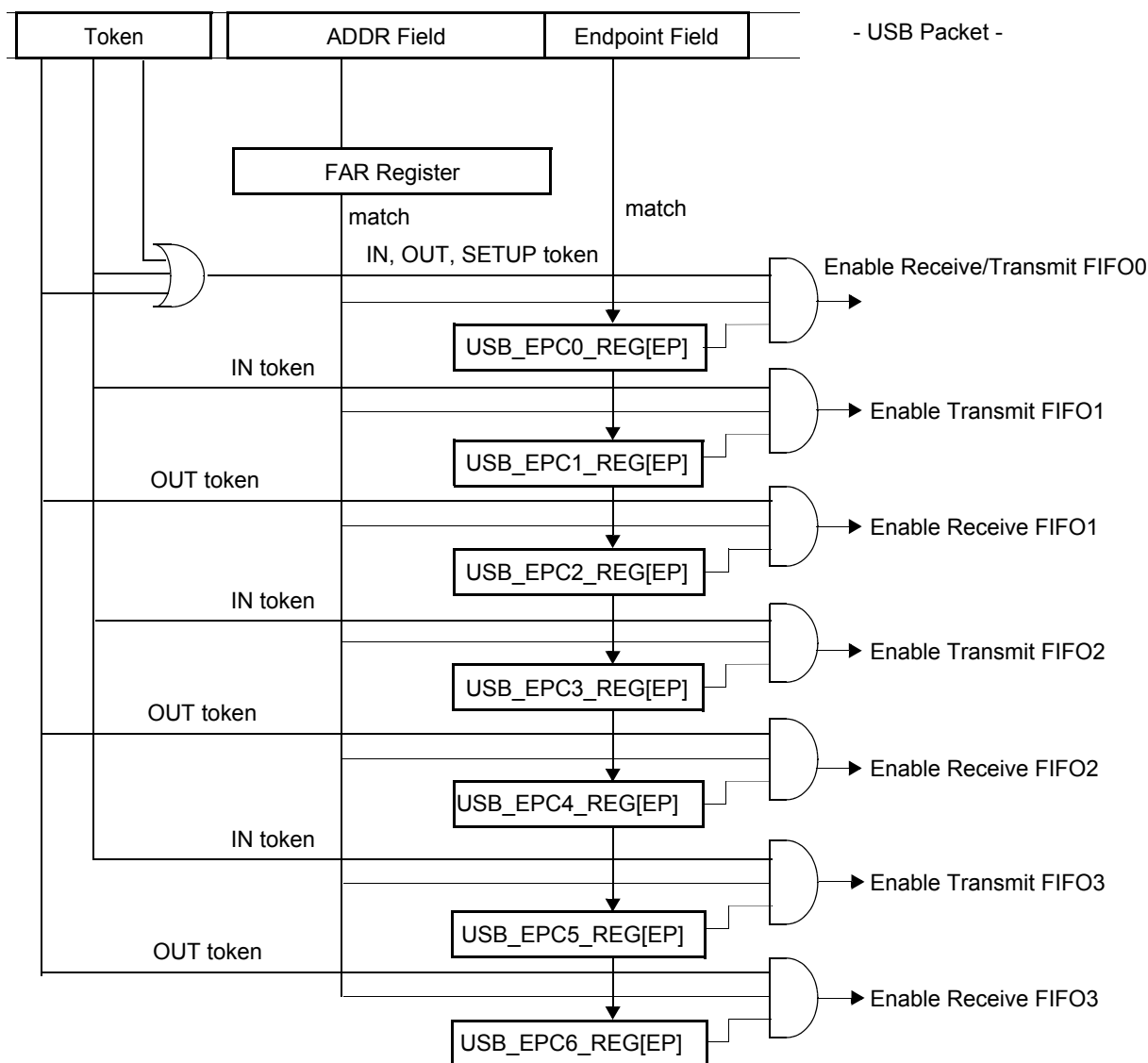


Figure 106: USB Function Address/Endpoint Decoding

### 31.6 TRANSMIT AND RECEIVE ENDPOINT FIFOS

The Full/Low Speed USB node uses a total of seven transmit and receive FIFOs: one bidirectional transmit and receive FIFO for the mandatory control endpoint, three transmit FIFOs and three receive FIFOs. As shown in Table 52, the bidirectional FIFO for the control endpoint is 8 bytes deep. The additional unidirectional FIFOs are 64 bytes each for both transmit and receive. Each FIFO can be programmed for one exclusive USB endpoint, used together with one globally decoded USB function address. The firmware must not enable both transmit and receive FIFOs for endpoint zero at any given time.

Table 52: USB Node Endpoint sizes

| Endpoint No. | TX FIFO      |              | RX FIFO      |               |
|--------------|--------------|--------------|--------------|---------------|
|              | Size (Bytes) | Name         | Size (Bytes) | Name          |
| 0            | 8 FIFO0      |              |              |               |
| 1            | 64           | TXFIFO1 (IN) |              |               |
| 2            |              |              | 64           | RXFIFO1 (OUT) |
| 3            | 64           | TXFIFO2 (IN) |              |               |
| 4            |              |              | 64           | RXFIFO2 (OUT) |
| 5            | 64           | TXFIFO3 (IN) |              |               |
| 6            |              |              | 64           | RXFIFO3 (OUT) |

If two endpoints in the same direction are programmed with the same endpoint number [EP field] and both are enabled, data is received or transmitted to/from the endpoint with the lower number, until that endpoint is disabled for bulk or interrupt transfers, or becomes full or empty for ISO transfers. For example, if receive EP1 and receive EP2 both use endpoint 3 and are both isochronous, the first OUT packet is received into EP1 and the second OUT packet into EP2, assuming no firmware interaction in between. For ISO endpoints, this allows implementing a ping-pong buffer scheme together with the frame number match logic.

Endpoints in different directions programmed with the same endpoint number operate independently.

If an OUT token is received for the FIFO, the firmware is informed that the FIFO has received data only if there was no error condition (CRC or STUFF error). Erroneous receptions are automatically discarded.

### 31.7 BIDIRECTIONAL CONTROL ENDPOINT FIFO0

FIFO0 should be used for the bidirectional control endpoint zero. It can be configured to receive data sent to the default address with the DEF bit in the EPC0 register.

The Endpoint 0 FIFO can hold a single receive or transmit packet with up to 8 bytes of data.

Note: A packet written to the FIFO is transmitted if an IN token for the respective endpoint is received. If an error condition is detected, the packet data remains in the FIFO and transmission is retried with the next IN token.

The FIFO contents can be flushed to allow response to an OUT token or to write new data into the FIFO for the next IN token.

Figure 107 shows the Endpoint 0 state machine. In state TXWAIT, if USB\_RXC0\_REG[SETUP\_FIX]=0, no state change will take place if a SETUP is received. With SETUP\_FIX=1, the state machine goes to IDLE, flushes to EP0 and receives the token in the RXWAIT state. If a SETUP is received in states TXFILL or RXDRAIN and SETUP\_FIX=1, the SETUP will be ignored and no ACK is send. This allows undisturbed FIFO filling/emptying. This state is usually present for a very short time and will force the host to retransmit the SETUP once.

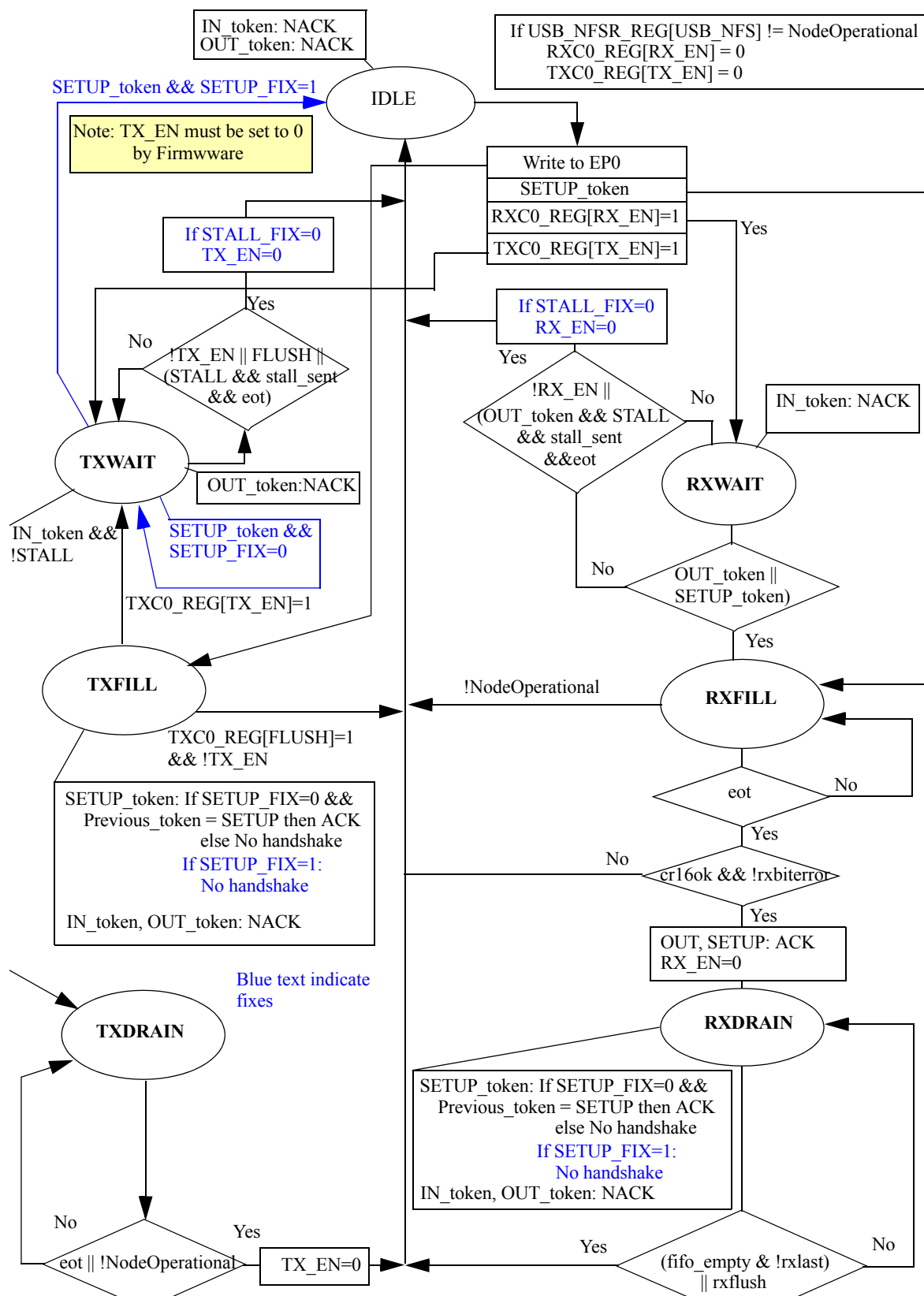


Figure 107: Endpoint 0 Operation

### 31.8 TRANSMIT ENDPOINT FIFO (TXFIFO1 TO TXFIFO5)

The Transmit FIFOs for Endpoints 1, 3 and 5 support bulk and interrupt USB packet transfers larger than the actual FIFO size. Therefore the firmware must update

the FIFO contents while the USB packet is transmitted on the bus.

Figure 108 illustrates the operation of the transmit FIFOs.

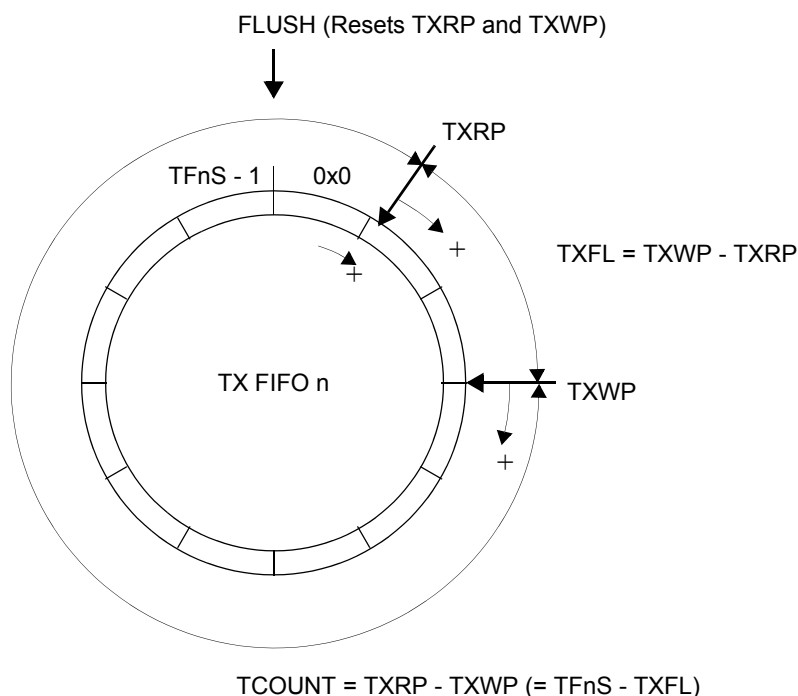


Figure 108: Tx FIFO operation

#### TFnS: Transmit FIFO n Size

This is the total number of bytes available within the FIFO.

#### TXRP: Transmit Read Pointer

This pointer is incremented every time the Endpoint Controller reads from the transmit FIFO. This pointer wraps around to zero if TFnS is reached. TXRP is never incremented beyond the value of the write pointer TXWP.

An underrun condition occurs if TXRP equals TXWP and an attempt is made to transmit more bytes when the LAST bit in the TXCMDx register is not set.

#### TXWP: Transmit Write Pointer

This pointer is incremented every time the firmware writes to the transmit FIFO. This pointer wraps around to zero if TFnS is reached.

If an attempt is made to write more bytes to the FIFO than actual space available (FIFO overrun), the write to the FIFO is ignored. If so, TCOUNT is checked for an indication of the number of empty bytes remaining.

#### TXFL: Transmit FIFO Level

This value indicates how many bytes are currently in the FIFO.

A FIFO warning is issued if TXFL decreases to a specific value. The respective WARNn bit in the FWR register is set if TXFL is equal to or less than the number specified by the TFWL bit in the TXCn register.

#### TCOUNT: Transmit FIFO Count

This value indicates how many empty bytes can be filled within the transmit FIFO. This value is accessible by firmware via the TXSn register.

### 31.9 RECEIVE ENDPOINT FIFO (RXFIFO2 TO RXFIFO6)

The Receive FIFOs for the Endpoints 2, 4 and 6 support bulk, interrupt USB packet transfers larger than the actual FIFO size. If the packet length exceeds the FIFO size, the firmware must read the FIFO contents while the USB packet is being received on the bus.

Figure 109 illustrates the operation of the Receive FIFOs.

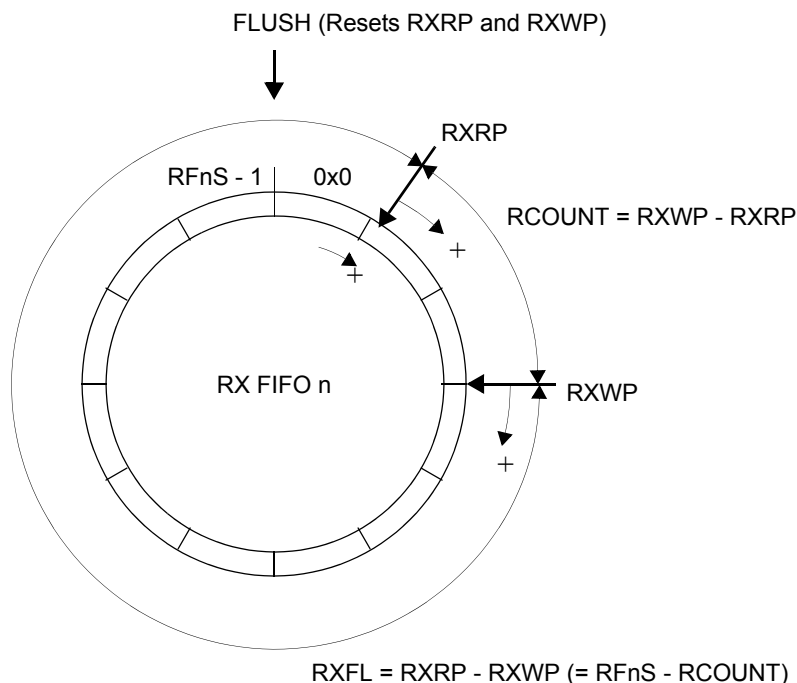


Figure 109: Rx FIFO operation

**RfNS: Receive FIFO n Size**

This is the total number of bytes available within the FIFO.

**RXRP: Receive Read Pointer**

This pointer is incremented with every read of the firmware from the receive FIFO. This pointer wraps around to zero if RfNS is reached. RXRP is never incremented beyond the value of RXWP.

If an attempt is made to read more bytes than are actually available (FIFO underrun), the last byte is read repeatedly.

**RXWP: Receive Write Pointer**

This pointer is incremented every time the Endpoint Controller writes to the receive FIFO. This pointer wraps around to zero if RfNS is reached.

An overrun condition occurs if RXRP equals RXWP and an attempt is made to write an additional byte.

**RXFL: Receive FIFO Level**

This value indicates how many more bytes can be received until an overrun condition occurs with the next write to the FIFO.

A FIFO warning is issued if RXFL decreases to a specific value. The respective WARNn bit in the FWR register is set if RXFL is equal to or less than the number specified by the RFWL bit in the RXCn register.

**RCOUNT: Receive FIFO Count**

This value indicates how many bytes can be read from the receive FIFO. This value is accessible by firmware via the RXSn register.

**31.10 INTERRUPT HIERARCHY**

Figure 110 shows the register hierarchy for generating USB interrupt events. Each bit in the event register can be masked with by setting the corresponding bit in the xxxMSK\_REG. A USBFS\_IRQ to the CPU is generated if one or more bits in the MAEV\_REG are set and the corresponding bits in the MAMSK\_REG are set to '1'. Bit 7 in the MAMSK\_REG is a global interrupt enabled for the USBFS\_IRQ.

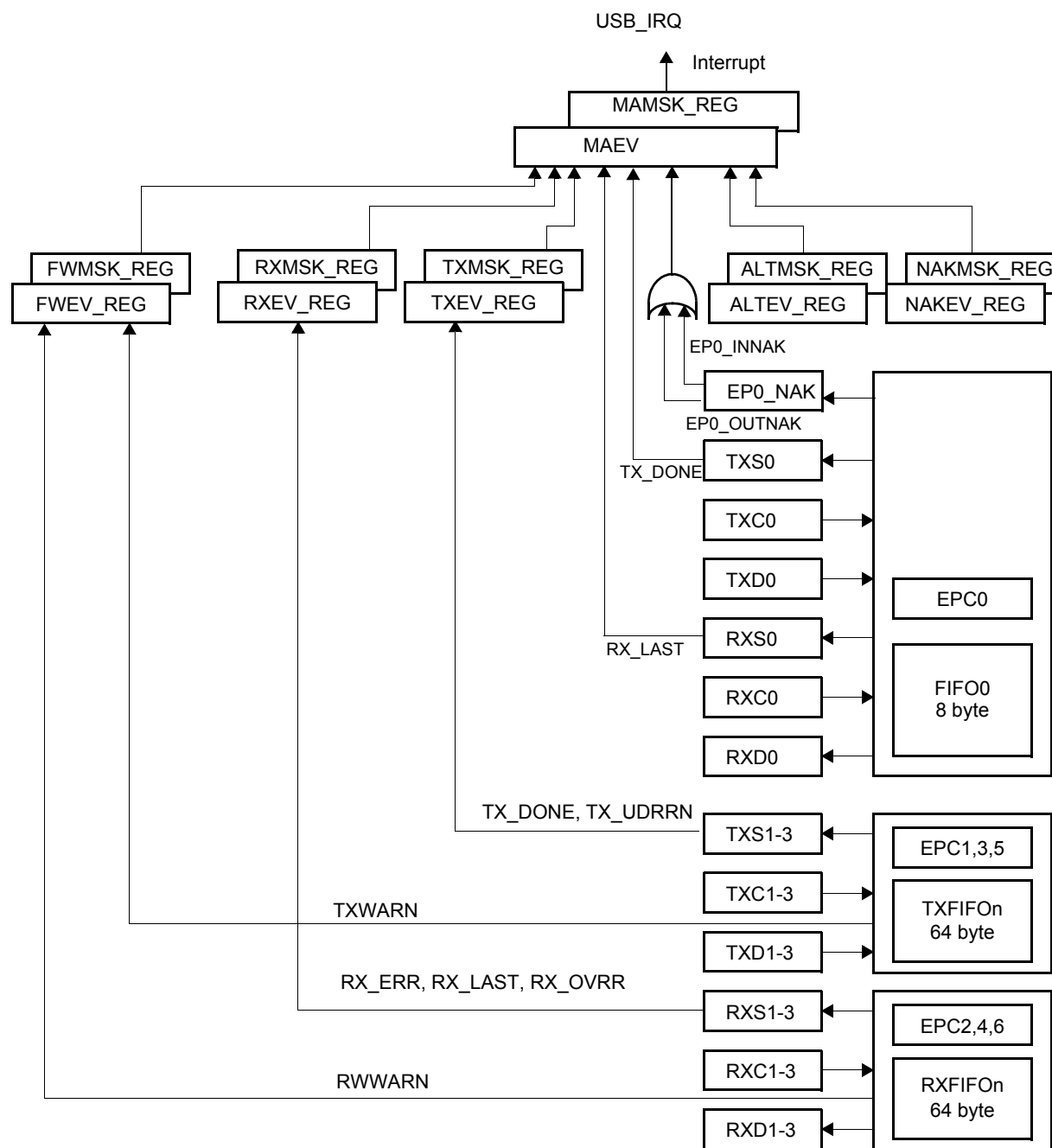


Figure 110: Interrupt Register hierarchy



## 32 Input/Output ports

The DA14680 has software-configurable I/O pin assignment, organized into ports Port 0, Port1, Port2 Port 3 and Port 4. Only ports 0, 1 and 2 are available at the WLCSP package. All ports are available at the QFN60 package.

### Features

- Port 0: 8 pins, Port 1: 8 pins, Port 2: 5 pins, Port 3: 8 pins, Port 4: 8 pins
- Fully programmable pin assignment

- Selectable 25 kΩ pull-up, pull-down resistors per pin
- Programmable open-drain functionality
- Pull-up to V33 or VDD1V8 (depends on selection)
- Fixed assignment for analog pins ADC[7:0], QSPI and SDW
- Pins retain their last state when system enters the Sleep or Deep Sleep mode.
- P0\_6 is kept powered while in Extended Sleep controlled by Timer1 PWM5 signal

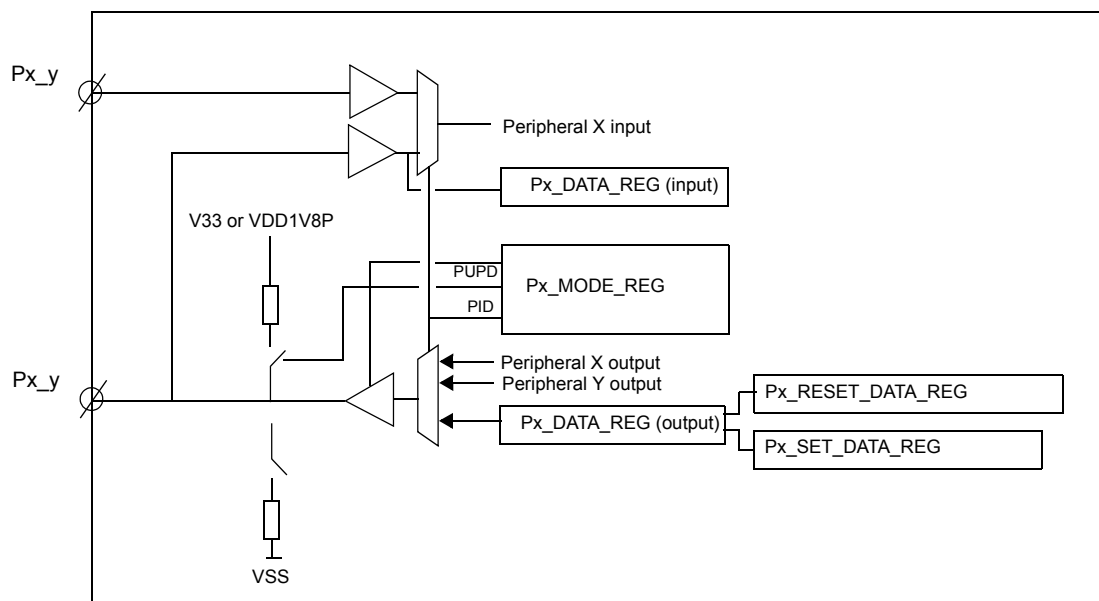


Figure 111: Port P0, P1, P2, P3 and P4 with Programmable Pin Assignment

### 32.1 PROGRAMMABLE PIN ASSIGNMENT

The Programmable Pin Assignment (PPA) provides a multiplexing function to the I/Os of on-chip peripherals. Any peripheral input or output signal can be freely mapped to any I/O port bit by setting `Pxy_MODE_REG[4-0]`.

Refer to the `Px_MODE_REGS` for an overview of the available PIDs. Analog ADC has fixed pin assignment in order to limit interference with the digital domain. The SWD interface (JTAG) is mapped on P0\_6 and P2\_4.

The firmware has the possibility to assign the same peripheral output to more than one pin. It is the responsibility of the user to make a unique assignment.

In case more than one input signal is assigned to a peripheral input, the left most pin in the lowest port pin number has priority. (e.g P00\_MODE\_REG has priority over P01\_MODE\_REG)

The port direction is controlled by setting:

`Pxy_MODE_REG[9:8]`

- 00 = Input, no resistors selected
- 01 = Input, pull-up selected
- 10 = Input, pull-down selected
- 11 = Output, no resistors selected

In output mode and analog mode the pull-up/down resistors are automatically disabled.

### 32.2 GENERAL PURPOSE PORT REGISTERS

The general purpose ports are selected with `PID=0`. The port function is accessible through registers:

- `Px_DATA_REG`: Port data input/output register
- `Px_SET_OUTPUT_DATA_REG`: Port set output register
- `Px_RESET_OUTPUT_DATA_REG`: Port reset output register

#### 32.2.1 Port Data Register

The registers input `Px_DATA_REG` and output `Px_DATA_REG` are mapped on the same address.

The data input register (`Px_DATA_REG`) is a read-only

register that returns the current state on each port pin even if the output direction is selected, regardless of the programmed PID, unless the analog function is selected (in this case it reads 0). The ARM CPU can read this register at any time even when the pin is configured as an output.

The data output register (Px\_DATA\_REG) holds the data to be driven on the output port pins. In this configuration, writing to the register changes the output value.

### 32.2.2 Port Set Data Output Register

Writing an 1 in the set data output register (Px\_SET\_DATA\_REG) sets the corresponding output pin. Writing a 0 is ignored.

### 32.2.3 Port Reset Data Output Register

Writing a 1 in the reset data output register (Px\_RESET\_DATA\_REG) resets the corresponding output pin. Writing a 0 is ignored.

## 32.3 FIXED ASSIGNMENT FUNCTIONALITY

There are certain signals that have a fixed mapping on specific general purpose IOs. This assignment is illustrated in the following table:

**Table 53: Fixed Assignment of Specific Signals**

| GPIO | SWD    | QSPI     | ADC   | PWM                       |
|------|--------|----------|-------|---------------------------|
| P0_0 |        | QSPI_CLK |       |                           |
| P0_1 |        | QSPI_D0  |       |                           |
| P0_2 |        | QSPI_D1  |       |                           |
| P0_3 |        | QSPI_D2  |       |                           |
| P0_4 |        | QSPI_D3  |       |                           |
| P0_5 |        | QSPI_CS  |       |                           |
| P0_6 | SWDIO  |          | ADC_4 | PWM5 (only in Sleep mode) |
| P0_7 |        |          | ADC_3 |                           |
| P1_0 |        |          | ADC_5 |                           |
| P1_1 |        |          |       |                           |
| P1_2 |        |          | ADC_0 |                           |
| P1_3 |        |          | ADC_2 |                           |
| P1_4 |        |          | ADC_1 |                           |
| P1_5 |        |          | ADC_6 |                           |
| P1_6 |        |          |       |                           |
| P1_7 |        |          |       |                           |
| P2_3 |        |          |       |                           |
| P2_4 | SW_CLK |          | ADC_3 |                           |

## 32.4 STATE RETENTION WHILE SLEEPING

Before setting the system to any of the sleep modes, the state of the pads needs to be retained so that no external components are affected by GPIOs changing state when the system goes to sleep, but also to avoid any floating driving signals from power domains that are shut off leading to increasing power dissipation.

The state of the pads is automatically latched by always-on latches by setting the PAD\_LATCH\_EN bit in the SYS\_CTRL\_REG. This bit will latch all digital control signals going into the pad as well as the data output, hence if the pad was set as an output driving high, it will retain its precise state.

The signals in red in the following figures are latched:

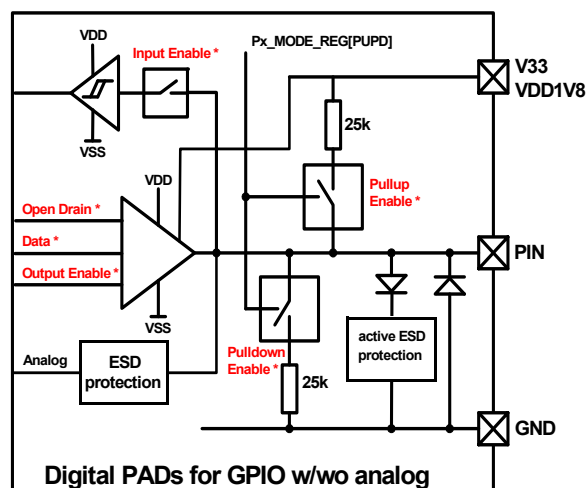


**Figure 112: Latching of digital pad signals**

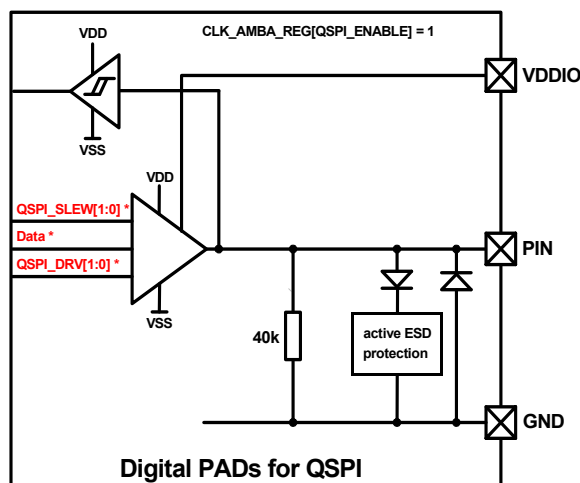


**Figure 113: Latching of QSPI pad signals**

After waking up, the software must ensure disabling of

the latching by de-asserting the PAD\_LATCH\_EN bit so that all pads are accessible and controllable again.

In the case of the QSPI pads, the pad latching will be overwritten by the QSPI controller as soon as the clock of the controller is enabled.

### 32.5 SPECIAL I/O CONSIDERATIONS

There are certain considerations in using the GPIOs as explained below:

- To use P1\_1 or P2\_2 in GPIO mode, USBPAD\_REG[USBPAD\_EN] must be set. However, the allowed levels on this pins are 0V and the voltage on V33 rail. If 1.8V is selected as the pin supply, then a current of 150 uA is to be expected. Moreover, these pins should not be used in sleep modes because the USBPAD\_REG will be powered off (belongs to the peripheral power domain).
- P1\_0, P1\_5 and P1\_7 might affect radio performance if toggling while RF activity. It is recommended to use them at low speed and not while radio is active.

### 33 BLE Core

The BLE (Bluetooth Low Energy) core is a qualified Bluetooth 4.2 baseband controller compatible with Bluetooth Low Energy specification and it is in charge of packet encoding/decoding and frame scheduling.

#### Features

- Bluetooth Low Energy compliant according to the specification of the Bluetooth System, v4.2, Bluetooth SIG.
  - Dual Topology
  - Low duty cycle advertising
  - L2CAP connection oriented channels
- All device classes support (Broadcaster, Central, Observer, Peripheral)
- All packet types (Advertising / Data / Control)
- Dedicated Encryption (AES / CCM)
- Bit stream processing (CRC, Whitening)
- FDMA / TDMA / events formatting and synchronization
- Frequency Hopping calculation
- Operating clock 16 or 8 MHz.
- Low power modes supporting 32.0 kHz, 32.768 kHz or 11.7 kHz
- Supports power down of the baseband during the protocol's idle periods.
- AHB Slave interface for register file access.
- AHB Slave interface for Exchange Memory access of CPU via BLE core.
- AHB Master interface for direct access of BLE core to Exchange Memory space

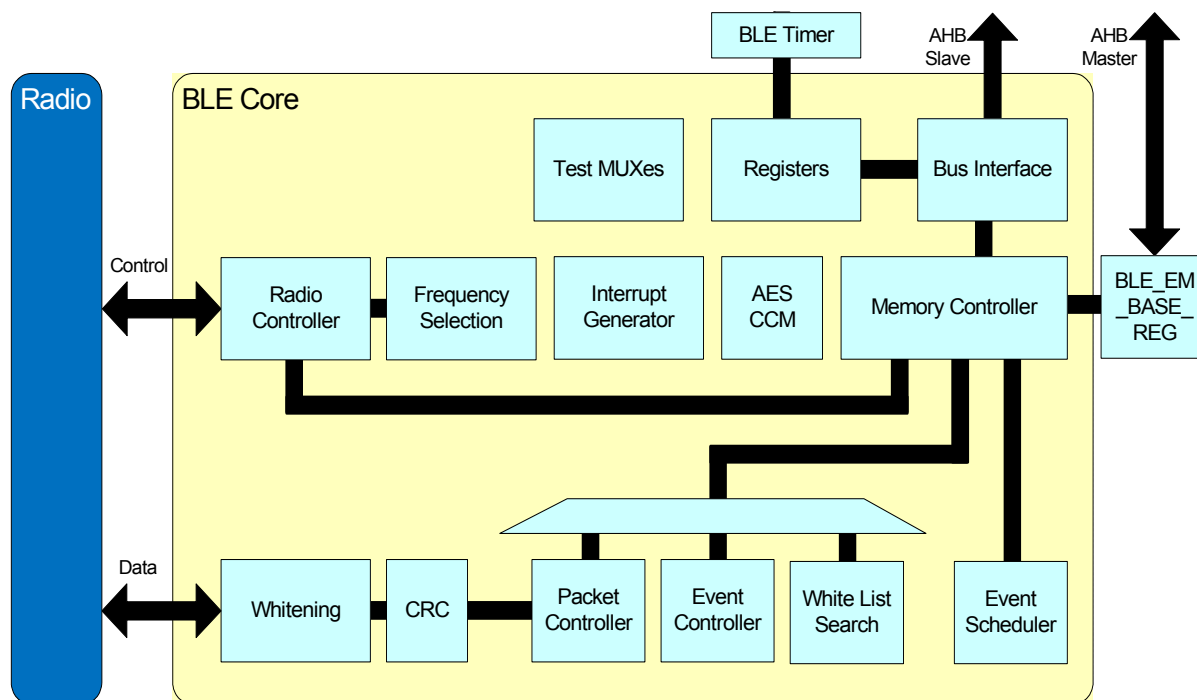


Figure 114: BLE Core Block Diagram

#### 33.1 ARCHITECTURE

##### 33.1.1 Exchange Memory

The BLE Core requires access to a memory space named “Exchange Memory” to store control structures and frame buffers. The access to Exchange Memory is performed via the AHB Master interface. The base address of the Exchange Memory is programmable by means of the BLE\_EM\_BASE register. The maximum addressable size of the BLE core is 64 kB.

#### 33.2 PROGRAMMING

##### 33.2.1 Wake up IRQ

Once BLE core switches to “BLE Deep Sleep Mode” the only way to correctly exit from this state is by initially generating the BLE\_WAKEUP\_LP\_IRQ and consecutively the BLE\_SLP\_IRQ. This sequence must be followed regardless of the cause of the termination of the “BLE Deep Sleep Mode”, i.e. either when if the BLE Timer expired or BLE Timer has been stopped due to the assertion of BLE\_WAKEUP\_REQ.

The assertion and de-assertion of BLE\_WAKEUP\_LP\_IRQ is fully controlled via the BLE\_ENBPRESET\_REG bit fields. Detailed description is following:

**TWIRQ\_SET:** Number of “ble\_lp\_clk” cycles before the expiration of the BLE Timer, when the BLE\_WAKEUP\_LP\_IRQ will be asserted. It is recommended to select a TWIRQ\_SET value larger than the time required for the XTAL16\_TRIM\_READY\_DELAY event, plus the IRQ Handler execution time. If the programmed value of TWIRQ\_SET is less than the minimum required value, then the actual BLE sleep duration (refer to BLE\_DEEPSLSTAT\_REG) will be larger than the programmed sleep duration (refer to BLE\_DEEPSLWKUP\_REG).

**TWIRQ\_RESET:** Number of “ble\_lp\_clk” cycles before the expiration of the sleep period, when the BLE\_WAKEUP\_LP\_IRQ will be de-asserted. It is recommended to always set to “1”.

**TWEXT:** Determines the high period of BLE\_WAKEUP\_LP\_IRQ, in the case of an external wake up event (refer to GP\_CONTROL\_REG[BLE\_WAKEUP\_REQ]). Minimum value is “TWIRQ\_RESET + X”, where X is the number of “ble\_lp\_clk” clock cycles that BLE\_WAKEUP\_LP\_IRQ will be held high. Recommended value is “TWIRQ\_RESET + 1”. Note that as soon as GP\_CONTROL\_REG[BLE\_WAKEUP\_REQ] is set to “1” the BLE\_WAKEUP\_LP\_IRQ will be asserted.

**Minimum BLE Sleep Duration:** The minimum value of BLE\_DEEPSLWKUP\_REG[DEEPSLTIME] time, measured in “ble\_lp\_clk” cycles, is the maximum of (a) “TWIRQ\_SET + 1” and (b) the SW execution time from setting BLE\_DEEPSLCNTL\_REG[DEEP\_SLEEP\_ON]

up to preparing CPU to accept the BLE\_WAKEUP\_LP\_IRQ (i.e. to call the ARM instruction WFI). If programmed DEEPSLTIME is less than the aforementioned minimum value, then BLE\_WAKEUP\_LP\_IRQ Handler may execute sooner than the call of ARM WFI instruction for example, causing SW instability.

### 33.2.2 Switch from Active Mode to Deep Sleep Mode

Software can set the BLE core into the “BLE Deep Sleep Mode”, by first programming the timing of BLE\_WAKEUP\_LP\_IRQ generation in BLE\_ENBPRESET\_REG, then program the desired sleep duration at BLE\_DEEPSLWKUP\_REG and finally set the register bit BLE\_DEEPSLCNTL\_REG[DEEP\_SLEEP\_ON]. The BLE Core will switch to the “ble\_lp\_clk” (32.0kHz or 32.768 kHz) in order to maintain its internal 625us timing reference. Software must poll the state of BLE\_CNTL2\_REG[RADIO\_PWRDN\_ALLOW] to detect the completion of this mode transition. Once the mode transition is completed, SW must disable the BLE clocks (“ble\_master1\_clk”, “ble\_master2\_clk” and “ble\_crypt\_clk”) by setting to “0” the CLK\_RADIO\_REG[BLE\_ENABLE] register bit. Finally, SW can optionally power down the BLE power domain by using the PMU\_CTRL\_REG[RADIO\_SLEEP] and the Peripheral and System power domains as well.

Figure 115 presents the waveforms while entering in BLE Deep Sleep Mode. In this case, SW, as soon as it detects that RADIO\_PWRDOWN\_ALLOW is “1”, it sets the PMU\_CTRL\_REG[BLE\_SLEEP] to power down the BLE domain. At the following figures, the corresponding BLE Core signals are marked with red while BLE domain is in power down state.

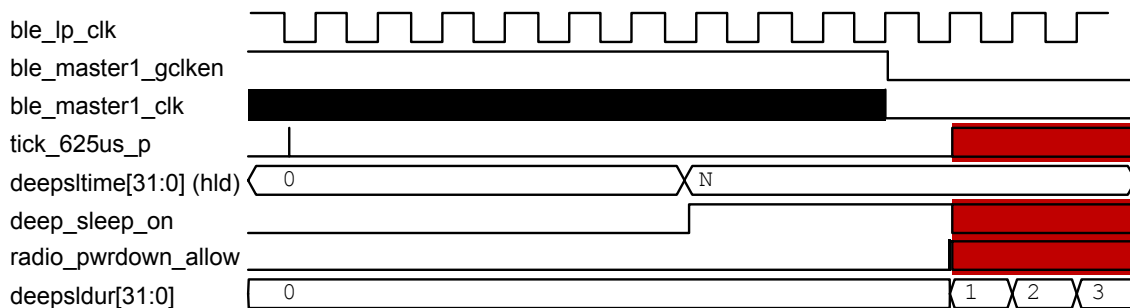


Figure 115: Entering into BLE Deep Sleep Mode

### 33.2.3 Switch from Deep Sleep Mode to Active Mode

There are two possibilities for BLE Core to terminate the BLE Deep Sleep mode:

1. Termination at the end of a predetermined time.
2. Termination on software wake-up request, due to an external event.

### 33.2.4 Switching on at anchor points.

Figure 118 shows a typical deep sleep phase that is terminated at predetermined time. After a configurable time before the scheduled wake up time (configured via BLE\_ENBPRESET\_REG register bit fields), the BLE Timer asserts the BLE\_WAKEUP\_LP\_IRQ in order to wake-up the CPU (powering up the System Power Domain). The BLE\_WAKEUP\_LP\_IRQ Interrupt

Handler will prepare the code environment and the XTAL 16MHz stabilization (refer to `SYS_STAT_REG[XTAL16_SETTLED]`) and will decide when the BLE Core will be ready to exit from the BLE Deep Sleep Mode.

Once the SW decides that BLE Core can wake up, it must enable the BLE clocks (via `CLK_RADIO_REG[BLE_ENABLE]`) and power up the BLE Power Domain (refer to `PMU_CTRL_REG[BLE_SLEEP]` and `SYS_STAT_REG[BLE_IS_UP]`).

After the expiration of the sleep period (as specified in `BLE_DEEPSLWKUP_REG[DEEPSLTIME]`) the BLE Timer will not exit the BLE Deep Sleep mode until it will

detect that BLE Core is powered up. That means that if the SW requires more time to power up the BLE Core, then the final sleep duration (provided by `BLE_DEEPSLSTAT_REG`) will be larger than the pre-programmed value.

When BLE Timer is expired, BLE clocks are enabled and BLE Core is powered up, the BLE Core exits the "BLE Core Deep Sleep mode" and asserts the `BLE_SLP_IRQ`.

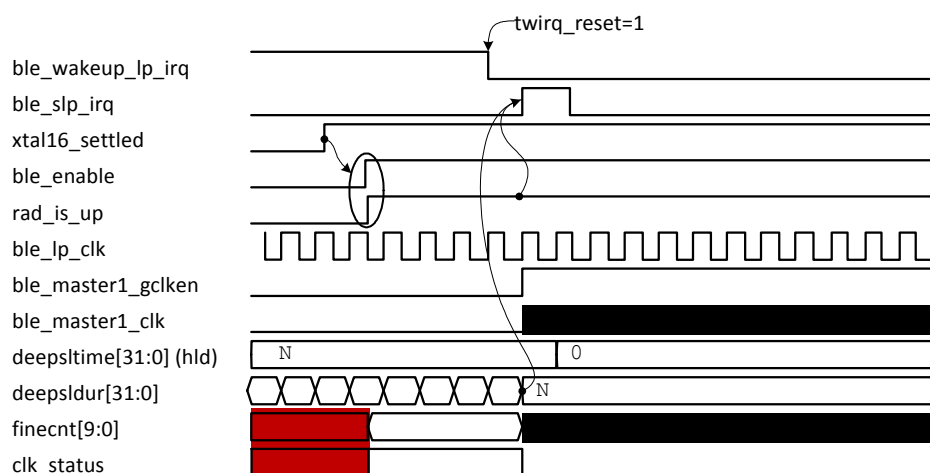


Figure 116: Exit BLE Deep Sleep Mode at predetermined time (zoom in)

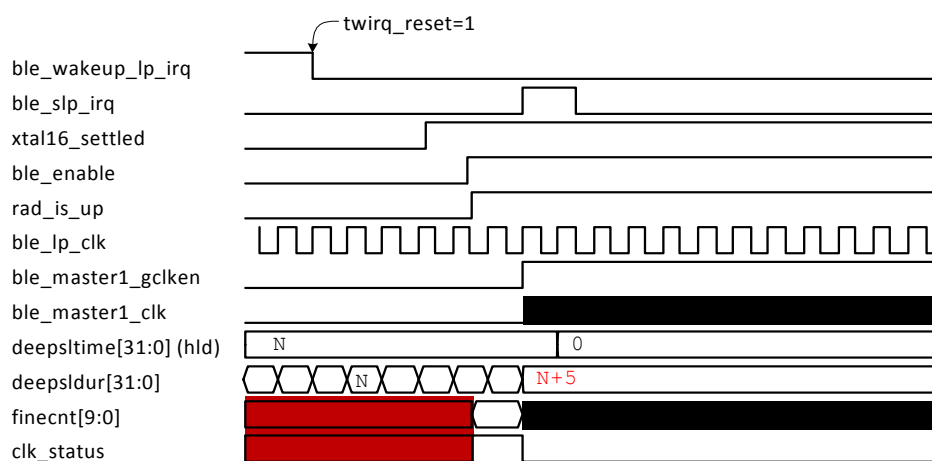


Figure 117: Exit BLE Deep Sleep Mode later than the predetermined time (zoom in)



Figure 118: Exit BLE Deep Sleep Mode at predetermined time (zoom out)

### 33.2.5 Switching on due to an external event.

Figure 119 shows a wake up from a deep sleep period forced by the assertion of register bit GP\_CONTROL\_REG[BLE\_WAKEUP\_REQ].

Assume that the system is in Deep Sleep state, i.e. all Power Domains have been switched off, and both the Wakeup Timer and Wakeup Controller have been programmed appropriately. Then assume that an event is detected at one of the GPIOs. In that case, the SW will decide to wake-up the BLE core, and will set the GP\_CONTROL\_REG[BLE\_WAKEUP\_REQ] to “1” in order to force the wake up sequence.

At Figure 119 the BLE\_WAKEUP\_REQ has been asserted by SW as soon as possible, causing BLE\_WAKEUP\_LP\_IRQ Handler to be executed as soon as possible. It is also possible to postpone the assertion of BLE\_WAKEUP\_REQ to occur after the detection of XTAL16\_TRIM\_READY, causing both BLE\_WAKEUP\_LP\_IRQ and BLE\_SLP\_IRQ Handlers to execute sequentially. The decision depends on the software structure and the application.

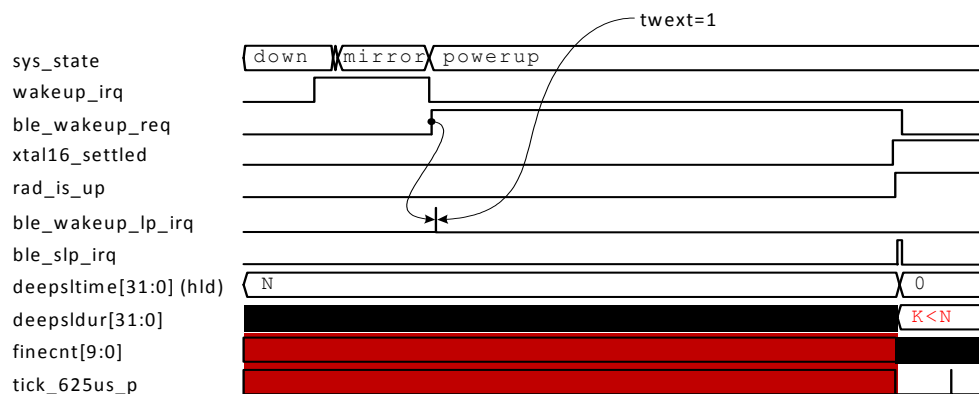


Figure 119: Exit BLE Deep Sleep Mode due to external event

As soon as bit field BLE\_WAKEUP\_REQ is set to “1” the BLE\_WAKEUP\_LP\_IRQ will be asserted. In that case, the high period of BLE\_WAKEUP\_LP\_IRQ is controlled via TWEXT. The recommended value of TWEXT is “TWIRQ\_RESET + 1”, meaning that BLE\_WAKEUP\_LP\_IRQ will remain high for one “ble\_lp\_clk” period.

As long as the BLE\_WAKEUP\_REQ is high, entering the sleep mode is prohibited. Please note that BLE\_WAKEUP\_REQ event can be disabled by setting BLE\_DEEPSLCNTL\_REG[EXTWKUPDSB].



### 33.3 DIAGNOSTIC SIGNALS

The BLE core provides several internal signals that can be mapped out on GPIOs to provide more insight on the real time operation. They can also be used for controlling external Front End Modules (FEMs) (e.g. as explained in [Figure 126](#). These signals are named as BLE diagnostics (ble\_diagx).

There are 8 fixed ports that can be assigned BLE diagnostic signals as presented in the following table:

**Table 54: BLE diagnostics mapping on GPIOs**

| BLE Diagnostics Port | GPIO port |
|----------------------|-----------|
| ble_diag0            | P2_0      |
| ble_diag1            | P2_1      |
| ble_diag2            | P2_2      |
| ble_diag3            | P1_0      |

**Table 54: BLE diagnostics mapping on GPIOs**

| BLE Diagnostics Port | GPIO port |
|----------------------|-----------|
| ble_diag4            | P1_1      |
| ble_diag5            | P1_2      |
| ble_diag6            | P1_3      |
| ble_diag7            | P2_3      |

There are 3 different configuration schemes which allow for specific internal signals to be mapped on the BLE Diagnostics Port. These are configured by programming the BLE\_DIAGCNTLx\_REG registers with specific values. The assignment of the internal signals on the diagnostic port depending on the configuration is presented in the following table:

**Table 55: BLE diagnostic signals per configuration**

| Registers value                                                                                   | Diagnostics Port | BLE signal       | Description                                                                                                                   |
|---------------------------------------------------------------------------------------------------|------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------|
| BLE_DIAGCNTL_REG = 0x83838383<br>BLE_DIAGCNTL2_REG = 0x83838383<br>BLE_DIAGCNTL3_REG = 0x76543210 | ble_diag0        | radcntl_txen     | Radio controller Tx enable signal                                                                                             |
|                                                                                                   | ble_diag1        | radcntl_rxen     | Radio controller Rx enable signal                                                                                             |
|                                                                                                   | ble_diag2        | sync_window      | Defines the correlation window for the access address. See timing in <a href="#">Figure 120</a>                               |
|                                                                                                   | ble_diag3        | sync_found_pulse | Access address detection pulse. Will be generated only if correlation is successful. See timing in <a href="#">Figure 120</a> |
|                                                                                                   | ble_diag4        | event_in_process | Indicates that an BLE event is currently in process. See timing in <a href="#">Figure 126</a>                                 |
|                                                                                                   | ble_diag5        | ble_event_irq    | A pulse designating the end of Advertising/Scanning/Connection events                                                         |
|                                                                                                   | ble_diag6        | ble_rx_irq       | A pulse designating that a packet is received depending on the configuration                                                  |
|                                                                                                   | ble_diag7        | ble_error_irq    | A pulse indicating that the BLE core and the CPU are trying to access the same memory space at the same time                  |
| BLE_DIAGCNTL_REG = 0x92929292<br>BLE_DIAGCNTL2_REG = 0x92929292<br>BLE_DIAGCNTL3_REG = 0x76543210 | ble_diag0        | rx_data          | BLE packet controller RX data bit                                                                                             |
|                                                                                                   | ble_diag1        | rx_data_en       | BLE packet controller RX data bit qualifier                                                                                   |
|                                                                                                   | ble_diag2        | rx_data_core     | BLE bit streaming RX data bit. Bit streaming engine consists of CRC and Whitening                                             |
|                                                                                                   | ble_diag3        | rx_data_core_en  | BLE bit streaming RX data bit qualifier                                                                                       |
|                                                                                                   | ble_diag4        | tx_data          | BLE packet controller TX data bit                                                                                             |
|                                                                                                   | ble_diag5        | tx_data_en       | BLE packet controller TX data bit qualifier                                                                                   |
|                                                                                                   | ble_diag6        | tx_data_core     | BLE bit streaming TX data bit.                                                                                                |
|                                                                                                   | ble_diag7        | tx_data_core_en  | BLE bit streaming TX data bit qualifier                                                                                       |

The sync\_window and sync\_found\_pulse timing is presented in the following figure:

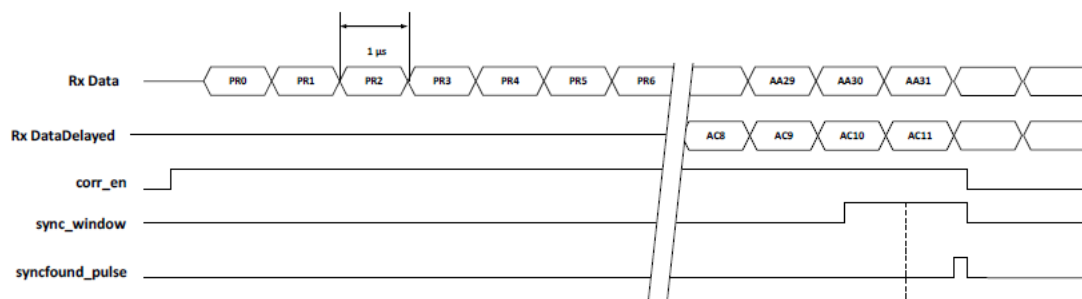


Figure 120: Diagnostics signals timing during correlation

### 33.4 POWER PROFILE

This section presents the current profile of the DA14680 when operating in the respective protocol mode as described in the following sections.

#### 33.4.1 Advertising Event

The current profile of a Bluetooth Advertising event is presented in the following figure:

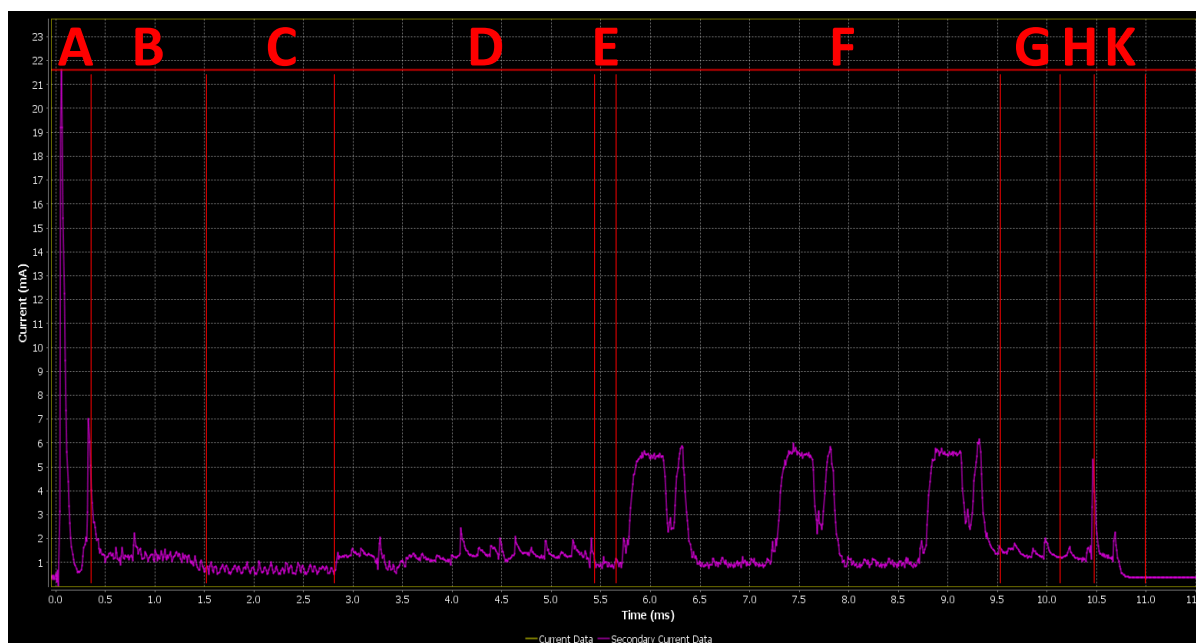


Figure 121: BLE Advertising power profile

The current profile figure is divided into sections which represent different operations within the DA14680 SoC. They are presented and explained in the following table:

Table 56: BLE Advertising profile breakdown

| Section | Description                                                   | Time ( $\mu$ s) |
|---------|---------------------------------------------------------------|-----------------|
| A       | Startup process                                               | 500             |
| B       | XTAL16M initial time                                          | 1000            |
| C       | XTAL16M settling time                                         | 1400            |
| D       | Switched to XTAL16, BLE core is enabled                       | 2600            |
| E       | WFI until the Radio starts                                    | 250             |
| F       | Radio activity (TX/RX) on channels 37, 38 and 39 <sup>1</sup> | 3800            |
| G       | BLE core sleep preparation                                    | 400             |
| H       | OS sleep preparation                                          | 350             |

Table 56: BLE Advertising profile breakdown

| Section | Description        | Time ( $\mu$ s) |
|---------|--------------------|-----------------|
| K       | Goto sleep process | 400             |

<sup>1</sup> Time between RX/TX on successive channels is 1.5 ms

The overall charge required for this operation at 3V is 20  $\mu$ C while executing code from FLASH and ROM (cached mode). The Advertising interval is 600 ms. Note, that overall time might vary depending on the protocol's parameters.

### 33.4.2 Connection Event

The current profile of a Bluetooth Connection event is presented in the following figure:

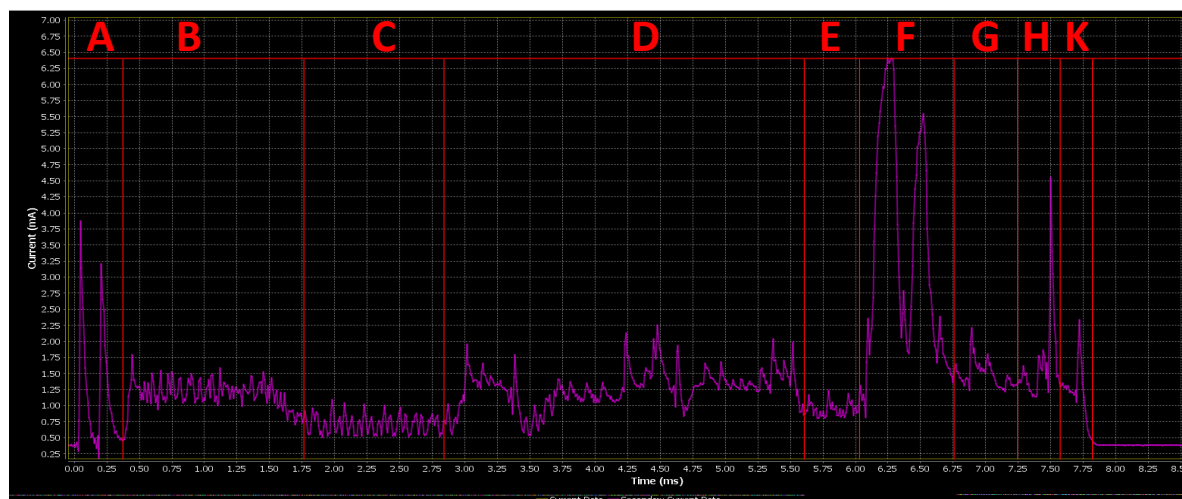


Figure 122: BLE Connection power profile

The current profile figure is divided into sections which represent different operations within the DA14680 SoC. They are presented and explained in the following table:

Table 57: BLE Connection profile breakdown

| Section | Description                             | Time ( $\mu$ s) |
|---------|-----------------------------------------|-----------------|
| A       | Startup process                         | 500             |
| B       | XTAL16M initial time                    | 1300            |
| C       | XTAL16M settling time                   | 1150            |
| D       | Switched to XTAL16, BLE core is enabled | 2600            |
| E       | WFI until the Radio starts              | 450             |

Table 57: BLE Connection profile breakdown

| Section | Description                | Time ( $\mu$ s) |
|---------|----------------------------|-----------------|
| F       | Radio activity (RX, TX)    | 500             |
| G       | BLE core sleep preparation | 400             |
| H       | OS sleep preparation       | 350             |
| K       | Goto sleep process         | 400             |

The overall charge required for this operation at 3V is 10.6  $\mu$ C while executing code from FLASH and ROM (cached mode). The Connection interval is 30 ms. Note, that overall time might vary depending on the protocol's parameters.

### 34 CoEx interface

The DA14680 implements a coexistence interface for signalling radio activity to external 2.4 GHz co-located devices. A three wire interface is delivering information on the priority and the RF transmit/receive events.

The CoEx interface and its connection to the rest of the system is displayed in [Figure 123](#).

#### Features

- 2.4 GHz Radio activity indication
- Priority indication
- Sensing of external module RF activity
- Supports up to 2 external 2.4 GHz devices

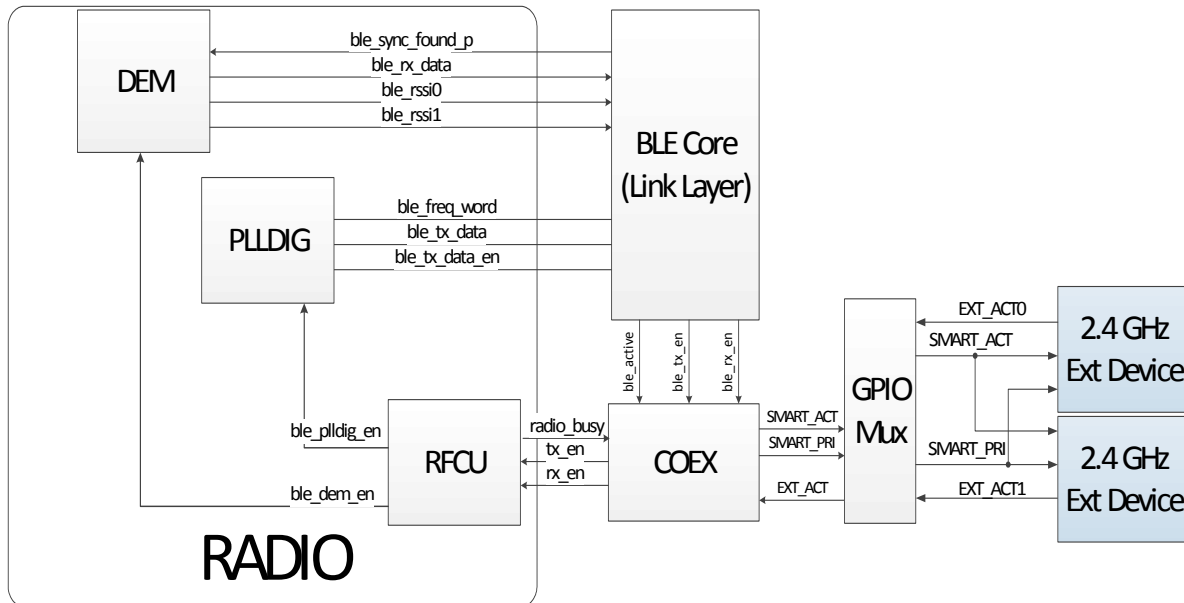


Figure 123: Coexistence interface

#### 34.1 ARCHITECTURE

The coexistence external interface contains three signals, namely:

1. EXT\_ACT0, EXT\_ACT1: this is an input to the DA14680 and when asserted, it designates that an external 2.4GHz device is about to issue RF activity.
2. SMART\_ACT: this is an output from the DA14680 and when asserted it designates that the DA14680 is transmitting or receiving hence an external device should be aware of the radio activity. The exact timing of the assertion of this signal is depicted in [Figure 124](#).
3. SMART\_PRI: this signal is communicating whether the DA14680 has priority over the external devices or not. If asserted, then the external devices could adjust their RF activity accordingly.

#### 34.2 PROGRAMMING

The CoEx block is in the same power domain as the Radio, namely PD\_RAD. Hence, the power domain has to be enabled prior to programming the registers. This can be done with PMU\_CTRL\_REG[PERIPH\_SLEEP]=0.

Furthermore, as depicted in [Figure 26](#), the CoEx block is clocked by the same clock as the Digital PHY does. Hence the CLK\_RADIO\_REG[RFCU\_EN] bit has to be asserted before programming the block's registers.

Any GPIO can be selected to act as EXT\_ACT0 or EXT\_ACT1, by just programming the Pxy\_MODE\_REG[PID]=48/49 (COEX\_EXT\_ACT0 or COEX\_EXT\_ACT1), hence two external devices can be supported with the same priority. The behaviour of the SMART\_ACT (Pxy\_MODE\_REG[PID]=50) and SMART\_PRI (Pxy\_MODE\_REG[PID]=51) output signals is configurable by means of COEX\_CTRL\_REG and COEX\_PRIx\_REG registers. The SMART\_ACT line will always be asserted if the DA14680 is about to start RF activity if COEX\_CTRL\_REG[SMART\_ACT\_IMPL]=0. On the contrary, if COEX\_CTRL\_REG[SMART\_ACT\_IMPL]=1 then it is asserted only if DA14680 has higher priority than the external devices as programmed in COEX\_PRIx\_REG[COEX\_PRI\_MAC]. if not, then both SMART\_ACT and SMART\_PRI will be de-asserted as long as the EXT\_ACT is high disabling the DA14680 RF activity to avoid collisions with the external device.

An overview of the aforementioned scenarios is presented in [Figure 124](#).

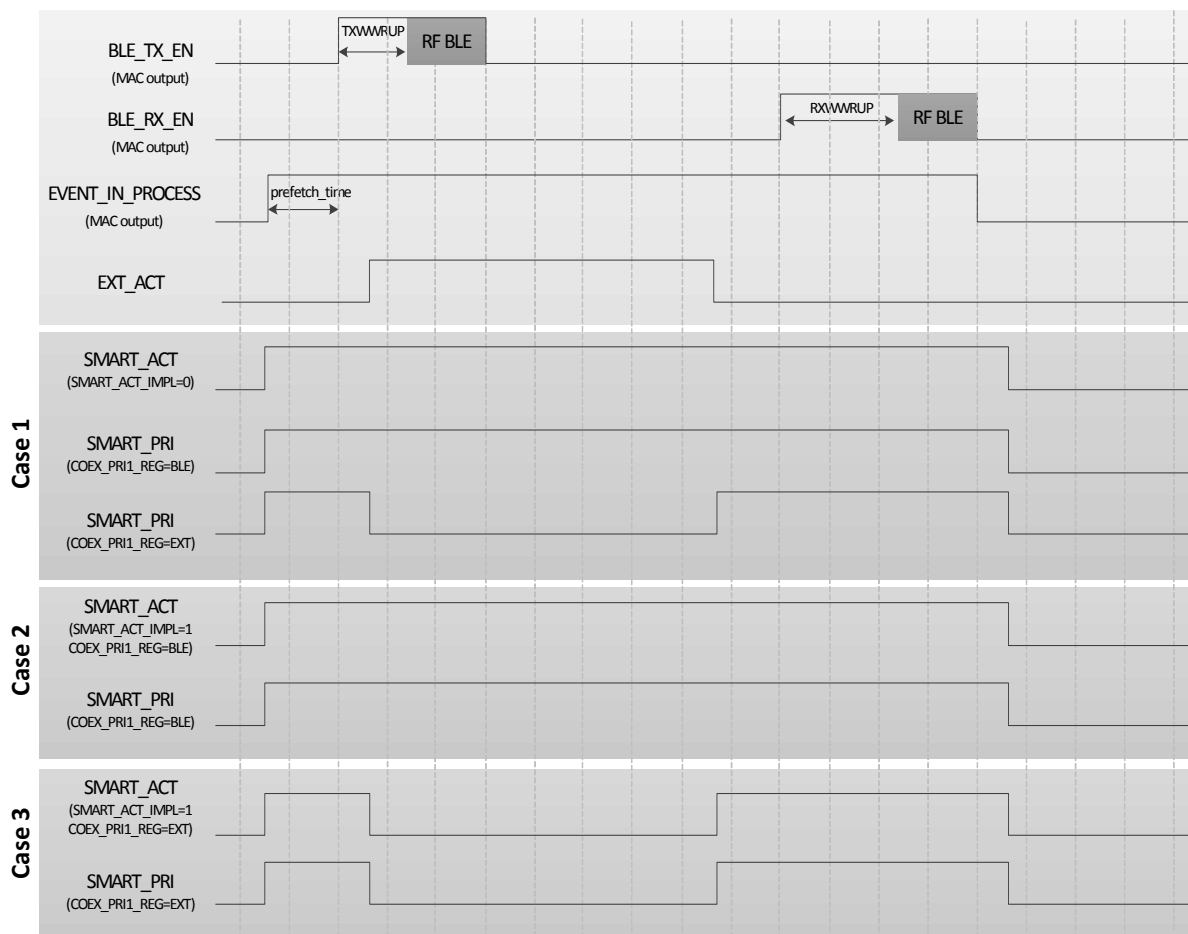


Figure 124: Coexistence signalling cases

**Case 1:**

Assuming COEX\_CTRL\_REG[SMART\_ACT\_IMPL]=0, the SMART\_ACT signal will be asserted with the event\_in\_process BLE MAC signal which is an envelope of an RX and TX event. It will be de-asserted after the de-assertion of the event\_in\_process and as soon as the latest BLE activity is completed designated by an internal signal. SMART\_PRI notifies whether the BLE has higher priority or not, hence if COEX\_PRI1\_REG[COEX\_PRI\_MAC]=BLE then it follows the SMART\_ACT waveform. If COEX\_PRI1\_REG[COEX\_PRI\_MAC]=EXT then it will be de-asserted whenever the external device notifies its RF activity via EXT\_ACT.

**Case 2:**

Assuming COEX\_CTRL\_REG[SMART\_ACT\_IMPL]=1, the SMART\_ACT will be asserted only if BLE has priority over the external devices and not always. In this case, the COEX\_PRI1\_REG[COEX\_PRI\_MAC]=BLE grants priority to the BLE hence the SMART\_ACT and SMART\_PRI are asserted at the same time.

**Case 3:**

In this case the external device has higher priority due to the COEX\_PRI1\_REG[COEX\_PRI\_MAC]=EXT. Thus, the SMART\_ACT will not be asserted if an external device signals RF activity via EXT\_ACT. SMART\_PRI will be also be low as long as EXT\_ACT is high.

## 35 Radio

The Radio Transceiver implements the RF part of the DA14680. It provides a 93dB RF link budget for reliable wireless communications.

All RF blocks are supplied by on-chip low-drop out-regulators (LDOs). The bias scheme is programmable per block and optimized for minimum power consumption.

The radio block diagram is given in Figure 125. It comprises the Receiver, Transmitter, Synthesizer, Rx/Tx combiner block, and Biasing LDOs.

### Features

- Single ended RFIO interface, 50  $\Omega$  matched
- Alignment free operation
- -94.5 dBm receiver sensitivity
- 0 dBm transmit output power
- Ultra low power consumption
- Fast frequency tuning minimises overhead

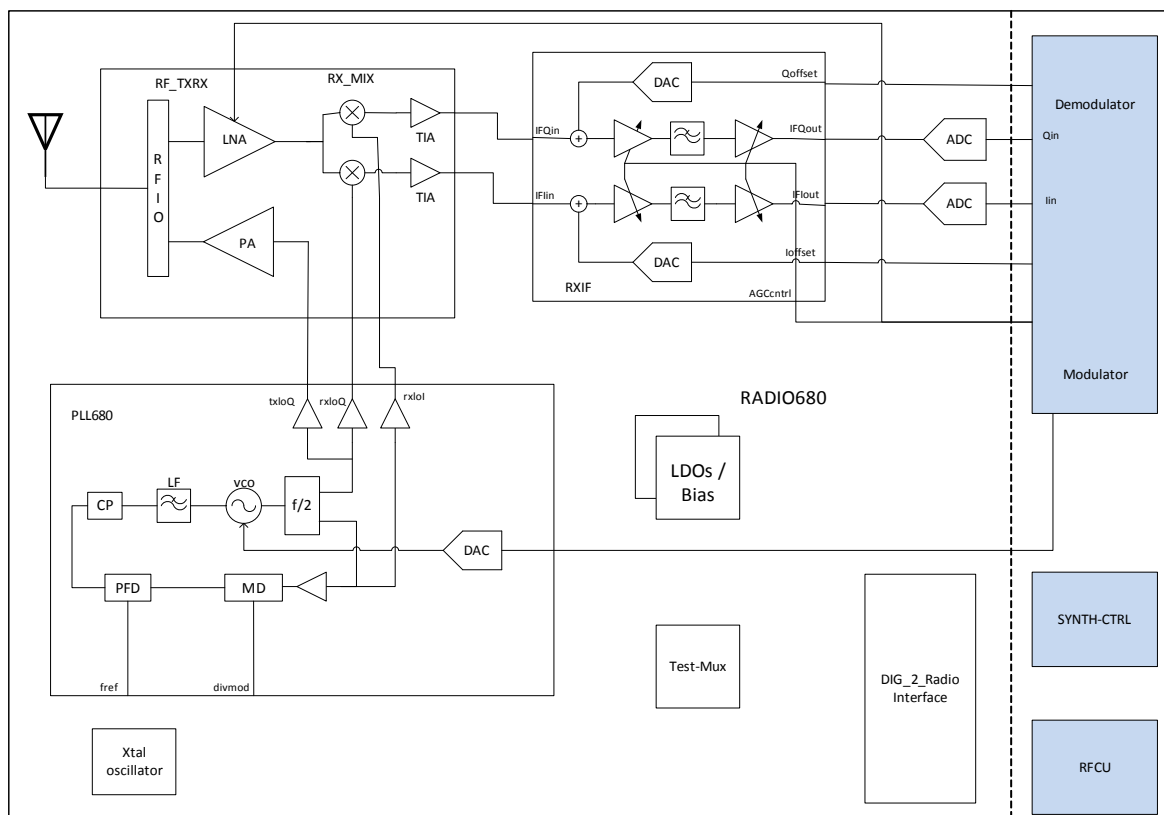


Figure 125: Radio Block Diagram

### 35.1 ARCHITECTURE

#### 35.1.1 Receiver

The RX frontend consists of a selective matching network, a low noise amplifier (LNA) and an image rejection down conversion mixer. The LNA gain is controlled by the AGC.

The intermediate frequency (IF) part of the receiver comprises a complex filter and 2 variable gain amplifiers. This provides the necessary signal conditioning prior to digitalization. The digital demodulator block (DEM) provides a synchronous bit stream.

#### 35.1.2 Synthesizer

The RF Synthesizer generates the quadrature LO signal for the mixer, but also generates the modulated TX

output signal. The VCO runs at twice the required frequency and a dedicated divide-by-2 circuit generates the 2.4 GHz signals in the required phase relations. Its frequency is controlled by a classic 3rd order type II PLL with a passive loop filter, operated in fractional- $N$  mode. The reference frequency is the 16 MHz crystal clock. The multi-modulus divider has a nominal divide ratio of 153 which is varied by a  $\Sigma\Delta$  modulator. The modulation of the TX frequency is performed by 2-point modulation. The fractional divide ratio also contains the shaped TX data stream. A second modulation path feeds the TX data stream directly to the VCO. The latter path is automatically calibrated from time to time to align the low and high frequency parts of the 2-point modulation scheme.

### 35.1.3 Transmitter

The RF power amplifier (RFPA) is an extremely efficient Class-D structure, providing typically 0 dBm to the antenna. It is fed by the VCO's divide-by-2 circuit and delivers its TX power to the antenna pin through the combined RX/TX matching circuit.

### 35.1.4 RFIO

The RX/TX combiner block is a unique feature of the DA14680. It makes sure that the received power is applied to the LNA with minimum losses towards the RFPA. In TX mode, the LNA poses a minimal load for the RFPA and its input pins are protected from the RFPA. In both modes, the single ended RFIO port is matched to 50  $\Omega$ , in order to provide the simplest possible interfacing to the antenna on the printed circuit board.

### 35.1.5 Biasing

All RF blocks are supplied by on-chip low-drop out-regulators (LDOs). The bias scheme is programmable per block and optimized for minimum power consumption.

### 35.1.6 Control

The radio control unit (RFCU), controls the block timing and configuration registers. The BLE interfaces directly with the RFCU. The DA14680 can be put in test mode using a standard Bluetooth tester (e.g. Rohde & Schwarz CBT with K57 option) by connecting the antenna terminal for the RF link and the UART as described in section 23.7.

## 35.2 DYNAMIC CONTROLLED FUNCTIONS

The RF control unit (RFCU) provides the capability of controlling 5 signals which can be used for controlling a Front End Module or an external Power Amplifier. The timing granularity of the DCF signals is 1  $\mu$ s. The DCFs can be output on any GPIO using PID numbers 55 to 59.

The programming of the DCF signals are with respect to the rising and falling edges of the TX\_EN or RX\_EN signals from the BLE MAC, as depicted in the following figure:

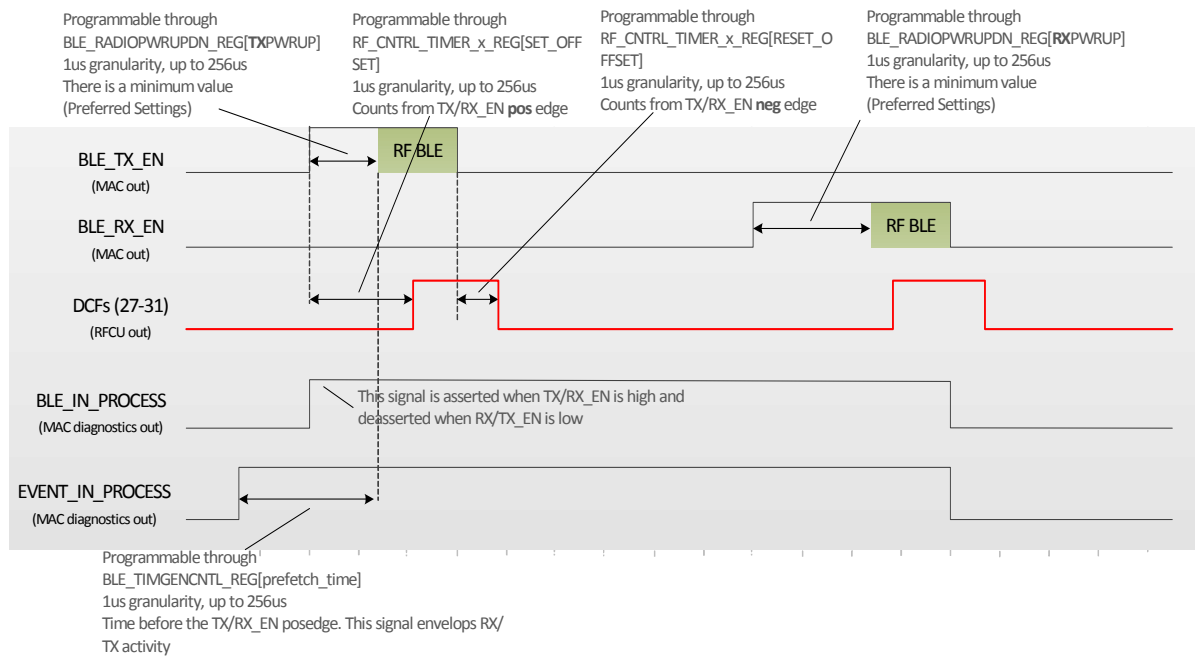


Figure 126: DCF signals programming

## 35.3 DIAGNOSTIC SIGNALS

There are diagnostic signals that can trigger the rf\_diag\_irq interrupt line. There are 2 signals from the BLE MAC and 2 signals from the Radio that can actually be programmed to act as this interrupt line sources. They are controlled by the RF\_DIAGIRQ01\_REG and RF\_DIAGIRQ23\_REG registers. The overview of the options is provided in the following table:

Table 58: RF\_DIAG\_IRQ source selection

| Word Sel<br>Bit Sel | WSELx= 0 or 1 | WSELx= 2 or 3 |
|---------------------|---------------|---------------|
| BSELx=7             | ble_diag7     | TX_EN         |
| BSELx=6             | ble_diag6     | RX_EN         |
| BSELx=5             | ble_diag5     | DCF26         |



Table 58: RF\_DIAG\_IRQ source selection

| Word Sel<br>Bit Sel | WSELx= 0 or 1 | WSELx= 2 or 3 |
|---------------------|---------------|---------------|
| BSELx=4             | ble_diag4     | DCF25         |
| BSELx=3             | ble_diag3     | DCF24         |
| BSELx=2             | ble_diag2     | DCF23         |
| BSELx=1             | ble_diag1     | DCF22         |
| BSELx=0             | ble_diag0     | DCF21         |

There are 4 different selections in these 2 registers namely DIAGIRQ\_WSEL\_0 to DIAGIRQ\_WSEL\_3 (designated as WSELx in the table) and DIAGIRQ\_BSEL\_0 to DIAGIRQ\_BSEL\_3 (designated as BSELx in the table). They are identical allowing for different signals combinations for the interrupt generation.

The exact definition of the ble\_diagx signals are described in [Table 55](#).

The DCF21 to DCF26 signals are programmable timers as explained in the previous section and are used to trigger events in the Radio circuitry.

### 36 Memory map

This section contains a detailed view of the DA14680 memory map.

**Table 59: Memory Map**

| Address    | Description     | Power Domain | AMBA  |
|------------|-----------------|--------------|-------|
| 0x0        | Remapped Device |              |       |
| 0x7F00000  | ROM             | SYS_PD       | AHB   |
| 0x7F40000  | OTPC            | SYS_PD       | AHB   |
| 0x7F80000  | OTP             | SYS_PD       | AHB   |
| 0x7FC0000  | DataRAM         | SYS_PD       | AHB   |
| 0x7FE0000  | CacheRAM        | SYS_PD       | AHB   |
| 0x8000000  | QSPI FLASH      | SYS_PD       | AHB   |
| 0xC000000  | QSPIC           | SYS_PD       | AHB   |
| 0xC0FFFFFF | Reserved        |              |       |
| 0x40000000 | BLEC            | BLE_PD       | AHB   |
| 0x40020000 | AES-HASH        | SYS_PD       | AHB   |
| 0x40030000 | ECC_M           | SYS_PD       | AHB   |
| 0x40040000 | TRNG_M          | SYS_PD       | AHB   |
| 0x40060000 | Reserved        |              |       |
| 0x400A0000 | CACHE_MTR       | SYS_PD       | AHB   |
| 0x400B0000 | CACHE_MDR       | SYS_PD       | AHB   |
| 0x400C3000 | CACHEC_MRM      | SYS_PD       | AHB   |
| 0x40098000 | Reserved        |              |       |
| 0x50000000 | CRG             | AON_PD       | APB16 |
| 0x50000100 | WKUPC           | AON_PD       | APB16 |
| 0x50000200 | TIM1            | AON_PD       | APB16 |
| 0x50000300 | Reserved        |              |       |
| 0x50001000 | UART            | PER_PD       | APB16 |
| 0x50001100 | UART2           | PER_PD       | APB16 |
| 0x50001200 | SPI             | PER_PD       | APB16 |
| 0x50001300 | SPI2            | PER_PD       | APB16 |
| 0x50001400 | I2C             | PER_PD       | APB16 |
| 0x50001500 | I2C2            | PER_PD       | APB16 |
| 0x50001600 | KEYSC           | PER_PD       | APB16 |
| 0x50001700 | IR              | PER_PD       | APB16 |
| 0x50001800 | USB             | PER_PD       | APB16 |
| 0x50001900 | ADC             | PER_PD       | APB16 |
| 0x50001A00 | QDEC            | PER_PD       | APB16 |
| 0x50001B00 | ANAMISC         | PER_PD       | APB16 |
| 0x50001C00 | CRG_PERIPH      | PER_PD       | APB16 |
| 0x50001C4A | Reserved        |              |       |
| 0x50002000 | RFCU            | RAD_PD       | APB16 |
| 0x50002D00 | PLLDIG          | RAD_PD       | APB16 |
| 0x50002E00 | DEMODO          | RAD_PD       | APB16 |
| 0x50002F00 | COEX            | RAD_PD       | APB16 |

Table 59: Memory Map

| Address    | Description      | Power Domain | AMBA  |
|------------|------------------|--------------|-------|
| 0x50003000 | GPIOMUX          | SYS_PD       | APB16 |
| 0x50003100 | WDOGTIM          | SYS_PD       | APB16 |
| 0x50003200 | VERSION          | SYS_PD       | APB16 |
| 0x50003300 | GPREG            | SYS_PD       | APB16 |
| 0x50003400 | TIM0/2           | SYS_PD       | APB16 |
| 0x50003500 | DMA              | SYS_PD       | APB16 |
| 0x50003600 | RFPT             | SYS_PD       | APB16 |
| 0x50004000 | APU              | PER_PD       | APB32 |
| 0x50005000 | TRNG             | SYS_PD       | APB32 |
| 0x50006000 | ECC              | RAD_PD       | APB16 |
| 0x50006100 | Reserved         |              |       |
| 0xE0000000 | ARM Internal Bus | SYS_PD       |       |

**Note:** AHB implies a full 32-bit aligned address space and a 32 bit data bus. APB16 implies a 16-bit aligned address space and 16-bit data bus. APB32 implies a 32-bit aligned address space and a 32-bit data bus. Byte accesses on either APB16 or APB32 are not allowed.

## 37 Registers

This section contains a detailed view of the DA14680 registers. It also describes the bitfields in certain registers that will retain their value even if the power domain they reside in, is shut off.

**Table 60: Retention Registers**

**Table 60: Retention Registers**

| Register             | Bit field          |
|----------------------|--------------------|
| BLE_CNTL2_REG        | EMACCERRSTAT       |
|                      | EMACCERRACK        |
|                      | EMACCERRMSK        |
|                      | BLE_DIAG_OVRWR_7_5 |
|                      | BLE_CLK_STAT       |
|                      | MON_LP_CLK         |
|                      | RADIO_PWRDN_ALLOW  |
|                      | BLE_CLK_SEL        |
|                      | BB_ONLY            |
|                      | SW_RPL_SPI         |
|                      | WAKEUPLPSTAT       |
|                      | BLE_RSSI_SEL       |
| BLE_EM_BASE_REG      | BLE_EM_BASE_10_16  |
| QSPIC_BURSTBRK_REG   | QSPIC_BRK_WRD      |
|                      | QSPIC_BRK_EN       |
|                      | QSPIC_BRK_SZ       |
|                      | QSPIC_BRK_TX_MD    |
|                      | QSPIC_SEC_HF_DS    |
| QSPIC_BURSTCMDA_REG  | QSPIC_INST         |
|                      | QSPIC_INST_WB      |
|                      | QSPIC_EXT_BYTE     |
|                      | QSPIC_INST_TX_MD   |
|                      | QSPIC_ADR_TX_MD    |
|                      | QSPIC_EXT_TX_MD    |
|                      | QSPIC_DMY_TX_MD    |
| QSPIC_BURSTCMD B_REG | QSPIC_DAT_RX_MD    |
|                      | QSPIC_EXT_BYTE_EN  |
|                      | QSPIC_EXT_HF_DS    |
|                      | QSPIC_DMY_NUM      |
|                      | QSPIC_INST_MD      |
|                      | QSPIC_WRAP_MD      |
|                      | QSPIC_WRAP_LEN     |
|                      | QSPIC_WRAP_SIZE    |
|                      | QSPIC_CS_HIGH_MIN  |
|                      | QSPIC_DMY_FORCE    |

| Register             | Bit field          |
|----------------------|--------------------|
| QSPIC_CTRLMODE_REG   | QSPIC_AUTO_MD      |
|                      | QSPIC_CLK_MD       |
|                      | QSPIC_IO2_OEN      |
|                      | QSPIC_IO3_OEN      |
|                      | QSPIC_IO2_DAT      |
|                      | QSPIC_IO3_DAT      |
|                      | QSPIC_HRDY_MD      |
|                      | QSPIC_RXD_NEG      |
|                      | QSPIC_RPIPE_EN     |
|                      | QSPIC_PCLK_MD      |
|                      | QSPIC_FORCENSEQ_EN |
|                      | QSPIC_USE_32BA     |
| QSPIC_ERASECMDA_REG  | QSPIC_ERS_INST     |
|                      | QSPIC_WEN_INST     |
|                      | QSPIC_SUS_INST     |
|                      | QSPIC_RES_INST     |
| QSPIC_ERASECMD B_REG | QSPIC_ERS_TX_MD    |
|                      | QSPIC_WEN_TX_MD    |
|                      | QSPIC_SUS_TX_MD    |
|                      | QSPIC_RES_TX_MD    |
|                      | QSPIC_EAD_TX_MD    |
|                      | QSPIC_ERS_CS_HI    |
|                      | QSPIC_ERSRES_HLD   |
|                      | QSPIC_RESSUS_DLY   |
| QSPIC_GP_REG         | QSPIC_PADS_DRV     |
|                      | QSPIC_PADS_SLEW    |
| QSPIC_STATUSCMD_REG  | QSPIC_RSTAT_INST   |
|                      | QSPIC_RSTAT_TX_MD  |
|                      | QSPIC_RSTAT_RX_MD  |
|                      | QSPIC_BUSY_POS     |
|                      | QSPIC_BUSY_VAL     |
|                      | QSPIC_RESSTS_DLY   |
|                      | QSPIC_STSDLY_SEL   |
| QSPIC_UCODE_STA RT   | QSPIC_UCODE_X      |
| CACHE ASSOCCFG_REG   | CACHE_ASSOC        |
| CACHE_CTRL1_REG      | CACHE_FLUSH        |
|                      | CACHE_RES1         |

Table 60: Retention Registers

1 : Cache registers are only retained if  
PMU\_CTRL\_REG[RETAIN\_CACHE=1]

| Register                     | Bit field                       |
|------------------------------|---------------------------------|
| CACHE_CTRL2_REG <sup>1</sup> | CACHE_LEN                       |
|                              | CACHE_WEN                       |
|                              | CACHE_CGEN                      |
|                              | ENABLE_ALSO_OTP_CACHED          |
|                              | ENABLE_ALSO_QSPIFLASH_CACHED    |
| CACHE_CTRL3_REG              | CACHE_ASSOCIATIVITY_RESET_VALUE |
|                              | CACHE_LINE_SIZE_RESET_VALUE     |
|                              | CACHE_RAM_SIZE_RESET_VALUE      |
|                              | CACHE_CONTROLLER_RESET          |
| CACHE_LNSIZECFG_REG          | CACHE_LINE                      |
| CACHE_MRM_CTRL_REG           | MRM_START                       |
|                              | MRM_IRQ_MASK                    |
|                              | MRM_IRQ_TINT_STATUS             |
|                              | MRM_IRQ_THRES_STATU S           |
| CACHE_MRM_HITS_REG           | MRM_HITS                        |
| CACHE_MRM_MISSES_REG         | MRM_MISSES                      |
| CACHE_MRM_THRES_REG          | MRM_THRES                       |
| CACHE_MRM_TINT_REG           | MRM_TINT                        |
| SWD_RESET_REG                | SWD_HW_RESET_REQ                |
| OTPC_MODE_REG                | OTPC_MODE_MODE                  |
| OTPC_NWORDS_REG              | OTPC_NWORDS                     |
| OTPC_TIM1_REG                | OTPC_TIM1_CC_T_CADX             |
|                              | OTPC_TIM1_CC_T_PW               |
|                              | OTPC_TIM1_CC_T_1US              |
|                              | OTPC_TIM1_CC_T_500NS            |
|                              | OTPC_TIM1_CC_T_200NS            |
|                              | OTPC_TIM1_CC_T_25NS             |
| OTPC_TIM2_REG                | OTPC_TIM2_CC_STBY_TH R          |
|                              | OTPC_TIM2_CC_T_BCHK             |
|                              | OTPC_TIM2_RDENL_PRO T           |
| DEBUG_REG                    | DEBUGS_FREEZE_EN                |

## 37.1 OTPC REGISTER FILE

Table 61: Register map OTPC

| Address    | Port            | Description                                                                                 |
|------------|-----------------|---------------------------------------------------------------------------------------------|
| 0x07F40000 | OTPC_MODE_REG   | Mode register                                                                               |
| 0x07F40004 | OTPC_PCTRL_REG  | Bit-programming control register                                                            |
| 0x07F40008 | OTPC_STAT_REG   | Status register                                                                             |
| 0x07F4000C | OTPC_AHBADR_REG | AHB master start address                                                                    |
| 0x07F40010 | OTPC_CELADR_REG | Macrocell start address                                                                     |
| 0x07F40014 | OTPC_NWORDS_REG | Number of words                                                                             |
| 0x07F40018 | OTPC_FFPRT_REG  | Ports access to fifo logic                                                                  |
| 0x07F4001C | OTPC_FFRD_REG   | The data which have taken with the latest read from the OTPC_FFPRT_REG                      |
| 0x07F40020 | OTPC_PWORDL_REG | The 32 lower bits of the 64-bit word that will be programmed, when the MPROG mode is used.  |
| 0x07F40024 | OTPC_PWORDH_REG | The 32 higher bits of the 64-bit word that will be programmed, when the MPROG mode is used. |
| 0x07F40028 | OTPC_TIM1_REG   | Various timing parameters of the OTP cell.                                                  |
| 0x07F4002C | OTPC_TIM2_REG   | Various timing parameters of the OTP cell.                                                  |

Table 62: OTPC\_MODE\_REG (0x07F40000)

| Bit   | Mode | Symbol                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|-------|------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:30 | -    | -                     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 29:28 | -    | -                     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 27:10 | -    | -                     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 9     | R/W  | OTPC_MODE_RLD_RR_REQ  | Write with 1 in order to be requested the reloading of the repair records. The reloading of the repair records will be performed at the next enabling of the OTP cell. That means that first the controller should be configured to the STBY mode and after should be activated any other mode. The hardware will clear this register, when the reloading will be performed.<br>The reloading has meaning only if the repair records have been updated manually (MPROG mode). | 0x0   |
| 8     | R/W  | OTPC_MODE_USE_SP_ROWS | Selects the memory area of the OTP cell that will be used.<br>0 - Uses the normal memory area of the OTP cell<br>1 - Uses the spare rows of the OTP cell<br>This selection has meaning only if the mode of the controller is not TDEC and TWR. The controller should be in STBY mode, in order to takes into account this bit. The selection will take effect at the next mode that will be enabled.                                                                          | 0x0   |
| 7     | -    | -                     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |

Table 62: OTPC\_MODE\_REG (0x07F40000)

| Bit | Mode | Symbol                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|-----|------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | OTPC_MODE_ERR_RESP_DIS | When is performed a read from the OTP memory in the MREAD mode, a double error is likely be detected during the retrieving of the data from the OTP. This error condition is always indicated in the status bit OTPC_STAT_REG[OTPC_STAT_ERROR]. However, the OTP controller has also the ability to indicates this error condition, by generating an ERROR response in the AHB bus. The generation of the ERROR response can be avoided with the help of this configuration bit.<br>0 - The OTP controller generates an ERROR response in the AHB bus, when a double error is detected during a reading in MREAD mode. The OTPC_STAT_REG[OTPC_STAT_ERROR] is also updated. The receiving of an ERROR response by the CPU causes a Hard Fault exception in the CPU.<br>1 - Only the OTPC_STAT_REG[OTPC_STAT_ERROR] is updated in a case of such error. The OTP controller will not generate an ERROR response in the AHB bus. | 0x0   |
| 5   | R/W  | OTPC_MODE_FIFO_FLUSH   | By writing with 1, removes any content from the fifo. This bit returns automatically to value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 4   | R/W  | OTPC_MODE_USE_DMA      | Selects the use of the dma, when the controller is configured in one of the modes: AREAD or APROG.<br>0 - The dma is not used. The data should be transferred from/to controller through the register OTPC_FFPRT_REG.<br>1 - The dma is used. The data transfers from/to controller are performed automatically, with the help of the internal DMA of the OTP controller. The AHB base address should be configured in register OTPC_AHBADR_REG, before the selection of one of the two modes: AREAD or APROG.                                                                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 3   | -    | -                      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 2:0 | R/W  | OTPC_MODE_MODE         | Defines the mode of operation of the OTPC controller. The encoding of the modes is as follows:<br>000 - STBY mode<br>001 - MREAD mode<br>010 - MPROG mode<br>011 - AREAD mode<br>100 - APROG mode<br>101 - TBLANK mode<br>110 - TDEC mode<br>111 - TWR mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |

Table 63: OTPC\_PCTRL\_REG (0x07F40004)

| Bit   | Mode | Symbol            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|-------|------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:16 | -    | -                 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 15    | R/W  | OTPC_PCTRL_PSTART | Write with '1' to trigger the programming of one OTP word, in the case where the MPROG mode is selected. The bit is cleared automatically. The 64-bits that will be programmed into the OTP memory are contained into the two registers OTPC_PWORDx_REG.<br>This bit should be used when a new programming is initiated, but also when the programming must be retried.<br>The OTPC_PCTRL_WADDR defines the OTP position where will be performed the programming. | 0x0   |



Table 63: OTPC\_PCTRL\_REG (0x07F40004)

| Bit  | Mode | Symbol            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 14   | R/W  | OTPC_PCTRL_PRETRY | It distinguishes the first attempt of a programming of an OTP position, from a retry of programming.<br>0 - A new value will be programmed in a blank OTP position. The hardware will try to write all the bits that are equal to '1'.<br>1 - The programming that is applied is not the first attempt, but is a request for reprogramming. Will be processed only the bits that were failed to be programmed during the previous attempt. The hardware knows the bits that were failed during the previous attempt.<br>The registers OTPC_PWORDx_REG should contain the 64 bits of the value that should be programmed, independent of the value of the OTPC_PCTRL_PRETRY bit.<br>Also, the OTPC_PCTRL_WADDR should contain always the required OTP address.<br>A retry of a programming should be requested only if the previous action was the first attempt of programming or a retry of programming. Should not be requested a retry if the first attempt has not been performed. | 0x0   |
| 13   | -    | -                 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 12:0 | R/W  | OTPC_PCTRL_WADDR  | Defines the OTP position where will be programmed the 64-bits that are contained into the registers OTPC_PWORDx_REG. It points to a physical 72 bits OTP word.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |

Table 64: OTPC\_STAT\_REG (0x07F40008)

| Bit   | Mode  | Symbol           | Description                                                                                                                                                                                                                                                                                               | Reset |
|-------|-------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:30 | -     | -                | Reserved                                                                                                                                                                                                                                                                                                  | 0x0   |
| 29:16 | R     | OTPC_STAT_NWORDS | It contains the "live" value of the number of (32 bits) words that remain to be processed by the controller.                                                                                                                                                                                              | 0x0   |
| 15:12 | -     | -                | Reserved                                                                                                                                                                                                                                                                                                  | 0x0   |
| 11:8  | R     | OTPC_STAT_FWORDS | Indicates the number of words which contained in the fifo of the controller.                                                                                                                                                                                                                              | 0x0   |
| 7     | R0/WC | OTPC_STAT_RERROR | Indicates that during a normal reading (MREAD or AREAD) was reported a double error by the SECEDED logic. That means that the data are corrupted.<br>0 - The read data are considered as correct.<br>1 - The SECEDED logic detects a double error.<br>This bit can be cleared only with a write with '1'. | 0x0   |
| 6     | R     | OTPC_STAT_ARDY   | Should be used to monitor the progress of the AREAD and APROG modes.<br>0 - One of the APROG or AREAD mode is selected. The controller is busy.<br>1 - The controller is not in an active AREAD or APROG mode.                                                                                            | 0x1   |
| 5     | R     | OTPC_STAT_TERROR | Indicates the result of a test sequence. Should be checked after the end of a TBLANK, TDEC and TWR mode (OTPC_STAT_TRDY = 1).<br>0 - The test sequence ends with no error.<br>1 - The test sequence has failed.                                                                                           | 0x0   |

Table 64: OTPC\_STAT\_REG (0x07F40008)

| Bit | Mode | Symbol             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4   | R    | OTPC_STAT_TRDY     | Indicates the state of a test mode. Should be used to monitor the progress of the TBLANK, TDEC and TWR modes.<br>0 - The controller is busy. One of the test modes is in progress.<br>1 - There is no active test mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x1   |
| 3   | R    | OTPC_STAT_PZERO    | Indicates that the programming sequence has been avoided during a programming request, due to that the word that should be programmed is equal to zero.<br>0 - At least one bit has been programmed into the OTP.<br>1 - The programming has not been performed. All the bits of the word that should be programmed are equal to zero.<br>When the controller is in MPROG mode, this bit can be checked after the end of the programming process (OTPC_STAT_PRDY = 1).<br>During APROG mode, the value of this field it is normal to changing periodically. After the end of the APROG mode (OTPC_STAT_ARDY = 1), this field indicates that one or more of words that have been processed are equal to zero.                                                                                                                                                                                    | 0x0   |
| 2   | R    | OTPC_STAT_PERR_COR | Indicates that a correctable error has been occurred during the word programming process.<br>0 - There is no correctable error in the word-programming process.<br>1 - The process of word - programming reported a correctable error.<br>The correctable error occurs when exactly one bit in an OTP position cannot take the required value. This is not a critical failure in the programming process. The data can still be retrieved correctly by the OTP memory, due to that the error correcting algorithm can repair the corrupted bit.<br>When the controller is in MPROG mode, this bit can be checked after the end of the programming process (OTPC_STAT_PRDY = 1).<br>During APROG mode, the value of this field it is normal to changing periodically. After the end of the APROG mode (OTPC_STAT_ARDY = 1), this field indicates that one or more words had a correctable error. | 0x0   |
| 1   | R    | OTPC_STAT_PERR_UNC | Indicates that an uncorrectable error has been occurred during the word programming process.<br>0 - There is no uncorrectable error in the word-programming process.<br>1 - The process of word-programming failed due to an uncorrectable error.<br>An uncorrectable error is considered when two or more of the bits in an OTP position cannot take the required values. This is a critical failure in the programming process, which means that the data cannot be corrected by the single error correcting algorithm.<br>When the controller is in MPROG mode, this bit should be checked after the end of the programming process (OTPC_STAT_PRDY = 1).<br>During APROG mode, the value of this field it is normal to changing periodically. After the end of the APROG mode (OTPC_STAT_ARDY = 1), this field indicates if the programming was failed or ended successfully.               | 0x0   |

Table 64: OTPC\_STAT\_REG (0x07F40008)

| Bit | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R    | OTPC_STAT_PRDY | Indicates the state of a bit-programming process.<br>0 - The controller is busy. A bit-programming is in progress<br>1 - The logic which performs bit-programming is idle.<br>When the controller is in MPROG mode, this bit should be used to monitor the progress of a programming request.<br>During APROG mode, the value of this field it is normal to changing periodically. | 0x1   |

Table 65: OTPC\_AHBADR\_REG (0x07F4000C)

| Bit  | Mode | Symbol      | Description                                                                                                                                                       | Reset     |
|------|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 31:2 | R/W  | OTPC_AHBADR | It is the AHB address used by the AHB master interface of the controller (the bits [31:2]). The bits [1:0] of the address are considered always as equal to zero. | 0x1FF0000 |
| 1:0  | -    | -           | Reserved                                                                                                                                                          | 0x0       |

Table 66: OTPC\_CELADR\_REG (0x07F40010)

| Bit   | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-------|------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:14 | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 13:0  | R/W  | OTPC_CELADR | It represents an OTP address, where the OTP word width should be considered equal to 32-bits.<br>The physical word width of the OTP memory is 72 bits. The 8-bits of them are used for the implementation of an error correcting code and are not available for the application. The remaining 64 bits of the physical word are available for the application.<br>The OTPC_CELADDR can distinguish the upper 32 bits from the lower 32 bits of the available for the application bits of the OTP word.<br>When OTPC_CELADDR[0] = 1 the address refers to the upper 32 bits of the physical OTP address<br>OTPC_CELADDR[14:1].<br>The register is used during the modes: AREAD and APROG. | 0x0   |

Table 67: OTPC\_NWORDS\_REG (0x07F40014)

| Bit   | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                   | Reset |
|-------|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:14 | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 13:0  | R/W  | OTPC_NWORDS | The number of words (minus one) for reading /programming during the AREAD/APROG mode.<br>The width of the word should be considered equal to 32-bits.<br>The value of the register remains unchanged, by the internal logic of the controller.<br>During mirroring, this register reflects the current amount of copied data. | 0x0   |

Table 68: OTPC\_FFPRT\_REG (0x07F40018)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|------|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | OTPC_FFPRT | Provides access to the fifo through an access port. Write to this register with the corresponding data, when the APROG mode is selected and the dma is disabled. Read from this register the corresponding data, when the AREAD mode is selected and the dma is disabled. The software should check the OTPCC_STAT_FWORDS register for the availability of data/space, before accessing the fifo. | 0x0   |

Table 69: OTPC\_FFRD\_REG (0x07F4001C)

| Bit  | Mode | Symbol    | Description                                                                                | Reset |
|------|------|-----------|--------------------------------------------------------------------------------------------|-------|
| 31:0 | R    | OTPC_FFRD | Contains the value which taken from the fifo, after a read of the OTPC_FFPRT_REG register. | 0x0   |

Table 70: OTPC\_PWORDL\_REG (0x07F40020)

| Bit  | Mode | Symbol      | Description                                                                                                                   | Reset |
|------|------|-------------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | OTPC_PWORDL | Contains the lower 32 bits that can be programmed with the help of the OTPC_PCTRL_REG, while the controller is in MPROG mode. | 0x0   |

Table 71: OTPC\_PWORDH\_REG (0x07F40024)

| Bit  | Mode | Symbol      | Description                                                                                                                   | Reset |
|------|------|-------------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | OTPC_PWORDH | Contains the upper 32 bits that can be programmed with the help of the OTPC_PCTRL_REG, while the controller is in MPROG mode. | 0x0   |

Table 72: OTPC\_TIM1\_REG (0x07F40028)

| Bit   | Mode | Symbol               | Description                                                                                                                                                                                                                                                                                                               | Reset |
|-------|------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31    | R/W  | OTPC_TIM1_CC_T_25NS  | The number of hclk_c clock periods (minus one) that give a time interval at least higher than 25ns.                                                                                                                                                                                                                       | 0x0   |
| 30:27 | R/W  | OTPC_TIM1_CC_T_200NS | The number of hclk_c clock periods (minus one) that give a time interval at least higher than 200ns.                                                                                                                                                                                                                      | 0x3   |
| 26:22 | R/W  | OTPC_TIM1_CC_T_500NS | The number of hclk_c clock periods (minus one) that give a time interval at least higher than 500ns                                                                                                                                                                                                                       | 0x8   |
| 21:16 | R/W  | OTPC_TIM1_CC_T_1US   | The number of hclk_c clock periods (minus one) that give a time interval at least higher than 1us.                                                                                                                                                                                                                        | 0x10  |
| 15:8  | R/W  | OTPC_TIM1_CC_T_PW    | The number of hclk_c clock periods (minus one) that give a time interval that is<br>- at least higher than 4.8us<br>- and lower than 5.2 us<br>It is preferred the programmed value to give a time interval equal to 5us.<br>It defines the duration of the programming pulse for every bit that written in the OTP cell. | 0x4F  |
| 7:0   | R/W  | OTPC_TIM1_CC_T_CAD_X | The number of hclk_c clock periods (minus one) that give a time interval at least higher than 2us. It is used as a wait time each time where the OTP cell is enabled.                                                                                                                                                     | 0x20  |

Table 73: OTPC\_TIM2\_REG (0x07F4002C)

| Bit   | Mode | Symbol                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-------|------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:24 | -    | -                     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 23    | R/W  | OTPC_TIM2_RDENL_PROT  | <p>This bit has meaning only when the OTPC_TIM1_CC_T_25NS = 1, otherwise has no functionality.</p> <p>0 - The minimum number of clock cycles for which the signal read_enable of the OTP memory stays inactive is one clock cycle. This is also applicable if OTPC_TIM1_CC_T_25NS = 0.</p> <p>1 - The minimum number of clock cycles for which the signal read_enable of the OTP memory stays inactive is two clock cycles. The controller adds one extra wait state in the AHB access, if it is required, in order to achieve this constraint. This setting is applicable only if OTPC_TIM1_CC_T_25NS = 1.</p>                                    | 0x0   |
| 22:16 | R/W  | OTPC_TIM2_CC_T_BCHK   | The number of hclk_c clock periods (minus one) that give a time interval between 100ns and 200ns. This time interval is used for the reading of the contents of the OTP cell during the TBLANK mode.                                                                                                                                                                                                                                                                                                                                                                                                                                               | 0x1   |
| 15:10 | -    | -                     | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 9:0   | R/W  | OTPC_TIM2_CC_STBY_THR | <p>This register controls a power saving feature, which is applicable only in MREAD mode. The controller monitors the accesses in the OTP cell. If there is no access for more than OTPC_TIM2_CC_STBY_THR hclk_c clock cycles, the OTP cell goes to the standby while the controller itself remains in the MREAD mode. The OTP cell will be enabled again when will be applied a new read request. The enabling of the OTP cell has a cost of 2us (OTPC_TIM1_CC_T_CADX hclk_c clock cycles).</p> <p>When OTPC_TIM2_CC_STBY_THR = 0 the power saving feature is disabled and the OTP cell remains active while the controller is in MREAD mode.</p> | 0x0   |

## 37.2 QSPIC REGISTER FILE

Table 74: Register map QSPIC

| Address    | Port                 | Description                                                   |
|------------|----------------------|---------------------------------------------------------------|
| 0x0C000000 | QSPIC_CTRLBUS_REG    | SPI Bus control register for the Manual mode                  |
| 0x0C000004 | QSPIC_CTRLMODE_REG   | Mode Control register                                         |
| 0x0C000008 | QSPIC_RECVDATA_REG   | Received data for the Manual mode                             |
| 0x0C00000C | QSPIC_BURSTCMDA_REG  | The way of reading in Auto mode (command register A)          |
| 0x0C000010 | QSPIC_BURSTCMDDB_REG | The way of reading in Auto mode (command register B)          |
| 0x0C000014 | QSPIC_STATUS_REG     | The status register of the QSPI controller                    |
| 0x0C000018 | QSPIC_WRITEDATA_REG  | Write data to SPI Bus for the Manual mode                     |
| 0x0C00001C | QSPIC_READDATA_REG   | Read data from SPI Bus for the Manual mode                    |
| 0x0C000020 | QSPIC_DUMMYDATA_REG  | Send dummy clocks to SPI Bus for the Manual mode              |
| 0x0C000024 | QSPIC_ERASECTRL_REG  | QSPI Erase control register                                   |
| 0x0C000028 | QSPIC_ERASECMDA_REG  | The way of erasing in Auto mode (command register A)          |
| 0x0C00002C | QSPIC_ERASECMDDB_REG | The way of erasing in Auto mode (command register B)          |
| 0x0C000030 | QSPIC_BURSTBRK_REG   | Read break sequence in Auto mode                              |
| 0x0C000034 | QSPIC_STATUSCMD_REG  | The way of reading the status of external device in Auto mode |
| 0x0C000038 | QSPIC_CHKERASE_REG   | Check erase progress in Auto mode                             |
| 0x0C00003C | QSPIC_GP_REG         | QSPI General Purpose control register                         |

Table 75: QSPIC\_CTRLBUS\_REG (0x0C000000)

| Bit  | Mode | Symbol            | Description                                                                            | Reset |
|------|------|-------------------|----------------------------------------------------------------------------------------|-------|
| 31:5 | -    | -                 | Reserved                                                                               | 0x0   |
| 4    | W    | QSPIC_DIS_CS      | Write 1 to disable the chip select (active low) when the controller is in Manual mode. | 0x0   |
| 3    | W    | QSPIC_EN_CS       | Write 1 to enable the chip select (active low) when the controller is in Manual mode.  | 0x0   |
| 2    | W    | QSPIC_SET_QUAD    | Write 1 to set the bus mode in Quad mode when the controller is in Manual mode.        | 0x0   |
| 1    | W    | QSPIC_SET_DUAL    | Write 1 to set the bus mode in Dual mode when the controller is in Manual mode.        | 0x0   |
| 0    | W    | QSPIC_SET_SINGL E | Write 1 to set the bus mode in Single SPI mode when the controller is in Manual mode.  | 0x0   |

Table 76: QSPIC\_CTRLMODE\_REG (0x0C000004)

| Bit   | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                               | Reset |
|-------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:14 | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 13    | R/W  | QSPIC_USE_32BA | Controls the length of the address that the external memory device uses.<br>0 - The external memory device uses 24 bits address.<br>1 - The external memory device uses 32 bits address.<br>The controller uses this bit in order to decide the number of the address bytes that has to transfer to the external device during Auto mode. | 0x0   |

Table 76: QSPIC\_CTRLMODE\_REG (0x0C000004)

| Bit  | Mode | Symbol             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|------|------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 12   | R/W  | QSPIC_FORCENSEQ_EN | Controls the way with which is addressed by the QSPI controller a burst request from the AMBA bus.<br>0: The controller translates a burst access on the AMBA bus as a burst access on the QSPI bus. That results to the minimum number of command/address phases.<br>1: The controller will split a burst access on the AMBA bus into a number of single accesses on the QSPI bus. That results to a separate command for each beat of the burst. E.g a 4-beat word incremental AMBA read access will be split into 4 different sequences on the QSPI bus: command/address/extra clock/read data. The QSPI_CS will be low only for the time that is needed for each of these single accesses. This configuration bit is usefull when the clock frequency of the QSPI bus is much higher than the clock of the AMBA bus. In this case the interval for which the CS remains low is minimized, achieving lower power dissipation with respect of the case where the QSPIC_FORCENSEQ_EN=0, at cost of performance. | 0x0   |
| 11:9 | R/W  | QSPIC_PCLK_MD      | Read pipe clock delay relative to the falling edge of QSPI_SCK.<br>Refer to QSPI Timing for timing parameters and recommended values: 0 to 7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 8    | R/W  | QSPIC_RPIPE_EN     | Controls the use of the data read pipe.<br>0 = The read pipe is disabled; the sampling clock is defined according to the QSPIC_RXD_NEG setting.<br>1 = The read pipe is enabled. The delay of the sampling clock is defined according to the QSPI_PCLK_MD setting. (Recommended)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 7    | R/W  | QSPIC_RXD_NEG      | Defines the clock edge that is used for the capturing of the received data, when the read pipe is not active (QSPIC_RPIPE_EN = 0).<br>0: Sampling of the received data with the positive edge of the QSPI_SCK<br>1: Sampling of the received data with the negative edge of the QSPI_SCK<br>The internal QSPI_SCK clock that is used by the controller for the capturing of the received data has a skew in respect of the QSPI_SCK that is received by the external memory device. In order to be improved the timing requirements of the read path, the controller supports a read pipe register with programmable clock delay. See also the QSPIC_RPIPE_EN register.                                                                                                                                                                                                                                                                                                                                          | 0x0   |



Table 76: QSPIC\_CTRLMODE\_REG (0x0C000004)

| Bit | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | QSPIC_HRDY_MD | This configuration bit is useful when the frequency of the QSPI clock is much lower than the clock of the AMBA bus, in order to not locks the AMBA bus for a long time.<br>0: Adds wait states via hready signal when an access is performed on the QSPIC_WRITEDATA, QSPIC_READDATA and QSPIC_DUMMYDATA registers. It is not needed to checked the QSPIC_BUSY of the QSPIC_STATUS_REG.<br>1: The controller don't adds wait states via the hready signal, when is performed access on the QSPIC_WRITEDATA, QSPIC_READDATA and QSPIC_DUMMYDATA registers. The QSPIC_BUSY bit of the QSPIC_STATUS_REG must be checked in order to be detected the completion of the requested access.<br>It is applicable only when the controller is in Manual mode. In the case of the Auto mode, the controller always adds wait states via the hready signal. | 0x0   |
| 5   | R/W  | QSPIC_IO3_DAT | The value of QSPI_IO3 pad if QSPI_IO3_OEN is 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 4   | R/W  | QSPIC_IO2_DAT | The value of QSPI_IO2 pad if QSPI_IO2_OEN is 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 3   | R/W  | QSPIC_IO3_OEN | QSPI_IO3 output enable. Use this only in SPI or Dual SPI mode to control /HOLD signal. When the Auto Mode is selected (QSPIC_AUTO_MD = 1) and the QUAD SPI is used, set this bit to zero.<br>0: The QSPI_IO3 pad is input.<br>1: The QSPI_IO3 pad is output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 2   | R/W  | QSPIC_IO2_OEN | QSPI_IO2 output enable. Use this only in SPI or Dual SPI mode to control /WP signal. When the Auto Mode is selected (QSPIC_AUTO_MD = 1) and the QUAD SPI is used, set this bit to zero.<br>0: The QSPI_IO2 pad is input.<br>1: The QSPI_IO2 pad is output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 1   | R/W  | QSPIC_CLK_MD  | Mode of the generated QSPI_SCK clock<br>0: Use Mode 0 for the QSPI_CLK. The QSPI_SCK is low when QSPI_CS is high.<br>1: Use Mode 3 for the QSPI_CLK. The QSPI_SCK is high when QSPI_CS is high.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 0   | R/W  | QSPIC_AUTO_MD | Mode of operation<br>0: The Manual Mode is selected.<br>1: The Auto Mode is selected.<br>During an erasing the QSPIC_AUTO_MD goes in read only mode (see QSPIC_ERASE_EN)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |

Table 77: QSPIC\_RECVDATA\_REG (0x0C000008)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                              | Reset |
|------|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R    | QSPIC_RECVDATA | This register contains the received data when the QSPIC_READDATA_REG register is used in Manual mode, in order to be retrieved data from the external memory device and QSPIC_HRDY_MD=1 && QSPIC_BUSY=0. | 0x0   |

Table 78: QSPIC\_BURSTCMDA\_REG (0x0C00000C)

| Bit   | Mode | Symbol           | Description                                                                                                                                                                                                                                      | Reset |
|-------|------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:30 | R/W  | QSPIC_DMY_TX_MD  | It describes the mode of the SPI bus during the Dummy bytes phase.<br>00 - Single SPI<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                 | 0x0   |
| 29:28 | R/W  | QSPIC_EXT_TX_MD  | It describes the mode of the SPI bus during the Extra Byte phase.<br>00 - Single SPI<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                  | 0x0   |
| 27:26 | R/W  | QSPIC_ADR_TX_MD  | It describes the mode of the SPI bus during the address phase.<br>00 - Single SPI<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                     | 0x0   |
| 25:24 | R/W  | QSPIC_INST_TX_MD | It describes the mode of the SPI bus during the instruction phase.<br>00 - Single SPI<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                 | 0x0   |
| 23:16 | R/W  | QSPIC_EXT_BYTE   | The value of an extra byte which will be transferred after address (only if QSPIC_EXT_BYTE_EN= 1). Usually this is the Mode Bits in Dual/Quad SPI I/O instructions.                                                                              | 0x0   |
| 15:8  | R/W  | QSPIC_INST_WB    | Instruction Value for Wrapping Burst. This value is the selected instruction when QSPIC_WRAP_MD is equal to 1 and the access is a wrapping burst of length and size described by the bit fields QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE respectively. | 0x0   |
| 7:0   | R/W  | QSPIC_INST       | Instruction Value for Incremental Burst or Single read access. This value is the selected instruction at the cases of incremental burst or single read access. Also this value is used when a wrapping burst is not supported (QSPIC_WRAP_MD)    | 0x0   |

Table 79: QSPIC\_BURSTCMDDB\_REG (0x0C000010)

| Bit   | Mode | Symbol           | Description                                                                                                                                                                                                                                                                                    | Reset |
|-------|------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:16 | -    | -                | Reserved                                                                                                                                                                                                                                                                                       | 0x0   |
| 15    | R/W  | QSPIC_DMY_FORC E | By setting this bit, the number of dummy bytes is forced to be equal to 3. In this case the QSPIC_DMY_NUM field is overruled and has no function.<br>0 - The number of dummy bytes is controlled by the QSPIC_DMY_NUM field<br>1 - Three dummy bytes are used. The QSPIC_DMY_NUM is overruled. | 0x0   |

Table 79: QSPIC\_BURSTCMD\_REG (0x0C000010)

| Bit   | Mode | Symbol                | Description                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|-------|------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 14:12 | R/W  | QSPIC_CS_HIGH_M<br>IN | Between the transmissions of two different instructions to the flash memory, the SPI bus stays in idle state (QSPI_CS high) for at least this number of QSPI_SCK clock cycles. See the QSPIC_ERS_CS_HI register for some exceptions.                                                                                                                                                                                                    | 0x0   |
| 11:10 | R/W  | QSPIC_WRAP_SIZE       | It describes the selected data size of a wrapping burst (QSPIC_WRAP_MD).<br>00 - byte access (8-bits)<br>01 - half word access (16 bits)<br>10 - word access (32-bits)<br>11 - Reserved                                                                                                                                                                                                                                                 | 0x0   |
| 9:8   | R/W  | QSPIC_WRAP_LEN        | It describes the selected length of a wrapping burst (QSPIC_WRAP_MD).<br>00 - 4 beat wrapping burst<br>01 - 8 beat wrapping burst<br>10 - 16 beat wrapping burst<br>11 - Reserved                                                                                                                                                                                                                                                       | 0x0   |
| 7     | R/W  | QSPIC_WRAP_MD         | Wrap mode<br>0 - The QSPIC_INST is the selected instruction at any access.<br>1 - The QSPIC_INST_WB is the selected instruction at any wrapping burst access of length and size described by the registers QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE respectively. In all other cases the QSPIC_INST is the selected instruction. Use this feature only when the serial FLASH memory supports a special instruction for wrapping burst access. | 0x0   |
| 6     | R/W  | QSPIC_INST_MD         | Instruction mode<br>0 - Transmit instruction at any burst access.<br>1 - Transmit instruction only in the first access after the selection of Auto Mode.                                                                                                                                                                                                                                                                                | 0x0   |
| 5:4   | R/W  | QSPIC_DMY_NUM         | Number of Dummy Bytes<br>00 - Zero Dummy Bytes (Don't Send Dummy Bytes)<br>01 - Send 1 Dummy Byte<br>10 - Send 2 Dummy Bytes<br>11 - Send 4 Dummy Bytes<br>When QSPIC_DMY_FORCE is enabled, the QSPIC_DMY_NUM is overruled. In this case the number of dummy bytes is defined by the QSPIC_DMY_FORCE and is equal to 3, independent of the value of the QSPIC_DMY_NUM.                                                                  | 0x0   |
| 3     | R/W  | QSPIC_EXT_HF_DS       | Extra Half Disable Output<br>0 - if QSPIC_EXT_BYTE_EN=1, is transmitted the complete QSPIC_EXT_BYTE<br>1 - if QSPIC_EXT_BYTE_EN=1, the output is disabled (hi-z) during the transmission of bits [3:0] of QSPIC_EXT_BYTE                                                                                                                                                                                                                | 0x0   |
| 2     | R/W  | QSPIC_EXT_BYTE_<br>EN | Extra Byte Enable<br>0 - Don't Send QSPIC_EXT_BYTE<br>1 - Send QSPIC_EXT_BYTE                                                                                                                                                                                                                                                                                                                                                           | 0x0   |

Table 79: QSPIC\_BURSTCMDDB\_REG (0x0C000010)

| Bit | Mode | Symbol          | Description                                                                                                               | Reset |
|-----|------|-----------------|---------------------------------------------------------------------------------------------------------------------------|-------|
| 1:0 | R/W  | QSPIC_DAT_RX_MD | It describes the mode of the SPI bus during the data phase.<br>00 - Single SPI<br>01 - Dual<br>10 - Quad<br>11 - Reserved | 0x0   |

Table 80: QSPIC\_STATUS\_REG (0x0C000014)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                          | Reset |
|------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:1 | -    | -          | Reserved                                                                                                                                                                                                             | 0x0   |
| 0    | R/W  | QSPIC_BUSY | The status of the SPI Bus.<br>0 - The SPI Bus is idle<br>1 - The SPI Bus is active. Read data, write data or dummy data activity is in progress.<br>Has meaning only in Manual mode and only when QSPIC_HRDY_MD = 1. | 0x0   |

Table 81: QSPIC\_WRITEDATA\_REG (0x0C000018)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|------|------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | W    | QSPIC_WRITEDATA | Writing to this register is generating a data transfer from the controller to the external memory device. The data written in this register is then transferred to the memory using the selected mode of the SPI bus (SPI, Dual SPI, Quad SPI). The data size of the access to this register can be 32-bits / 16-bits/ 8-bits and is equal to the number of the transferred bits.<br>This register has meaning only when the controller is in Manual mode. | 0x0   |

Table 82: QSPIC\_READDATA\_REG (0x0C00001C)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R    | QSPIC_READDATA | A read access at this register generates a data transfer from the external memory device to the QSPIC controller. The data is transferred using the selected mode of the SPI bus (SPI, Dual SPI, Quad SPI). The data size of the access to this register can be 32-bits / 16-bits / 8-bits and is equal to the number of the transferred bits.<br>This register has meaning only when the controller is in Manual mode. | 0x0   |

Table 83: QSPIC\_DUMMYDATA\_REG (0x0C000020)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | W    | QSPIC_DUMMYDATA | Writing to this register generates a number of clock pulses to the SPI bus. During the last clock of this activity in the SPI bus, the QSPI_IOx data pads are in hi-z state. The data size of the access to this register can be 32-bits / 16-bits/ 8-bits. The number of generated pulses is equal to: (size of AHB bus access) / (size of SPI bus). The size of SPI bus is equal to 1, 2 or 4 for Single, Dual or Quad SPI mode respectively. This register has meaning only when the controller is in Manual mode. | 0x0   |

Table 84: QSPIC\_ERASECTRL\_REG (0x0C000024)

| Bit   | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-------|------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:28 | -    | -               | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 27:25 | R    | QSPIC_ERS_STATE | It shows the progress of sector/block erasing (read only).<br>000 = No Erase.<br>001 = Pending erase request<br>010 = Erase procedure is running<br>011 = Suspended Erase procedure<br>100 = Finishing the Erase procedure<br>101..111 = Reserved                                                                                                                                                                                                        | 0x0   |
| 24    | R/W  | QSPIC_ERASE_EN  | During Manual mode (QSPIC_AUTO_MD = 0). This bit is in read only mode.<br>During Auto mode (QSPIC_AUTO_MD = 1). To request the erasing of the block/sector (QSPIC_ERS_ADDR, 12'b0) write 1 to this bit. This bit is cleared automatically with the end of the erasing. Until the end of erasing the QSPIC_ERASE_EN remains in read only mode. During the same period of time the controller remains in Auto Mode (QSPIC_AUTO_MD goes in read only mode). | 0x0   |
| 23:4  | R/W  | QSPIC_ERS_ADDR  | Defines the address of the block/sector that is requested to be erased.<br>If QSPIC_USE_32BA = 0 (24 bits addressing), bits QSPIC_ERASECTRL_REG[23-12] determine the block/sector address bits [23-12].<br>QSPIC_ERASECTRL_REG[11-4] are ignored by the controller.<br>If QSPIC_USE_32BA = 1 (32 bits addressing) bits QSPIC_ERASECTRL_REG[23-4] determine the block / sectors address bits [31:12]                                                      | 0x0   |
| 3:0   | -    | -               | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |

Table 85: QSPIC\_ERASECMDA\_REG (0x0C000028)

| Bit   | Mode | Symbol         | Description                                      | Reset |
|-------|------|----------------|--------------------------------------------------|-------|
| 31:24 | R/W  | QSPIC_RES_INST | The code value of the erase resume instruction   | 0x0   |
| 23:16 | R/W  | QSPIC_SUS_INST | The code value of the erase suspend instruction. | 0x0   |
| 15:8  | R/W  | QSPIC_WEN_INST | The code value of the write enable instruction   | 0x0   |
| 7:0   | R/W  | QSPIC_ERS_INST | The code value of the erase instruction.         | 0x0   |

Table 86: QSPIC\_ERASECMDB\_REG (0x0C00002C)

| Bit   | Mode | Symbol           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|-------|------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:30 | -    | -                | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 29:24 | R/W  | QSPIC_RESSUS_DLY | Defines a timer that counts the minimum allowed delay between an erase suspend command and the previous erase resume command (or the initial erase command).<br>0 = Dont wait. The controller starts immediately to suspend the erase procedure.<br>1..63 = The controller waits for at least this number of 222kHz clock cycles before the suspension of erasing. Time starts counting after the end of the previous erase resume command (or the initial erase command) | 0x0   |
| 23:20 | -    | -                | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 19:16 | R/W  | QSPIC_ERSRES_HLD | The controller must stay without flash memory reading requests for this number of AMBA hclk clock cycles, before to perform the command of erase or erase resume<br>15 - 0                                                                                                                                                                                                                                                                                                | 0x0   |
| 15    | -    | -                | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 14:10 | R/W  | QSPIC_ERS_CS_HI  | After the execution of instructions: write enable, erase, erase suspend and erase resume, the QSPI_CS remains high for at least this number of qspi bus clock cycles.                                                                                                                                                                                                                                                                                                     | 0x0   |
| 9:8   | R/W  | QSPIC_EAD_TX_MD  | The mode of the QSPI Bus during the address phase of the erase instruction<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 7:6   | R/W  | QSPIC_RES_TX_MD  | The mode of the QSPI Bus during the transmission of the resume instruction<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 5:4   | R/W  | QSPIC_SUS_TX_MD  | The mode of the QSPI Bus during the transmission of the suspend instruction.<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 3:2   | R/W  | QSPIC_WEN_TX_MD  | The mode of the QSPI Bus during the transmission of the write enable instruction.<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                                                                                                                                                                                                                               | 0x0   |

Table 86: QSPIC\_ERASECMD\_REG (0x0C00002C)

| Bit | Mode | Symbol          | Description                                                                                                                              | Reset |
|-----|------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1:0 | R/W  | QSPIC_ERS_TX_MD | The mode of the QSPI Bus during the instruction phase of the erase instruction<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved | 0x0   |

Table 87: QSPIC\_BURSTBRK\_REG (0x0C000030)

| Bit   | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:21 | -    | -               | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 20    | R/W  | QSPIC_SEC_HF_DS | Disable output during the transmission of the second half (QSPIC_BRK_WRD[3:0]). Setting this bit is only useful if QSPIC_BRK_EN = 1 and QSPIC_BRK_SZ = 1.<br>0 - The controller drives the QSPI bus during the transmission of the QSPIC_BRK_WRD[3:0].<br>1 - The controller leaves the QSPI bus in Hi-Z during the transmission of the QSPIC_BRK_WORD[3:0].                                                                                                                                                                                                                                    | 0x0   |
| 19:18 | R/W  | QSPIC_BRK_TX_MD | The mode of the QSPI Bus during the transmission of the burst break sequence.<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 17    | R/W  | QSPIC_BRK_SZ    | The size of Burst Break Sequence<br>0 - One byte (Send QSPIC_BRK_WRD[15:8])<br>1 - Two bytes (Send QSPIC_BRK_WRD[15:0])                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 16    | R/W  | QSPIC_BRK_EN    | Controls the application of a special command (read burst break sequence) that is used in order to force the device to abandon the continuous read mode.<br>0 - The special command is not applied<br>1 - The special command is applied<br>This special command is applied by the controller to the external device under the following conditions:<br>- the controller is in Auto mode<br>- the QSPIC_INST_MD = 1<br>- the previous command that has been applied in the external device was read<br>- the controller want to apply to the external device a command different than the read. | 0x0   |
| 15:0  | R/W  | QSPIC_BRK_WRD   | This is the value of a special command (read burst break sequence) that is applied by the controller to the external memory device, in order to force the memory device to abandon the continuous read mode.                                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |

Table 88: QSPIC\_STATUSCMD\_REG (0x0C000034)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 31:23 | -    | -      | Reserved    | 0x0   |



Table 88: QSPIC\_STATUSCMD\_REG (0x0C000034)

| Bit   | Mode | Symbol            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-------|------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 22    | R/W  | QSPIC_STSDLY_SEL  | Defines the timer which is used to count the delay that it has to wait before to read the FLASH Status Register, after an erase or an erase resume command.<br>0 - The delay is controlled by the QSPIC_RESSTS_DLY which counts on the qspi clock.<br>1 - The delay is controlled by the QSPIC_RESSUS_DLY which counts on the 222 kHz clock.                                                                                                                                                                                                                                                                                                       | 0x0   |
| 21:16 | R/W  | QSPIC_RESSTS_DLY  | Defines a timer that counts the minimum required delay between the reading of the status register and of the previous erase or erase resume instruction.<br>0 - Dont wait. The controller starts to reading the Flash memory status register immediately.<br>1..63 - The controller waits for at least this number of QSPI_CLK cycles and afterwards it starts to reading the Flash memory status register. The timer starts to count after the end of the previous erase or erase resume command. The actual timer that will be used by the controller before the reading of the Flash memory status register is defined by the QSPIC_STSDLY_SEL. | 0x0   |
| 15    | R/W  | QSPIC_BUSY_VAL    | Defines the value of the Busy bit which means that the flash is busy.<br>0 - The flash is busy when the Busy bit is equal to 0.<br>1 - The flash is busy when the Busy bit is equal to 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 14:12 | R/W  | QSPIC_BUSY_POS    | It describes who from the bits of status represents the Busy bit (7 - 0).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 11:10 | R/W  | QSPIC_RSTAT_RX_MD | The mode of the QSPI Bus during the receive status phase of the read status instruction<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 9:8   | R/W  | QSPIC_RSTAT_TX_MD | The mode of the QSPI Bus during the instruction phase of the read status instruction.<br>00 - Single<br>01 - Dual<br>10 - Quad<br>11 - Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 7:0   | R/W  | QSPIC_RSTAT_INST  | The code value of the read status instruction. It is transmitted during the instruction phase of the read status instruction.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |

Table 89: QSPIC\_CHKERASE\_REG (0x0C000038)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                               | Reset |
|------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | W    | QSPIC_CHKERASE | Writing any value to this register during erasing, forces the controller to read the flash memory status register. Depending on the value of the Busy bit, it updates the QSPIC_ERASE_EN. | 0x0   |

Table 90: QSPIC\_GP\_REG (0x0C00003C)

| Bit | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4:3 | R/W  | QSPIC_PADS_SLEW | QSPI pads slew rate control. Indicative values under certain conditions:<br>0: Rise=1.7 V/ns, Fall=1.9 V/ns (weak)<br>1: Rise=2.0 V/ns, Fall=2.3 V/ns<br>2: Rise=2.3 V/ns, Fall=2.6 V/ns<br>3: Rise=2.4 V/ns, Fall=2.7 V/ns (strong)<br>Conditions: FLASH pin capacitance 6 pF, Vcc=1.8V, T=25C and Idrive=16mA. | 0x0   |
| 2:1 | R/W  | QSPIC_PADS_DRV  | QSPI pads drive current<br>0: 4 mA<br>1: 8 mA<br>2: 12 mA<br>3: 16 mA                                                                                                                                                                                                                                            | 0x0   |
| 0   | -    | -               | Reserved                                                                                                                                                                                                                                                                                                         | 0x0   |

## 37.3 BLE REGISTER FILE

Table 91: Register map BLE

| Address    | Port                     | Description                                                                              |
|------------|--------------------------|------------------------------------------------------------------------------------------|
| 0x40000000 | BLE_RWBLECNTRL_REG       | BLE Control register                                                                     |
| 0x40000004 | BLE_VERSION_REG          | Version register                                                                         |
| 0x40000008 | BLE_RWBLECONF_REG        | Configuration register                                                                   |
| 0x4000000C | BLE_INTCNTRL_REG         | Interrupt controller register                                                            |
| 0x40000010 | BLE_INTSTAT_REG          | Interrupt status register                                                                |
| 0x40000014 | BLE_INTRAWSTAT_REG       | Interrupt raw status register                                                            |
| 0x40000018 | BLE_INTACK_REG           | Interrupt acknowledge register                                                           |
| 0x4000001C | BLE_BASETIMECNT_REG      | Base time reference counter                                                              |
| 0x40000020 | BLE_FINETIMECNT_REG      | Fine time reference counter                                                              |
| 0x40000024 | BLE_BDADDRL_REG          | BLE device address LSB register                                                          |
| 0x40000028 | BLE_BDADDRU_REG          | BLE device address MSB register                                                          |
| 0x4000002C | BLE_CURRENTRXDESCPTR_REG | Rx Descriptor Pointer for the Receive Buffer Chained List                                |
| 0x40000030 | BLE_DEEPSLCNTRL_REG      | Deep-Sleep control register                                                              |
| 0x40000034 | BLE_DEEPSLWKUP_REG       | Time (measured in Low Power clock cycles) in Deep Sleep Mode before waking-up the device |
| 0x40000038 | BLE_DEEPSLSTAT_REG       | Duration of the last deep sleep phase register                                           |
| 0x4000003C | BLE_ENBPRESET_REG        | Time in low power oscillator cycles register                                             |
| 0x40000040 | BLE_FINECNTCORR_REG      | Phase correction value register                                                          |
| 0x40000044 | BLE_BASETIMECNTCORR_REG  | Base Time Counter                                                                        |
| 0x40000050 | BLE_DIAGCNTRL_REG        | Diagnostics Register                                                                     |
| 0x40000054 | BLE_DIAGSTAT_REG         | Debug use only                                                                           |
| 0x40000058 | BLE_DEBUGADDMAX_REG      | Upper limit for the memory zone                                                          |
| 0x4000005C | BLE_DEBUGADDMIN_REG      | Lower limit for the memory zone                                                          |
| 0x40000060 | BLE_ERRORYPESTAT_REG     | Error Type Status registers                                                              |
| 0x40000064 | BLE_SWPROFILING_REG      | Software Profiling register                                                              |
| 0x40000070 | BLE_RADIOCNTRL0_REG      | Radio interface control register                                                         |
| 0x40000074 | BLE_RADIOCNTRL1_REG      | Radio interface control register                                                         |
| 0x40000078 | BLE_RADIOCNTRL2_REG      | Radio interface control register                                                         |
| 0x4000007C | BLE_RADIOCNTRL3_REG      | Radio interface control register                                                         |
| 0x40000080 | BLE_RADIOPWUPDN_REG      | RX/TX power up/down phase register                                                       |
| 0x40000090 | BLE_ADVCHMAP_REG         | Advertising Channel Map                                                                  |
| 0x400000A0 | BLE_ADVTIM_REG           | Advertising Packet Interval                                                              |
| 0x400000A4 | BLE_ACTSCANSTAT_REG      | Active scan register                                                                     |
| 0x400000B0 | BLE_WLPUBADDPTR_REG      | Start address of public devices list                                                     |
| 0x400000B4 | BLE_WLPRIVADDPTR_REG     | Start address of private devices list                                                    |
| 0x400000B8 | BLE_WLNBDEV_REG          | Devices in white list                                                                    |
| 0x400000C0 | BLE_AESCNTRL_REG         | Start AES register                                                                       |
| 0x400000C4 | BLE_AESKEY31_0_REG       | AES encryption key                                                                       |
| 0x400000C8 | BLE_AESKEY63_32_REG      | AES encryption key                                                                       |
| 0x400000CC | BLE_AESKEY95_64_REG      | AES encryption key                                                                       |
| 0x400000D0 | BLE_AESKEY127_96_REG     | AES encryption key                                                                       |
| 0x400000D4 | BLE_AESPTR_REG           | Pointer to the block to encrypt/decrypt                                                  |
| 0x400000D8 | BLE_TXMICVAL_REG         | AES / CCM plain MIC value                                                                |

Table 91: Register map BLE

| Address    | Port                  | Description                                  |
|------------|-----------------------|----------------------------------------------|
| 0x400000DC | BLE_RXMICVAL_REG      | AES / CCM plain MIC value                    |
| 0x400000E0 | BLE_RFTESTCNTL_REG    | RF Testing Register                          |
| 0x400000E4 | BLE_RFTESTTXSTAT_REG  | RF Testing Register                          |
| 0x400000E8 | BLE_RFTESTRXSTAT_REG  | RF Testing Register                          |
| 0x400000F0 | BLE_TIMGENCNTL_REG    | Timing Generator Register                    |
| 0x400000F4 | BLE_GROSSTIMTGT_REG   | Gross Timer Target value                     |
| 0x400000F8 | BLE_FINETIMTGT_REG    | Fine Timer Target value                      |
| 0x400000FC | BLE_SAMPLECLK_REG     | Samples the Base Time Counter                |
| 0x40000100 | BLE_COEXIFCNTL0_REG   | Coexistence interface Control 0 Register     |
| 0x40000104 | BLE_COEXIFCNTL1_REG   | Coexistence interface Control 1 Register     |
| 0x40000108 | BLE_BLEMPRIO0_REG     | Coexistence interface Priority 0 Register    |
| 0x4000010C | BLE_BLEMPRIO1_REG     | Coexistence interface Priority 1 Register    |
| 0x40000110 | BLE_BLEPRIOSCHARB_REG | Priority Scheduling Arbiter Control Register |
| 0x40000200 | BLE_CNTL2_REG         | BLE Control Register 2                       |
| 0x40000208 | BLE_EM_BASE_REG       | Exchange Memory Base Register                |
| 0x4000020C | BLE_DIAGCNTL2_REG     | Debug use only                               |
| 0x40000210 | BLE_DIAGCNTL3_REG     | Debug use only                               |

Table 92: BLE\_RWBLECNTL\_REG (0x40000000)

| Bit | Mode | Symbol            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-----|------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31  | R0/W | MASTER_SOFT_RST   | Reset the complete BLE Core except registers and timing generator, when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.                                                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 30  | R0/W | MASTER_TGSOFT_RST | Reset the timing generator, when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 29  | R/W  | REG_SOFT_RST      | Reset the complete register block, when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.<br>Note that INT_STAT will not be cleared, so the user should also write to BLE_INTACK_REG after the SW Reset                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 28  | R0/W | SWINT_REQ         | Forces the generation of ble_sw_irq when written with a 1, and proper masking is set. Resets at 0 when action is performed. No action happens if it is written with 0.                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 26  | R0/W | RFTEST_ABORT      | Abort the current RF Testing defined as per CS-FORMAT when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.<br>Note that when RFTEST_ABORT is requested:<br>1) In case of infinite Tx, the Packet Controller FSM stops at the end of the current byte in process, and processes accordingly the packet CRC.<br>2) In case of Infinite Rx, the Packet Controller FSM either stops as the end of the current Packet reception (if Access address has been detected), or simply stop the processing switching off the RF. | 0x0   |
| 25  | R0/W | ADVERT_ABORT      | Abort the current Advertising event when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 92: BLE\_RWBLECNTL\_REG (0x40000000)

| Bit   | Mode | Symbol        | Description                                                                                                                                                                                                                                      | Reset |
|-------|------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 24    | R0/W | SCAN_ABORT    | Abort the current scan window when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.                                                                                                            | 0x0   |
| 22    | R/W  | MD_DSB        | 0: Normal operation of MD bits management<br>1: Allow a single Tx/Rx exchange whatever the MD bits are.<br>- value forced by SW from Tx Descriptor<br>- value just saved in Rx Descriptor during reception                                       | 0x0   |
| 21    | R/W  | SN_DSB        | 0: Normal operation of Sequence number<br>1: Sequence Number Management disabled:<br>- value forced by SW from Tx Descriptor<br>- value ignored in Rx, where no SN error reported.                                                               | 0x0   |
| 20    | R/W  | NESN_DSB      | 0: Normal operation of Acknowledge<br>1: Acknowledge scheme disabled:<br>- value forced by SW from Tx Descriptor<br>- value ignored in Rx, where no NESN error reported.                                                                         | 0x0   |
| 19    | R/W  | CRYPT_DSB     | 0: Normal operation. Encryption / Decryption enabled.<br>1: Encryption / Decryption disabled.<br>Note that if CS-CRYPT_EN is set, then MIC is generated, and only data encryption is disabled, meaning data sent are plain data.                 | 0x0   |
| 18    | R/W  | WHIT_DSB      | 0: Normal operation. Whitening enabled.<br>1: Whitening disabled.                                                                                                                                                                                | 0x0   |
| 17    | R/W  | CRC_DSB       | 0: Normal operation. CRC removed from data stream.<br>1: CRC stripping disabled on Rx packets, CRC replaced by 0x000 in Tx.                                                                                                                      | 0x0   |
| 16    | R/W  | HOP_REMAP_DSB | 0: Normal operation. Frequency Hopping Remapping algorithm enabled.<br>1: Frequency Hopping Remapping algorithm disabled                                                                                                                         | 0x0   |
| 13:12 | R/W  | CORR_MODE     | Defines correlation mode, meaningful only if RW_BLE_CORR_PREAMBLE_ENABLE is set<br>00: Correlates onto Access Address<br>01: Correlates onto half preamble and Access Address<br>10: Correlates onto full preamble and Access Address<br>11: n/a | 0x0   |
| 9     | R/W  | ADVERTFILT_EN | Advertising Channels Error Filtering Enable control<br>0: RW-BLE Core reports all errors to RW-BLE Software<br>1: RW-BLE Core reports only correctly received packet, without error to RW-BLE Software                                           | 0x0   |
| 8     | R/W  | RWBLE_EN      | 0: Disable RW-BLE Core Exchange Table pre-fetch mechanism.<br>1: Enable RW-BLE Core Exchange table pre-fetch mechanism.                                                                                                                          | 0x0   |
| 7:4   | R/W  | RXWINSZDEF    | Default Rx Window size in us. Used when device:<br>• is master connected<br>• performs its second receipt.<br>0 is not a valid value. Recommended value is 10 (in decimal).                                                                      | 0x0   |
| 2:0   | R/W  | SYNCERR       | Indicates the maximum number of errors allowed to recognize the synchronization word.                                                                                                                                                            | 0x0   |

Table 93: BLE\_VERSION\_REG (0x40000004)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---------------|-------|
| 31:24 | R    | TYP    | BLE Core Type | 0x1   |

Table 93: BLE\_VERSION\_REG (0x40000004)

| Bit   | Mode | Symbol | Description                            | Reset |
|-------|------|--------|----------------------------------------|-------|
| 23:16 | R    | REL    | BLE Core version Major release number. | 0x0   |
| 15:8  | R    | UPG    | BLE Core upgrade Upgrade number.       | 0x0   |
| 7:0   | R    | BUILD  | BLE Core Build Build number.           | 0x0   |

Table 94: BLE\_RWBLECONF\_REG (0x40000008)

| Bit   | Mode | Symbol    | Description                                                                            | Reset |
|-------|------|-----------|----------------------------------------------------------------------------------------|-------|
| 29:24 | R    | ADD_WIDTH | Value of the RW_BLE_ADDRESS_WIDTH parameter converted into binary.                     | 0x10  |
| 22:16 | R    | RFIF      | Radio Interface ID                                                                     | 0x2   |
| 13:8  | R    | CLK_SEL   | Operating Frequency (in MHz)                                                           | 0x0   |
| 6     | R/W  | DECIPHER  | 0: AES deciphering not present                                                         | 0x0   |
| 5     | R    | DMODE     | 0: RW-BLE Core is used as a standalone BLE device                                      | 0x0   |
| 4     | R    | INTMODE   | 1: Interrupts are trigger level generated, i.e. stays active at 1 till acknowledgement | 0x1   |
| 3     | R    | COEX      | 1: WLAN Coexistence mechanism present                                                  | 0x1   |
| 2     | R    | USEDDBG   | 1: Diagnostic port instantiated                                                        | 0x1   |
| 1     | R    | USECRYPT  | 1: AES-CCM Encryption block present                                                    | 0x1   |
| 0     | R    | BUSWIDTH  | Processor bus width:<br>1: 32 bits                                                     | 0x1   |

Table 95: BLE\_INTCNTL\_REG (0x4000000C)

| Bit | Mode | Symbol            | Description                                                                                                                                                                                                                                                      | Reset |
|-----|------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R/W  | CSCNTDEVMSK       | CSCNT interrupt mask during event. This bit allows to enable CSCNT interrupt generation during events (i.e. advertising, scanning, initiating, and connection)<br>0: CSCNT Interrupt not generated during events.<br>1: CSCNT Interrupt generated during events. | 0x1   |
| 9   | R/W  | SWINTMSK          | SW triggered interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated                                                                                                                                                                              | 0x0   |
| 8   | R/W  | EVENTAPFAINTMSK   | End of event / anticipated pre-fetch abort interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated                                                                                                                                                | 0x1   |
| 7   | R/W  | FINETGTIMINTMSK   | Fine Target Timer Mask<br>0: Interrupt not generated<br>1: Interrupt generated                                                                                                                                                                                   | 0x0   |
| 6   | R/W  | GROSSTGTIMINT-MSK | Gross Target Timer Mask<br>0: Interrupt not generated<br>1: Interrupt generated                                                                                                                                                                                  | 0x0   |
| 5   | R/W  | ERRORINTMSK       | Error Interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated                                                                                                                                                                                     | 0x0   |
| 4   | R/W  | CRYPTINTMSK       | Encryption engine Interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated                                                                                                                                                                         | 0x1   |
| 3   | R/W  | EVENTINTMSK       | End of event Interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated                                                                                                                                                                              | 0x1   |

Table 95: BLE\_INTCNTRL\_REG (0x4000000C)

| Bit | Mode | Symbol      | Description                                                                            | Reset |
|-----|------|-------------|----------------------------------------------------------------------------------------|-------|
| 2   | R/W  | SLPINTMSK   | Sleep Mode Interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated      | 0x1   |
| 1   | R/W  | RXINTMSK    | Rx Interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated              | 0x1   |
| 0   | R/W  | CSCNTINTMSK | 625us Base Time Interrupt Mask<br>0: Interrupt not generated<br>1: Interrupt generated | 0x1   |

Table 96: BLE\_INTSTAT\_REG (0x40000010)

| Bit | Mode | Symbol             | Description                                                                                                                                                                   | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 9   | R    | SWINTSTAT          | SW triggered interrupt status<br>0: No SW triggered interrupt.<br>1: A SW triggered interrupt is pending                                                                      | 0x0   |
| 8   | R    | EVENTAPFAINT-STAT  | End of event / Anticipated Pre-Fetch Abort interrupt status<br>0: No End of Event interrupt.<br>1: An End of Event interrupt is pending.                                      | 0x0   |
| 7   | R    | FINETGTIMINTSTAT   | Masked Fine Target Timer Error interrupt status<br>0: No Fine Target Timer interrupt.<br>1: A Fine Target Timer interrupt is pending.                                         | 0x0   |
| 6   | R    | GROSSTGTIMINT-STAT | Masked Gross Target Timer interrupt status<br>0: No Gross Target Timer interrupt.<br>1: A Gross Target Timer interrupt is pending.                                            | 0x0   |
| 5   | R    | ERRORINTSTAT       | Masked Error interrupt status<br>0: No Error interrupt.<br>1: An Error interrupt is pending.                                                                                  | 0x0   |
| 4   | R    | CRYPTINTSTAT       | Masked Encryption engine interrupt status<br>0: No Encryption / Decryption interrupt.<br>1: An Encryption / Decryption interrupt is pending.                                  | 0x0   |
| 3   | R    | EVENTINTSTAT       | Masked End of Event interrupt status<br>0: No End of Advertising / Scanning / Connection interrupt.<br>1: An End of Advertising / Scanning / Connection interrupt is pending. | 0x0   |
| 2   | R    | SLPINTSTAT         | Masked Sleep interrupt status<br>0: No End of Sleep Mode interrupt.<br>1: An End of Sleep Mode interrupt is pending.                                                          | 0x0   |
| 1   | R    | RXINTSTAT          | Masked Packet Reception interrupt status<br>0: No Rx interrupt.<br>1: An Rx interrupt is pending.                                                                             | 0x0   |
| 0   | R    | CSCNTINTSTAT       | Masked 625us base time reference interrupt status<br>0: No 625us Base Time interrupt.<br>1: A 625us Base Time interrupt is pending.                                           | 0x0   |

Table 97: BLE\_INTRAWSTAT\_REG (0x40000014)

| Bit | Mode | Symbol       | Description                                                                                                   | Reset |
|-----|------|--------------|---------------------------------------------------------------------------------------------------------------|-------|
| 9   | R    | SWINTRAWSTAT | SW triggered interrupt raw status<br>0: No SW triggered interrupt.<br>1: A SW triggered interrupt is pending. | 0x0   |



Table 97: BLE\_INTRAWSTAT\_REG (0x40000014)

| Bit | Mode | Symbol               | Description                                                                                                                                                                | Reset |
|-----|------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 8   | R    | EVENTAPFAINT-RAWSTAT | End of event / Anticipated Pre-Fetch Abort interrupt raw status<br>0: No End of Event interrupt.<br>1: An End of Event interrupt is pending.                               | 0x0   |
| 7   | R    | FINETGTIMINTRA-STAT  | Fine Target Timer Error interrupt raw status<br>0: No Fine Target Timer interrupt.<br>1: A Fine Target Timer interrupt is pending.                                         | 0x0   |
| 6   | R    | GROSSTGTIMINTRA-STAT | Gross Target Timer interrupt raw status<br>0: No Gross Target Timer interrupt.<br>1: A Gross Target Timer interrupt is pending.                                            | 0x0   |
| 5   | R    | ERRORINTRA-STAT      | Error interrupt raw status<br>0: No Error interrupt.<br>1: An Error interrupt is pending.                                                                                  | 0x0   |
| 4   | R    | CRYPTINTRA-STAT      | Encryption engine interrupt raw status<br>0: No Encryption / Decryption interrupt.<br>1: An Encryption / Decryption interrupt is pending.                                  | 0x0   |
| 3   | R    | EVENTINTRA-STAT      | End of Event interrupt raw status<br>0: No End of Advertising / Scanning / Connection interrupt.<br>1: An End of Advertising / Scanning / Connection interrupt is pending. | 0x0   |
| 2   | R    | SLPINTRA-STAT        | Sleep interrupt raw status<br>0: No End of Sleep Mode interrupt.<br>1: An End of Sleep Mode interrupt is pending.                                                          | 0x0   |
| 1   | R    | RXINTRA-STAT         | Packet Reception interrupt raw status<br>0: No Rx interrupt.<br>1: An Rx interrupt is pending.                                                                             | 0x0   |
| 0   | R    | CSCNTINTRA-STAT      | 625us base time reference interrupt raw status<br>0: No 625us Base Time interrupt.<br>1: A 625us Base Time interrupt is pending.                                           | 0x0   |

Table 98: BLE\_INTACK\_REG (0x40000018)

| Bit | Mode | Symbol           | Description                                                                                                                                                                                                                                                                | Reset |
|-----|------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 9   | R0/W | SWINTACK         | SW triggered interrupt acknowledgement bit<br>Software writing 1 acknowledges the SW triggered interrupt. This bit resets SWINTSTAT and SWINTRA-STAT flags. Resets at 0 when action is performed                                                                           | 0x0   |
| 8   | R0/W | EVENTAPFAINTACK  | End of event / Anticipated Pre-Fetch Abort interrupt acknowledgement bit<br>Software writing 1 acknowledges the End of event / Anticipated Pre-Fetch Abort interrupt. This bit resets EVENTAPFAINTSTAT and EVENTAPFAINTRAWSTAT flags. Resets at 0 when action is performed | 0x0   |
| 7   | R0/W | FINETGTIMINTACK  | Fine Target Timer interrupt acknowledgement bit<br>Software writing 1 acknowledges the Fine Timer interrupt. This bit resets FINETGTIMINTSTAT and FINETGTIMINTRA-STAT flags. Resets at 0 when action is performed                                                          | 0x0   |
| 6   | R0/W | GROSSTGTIMINTACK | Gross Target Timer interrupt acknowledgement bit<br>Software writing 1 acknowledges the Gross Timer interrupt. This bit resets GROSSTGTIMINTSTAT and GROSSTGTIMINTRA-STAT flags. Resets at 0 when action is performed                                                      | 0x0   |

Table 98: BLE\_INTACK\_REG (0x40000018)

| Bit | Mode | Symbol      | Description                                                                                                                                                                                                                        | Reset |
|-----|------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R0/W | ERRORINTACK | Error interrupt acknowledgement bit<br>Software writing 1 acknowledges the Error interrupt. This bit resets ERRORINTSTAT and ERRORINTRAWSTAT flags.<br>Resets at 0 when action is performed                                        | 0x0   |
| 4   | R0/W | CRYPTINTACK | Encryption engine interrupt acknowledgement bit<br>Software writing 1 acknowledges the Encryption engine interrupt. This bit resets CRYPTINTSTAT and CRYPTINTRAWSTAT flags.<br>Resets at 0 when action is performed                | 0x0   |
| 3   | R0/W | EVENTINTACK | End of Event interrupt acknowledgment bit<br>Software writing 1 acknowledges the End of Advertising / Scanning / Connection interrupt. This bit resets SLPINTSTAT and SLPINTRAWSTAT flags.<br>Resets at 0 when action is performed | 0x0   |
| 2   | R0/W | SLPINTACK   | End of Deep Sleep interrupt acknowledgment bit<br>Software writing 1 acknowledges the End of Sleep Mode interrupt. This bit resets SLPINTSTAT and SLPINTRAWSTAT flags.<br>Resets at 0 when action is performed                     | 0x0   |
| 1   | R0/W | RXINTACK    | Packet Reception interrupt acknowledgment bit<br>Software writing 1 acknowledges the Rx interrupt. This bit resets RXINTSTAT and RXINTRAWSTAT flags.<br>Resets at 0 when action is performed                                       | 0x0   |
| 0   | R0/W | CSCNTINTACK | 625us base time reference interrupt acknowledgment bit<br>Software writing 1 acknowledges the CLKN interrupt. This bit resets CLKINTSTAT and CLKINTRAWSTAT flags.<br>Resets at 0 when action is performed                          | 0x0   |

Table 99: BLE\_BASETIMECNT\_REG (0x4000001C)

| Bit  | Mode | Symbol      | Description                                                                                                                              | Reset |
|------|------|-------------|------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 26:0 | R    | BASETIMECNT | Value of the 625us base time reference counter. Updated each time SAMPCLK is written. Used by the SW in order to synchronize with the HW | 0x0   |

Table 100: BLE\_FINETIMECNT\_REG (0x40000020)

| Bit | Mode | Symbol  | Description                                                                                                                                                                            | Reset |
|-----|------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 9:0 | R    | FINECNT | Value of the current s fine time reference counter. Updated each time SAMPCLK is written. Used by the SW in order to synchronize with the HW, and obtain a more precise sleep duration | 0x0   |

Table 101: BLE\_BDADDRL\_REG (0x40000024)

| Bit  | Mode | Symbol  | Description                                    | Reset |
|------|------|---------|------------------------------------------------|-------|
| 31:0 | R/W  | BDADDRL | Bluetooth Low Energy Device Address. LSB part. | 0x0   |

Table 102: BLE\_BDADDRU\_REG (0x40000028)

| Bit | Mode | Symbol    | Description                                                                                                                        | Reset |
|-----|------|-----------|------------------------------------------------------------------------------------------------------------------------------------|-------|
| 16  | R/W  | PRIV_NPUB | Bluetooth Low Energy Device Address privacy indicator<br>0: Public Bluetooth Device Address<br>1: Private Bluetooth Device Address | 0x0   |

**Table 102: BLE\_BDADDRU\_REG (0x40000028)**

| Bit  | Mode | Symbol  | Description                                    | Reset |
|------|------|---------|------------------------------------------------|-------|
| 15:0 | R/W  | BDADDRU | Bluetooth Low Energy Device Address. MSB part. | 0x0   |

**Table 103: BLE\_CURRENTRXDESCPTR\_REG (0x4000002C)**

| Bit   | Mode | Symbol            | Description                                                                                 | Reset |
|-------|------|-------------------|---------------------------------------------------------------------------------------------|-------|
| 31:16 | R/W  | ETPTR             | Exchange Table Pointer that determines the starting point of the Exchange Table             | 0x0   |
| 14:0  | R/W  | CURRENTRX-DESCPTR | Rx Descriptor Pointer that determines the starting point of the Receive Buffer Chained List | 0x0   |

**Table 104: BLE\_DEEPSLCNTL\_REG (0x40000030)**

| Bit | Mode | Symbol             | Description                                                                                                                                                                                                                                                                         | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31  | R/W  | EXTWKUPDSB         | External Wake-Up disable<br>0: RW-BLE Core can be woken by external wake-up<br>1: RW-BLE Core cannot be woken up by external wake-up                                                                                                                                                | 0x0   |
| 15  | R    | DEEP_SLEEP_STAT    | Indicator of current Deep Sleep clock mux status:<br>0: RW-BLE Core is not yet in Deep Sleep Mode<br>1: RW-BLE Core is in Deep Sleep Mode (only low_power_clk is running)                                                                                                           | 0x0   |
| 4   | R0/W | SOFT_WAKEUP_REQ    | Wake Up Request from RW-BLE Software. Applies when system is in Deep Sleep Mode. It wakes up the RW-BLE Core when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.                                                                | 0x0   |
| 3   | R0/W | DEEP_SLEEP_CORR_EN | 625us base time reference integer and fractional part correction. Applies when system has been woken-up from Deep Sleep Mode. It enables Fine Counter and Base Time counter when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0. | 0x0   |
| 2   | R0/W | DEEP_SLEEP_ON      | 0: RW-BLE Core in normal active mode<br>1: Request RW-BLE Core to switch in deep sleep mode. This bit is reset on DEEP_SLEEP_STAT falling edge.                                                                                                                                     | 0x0   |
| 1:0 | R/W  | DEEP_SLEEP_IRQ_EN  | Always set to "3" when DEEP_SLEEP_ON is set to "1". It controls the generation of BLE_WAKEUP_LP_IRQ.                                                                                                                                                                                | 0x0   |

**Table 105: BLE\_DEEPSLWKUP\_REG (0x40000034)**

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                                                               | Reset |
|------|------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | DEEPSLTIME | Determines the time in low_power_clk clock cycles to spend in Deep Sleep Mode before waking-up the device. This ensures a maximum of 37 hours and 16mn sleep mode capabilities at 32kHz. This ensures a maximum of 36 hours and 16mn sleep mode capabilities at 32.768kHz | 0x0   |

**Table 106: BLE\_DEEPSLSTAT\_REG (0x40000038)**

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                        | Reset |
|------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R    | DEEPSLDUR | Actual duration of the last deep sleep phase measured in low_power_clk clock cycle. DEEPSLDUR is set to zero at the beginning of the deep sleep phase, and is incremented at each low_power_clk clock cycle until the end of the deep sleep phase. | 0x0   |

Table 107: BLE\_ENBPRESET\_REG (0x4000003C)

| Bit   | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                           | Reset |
|-------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:21 | R/W  | TWEXT       | Minimum and recommended value is "TWIRQ_RESET + 1". In the case of wake-up due to an external wake-up request, TWEXT specifies the time delay in low power oscillator cycles to deassert BLE_WAKEUP_LP_IRQ. Refer also to GP_CONTROL_REG[BLE_WAKEUP_REQ]. Range is [0...64 ms] for 32kHz; [0...62.5 ms] for 32.768kHz | 0x0   |
| 20:10 | R/W  | TWIRQ_SET   | Minimum value is "TWIRQ_RESET + 1". Time in low power oscillator cycles to set BLE_WAKEUP_LP_IRQ before the BLE sleep timer expiration. Refer also to BLE_DEEPSLWKUP_REG[DEEPSLTIME]. Range is [0...64 ms] for 32kHz; [0...62.5 ms] for 32.768kHz                                                                     | 0x0   |
| 9:0   | R/W  | TWIRQ_RESET | Recommended value is 1. Time in low power oscillator cycles to reset BLE_WAKEUP_LP_IRQ before the BLE sleep timer expiration. Refer also to BLE_DEEPSLWKUP_REG[DEEPSLTIME]. Range is [0...32 ms] for 32kHz; [0...31.25 ms] for 32.768kHz.                                                                             | 0x0   |

Table 108: BLE\_FINECNTCORR\_REG (0x40000040)

| Bit | Mode | Symbol      | Description                                                                       | Reset |
|-----|------|-------------|-----------------------------------------------------------------------------------|-------|
| 9:0 | R/W  | FINECNTCORR | Phase correction value for the 625us reference counter (i.e. Fine Counter) in us. | 0x0   |

Table 109: BLE\_BASETIMECNTCORR\_REG (0x40000044)

| Bit  | Mode | Symbol           | Description                         | Reset |
|------|------|------------------|-------------------------------------|-------|
| 26:0 | R/W  | BASETIMECNT-CORR | Base Time Counter correction value. | 0x0   |

Table 110: BLE\_DIAGCNTL\_REG (0x40000050)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 111: BLE\_DIAGSTAT\_REG (0x40000054)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 112: BLE\_DEBUGADDMAX\_REG (0x40000058)

| Bit   | Mode | Symbol     | Description                                                              | Reset |
|-------|------|------------|--------------------------------------------------------------------------|-------|
| 31:16 | R/W  | REG_ADDMAX | Upper limit for the Register zone indicated by the reg_inzone flag       | 0x0   |
| 15:0  | R/W  | EM_ADDMAX  | Upper limit for the Exchange Memory zone indicated by the em_inzone flag | 0x0   |

Table 113: BLE\_DEBUGADMIN\_REG (0x4000005C)

| Bit   | Mode | Symbol     | Description                                                              | Reset |
|-------|------|------------|--------------------------------------------------------------------------|-------|
| 31:16 | R/W  | REG_ADDMIN | Lower limit for the Register zone indicated by the reg_inzone flag       | 0x0   |
| 15:0  | R/W  | EM_ADDMIN  | Lower limit for the Exchange Memory zone indicated by the em_inzone flag | 0x0   |

Table 114: BLE\_ERRORTYPESTAT\_REG (0x40000060)

| Bit | Mode | Symbol             | Description                                                                                                                                                                                                                                                       | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 17  | R/W  | CONCEVTIRQ_ERR OR  | Indicates whether two consecutive and concurrent ble_event_irq have been generated, and not acknowledged in time by the RW-BLE Software.<br>0: No error<br>1: Error occurred                                                                                      | 0x0   |
| 16  | R    | RXDATA_PTR_ERR OR  | Indicates whether Rx data buffer pointer value programmed is null: this is a major programming failure.<br>0: No error<br>1: Error occurred                                                                                                                       | 0x0   |
| 15  | R    | TXDATA_PTR_ERR OR  | Indicates whether Tx data buffer pointer value programmed is null during Advertising / Scanning / Initiating events, or during Master / Slave connections with non-null packet length: this is a major programming failure.<br>0: No error<br>1: Error occurred   | 0x0   |
| 14  | R    | RXDESC_EMPTY_ERROR | Indicates whether Rx Descriptor pointer value programmed in register is null: this is a major programming failure.<br>0: No error<br>1: Error occurred                                                                                                            | 0x0   |
| 13  | R    | TXDESC_EMPTY_ERROR | Indicates whether Tx Descriptor pointer value programmed in Control Structure is null during Advertising / Scanning / Initiating events: this is a major programming failure.<br>0: No error<br>1: Error occurred                                                 | 0x0   |
| 12  | R    | CSFORMAT_ERROR     | Indicates whether CS-FORMAT has been programmed with an invalid value: this is a major software programming failure.<br>0: No error<br>1: Error occurred                                                                                                          | 0x0   |
| 11  | R    | LLCHMAP_ERROR      | Indicates Link Layer Channel Map error, happens when actual number of CS-LLCHMAP bit set to one is different from CS-NBCHGOOD at the beginning of Frequency Hopping process<br>0: No error<br>1: Error occurred                                                   | 0x0   |
| 10  | R    | ADV_UNDERRUN       | Indicates Advertising Interval Under run, occurs if time between two consecutive Advertising packet (in Advertising mode) is lower than the expected value.<br>0: No error<br>1: Error occurred                                                                   | 0x0   |
| 9   | R    | IFS_UNDERRUN       | Indicates Inter Frame Space Under run, occurs if IFS time is not enough to update and read Control Structure/Descriptors, and/or White List parsing is not finished and/or Decryption time is too long to be finished on time<br>0: No error<br>1: Error occurred | 0x0   |

Table 114: BLE\_ERRORTYPESTAT\_REG (0x40000060)

| Bit | Mode | Symbol                 | Description                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 8   | R    | WHITELIST_ERROR        | Indicates White List Timeout error, occurs if White List parsing is not finished on time<br>0: No error<br>1: Error occurred                                                                                                                                                                                    | 0x0   |
| 7   | R    | EVT_CNTL_APFM_ERROR    | Indicates Anticipated Pre-Fetch Mechanism error: happens when 2 consecutive events are programmed, and when the first event is not completely finished while second pre-fetch instant is reached.<br>0: No error<br>1: Error occurred                                                                           | 0x0   |
| 6   | R    | EVT_SCHDL_APFM_ERROR   | Indicates Anticipated Pre-Fetch Mechanism error: happens when 2 consecutive events are programmed, and when the first event is not completely finished while second pre-fetch instant is reached.<br>0: No error<br>1: Error occurred                                                                           | 0x0   |
| 5   | R    | EVT_SCHDL_ENTR_Y_ERROR | Indicates Event Scheduler faced Invalid timing programming on two consecutive ET entries (e.g first one with 624s offset and second one with no offset)<br>0: No error<br>1: Error occurred                                                                                                                     | 0x0   |
| 4   | R/W  | EVT_SCHDL_EMACC_ERROR  | Indicates Event Scheduler Exchange Memory access error, happens when Exchange Memory accesses are not served in time, and blocks the Exchange Table entry read<br>0: No error<br>1: Error occurred                                                                                                              | 0x0   |
| 3   | R    | RADIO_EMACC_ERROR      | Indicates Radio Controller Exchange Memory access error, happens when Exchange Memory accesses are not served in time and data are corrupted.<br>0: No error<br>1: Error occurred                                                                                                                               | 0x0   |
| 2   | R    | PKTCNTL_EMACC_ERROR    | Indicates Packet Controller Exchange Memory access error, happens when Exchange Memory accesses are not served in time and Tx/Rx data are corrupted<br>0: No error<br>1: Error occurred                                                                                                                         | 0x0   |
| 1   | R    | RXCRIPT_ERROR          | Indicates real time decryption error, happens when AES-CCM decryption is too slow compared to Packet Controller requests. A 16-bytes block has to be decrypted prior the next block is received by the Packet Controller<br>0: No error<br>1: Error occurred                                                    | 0x0   |
| 0   | R    | TXCRYPT_ERROR          | Indicates Real Time encryption error, happens when AES-CCM encryption is too slow compared to Packet Controller requests. A 16-bytes block has to be encrypted and prepared on Packet Controller request, and needs to be ready before the Packet Controller has to send it<br>0: No error<br>1: Error occurred | 0x0   |

Table 115: BLE\_SWPROFILING\_REG (0x40000064)

| Bit  | Mode | Symbol    | Description                                                                                                         | Reset |
|------|------|-----------|---------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | SWPROFVAL | Software Profiling register: used by RW-BLE Software for profiling purpose: this value is copied on Diagnostic port | 0x0   |

Table 116: BLE\_RADIOCNTRL0\_REG (0x40000070)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 31:0 | -    | -      | Reserved    | 0x0   |

Table 117: BLE\_RADIOCNTRL1\_REG (0x40000074)

| Bit   | Mode | Symbol | Description                                         | Reset |
|-------|------|--------|-----------------------------------------------------|-------|
| 20:16 | R/W  | XRFSEL | Extended radio selection field, Must be set to "2". | 0x0   |

Table 118: BLE\_RADIOCNTRL2\_REG (0x40000078)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 31:0 | -    | -      | Reserved    | 0x0   |

Table 119: BLE\_RADIOCNTRL3\_REG (0x4000007C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 31:0 | -    | -      | Reserved    | 0x0   |

Table 120: BLE\_RADIOPWUPDN\_REG (0x40000080)

| Bit   | Mode | Symbol      | Description                                                                                                                                                                       | Reset |
|-------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 30:24 | R/W  | RTRIP_DELAY | Defines round trip delay value. This value correspond to the addition of data latency in Tx and data latency in Rx. Value is in us                                                | 0x0   |
| 23:16 | R/W  | RXPWRUP     | This register holds the length in s of the RX power up phase for the current radio device. Default value is 210us (reset value). Operating range depends on the selected radio.   | 0xD2  |
| 11:8  | R/W  | TXPWDRN     | This register extends the length in s of the TX power down phase for the current radio device. Default value is 3us (reset value). Operating range depends on the selected radio. | 0x3   |
| 7:0   | R/W  | TXPWUP      | This register holds the length in s of the TX power up phase for the current radio device. Default value is 210us (reset value). Operating range depends on the selected radio.   | 0xD2  |

Table 121: BLE\_ADVCHMAP\_REG (0x40000090)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                        | Reset |
|-----|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2:0 | R/W  | ADVCHMAP | Advertising Channel Map, defined as per the advertising connection settings. Contains advertising channels index 37 to 39. If ADVCHMAP[i] equals:<br>0: Do not use data channel i+37.<br>1: Use data channel i+37. | 0x7   |

Table 122: BLE\_ADVTIM\_REG (0x400000A0)

| Bit  | Mode | Symbol | Description                                                                                                                                                                                             | Reset |
|------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:0 | R/W  | ADVINT | Advertising Packet Interval defines the time interval in between two ADV_XXX packet sent. Value is in us. Value to program depends on the used Advertising Packet type and the device filtering policy. | 0x0   |



Table 123: BLE\_ACTSCANSTAT\_REG (0x400000A4)

| Bit   | Mode | Symbol     | Description                                             | Reset |
|-------|------|------------|---------------------------------------------------------|-------|
| 24:16 | R    | BACKOFF    | Active scan mode back-off counter initialization value. | 0x1   |
| 8:0   | R    | UPPERLIMIT | Active scan mode upper limit counter value.             | 0x1   |

Table 124: BLE\_WLPUBADDPTR\_REG (0x400000B0)

| Bit  | Mode | Symbol      | Description                                             | Reset |
|------|------|-------------|---------------------------------------------------------|-------|
| 15:0 | R/W  | WLPUBADDPTR | Start address pointer of the public devices white list. | 0x0   |

Table 125: BLE\_WLPRIVADDPTR\_REG (0x400000B4)

| Bit  | Mode | Symbol       | Description                                              | Reset |
|------|------|--------------|----------------------------------------------------------|-------|
| 15:0 | R/W  | WLPRIVADDPTR | Start address pointer of the private devices white list. | 0x0   |

Table 126: BLE\_WLNBDEV\_REG (0x400000B8)

| Bit  | Mode | Symbol    | Description                                  | Reset |
|------|------|-----------|----------------------------------------------|-------|
| 15:8 | R/W  | NBPRIVDEV | Number of private devices in the white list. | 0x0   |
| 7:0  | R/W  | NBPUBDEV  | Number of public devices in the white list.  | 0x0   |

Table 127: BLE\_AESCNTL\_REG (0x400000C0)

| Bit | Mode | Symbol    | Description                                                                                                                                                 | Reset |
|-----|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | AES_MODE  | 0: Cipher mode<br>1: Decipher mode                                                                                                                          | 0x0   |
| 0   | R0/W | AES_START | Writing a 1 starts AES-128 ciphering/deciphering process. This bit is reset once the process is finished (i.e. ble_crypt_irq interrupt occurs, even masked) | 0x0   |

Table 128: BLE\_AESKEY31\_0\_REG (0x400000C4)

| Bit  | Mode | Symbol     | Description                                  | Reset |
|------|------|------------|----------------------------------------------|-------|
| 31:0 | R/W  | AESKEY31_0 | AES encryption 128-bit key. Bit 31 down to 0 | 0x0   |

Table 129: BLE\_AESKEY63\_32\_REG (0x400000C8)

| Bit  | Mode | Symbol      | Description                                   | Reset |
|------|------|-------------|-----------------------------------------------|-------|
| 31:0 | R/W  | AESKEY63_32 | AES encryption 128-bit key. Bit 63 down to 32 | 0x0   |

Table 130: BLE\_AESKEY95\_64\_REG (0x400000CC)

| Bit  | Mode | Symbol      | Description                                   | Reset |
|------|------|-------------|-----------------------------------------------|-------|
| 31:0 | R/W  | AESKEY95_64 | AES encryption 128-bit key. Bit 95 down to 64 | 0x0   |

Table 131: BLE\_AESKEY127\_96\_REG (0x400000D0)

| Bit  | Mode | Symbol       | Description                                    | Reset |
|------|------|--------------|------------------------------------------------|-------|
| 31:0 | R/W  | AESKEY127_96 | AES encryption 128-bit key. Bit 127 down to 96 | 0x0   |

Table 132: BLE\_AESPTR\_REG (0x400000D4)

| Bit  | Mode | Symbol | Description                                                                            | Reset |
|------|------|--------|----------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | AESPTR | Pointer to the memory zone where the block to cipher/decipher using AES-128 is stored. | 0x0   |

Table 133: BLE\_TXMICVAL\_REG (0x400000D8)

| Bit  | Mode | Symbol   | Description                                                            | Reset |
|------|------|----------|------------------------------------------------------------------------|-------|
| 31:0 | R    | TXMICVAL | AES-CCM plain MIC value. Valid on when MIC has been calculated (in Tx) | 0x0   |

Table 134: BLE\_RXMICVAL\_REG (0x400000DC)

| Bit  | Mode | Symbol   | Description                                                                   | Reset |
|------|------|----------|-------------------------------------------------------------------------------|-------|
| 31:0 | R    | RXMICVAL | AES-CCM plain MIC value. Valid on once MIC has been extracted from Rx packet. | 0x0   |

Table 135: BLE\_RFTESTCNTL\_REG (0x400000E0)

| Bit | Mode | Symbol      | Description                                                                                                                                                                                                   | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31  | R/W  | INFINITERX  | Applicable in RF Test Mode only<br>0: Normal mode of operation<br>1: Infinite Rx window                                                                                                                       | 0x0   |
| 27  | R/W  | RXPKTCNTEN  | Applicable in RF Test Mode only<br>0: Rx packet count disabled<br>1: Rx packet count enabled, and reported in CS-RXCCMPKTCNT and RFTESTRXSTAT-RXPKTCNT on RF abort command                                    | 0x0   |
| 15  | R/W  | INFINITETX  | Applicable in RF Test Mode only<br>0: Normal mode of operation.<br>1: Infinite Tx packet / Normal start of a packet but endless payload                                                                       | 0x0   |
| 14  | R/W  | TXLENGTHSRC | Applicable only in Tx/Rx RF Test mode<br>0: Normal mode of operation: TxDESC-TXADVLEN controls the Tx packet payload size<br>1: Uses RFTESTCNTL-TXLENGTH packet length (can support up to 512 bytes transmit) | 0x0   |
| 13  | R/W  | PRBSTYPE    | Applicable only in Tx/Rx RF Test mode<br>0: Tx Packet Payload are PRBS9 type<br>1: Tx Packet Payload are PRBS15 type                                                                                          | 0x0   |
| 12  | R/W  | TXPLDSRC    | Applicable only in Tx/Rx RF Test mode<br>0: Tx Packet Payload source is the Control Structure<br>1: Tx Packet Payload are PRBS generator                                                                      | 0x0   |
| 11  | R/W  | TXPKTCNTEN  | Applicable in RF Test Mode only<br>0: Tx packet count disabled<br>1: Tx packet count enabled, and reported in CS-TXCCMPKTCNT and RFTESTTXSTAT-TXPKTCNT on RF abort command                                    | 0x0   |
| 8:0 | R/W  | TXLENGTH    | Applicable only for Tx/Rx RF Test mode, and valid when RFTESTCNTL-TXLENGTHSRC = 1<br>Tx packet length in number of byte                                                                                       | 0x0   |

Table 136: BLE\_RFTESTTXSTAT\_REG (0x400000E4)

| Bit  | Mode | Symbol   | Description                                                                                            | Reset |
|------|------|----------|--------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | TXPKTCNT | Reports number of transmitted packet during Test Modes. Value is valid if RFTESTCNTL-TXPKTCNTEN is set | 0x0   |

Table 137: BLE\_RFTESTRXSTAT\_REG (0x400000E8)

| Bit  | Mode | Symbol    | Description                                                                                                                                  | Reset |
|------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | RXPCKTCNT | Reports number of correctly received packet during Test Modes (no sync error, no CRC error). Value is valid if RFTESTCNTL-RXPCKTCNTEN is set | 0x0   |

Table 138: BLE\_TIMGENCNTL\_REG (0x400000F0)

| Bit   | Mode | Symbol             | Description                                                                                     | Reset |
|-------|------|--------------------|-------------------------------------------------------------------------------------------------|-------|
| 31    | R/W  | APFM_EN            | Controls the Anticipated pre-Fetch Abort mechanism<br>0: Disabled<br>1: Enabled                 | 0x1   |
| 25:16 | R/W  | PREFETCHABORT_TIME | Defines the instant in s at which immediate abort is required after anticipated pre-fetch abort | 0x1FE |
| 8:0   | R/W  | PREFETCH_TIME      | Defines Exchange Table pre-fetch instant in us                                                  | 0x96  |

Table 139: BLE\_GROSSTIMTGT\_REG (0x400000F4)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                    | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 22:0 | R/W  | GROSSTARGET | Gross Timer Target value on which a ble_grosstgtim_irq must be generated. This timer has a precision of 10ms: interrupt is generated only when GROSSTARGET[22:0] = BASETIMECNT[26:4] and BASETIMECNT[3:0] = 0. | 0x0   |

Table 140: BLE\_FINETIMTGT\_REG (0x400000F8)

| Bit  | Mode | Symbol     | Description                                                                                                                                                            | Reset |
|------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 26:0 | R/W  | FINETARGET | Fine Timer Target value on which a ble_finetgtim_irq must be generated. This timer has a precision of 625us: interrupt is generated only when FINETARGET = BASETIMECNT | 0x0   |

Table 141: BLE\_SAMPLECLK\_REG (0x400000FC)

| Bit | Mode | Symbol | Description                                                                                                    | Reset |
|-----|------|--------|----------------------------------------------------------------------------------------------------------------|-------|
| 0   | R0/W | SAMP   | Writing a 1 samples the Base Time Counter value in BASETIMECNT register. Resets at 0 when action is performed. | 0x0   |

Table 142: BLE\_COEXIFCNTL0\_REG (0x40000100)

| Bit   | Mode | Symbol        | Description                                                                                                                                                                                                                                  | Reset |
|-------|------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 21:20 | R/W  | WLCRXPRIOMODE | Defines Bluetooth Low Energy packet ble_rx mode behavior.<br>00: Rx indication excluding Rx Power up delay (starts when correlator is enabled)<br>01: Rx indication including Rx Power up delay<br>10: Rx High priority indicator<br>11: n/a | 0x0   |

Table 142: BLE\_COEXIFCNTL0\_REG (0x40000100)

| Bit   | Mode | Symbol        | Description                                                                                                                                                                                                                                    | Reset |
|-------|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 17:16 | R/W  | WLCTXPRIOMODE | Defines Bluetooth Low Energy packet ble_tx mode behavior<br>00: Tx indication excluding Tx Power up delay<br>01: Tx indication including Tx Power up delay<br>10: Tx High priority indicator<br>11: n/a                                        | 0x0   |
| 7:6   | R/W  | WLANTXMSK     | Determines how wlan_tx impact BLE Tx and Rx<br>00: wlan_tx has no impact (default mode)<br>01: wlan_tx can stop BLE Tx, no impact on BLE Rx<br>10: wlan_tx can stop BLE Rx, no impact on BLE Tx<br>11: wlan_tx can stop both BLE Tx and BLE Rx | 0x0   |
| 5:4   | R/W  | WLANRXMSK     | Determines how wlan_rx impact BLE Tx and Rx<br>00: wlan_rx has no impact<br>01: wlan_rx can stop BLE Tx, no impact on BLE Rx (default mode)<br>10: wlan_rx can stop BLE Rx, no impact on BLE Tx<br>11: wlan_rx can stop both BLE Tx and BLE Rx | 0x0   |
| 1     | R/W  | SYNCGEN_EN    | Determines whether ble_sync is generated or not.<br>0: ble_sync pulse not generated<br>1: ble_sync pulse generated                                                                                                                             | 0x0   |
| 0     | R/W  | COEX_EN       | Enable / Disable control of the MWS/WLAN Coexistence control<br>0: Coexistence interface disabled<br>1: Coexistence interface enabled                                                                                                          | 0x0   |

Table 143: BLE\_COEXIFCNTL1\_REG (0x40000104)

| Bit   | Mode | Symbol       | Description                                                                                                                                                                                                                                                                      | Reset |
|-------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 28:24 | R/W  | WLCPRXTHR    | Applies on ble_rx if WLCRXPRIOMODE equals 10<br>Determines the threshold for Rx priority setting.<br>If ble_pti[3:0] output value is greater than WLCPRXTHR, then Rx Bluetooth Low Energy priority is considered as high, and must be provided to the WLAN coexistence interface | 0x0   |
| 20:16 | R/W  | WLCPTXTHR    | Applies on ble_tx if WLCTXPRIOMODE equals 10<br>Determines the threshold for priority setting.<br>If ble_pti[3:0] output value is greater than WLCPTXTHR, then Tx Bluetooth Low Energy priority is considered as high, and must be provided to the WLAN coexistence interface    | 0x0   |
| 14:8  | R/W  | WLCPDURATION | Applies on ble_tx if WLCTXPRIOMODE equals 10<br>Applies on ble_rx if WLCRXPRIOMODE equals 10<br>Determines how many s the priority information must be maintained<br>Note that if WLCPDURATION = 0x00, then Tx/Rx priority levels are maintained till Tx/Rx EN are de-asserted.  | 0x0   |
| 6:0   | R/W  | WLCPDELAY    | Applies on ble_tx if WLCTXPRIOMODE equals 10.<br>Applies on ble_rx if WLCRXPRIOMODE equals 10.<br>Determines the delay (in us) in Tx/Rx enables rises the time Bluetooth Low energy Tx/Rx priority has to be provided .                                                          | 0x0   |

Table 144: BLE\_BLEMPRIO0\_REG (0x40000108)

| Bit   | Mode | Symbol | Description                                                | Reset |
|-------|------|--------|------------------------------------------------------------|-------|
| 31:28 | R/W  | BLEM7  | Set Priority value for Passive Scanning                    | 0x3   |
| 27:24 | R/W  | BLEM6  | Set Priority value for Non-Connectable Advertising         | 0x4   |
| 23:20 | R/W  | BLEM5  | Set Priority value for Connectable Advertising BLE message | 0x4   |

Table 144: BLE\_BLEMPRIO0\_REG (0x40000108)

| Bit   | Mode | Symbol | Description                                                                 | Reset |
|-------|------|--------|-----------------------------------------------------------------------------|-------|
| 19:16 | R/W  | BLEM4  | Set Priority value for Active Scanning BLE message                          | 0x9   |
| 15:12 | R/W  | BLEM3  | Set Priority value for Initiating (Scanning) BLE message                    | 0xA   |
| 11:8  | R/W  | BLEM2  | Set Priority value for Data Channel transmission BLE message                | 0xD   |
| 7:4   | R/W  | BLEM1  | Set Priority value for LLCP BLE message                                     | 0xE   |
| 3:0   | R/W  | BLEM0  | Set Priority value for Initiating (Connection Request Response) BLE message | 0xF   |

Table 145: BLE\_BLEMPRIO1\_REG (0x4000010C)

| Bit   | Mode | Symbol      | Description                                                               | Reset |
|-------|------|-------------|---------------------------------------------------------------------------|-------|
| 31:28 | R/W  | BLEMDEFAULT | Set default priority value for other BLE message than those defined above | 0x3   |

Table 146: BLE\_BLEPRIOSCHARB\_REG (0x40000110)

| Bit | Mode | Symbol      | Description                                                                                                            | Reset |
|-----|------|-------------|------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R/W  | BLEPRIOMODE | Determine BLE Priority Scheduling Arbitration Mode<br>0: BLE Decision instant not used<br>1: BLE Decision instant used | 0x0   |
| 7:0 | R/W  | BLEMARGIN   | Determine the decision instant margin for Priority Scheduling Arbitration.                                             | 0x0   |

Table 147: BLE\_CNTL2\_REG (0x40000200)

| Bit   | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|-------|------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:22 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 21    | R/W  | BLE_RSSI_SEL | 0: Select Peak-hold RSSI value (default).<br>1: Select current Average RSSI value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 20    | R/W  | WAKEUPLPSTAT | The status of the BLE_WAKEUP_LP_IRQ. The Interrupt Service Routine of BLE_WAKEUP_LP_IRQ should return only when the WAKEUPLPSTAT is cleared.<br>Note that BLE_WAKEUP_LP_IRQ is automatically acknowledged after the power up of the Radio Subsystem, plus one Low Power Clock period.                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 19    | R/W  | SW_RPL_SPI   | Keep to 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 18:15 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 14:9  | R/W  | BLE_CLK_SEL  | BLE Clock Select.<br>Specifies the BLE master clock absolute frequency in MHz. Typical values are 16 and 8.<br>Value depends on the selected XTAL frequency and the value of CLK_RADIO_REG[BLE_DIV] bitfield. For example, if XTAL oscillates at 16MHz and CLK_RADIO_REG[BLE_DIV] = 1 (divide by 2), then BLE master clock frequency is 8MHz and BLE_CLK_SEL should be set to value 8.<br>The selected BLE master clock frequency (affected by BLE_DIV and BLE_CLK_SEL) must be modified and set only during the initialization time, i.e. before setting BLE_RWBTLCTL_REG[RWBLE_EN] to 1.<br>Refer also to BLE_RWBTLCONF_REG[CLK_SEL]. | 0x0   |

Table 147: BLE\_CNTL2\_REG (0x40000200)

| Bit | Mode | Symbol            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 8   | R/W  | RADIO_PWRDN_ALLOW | This active high signal indicates when it is allowed for the BLE core (embedded in the Radio sub-System power domain) to be powered down.<br>After the assertion of the BLE_DEEPSLCTL2_REG[DEEP_SLEEP_ON] a hardware sequence based on the Low Power clock will cause the assertion of RADIO_PWRDN_ALLOW. The RADIO_PWRDN_ALLOW will be cleared to "0" when the BLE core exits from the sleep state, i.e. when the BLE_SLP_IRQ will be asserted. | 0x0   |
| 7   | R/W  | MON_LP_CLK        | The SW can only write a "0" to this bit.<br>Whenever a positive edge of the low power clock used by the BLE Timers is detected, then the HW will automatically set this bit to "1". This functionality will not work if BLE Timer is in reset state (refer to CLK_RADIO_REG[BLE_LP_RESET]).<br>This bit can be used for SW synchronization, to debug the low power clock, etc.                                                                   | 0x0   |
| 6   | R/W  | BLE_CLK_STAT      | 0: BLE uses low power clock<br>1: BLE uses master clock                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 5:3 | -    | -                 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 2   | R/W  | EMACCERRMSK       | Exchange Memory Access Error Mask:<br>When cleared to "0" the EM_ACC_ERR will not cause an BLE_ERROR_IRQ interrupt.<br>When set to "1" an BLE_ERROR_IRQ will be generated as long as EM_ACC_ERR is "1".                                                                                                                                                                                                                                          | 0x0   |
| 1   | R/W  | EMACCERRACK       | Exchange Memory Access Error Acknowledge.<br>When the SW writes a "1" to this bit then the EMACCERRSTAT bit will be cleared.<br>When the SW writes "0" it will have no affect.<br>The read value is always "0".                                                                                                                                                                                                                                  | 0x0   |
| 0   | R/W  | EMACCERRSTAT      | Exchange Memory Access Error Status:<br>The bit is read-only and can be cleared only by writing a "1" at EMACCERRACK bitfield.<br>This bit will be set to "1" by the hardware when the controller will access an EM page that is not mapped according to the EM_MAPPING value.<br>When this bit is "1" then the BLE_ERROR_IRQ will be asserted as long as EMACCERRMSK is "1".                                                                    | 0x0   |

Table 148: BLE\_EM\_BASE\_REG (0x40000208)

| Bit   | Mode | Symbol            | Description                                                                       | Reset |
|-------|------|-------------------|-----------------------------------------------------------------------------------|-------|
| 31:17 | -    | -                 | Reserved                                                                          | 0x0   |
| 16:10 | R/W  | BLE_EM_BASE_16_10 | The physical address on the system memory map of the base of the Exchange Memory. | 0x0   |
| 9:0   | -    | -                 | Reserved                                                                          | 0x0   |

Table 149: BLE\_DIAGNTL2\_REG (0x4000020C)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 150: BLE\_DIAGCNTL3\_REG (0x40000210)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |



## 37.4 AES\_HASH REGISTER FILE

Table 151: Register map AES\_HASH

| Address    | Port                  | Description                                          |
|------------|-----------------------|------------------------------------------------------|
| 0x40020000 | CRYPTO_CTRL_REG       | Crypto Control register                              |
| 0x40020004 | CRYPTO_START_REG      | Crypto Start calculation                             |
| 0x40020008 | CRYPTO_FETCH_ADDR_REG | Crypto DMA fetch register                            |
| 0x4002000C | CRYPTO_LEN_REG        | Crypto Length of the input block in bytes            |
| 0x40020010 | CRYPTO_DEST_ADDR_REG  | Crypto DMA destination memory                        |
| 0x40020014 | CRYPTO_STATUS_REG     | Crypto Status register                               |
| 0x40020018 | CRYPTO_CLRIRQ_REG     | Crypto Clear interrupt request                       |
| 0x4002001C | CRYPTO_MREG0_REG      | Crypto Mode depended register 0                      |
| 0x40020020 | CRYPTO_MREG1_REG      | Crypto Mode depended register 1                      |
| 0x40020024 | CRYPTO_MREG2_REG      | Crypto Mode depended register 2                      |
| 0x40020028 | CRYPTO_MREG3_REG      | Crypto Mode depended register 3                      |
| 0x40020100 | CRYPTO_KEYS_START     | Crypto First position of the AES keys storage memory |

Table 152: CRYPTO\_CTRL\_REG (0x40020000)

| Bit   | Mode | Symbol              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|-------|------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:18 | -    | -                   | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 17    | R/W  | CRYPTO_AES_KEXP     | It forces (active high) the execution of the key expansion process with the starting of the AES encryption/decryption process. The bit will be cleared automatically by the hardware, after the completion of the AES key expansion process.                                                                                                                                                                                               | 0x0   |
| 16    | R/W  | CRYPTO_MORE_IN      | 0 - Define that this is the last input block. When the current input is consumed by the crypto engine and the output data is written to the memory, the calculation ends (CRYPTO_INACTIVE goes to one).<br>1 - The current input data block is not the last. More input data will follow. When the current input is consumed, the engine stops and waits for more data (CRYPTO_WAIT_FOR_IN goes to one).                                   | 0x0   |
| 15:10 | R/W  | CRYPTO_HASH_OUT_LEN | The number of bytes minus one of the hash result which will be saved at the memory by the DMA. In relation with the selected hash algorithm the accepted values are:<br>MD5: 0..15 -> 1-16 bytes<br>SHA-1: 0..19 -> 1-20 bytes<br>SHA-256: 0..31 -> 1-32 bytes<br>SHA-256/224: 0..27 -> 1-28 bytes<br>SHA-384: 0..47 -> 1-48 bytes<br>SHA-512: 0..63 -> 1-64 bytes<br>SHA-512/224: 0..27 -> 1-28 bytes<br>SHA-512/256: 0..31 -> 1-32 bytes | 0x0   |
| 9     | R/W  | CRYPTO_HASH_SEL     | Selects the type of the algorithm<br>0 - The encryption algorithm (AES)<br>1 - A hash algorithm.<br>The exact algorithm is defined by the fields CRYPTO_ALG and CRYPTO_ALG_MD.                                                                                                                                                                                                                                                             | 0x0   |
| 8     | R/W  | CRYPTO_IRQ_EN       | Interrupt Request Enable<br>0 - The interrupt generation ability is disabled.<br>1 - The interrupt generation ability is enabled. Generates an interrupt request at the end of operation.                                                                                                                                                                                                                                                  | 0x0   |
| 7     | R/W  | CRYPTO_ENCDEC       | Encryption/Decryption<br>0 - Decryption<br>1 - Encryption                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |

Table 152: CRYPTO\_CTRL\_REG (0x40020000)

| Bit | Mode | Symbol            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6:5 | R/W  | CRYPTO_AES_KEY_SZ | The size of AES Key<br>00 - 128 bits AES Key<br>01 - 192 bits AES Key<br>10 - 256 bits AES Key<br>11 - 256 bits AES Key                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 4   | R/W  | CRYPTO_OUT_MD     | Output Mode. This field makes sense only when the AES algorithm is selected (CRYPTO_HASH_SEL = 0)<br>0 - Write back to memory all the resulting data<br>1 - Write back to memory only the final block of the resulting data                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 3:2 | R/W  | CRYPTO_ALG_MD     | It defines the mode of operation of the AES algorithm when the controller is configured for an encryption/decryption processing (CRYPTO_HASH_SEL = 0).<br>00 - ECB<br>01 - ECB<br>10 - CTR<br>11 - CBC<br>When the controller is configured to applies a HASH function, this field selects the desired HASH algorithm with the help of the CRYPTO_ALG.<br>00 - HASH algorithms that are based on 32 bits operations<br>01 - HASH algorithms that are based on 64 bits operations<br>10 - Reserved<br>11 - Reserved<br>See also the CRYPTO_ALG field. | 0x0   |
| 1:0 | R/W  | CRYPTO_ALG        | Algorithm selection. When CRYPTO_HASH_SEL = 0 the only available choice is the AES algorithm.<br>00 - AES<br>01 - Reserved<br>10 - Reserved<br>11 - Reserved<br>When CRYPTO_HASH_SEL = 1, this field selects the desired hash algorithms, with the help of the CRYPTO_ALG_MD field.<br>If CRYPTO_ALG_MD = 00<br>00 - MD5<br>01 - SHA-1<br>10 - SHA-256/224<br>11 - SHA-256<br>If CRYPTO_ALG_MD = 01<br>00 - SHA-384<br>01 - SHA-512<br>10 - SHA-512/224<br>11 - SHA-512/256                                                                          | 0x0   |

Table 153: CRYPTO\_START\_REG (0x40020004)

| Bit  | Mode | Symbol       | Description                                                                          | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------|-------|
| 31:1 | -    | -            | Reserved                                                                             | 0x0   |
| 0    | R0/W | CRYPTO_START | Write 1 to initiate the processing of the input data. This register is auto-cleared. | 0x0   |

Table 154: CRYPTO\_FETCH\_ADDR\_REG (0x40020008)

| Bit  | Mode | Symbol            | Description                                                                                                                                                                                       | Reset |
|------|------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | CRYPTO_FETCH_ADDR | The memory address from where will be retrieved the data that will be processed. The value of this register is updated as the calculation proceeds and the output data are written to the memory. | 0x0   |

Table 155: CRYPTO\_LEN\_REG (0x4002000C)

| Bit   | Mode | Symbol     | Description                                                                                                                                                                                                                                                      | Reset |
|-------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:24 | -    | -          | Reserved                                                                                                                                                                                                                                                         | 0x0   |
| 23:0  | R/W  | CRYPTO_LEN | It contains the number of bytes of input data. If this number is not a multiple of a block size, the data is automatically extended with zeros. The value of this register is updated as the calculation proceeds and the output data are written to the memory. | 0x0   |

Table 156: CRYPTO\_DEST\_ADDR\_REG (0x40020010)

| Bit  | Mode | Symbol           | Description                                                                                                                                                                           | Reset |
|------|------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | CRYPTO_DEST_ADDR | Destination address at where the result of the processing is stored. The value of this register is updated as the calculation proceeds and the output data are written to the memory. | 0x0   |

Table 157: CRYPTO\_STATUS\_REG (0x40020014)

| Bit  | Mode | Symbol             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|------|------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:3 | -    | -                  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 2    | R    | CRYPTO_IRQ_ST      | The status of the interrupt request line of the CRYPTO block.<br>0 - There is no active interrupt request.<br>1 - An interrupt request is pending.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 1    | R    | CRYPTO_WAIT_FOR_IN | Indicates the situation where the engine waits for more input data. This is applicable when the CRYPTO_MORE_IN= 1, so the input data are fragmented in the memory.<br>0 - The crypto is not waiting for more input data.<br>1 - The crypto waits for more input data.<br>The CRYPTO_INACTIVE flag remains to zero to indicate that the calculation is not finished. The supervisor of the CRYPTO must program to the CRYPTO_FETCH_ADDR and CRYPTO_LEN a new input data fragment. The calculation will be continued as soon as the CRYPTO_START register will be written with 1. This action will clear the CRYPTO_WAIT_FOR_IN flag. | 0x0   |
| 0    | R    | CRYPTO_INACTIVE    | 0 - The CRYPTO is active. The processing is in progress.<br>1 - The CRYPTO is inactive. The processing has finished.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x1   |

Table 158: CRYPTO\_CLRIRQ\_REG (0x40020018)

| Bit  | Mode | Symbol        | Description                                   | Reset |
|------|------|---------------|-----------------------------------------------|-------|
| 31:1 | -    | -             | Reserved                                      | 0x0   |
| 0    | R0/W | CRYPTO_CLRIRQ | Write 1 to clear a pending interrupt request. | 0x0   |

Table 159: CRYPTO\_MREG0\_REG (0x4002001C)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                      | Reset |
|------|------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | CRYPTO_MREG0 | It contains information that are depended by the mode of operation, when is used the AES algorithm:<br>CBC - IV[31:0]<br>CTR - CTRBLK[31:0]. It is the initial value of the 32 bits counter.<br>At any other mode, the contents of this register has no meaning. | 0x0   |

Table 160: CRYPTO\_MREG1\_REG (0x40020020)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                       | Reset |
|------|------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | CRYPTO_MREG1 | It contains information that are depended by the mode of operation, when is used the AES algorithm:<br>CBC - IV[63:32]<br>CTR - CTRBLK[63:32]<br>At any other mode, the contents of this register has no meaning. | 0x0   |

Table 161: CRYPTO\_MREG2\_REG (0x40020024)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                       | Reset |
|------|------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | CRYPTO_MREG2 | It contains information that are depended by the mode of operation, when is used the AES algorithm:<br>CBC - IV[95:64]<br>CTR - CTRBLK[95:64]<br>At any other mode, the contents of this register has no meaning. | 0x0   |

Table 162: CRYPTO\_MREG3\_REG (0x40020028)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                         | Reset |
|------|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | R/W  | CRYPTO_MREG3 | It contains information that are depended by the mode of operation, when is used the AES algorithm:<br>CBC - IV[127:96]<br>CTR - CTRBLK[127:96]<br>At any other mode, the contents of this register has no meaning. | 0x0   |

Table 163: CRYPTO\_KEYS\_START (0x40020100)

| Bit  | Mode | Symbol       | Description                                                                                                                                                          | Reset |
|------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:0 | W    | CRYPTO_KEY_X | CRYPTO_KEY_(0-63)<br>This is the AES keys storage memory. This memory is accessible via AHB slave interface, only when the CRYPTO is inactive (CRYPTO_INACTIVE = 1). | 0x0   |

## 37.5 CACHE REGISTER FILE

Table 164: Register map CACHE

| Address    | Port                 | Description                                          |
|------------|----------------------|------------------------------------------------------|
| 0x400C3000 | CACHE_CTRL1_REG      | Cache control register 1                             |
| 0x400C3004 | CACHE_LNSIZECFG_REG  | Cache line size configuration register               |
| 0x400C3008 | CACHE ASSOCCFG_REG   | Cache associativity configuration register           |
| 0x400C3020 | CACHE_CTRL2_REG      | Cache control register 2                             |
| 0x400C3028 | CACHE_MRM_HITS_REG   | Cache MRM (Miss Rate Monitor) HITS register          |
| 0x400C302C | CACHE_MRM_MISSES_REG | Cache MRM (Miss Rate Monitor) MISSES register        |
| 0x400C3030 | CACHE_MRM_CTRL_REG   | Cache MRM (Miss Rate Monitor) CONTROL register       |
| 0x400C3034 | CACHE_MRM_TINT_REG   | Cache MRM (Miss Rate Monitor) TIME INTERVAL register |
| 0x400C3038 | CACHE_MRM_THRES_REG  | Cache MRM (Miss Rate Monitor) THRESHOLD register     |
| 0x400C3050 | SWD_RESET_REG        | SWD HW reset control register                        |

Table 165: CACHE\_CTRL1\_REG (0x400C3000)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                | Reset |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:2 | -    | -           | Reserved                                                                                                                                                                                                                                                                                   | 0     |
| 1    | R/W  | CACHE_RES1  | Reserved. Always keep 0.                                                                                                                                                                                                                                                                   | 0     |
| 0    | R0/W | CACHE_FLUSH | Writing a '1' into this bit, flushes the contents of the tag memories which invalidates the content of the cache memory. The read of this bit is always '0'.<br>Note: The flushing of the cache TAG memory takes 0x100 or 0x200 HCLK cycles for a Cache Data RAM size of 8 KB resp. 16 KB. | 0     |

Table 166: CACHE\_LNSIZECFG\_REG (0x400C3004)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:2 | -    | -          | Reserved                                                                                                                                                                                                                                                                                                                                                                                  | 0     |
| 1:0  | R/W  | CACHE_LINE | Cache line size:<br>0: 8 bytes,<br>1: 16 bytes,<br>2: 32 bytes,<br>3: reserved.<br>Note: Flush the cache just after the dynamic (run-time) reconfiguration of the cache with an 8 bytes cache line size: write the value "01" into the cache control register CACHE_CTRL1_REG just after the write of the value "00" into the cache line size configuration register CACHE_LNSIZECFG_REG. | 0     |

Table 167: CACHE ASSOCCFG\_REG (0x400C3008)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 31:2 | -    | -      | Reserved    | 0     |

Table 167: CACHE ASSOCCFG\_REG (0x400C3008)

| Bit | Mode | Symbol      | Description                                                                                                                                                                          | Reset |
|-----|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1:0 | R/W  | CACHE_ASSOC | Cache associativity:<br>0: 1-way (direct mapped)<br>1: 2-way<br>2: 4-way<br>3: reserved.<br>Note: Flush the cache controller <u>before</u> dynamically decreasing the associativity. | 2     |

Table 168: CACHE\_CTRL2\_REG (0x400C3020)

| Bit   | Mode | Symbol                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|-------|------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:13 | -    | -                            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0     |
| 12    | R/W  | ENABLE_ALSO_QSPIFLASH_CACHED | Enable also the QSPI FLASH cacheability when remapped to OTP (cached).<br>See also the notes at "CACHE_LEN".                                                                                                                                                                                                                                                                                                                                      | 0     |
| 11    | R/W  | ENABLE_ALSO_OTP_CACHED       | Enable also the OTP cacheability when remapped to QSPI FLASH (cached).<br>See also the notes at "CACHE_LEN".                                                                                                                                                                                                                                                                                                                                      | 0     |
| 10    | R/W  | CACHE_CGEN                   | 0: Cache controller clock gating is not enabled.<br>1: Cache controller clock gating is enabled (enabling power saving).<br>Note: This bit must be set to '0' (default) when setting the CACHE_FLUSH bit while executing from other than QSPI FLASH cached or OTP cached, e.g. from Booter or SYS-RAM.                                                                                                                                            | 0     |
| 9     | R/W  | CACHE_WEN                    | 0: Cache Data and TAG memory read only.<br>1: Cache Data and TAG memory read/write.<br>The TAG and Data memory are only updated by the cache controller.<br>There is no HW protection to prevent unauthorized access by the ARM.<br>Note: When accessing the memory mapped Cache Data and TAG memory (for debugging purposes) only 32 bits access is allowed to the Cache Data memory and only 16 bits access is allowed to the Cache TAG memory. | 0     |
| 8:0   | R/W  | CACHE_LEN                    | Length of QSPI FLASH cacheable memory.<br>(N+1)*64kbyte. N= 0 to 512 (Max of 32 Mbyte).<br>Setting CACHE_LEN=0 disables the cache.<br>Note 1: The OTP memory is completely cacheable (when enabled).<br>Note 2: The max. size/length of QSPI FLASH cacheable memory is 16 Mbyte when also OTP is cached.                                                                                                                                          | 0     |

Table 169: CACHE\_MRM\_HITS\_REG (0x400C3028)

| Bit   | Mode | Symbol   | Description                        | Reset |
|-------|------|----------|------------------------------------|-------|
| 31:19 | -    | -        | Reserved                           | 0     |
| 18:0  | R/W  | MRM_HITS | Contains the amount of cache hits. | 0     |

Table 170: CACHE\_MRM\_MISSES\_REG (0x400C302C)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 31:18 | -    | -      | Reserved    | 0     |

Table 170: CACHE\_MRM\_MISSES\_REG (0x400C302C)

| Bit  | Mode | Symbol     | Description                          | Reset |
|------|------|------------|--------------------------------------|-------|
| 17:0 | R/W  | MRM_MISSES | Contains the amount of cache misses. | 0     |

Table 171: CACHE\_MRM\_CTRL\_REG (0x400C3030)

| Bit  | Mode | Symbol               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:4 | -    | -                    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0     |
| 3    | R/W  | MRM_IRQ_THRES_STATUS | 0: No interrupt is generated.<br>1: Interrupt (pulse-sensitive) is generated because the number of cache misses reached the programmed threshold (threshold != 0).                                                                                                                                                                                                                                                                                                                                                              | 0     |
| 2    | R/W  | MRM_IRQ_TINT_STATUS  | 0: No interrupt is generated.<br>1: Interrupt (pulse-sensitive) is generated because the time interval counter reached the end (time interval != 0).                                                                                                                                                                                                                                                                                                                                                                            | 0     |
| 1    | R/W  | MRM_IRQ_MASK         | 0: Disables interrupt generation.<br>1: Enables interrupt generation.<br>Note: The Cache MRM generates a pulse-sensitive interrupt towards the ARM processor,                                                                                                                                                                                                                                                                                                                                                                   | 0     |
| 0    | R/W  | MRM_START            | 0: Freeze the "misses/hits" counters and reset the time interval counter to the programmed value in CACHE_MRM_TINT_REG.<br>1: Enables the counters.<br>Note: In case CACHE_MRM_CTRL_REG[MRM_START] is set to '1' and CACHE_MRM_TINT_REG (!=0) is used for the MRM interrupt generation, the time interval counter counts down (on a fixed reference clock of 16 MHz) until it's '0'. At that time CACHE_MRM_CTRL_REG[MRM_START] will be reset automatically to '0' by the MRM hardware and the MRM interrupt will be generated. | 0     |

Table 172: CACHE\_MRM\_TINT\_REG (0x400C3034)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                             | Reset |
|-------|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:18 | -    | -        | Reserved                                                                                                                                                                                                                | 0     |
| 17:0  | R/W  | MRM_TINT | Defines the time interval for the monitoring in 16 MHz clock cycles. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_TINT_STATUS].<br>Note: When MRM_TINT=0 (unrealistic value), no interrupt will be generated. | 0     |

Table 173: CACHE\_MRM\_THRES\_REG (0x400C3038)

| Bit   | Mode | Symbol    | Description                                                                                                                                                                                                     | Reset |
|-------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:18 | -    | -         | Reserved                                                                                                                                                                                                        | 0     |
| 17:0  | R/W  | MRM_THRES | Defines the threshold to trigger the interrupt generation. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_THRES_STATUS].<br>Note: When MRM_THRES=0 (unrealistic value), no interrupt will be generated. | 0     |

Table 174: SWD\_RESET\_REG (0x400C3050)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 31:1 | -    | -      | Reserved    | 0     |



Table 174: SWD\_RESET\_REG (0x400C3050)

| Bit | Mode | Symbol           | Description                                                                                                                                                                                                                 | Reset |
|-----|------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R0/W | SWD_HW_RESET_REQ | 0: default.<br>1: HW reset request without resetting the SWD and DAP controller. The register is automatically reset with a HW_RESET.<br>This bit can only be accessed by the debugger software and not by the application. | 0     |

## 37.6 CRG REGISTER FILE

Table 175: Register map CRG

| Address    | Port               | Description                                      |
|------------|--------------------|--------------------------------------------------|
| 0x50000000 | CLK_AMBA_REG       | HCLK, PCLK, divider and clock gates              |
| 0x50000002 | CLK_FREQ_TRIM_REG  | Xtal frequency trimming register.                |
| 0x50000008 | CLK_RADIO_REG      | Radio PLL control register                       |
| 0x5000000A | CLK_CTRL_REG       | Clock control register                           |
| 0x5000000C | CLK_TMR_REG        | Clock control for the timers                     |
| 0x50000010 | PMU_CTRL_REG       | Power Management Unit control register           |
| 0x50000012 | SYS_CTRL_REG       | System Control register                          |
| 0x50000014 | SYS_STAT_REG       | System status register                           |
| 0x50000020 | CLK_32K_REG        | 32 kHz oscillator register                       |
| 0x50000022 | CLK_16M_REG        | 16 MHz RC and xtal oscillator register           |
| 0x50000024 | CLK_RCX20K_REG     | 20KHz RXC-oscillator control register            |
| 0x50000028 | BANDGAP_REG        | bandgap trimming                                 |
| 0x5000002A | ANA_STATUS_REG     | status bit of analog (power management) circuits |
| 0x50000030 | VBUS_IRQ_MASK_REG  | IRQ masking                                      |
| 0x50000032 | VBUS_IRQ_CLEAR_REG | Clear pending IRQ register                       |
| 0x50000034 | BOD_CTRL_REG       | Brown Out Detection control register             |
| 0x50000036 | BOD_CTRL2_REG      | Brown Out Detection control register             |
| 0x50000038 | BOD_STATUS_REG     | Brown Out Detection status register              |
| 0x5000003A | LDO_CTRL1_REG      | LDO control register                             |
| 0x5000003C | LDO_CTRL2_REG      | LDO control register                             |
| 0x5000003E | SLEEP_TIMER_REG    | Timer for regulated sleep                        |
| 0x50000050 | XTALRDY_CTRL_REG   | Control register for XTALRDY IRQ                 |

Table 176: CLK\_AMBA\_REG (0x50000000)

| Bit   | Mode | Symbol          | Description                                                                                                                                  | Reset |
|-------|------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 12    | R/W  | QSPI_ENABLE     | Clock enable for QSPI controller                                                                                                             | 0x0   |
| 11:10 | R/W  | QSPI_DIV        | QSPI divider<br>00 = divide by 1<br>01 = divide by 2<br>10 = divide by 4<br>11 = divide by 8                                                 | 0x0   |
| 9     | R/W  | OTP_ENABLE      | Clock enable for OTP controller                                                                                                              | 0x0   |
| 8     | R/W  | TRNG_CLK_ENABLE | Clock enable for TRNG block                                                                                                                  | 0x0   |
| 7     | R/W  | ECC_CLK_ENABLE  | Clock enable for ECC block                                                                                                                   | 0x0   |
| 6     | R/W  | AES_CLK_ENABLE  | Clock enable for AES crypto block                                                                                                            | 0x0   |
| 5:4   | R/W  | PCLK_DIV        | APB interface clock, Cascaded with HCLK:<br>00 = divide hclk by 1<br>01 = divide hclk by 2<br>10 = divide hclk by 4<br>11 = divide hclk by 8 | 0x2   |
| 3     | -    | -               | Reserved                                                                                                                                     | 0x0   |

Table 176: CLK\_AMBA\_REG (0x50000000)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                             | Reset |
|-----|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2:0 | R/W  | HCLK_DIV | AHB interface and microprocessor clock. Source clock divided by:<br><br>000 = divide hclk by 1<br>001 = divide hclk by 2<br>010 = divide hclk by 4<br>011 = divide hclk by 8<br>1xx = divide hclk by 16 | 0x2   |

Table 177: CLK\_FREQ\_TRIM\_REG (0x50000002)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                    | Reset |
|------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10:8 | R/W  | COARSE_ADJ | Xtal frequency course trimming register.<br>0x0 = lowest frequency<br>0x7 = highest frequency<br>Increment or decrement the binary value with 1. Wait approximately 200usec to allow the adjustment to settle. | 0x0   |
| 7:0  | R/W  | FINE_ADJ   | Xtal frequency fine trimming register.<br>0x00 = lowest frequency<br>0xFF = highest frequency                                                                                                                  | 0x0   |

Table 178: CLK\_RADIO\_REG (0x50000008)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                 | Reset |
|-----|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 11  | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 9:8 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 7   | R/W  | BLE_ENABLE   | Enable the BLE core clocks.<br>When the BLE system clock is disabled, either due to the CLK_RADIO_REG[BLE_ENABLE] or due to the PMU_CTRL_REG[BLE_SLEEP], then any access to the BLE Register file will issue a hard fault to the CPU.                                                                                                       | 0x0   |
| 6   | R/W  | BLE_LP_RESET | Reset for the BLE LP timer                                                                                                                                                                                                                                                                                                                  | 0x1   |
| 5:4 | R/W  | BLE_DIV      | Division factor for BLE core blocks, having as reference the DIVN clock:<br>00 = Divide by 1<br>01 = Divide by 2<br>10 = Divide by 4<br>11 = Divide by 8<br>The programmed frequency should not be lower than 8MHz, not faster than 16MHz and not faster than the programmed CPU clock frequency. Refer also to BLE_CNTL2_REG[BLE_CLK_SEL]. | 0x0   |
| 3   | R/W  | RFCU_ENABLE  | Enable the RF control Unit clock                                                                                                                                                                                                                                                                                                            | 0x0   |
| 1:0 | R/W  | RFCU_DIV     | Division factor for RF Control Unit<br>0x0 = divide by 1<br>0x1 = divide by 2<br>0x2 = divide by 4<br>0x3 = divide by 8<br><br>The programmed frequency must be exactly 8MHz.                                                                                                                                                               | 0x0   |

Table 179: CLK\_CTRL\_REG (0x5000000A)

| Bit | Mode | Symbol             | Description                                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R    | RUNNING_AT_PLL96M  | Indicates that the PLL96MHz clock is used as clock, and may not be switched off                                                                                                                                                                                                                                                  | 0x0   |
| 14  | R    | RUNNING_AT_XTAL16M | Indicates that the XTAL16M clock is used as clock, and may not be switched off                                                                                                                                                                                                                                                   | 0x0   |
| 13  | R    | RUNNING_AT_RC16M   | Indicates that the RC16M clock is used as clock                                                                                                                                                                                                                                                                                  | 0x1   |
| 12  | R    | RUNNING_AT_32K     | Indicates that either the RC32k or XTAL32k is being used as clock                                                                                                                                                                                                                                                                | 0x0   |
| 9:8 | R/W  | CLK32K_SOURCE      | Sets the clock source of the LowerPower clock<br>'00': 32 KHz RC Oscillator<br>'01': RCX Oscillator<br>'10': XTAL32kHz, when using an external crystal i.c.w. the internal oscillator (set P20 and P21 to FUNC_XTAL32)<br>'11': XTAL32kHz, when an external generator or MCU applies a square wave on P20 (set P20 to FUNC_GPIO) | 0x0   |
| 6   | R/W  | DIVN_XTAL32M_MODE  | Enables the DIVN divide-by-2, in case of a 32 MHz crystal (See also XTAL32M_MODE), to keep the DIVN clock at 16 MHz.                                                                                                                                                                                                             | 0x0   |
| 5   | R/W  | PLL_DIV2           | Divides the PLL clock by 2 before being used                                                                                                                                                                                                                                                                                     | 0x0   |
| 4   | R/W  | USB_CLK_SRC        | Selects the USB source clock<br>0 : PLL clock, divided by 2<br>1 : HCLK                                                                                                                                                                                                                                                          | 0x0   |
| 3   | R/W  | XTAL32M_MODE       | Enables dividers in the XTAL for both the RF and the BB PLL.                                                                                                                                                                                                                                                                     | 0x0   |
| 2   | R/W  | XTAL16M_DISABLE    | Setting this bit instantaneously disables the 16 MHz crystal oscillator. This bit may not be set to '1' when "RUNNING_AT_XTAL16M is '1' to prevent deadlock. After resetting this bit, wait for XTAL16_TRIM_READY to become '1' before switching to XTAL16 clock source.                                                         | 0x0   |
| 1:0 | R/W  | SYS_CLK_SEL        | Selects the clock source.<br>0x0 : XTAL16M (check the XTAL16_TRIM_READY bit!!)<br>0x1 : RC16M<br>0x2 : The Low Power clock is used<br>0x3 : The PLL96Mhz is used                                                                                                                                                                 | 0x1   |

Table 180: CLK\_TMR\_REG (0x5000000C)

| Bit | Mode | Symbol            | Description                                                                                                                                      | Reset |
|-----|------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 14  | R/W  | P06_TMR1_PWM_MODE | Maps Timer1_pwm onto P0_6, when DEBUGGER_EN = '0'. This state is preserved during deep sleep, to allow PWM5 output on the pad during deep sleep. | 0x0   |
| 13  | R/W  | WAKEUPCT_ENABLE   | Enables the clock                                                                                                                                | 0x0   |
| 12  | R/W  | BREATH_ENABLE     | Enables the clock                                                                                                                                | 0x0   |
| 11  | R/W  | TMR2_CLK_SEL      | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock                                                                                     | 0x0   |
| 10  | R/W  | TMR2_ENABLE       | Enable timer clock                                                                                                                               | 0x0   |

Table 180: CLK\_TMR\_REG (0x5000000C)

| Bit | Mode | Symbol       | Description                                                                                                   | Reset |
|-----|------|--------------|---------------------------------------------------------------------------------------------------------------|-------|
| 9:8 | R/W  | TMR2_DIV     | Division factor for Timer<br>0x0 = divide by 1<br>0x1 = divide by 2<br>0x2 = divide by 4<br>0x3 = divide by 8 | 0x0   |
| 7   | R/W  | TMR1_CLK_SEL | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock                                                  | 0x0   |
| 6   | R/W  | TMR1_ENABLE  | Enable timer clock                                                                                            | 0x0   |
| 5:4 | R/W  | TMR1_DIV     | Division factor for Timer<br>0x0 = divide by 1<br>0x1 = divide by 2<br>0x2 = divide by 4<br>0x3 = divide by 8 | 0x0   |
| 3   | R/W  | TMR0_CLK_SEL | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock                                                  | 0x0   |
| 2   | R/W  | TMR0_ENABLE  | Enable timer clock                                                                                            | 0x0   |
| 1:0 | R/W  | TMR0_DIV     | Division factor for Timer<br>0x0 = divide by 1<br>0x1 = divide by 2<br>0x2 = divide by 4<br>0x3 = divide by 8 | 0x0   |

Table 181: PMU\_CTRL\_REG (0x50000010)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15   | R/W  | RETAIN_ECCRAM   | Selects the retainability of the ECC RAM during deep sleep.<br>'1' is retainable, '0' is power gated                                                                                                                                                                                                                                                                                               | 0x0   |
| 14   | R/W  | RETAIN_CACHE    | Selects the retainability of the cache block during deep sleep.<br>'1' is retainable, '0' is power gated                                                                                                                                                                                                                                                                                           | 0x0   |
| 13   | R/W  | ENABLE_CLKLESS  | Selects the clockless sleep mode. Wakeup is done asynchronously.<br>When set to '1', the lp_clk is stopped during deep sleep, until a wakeup event (not debounced) is detected by the WAKUPCT block.<br>When set to '0', the lp_clk continues running, so the MAC counters keep on running.<br>This mode cannot be combined with regulated sleep, so keep SLEEP_TIMER=0 when using ENABLE_CLKLESS. | 0x0   |
| 12:8 | R/W  | RETAIN_RAM      | Select the retainability of the 5 system memory RAM macros during deep sleep.<br>'1' is retainable, '0' is power gated<br>(4) is SYSRAM5<br>(3) is SYSRAM4<br>(2) is SYSRAM3<br>(1) is SYSRAM2<br>(0) is SYSRAM1                                                                                                                                                                                   | 0x0   |
| 7:6  | R/W  | OTP_COPY_DIV    | Sets the HCLK division during OTP mirroring                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 5    | R/W  | RESET_ON_WAKEUP | Perform a Hardware Reset after waking up. Booter will be started.                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 3    | -    | -               | Reserved                                                                                                                                                                                                                                                                                                                                                                                           | 0x1   |

Table 181: PMU\_CTRL\_REG (0x50000010)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                                                         | Reset |
|-----|------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2   | R/W  | BLE_SLEEP    | Put the BLE in powerdown.<br>When the BLE system clock is disabled, either due to the CLK_RADIO_REG[BLE_ENABLE] or due to the PMU_CTRL_REG[BLE_SLEEP], then any access to the BLE Register file will issue a hard fault to the CPU. | 0x1   |
| 1   | R/W  | RADIO_SLEEP  | Put the digital part of the radio in powerdown                                                                                                                                                                                      | 0x1   |
| 0   | R/W  | PERIPH_SLEEP | Put all peripherals (I2C, UART, SPI, ADC) in powerdown                                                                                                                                                                              | 0x1   |

Table 182: SYS\_CTRL\_REG (0x50000012)

| Bit | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                  | Reset |
|-----|------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | W    | SW_RESET        | Writing a '1' to this bit will generate a SW_RESET.                                                                                                                                                                                                                                          | 0x0   |
| 14  | R/W  | REMAP_INTVECT   | 0: normal operation<br>1: If ARM is in address range 0 to 0xFF then the address is remapped to SYS-RAM 0x07FC.0000 to 0x07FC.00FF. This allows to put the interrupt vector table to be placed in RAM while executing from QSPI                                                               | 0x0   |
| 13  | R/W  | OTP_COPY        | Enables OTP to SysRAM copy action after waking up PD_SYS                                                                                                                                                                                                                                     | 0x0   |
| 12  | R/W  | QSPI_INIT       | Enables QSPI initialization after wakeup                                                                                                                                                                                                                                                     | 0x0   |
| 11  | R/W  | DEV_PHASE       | Sets the development phase mode, used in combination with OTP_COPY<br>No copy action to SysRAM is done when the system wakes up.<br>For emulating startup time, the OTP_COPY bit still needs to be set.                                                                                      | 0x0   |
| 10  | R/W  | CACHERAM_MUX    | Controls accessibility of Cache RAM:<br>0: the cache controller is bypassed, the cacheRAM is visible in the memory space next to the DataRAMs<br>1: the cache controller is enabled, the cacheRAM is not visible anymore in the memory space                                                 | 0x0   |
| 9   | R/W  | TIMEOUT_DISABLE | Disables timeout in Power statemachine. By default, the statemachine continues if after 2 ms the blocks are not started up. This can be read back from ANA_STATUS_REG                                                                                                                        | 0x0   |
| 7   | R/W  | DEBUGGER_ENABLE | Enable the debugger. This bit is set by the booter according to the OTP header. If not set, the SWDIO and SW_CLK can be used as gpio ports.                                                                                                                                                  | 0x0   |
| 6   | R/W  | OTPC_RESET_REQ  | Reset request for the OTP controller.                                                                                                                                                                                                                                                        | 0x0   |
| 5   | R/W  | PAD_LATCH_EN    | Latches the control signals of the pads for state retention in powerdown mode.<br>0 = Control signals are retained<br>1 = Latch is transparent, pad can be recontrolled                                                                                                                      | 0x1   |
| 4:3 | R/W  | REMAP_RAMs      | Defines the sequence of the 3 first DataRAMs in the memory space. DataRAM4, DataRAM5 and potentially CacheRAM, cannot not be reshuffled.<br>0x0: DataRAM1, DataRAM2, DataRAM3<br>0x1: DataRAM2, DataRAM1, DataRAM3<br>0x2: DataRAM3, DataRAM1, DataRAM2<br>0x3: DataRAM3, DataRAM2, DataRAM1 | 0x0   |

Table 182: SYS\_CTRL\_REG (0x50000012)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|-----|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2:0 | R/W  | REMAP_ADR0 | Controls which memory is located at address 0x0000 for execution.<br>0x0: ROM<br>0x1: OTP<br>0x2: FLASH<br>0x3: RAMS (for the exact configuration see REMAP_RAM)<br>0x4: FLASH un-cached (for verification only)<br>0x5: OTP un-cached (for verification only)<br>0x6: Cache Data RAM (CACHERAM_MUX=0, for testing purposes only)<br>Note 1: DWord (64 bits) access is not supported by the Cache Data RAM interface in mirrored mode (only 32, 16 and 8 bits).<br>Note 2: DMA access is not supported by the Cache Data RAM interface when REMAP_ADR0=0x6. | 0x0   |

Table 183: SYS\_STAT\_REG (0x50000014)

| Bit | Mode | Symbol            | Description                                                                                  | Reset |
|-----|------|-------------------|----------------------------------------------------------------------------------------------|-------|
| 11  | -    | -                 | Reserved                                                                                     | 0x0   |
| 10  | -    | -                 | Reserved                                                                                     | 0x1   |
| 9   | R    | BLE_IS_UP         | Indicates that PD_DBG is functional                                                          | 0x0   |
| 8   | R    | BLE_IS_DOWN       | Indicates that PD_DBG is in power down                                                       | 0x1   |
| 7   | -    | -                 | Reserved                                                                                     | 0x0   |
| 6   | R    | XTAL16_TRIM_READY | Indicates that XTAL trimming mechanism is ready, i.e. the trimming equals CLK_FREQ_TRIM_REG. | 0x1   |
| 5   | R    | DBG_IS_ACTIVE     | Indicates that a debugger is attached.                                                       | 0x0   |
| 4   | -    | -                 | Reserved                                                                                     | 0x0   |
| 3   | R    | PER_IS_UP         | Indicates that PD_PER is functional                                                          | 0x0   |
| 2   | R    | PER_IS_DOWN       | Indicates that PD_PER is in power down                                                       | 0x1   |
| 1   | R    | RAD_IS_UP         | Indicates that PD_RAD is functional                                                          | 0x0   |
| 0   | R    | RAD_IS_DOWN       | Indicates that PD_RAD is in power down                                                       | 0x1   |

Table 184: CLK\_32K\_REG (0x50000020)

| Bit  | Mode | Symbol                 | Description                                                                                                                                                                                                        | Reset |
|------|------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 12   | R/W  | XTAL32K_DISABLE_AMPREG | Setting this bit disables the amplitude regulation of the XTAL32kHz oscillator.<br><br>Set this bit to '1' for an external clock to XTAL32Kp<br><br>Keep this bit '0' with a crystal between XTAL32Kp and XTAL32Km | 0x0   |
| 11:8 | R/W  | RC32K_TRIM             | 0000 = lowest frequency<br>0111 = default<br>1111 = highest frequency                                                                                                                                              | 0x7   |
| 7    | R/W  | RC32K_ENABLE           | Enables the 32kHz RC oscillator                                                                                                                                                                                    | 0x1   |
| 6:3  | R/W  | XTAL32K_CUR            | Bias current for the 32kHz XTAL oscillator. 0000 is minimum, 1111 is maximum, 0011 is default. For each application there is an optimal setting for which the start-up behavior is optimal                         | 0x3   |



Table 184: CLK\_32K\_REG (0x50000020)

| Bit | Mode | Symbol         | Description                                                                                               | Reset |
|-----|------|----------------|-----------------------------------------------------------------------------------------------------------|-------|
| 2:1 | R/W  | XTAL32K_RBIAS  | Setting for the bias resistor. 00 is maximum, 11 is minimum. Preferred setting will be provided by Dialog | 0x2   |
| 0   | R/W  | XTAL32K_ENABLE | Enables the 32kHz XTAL oscillator                                                                         | 0x0   |

Table 185: CLK\_16M\_REG (0x50000022)

| Bit   | Mode | Symbol                  | Description                                                                     | Reset |
|-------|------|-------------------------|---------------------------------------------------------------------------------|-------|
| 14    | R/W  | XTAL16_HPASS_FLT_EN     | enables high pass filter                                                        | 0x0   |
| 13    | R/W  | XTAL16_SPIKE_FLT_BYPASS | bypasses spikefilter                                                            | 0x0   |
| 12:10 | R/W  | XTAL16_AMP_TRIM         | sets xtal amplitude, 0 is minimum, 101 is maximum                               | 0x3   |
| 7:5   | R/W  | XTAL16_CUR_SET          | start-up current for the 16MHz XTAL oscillator. 000 is minimum, 110 is maximum. | 0x0   |
| 4:1   | R/W  | RC16M_TRIM              | 0000 = lowest frequency<br>1111 = highest frequency                             | 0x0   |
| 0     | R/W  | RC16M_ENABLE            | Enables the 16MHz RC oscillator                                                 | 0x0   |

Table 186: CLK\_RCX20K\_REG (0x50000024)

| Bit | Mode | Symbol        | Description                                                           | Reset |
|-----|------|---------------|-----------------------------------------------------------------------|-------|
| 11  | R/W  | RCX20K_ENABLE | Enable the RCX32 oscillator                                           | 0     |
| 10  | R/W  | RCX20K_LOWF   | Extra low frequency                                                   | 0     |
| 9:8 | R/W  | RCX20K_BIAS   | Bias control                                                          | 1     |
| 7:4 | R/W  | RCX20K_NTC    | Temperature control                                                   | 7     |
| 3:0 | R/W  | RCX20K_TRIM   | 0000 = lowest frequency<br>0111 = default<br>1111 = highest frequency | 8     |

Table 187: BANDGAP\_REG (0x50000028)

| Bit   | Mode | Symbol                   | Description                                                                                                                                                                                                                                                                                                                                  | Reset |
|-------|------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 14    | R/W  | BYPASS_COLD_BOOT_DISABLE | 0x1 -> Switch to LDO_SUPPLY_USB on vbus_available & vbus_high & wakeup ( <b>SET to 0x1 after boot</b> )<br>0x0 -> Switch to LDO_SUPPLY_USB on vbus_available                                                                                                                                                                                 | 0x0   |
| 13:10 | R/W  | LDO_SLEEP_TRIM           | 0x4 -> 1120 mV<br>0x5 -> 1089 mV<br>0x6 -> 1058 mV<br>0x7 -> 1030 mV<br>0x0 -> 1037 mV<br>0x1 -> 1005 mV<br>0x2 -> 978 mV<br>0x3 -> 946 mV<br>0x8 -> 952 mV<br>0x9 -> 918 mV<br>0xA -> 889 mV<br>0xB -> 861 mV<br>0xC -> 862 mV<br>0xD -> 828 mV<br>0xE -> 798 mV<br>0xF -> 770 mV<br>These values are from simulation and vary over corners | 0x0   |

Table 187: BANDGAP\_REG (0x50000028)

| Bit | Mode | Symbol    | Description               | Reset |
|-----|------|-----------|---------------------------|-------|
| 9:5 | R/W  | BGR_ITRIM | Current trimming for bias | 0x0   |
| 4:0 | R/W  | BGR_TRIM  | Trim register for bandgap | 0x0   |

Table 188: ANA\_STATUS\_REG (0x5000002A)

| Bit | Mode | Symbol              | Description                     | Reset |
|-----|------|---------------------|---------------------------------|-------|
| 15  | R    | COMP_1V8_PA_HIGH    | VDD1V8P > 1.7V                  | 0x0   |
| 14  | R    | COMP_1V8_FLASH_HIGH | VDD1V8 > 1.7V                   | 0x0   |
| 13  | R    | COMP_V33_HIGH       | V33 > 1.7V                      | 0x0   |
| 12  | R    | COMP_VBUS_LOW       | VBUS > 3.4V                     | 0x0   |
| 11  | R    | COMP_VBUS_HIGH      | VBUS > 4V                       | 0x0   |
| 10  | R    | LDO_1V8_FLASH_OK    | ldo_vdd1v8 = ok                 | 0x0   |
| 9   | R    | LDO_1V8_PA_OK       | ldo_vdd1v8P = ok                | 0x0   |
| 8   | R    | LDO_CORE_OK         | ldo_core = ok                   | 0x0   |
| 7   | R    | COMP_VDD_HIGH       | VDD > 1.13V                     | 0x1   |
| 6   | R    | BANDGAP_OK          | bandgap = ok                    | 0x0   |
| 5   | R    | LDO_SUPPLY_USB_OK   | ldo_supply_usb = ok             | 0x0   |
| 4   | R    | LDO_SUPPLY_VBAT_OK  | ldo_supply_vbat = ok            | 0x1   |
| 3   | R    | NEWBAT              | new battery has been detected   | 0x0   |
| 2   | R    | VBUS_AVAILABLE      | vbus is available (vbus > vbat) | 0x0   |
| 1   | R    | COMP_VBAT_OK        | vbat > 1.7V                     | 0x0   |
| 0   | R    | LDO_RADIO_OK        | ldo_radio = ok                  | 0x0   |

Table 189: VBUS\_IRQ\_MASK\_REG (0x50000030)

| Bit | Mode | Symbol           | Description                                                                                      | Reset |
|-----|------|------------------|--------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | VBUS_IRQ_EN_RISE | Setting this bit to '1' enables VBUS_IRQ generation when the VBUS starts to ramp above threshold | 0x0   |
| 0   | R/W  | VBUS_IRQ_EN_FALL | Setting this bit to '1' enables VBUS_IRQ generation when the VBUS starts to fall below threshold | 0x0   |

Table 190: VBUS\_IRQ\_CLEAR\_REG (0x50000032)

| Bit  | Mode | Symbol         | Description                                                     | Reset |
|------|------|----------------|-----------------------------------------------------------------|-------|
| 15:0 | R/W  | VBUS_IRQ_CLEAR | Writing any value to this register will reset the VBUS_IRQ line | 0x0   |

Table 191: BOD\_CTRL\_REG (0x50000034)

| Bit  | Mode | Symbol      | Description                                       | Reset |
|------|------|-------------|---------------------------------------------------|-------|
| 10:8 | R/W  | BOD_VDD_LVL | VDD BOD Level; 0=700mV; 1=700mV; 3=800mV; 7=1.05V | 0x7   |
| 7:0  | -    | Reserved    | Reserved                                          | 0x0   |

Table 192: BOD\_CTRL2\_REG (0x50000036)

| Bit | Mode | Symbol           | Description                        | Reset |
|-----|------|------------------|------------------------------------|-------|
| 5   | R/W  | BOD_VBAT_EN      | VBAT BOD Enable                    | 0x0   |
| 4   | R/W  | BOD_1V8_FLASH_EN | 1V8 Flash BOD Enable               | 0x0   |
| 3   | R/W  | BOD_1V8_PA_EN    | 1V8 PA BOD Enable                  | 0x0   |
| 2   | R/W  | BOD_V33_EN       | V33 BOD Enable                     | 0x0   |
| 1   | R/W  | BOD_VDD_EN       | VDD BOD Enable                     | 0x0   |
| 0   | R/W  | BOD_RESET_EN     | Generate a chip reset on BOD event | 0x0   |

Table 193: BOD\_STATUS\_REG (0x50000038)

| Bit | Mode | Symbol            | Description                             | Reset |
|-----|------|-------------------|-----------------------------------------|-------|
| 4   | R/W  | BOD_VBAT_LOW      | Indicates VBAT > VBAT_Trigger           | 0x0   |
| 3   | R/W  | BOD_V33_LOW       | Indicates V33 > V33_Trigger             | 0x0   |
| 2   | R/W  | BOD_1V8_FLASH_LOW | Indicates V18_Flash > V18_Flash_Trigger | 0x0   |
| 1   | R/W  | BOD_1V8_PA_LOW    | Indicates V18_PA > V18_PA_Trigger       | 0x0   |
| 0   | R/W  | BOD_VDD_LOW       | Indicates VDD > VDD_Trigger             | 0x0   |

Table 194: LDO\_CTRL1\_REG (0x5000003A)

| Bit   | Mode | Symbol                    | Description                                                                                                                                                                                                | Reset |
|-------|------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 14    | R/W  | LDO_RADIO_ENAB<br>LE      | Enables (1) or disables (0) LDO_RADIO (V14)<br>For fast XTAL startup, this bit may be kept to '1' during deep sleep. The LDO is switched off automatically when in deep sleep, and enabled when waking up. | 0x0   |
| 13:11 | R/W  | LDO_RADIO_SETV<br>D       | Sets the output voltage of LDO_RADIO<br>000 = 1.30 V<br>001 = 1.35 V<br>010 = 1.40 V<br>011 = 1.45 V<br>1XX = 1.50 V                                                                                       | 0x0   |
| 10:8  | R/W  | LDO_CORE_SETV<br>D        | Sets the output voltage of LDO_CORE<br>000 = 1.20 V<br>001 = 1.15 V<br>010 = 1.10 V<br>011 = 1.05 V<br>1XX = 1.32 V                                                                                        | 0x0   |
| 7:6   | R/W  | LDO_SUPPLY_USB_<br>LEVEL  | Sets the output voltage of LDO_SUPPLY_USB<br>00 = 2.40 V<br>01 = 3.30 V<br>10 = 3.45 V<br>11 = 3.60 V                                                                                                      | 0x1   |
| 5:4   | R/W  | LDO_SUPPLY_VBAT_<br>LEVEL | Sets the output voltage of LDO_SUPPLY_VBAT<br>00 = 2.40 V<br>01 = 3.30 V<br>10 = 3.45 V<br>11 = 3.60 V                                                                                                     | 0x1   |

Table 194: LDO\_CTRL1\_REG (0x5000003A)

| Bit | Mode | Symbol                 | Description                                                                                                  | Reset |
|-----|------|------------------------|--------------------------------------------------------------------------------------------------------------|-------|
| 3:2 | R/W  | LDO_VBAT_RET_LE<br>VEL | Sets the output voltage of LDO_VBAT_RET<br>00 = 2.40 V<br>01 = 3.30 V<br>10 = 3.45 V<br>11 = 3.60 V          | 0x1   |
| 1:0 | R/W  | LDO_CORE_CURLI<br>M    | Sets the current limit of LDO_CORE<br>00 = Current limiter disabled<br>01 = 8 mA<br>10 = 60 mA<br>11 = 80 mA | 0x3   |

Table 195: LDO\_CTRL2\_REG (0x5000003C)

| Bit | Mode | Symbol                        | Description                                                       | Reset |
|-----|------|-------------------------------|-------------------------------------------------------------------|-------|
| 6   | R/W  | LDO_1V8_PA_RET_<br>DISABLE    | Disables (1) or enables (0) LDO_1V8_PA_RET                        | 0x0   |
| 5   | R/W  | LDO_1V8_FLASH_R<br>ET_DISABLE | Disables (1) or enables (0) LDO_1V8_FLASH_RET                     | 0x0   |
| 4   | R/W  | LDO_VBAT_RET_DI<br>SABLE      | Disables (1) or enables (0) LDO_VBAT_RET                          | 0x0   |
| 3   | R/W  | LDO_1V8_PA_ON                 | Enables (1) or disables (0) LDO_1V8_PA                            | 0x1   |
| 2   | R/W  | LDO_1V8_FLASH_O<br>N          | Enables (1) or disables (0) LDO_1V8_FLASH                         | 0x1   |
| 1   | R/W  | LDO_3V3_ON                    | Enables (1) or disables (0) LDO_SUPPLY_VBAT and<br>LDO_SUPPLY_USB | 0x1   |
| 0   | R/W  | LDO_1V2_ON                    | Enables (1) or disables (0) LDO_CORE                              | 0x1   |

Table 196: SLEEP\_TIMER\_REG (0x5000003E)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                         | Reset |
|------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | SLEEP_TIMER | Defines the amount of ticks of the sleep clock between ena-<br>bling the bandgap for re-charging the retention LDOs. This<br>value depends on the load and should be calibrated on a per<br>application basis. If set to 0, no recharging cycle will happen<br>at all.<br>Keep this value to 0 (no recharging) when using the clock-<br>less sleep. | 0x0   |

Table 197: XTALRDY\_CTRL\_REG (0x50000050)

| Bit | Mode | Symbol      | Description                                                                                                | Reset |
|-----|------|-------------|------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | XTALRDY_CNT | Number of LP cycles between the crystal is enabled, and the<br>XTALRDY_IRQ is fired.<br>0x00: no interrupt | 0x0   |

## 37.7 DCDC REGISTER FILE

Table 198: Register map DCDC

| Address    | Port                | Description                         |
|------------|---------------------|-------------------------------------|
| 0x50000082 | DCDC_CTRL_0_REG     | DCDC First Control Register         |
| 0x50000084 | DCDC_CTRL_1_REG     | DCDC Second Control Register        |
| 0x50000086 | DCDC_CTRL_2_REG     | DCDC Third Control Register         |
| 0x50000088 | DCDC_V14_0_REG      | DCDC V14 First Control Register     |
| 0x5000008A | DCDC_V14_1_REG      | DCDC V14 Second Control Register    |
| 0x5000008C | DCDC_V18_0_REG      | DCDC V18 First Control Register     |
| 0x5000008E | DCDC_V18_1_REG      | DCDC V18 Second Control Register    |
| 0x50000090 | DCDC_VDD_0_REG      | DCDC VDD First Control Register     |
| 0x50000092 | DCDC_VDD_1_REG      | DCDC VDD Second Control Register    |
| 0x50000094 | DCDC_V18P_0_REG     | DCDC VPA First Control Register     |
| 0x50000096 | DCDC_V18P_1_REG     | DCDC VPA Second Control Register    |
| 0x50000098 | DCDC_RET_0_REG      | DCDC First Retention Mode Register  |
| 0x5000009A | DCDC_RET_1_REG      | DCDC Second Retention Mode Register |
| 0x5000009C | DCDC_TRIM_REG       | DCDC Comparator Trim Register       |
| 0x5000009E | DCDC_TEST_0_REG     | DCDC Test Register                  |
| 0x500000A0 | DCDC_TEST_1_REG     | DCDC Test Register                  |
| 0x500000A2 | DCDC_STATUS_0_REG   | DCDC First Status Register          |
| 0x500000A4 | DCDC_STATUS_1_REG   | DCDC Second Status Register         |
| 0x500000A6 | DCDC_STATUS_2_REG   | DCDC Third Status Register          |
| 0x500000A8 | DCDC_STATUS_3_REG   | DCDC Fourth Status Register         |
| 0x500000AA | DCDC_STATUS_4_REG   | DCDC Fifth Status Register          |
| 0x500000AC | DCDC_TRIM_0_REG     | DCDC V14 Comparator Trim Register   |
| 0x500000AE | DCDC_TRIM_1_REG     | DCDC V18 Comparator Trim Register   |
| 0x500000B0 | DCDC_TRIM_2_REG     | DCDC VDD Comparator Trim Register   |
| 0x500000B2 | DCDC_TRIM_3_REG     | DCDC VPA Comparator Trim Register   |
| 0x500000B4 | DCDC_IRQ_STATUS_REG | DCDC Interrupt Status Register      |
| 0x500000B6 | DCDC_IRQ_CLEAR_REG  | DCDC Interrupt Clear Register       |
| 0x500000B8 | DCDC_IRQ_MASK_REG   | DCDC Interrupt Clear Register       |

Table 199: DCDC\_CTRL\_0\_REG (0x50000082)

| Bit   | Mode | Symbol                    | Description                                                                                                                                                 | Reset |
|-------|------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 14    | R/W  | DCDC_FAST_START<br>UP     | Set current limit to maximum during initial startup                                                                                                         | 0x0   |
| 13    | R/W  | DCDC_BROWNOUT<br>_LV_MODE | Switches to low voltage settings when battery voltage drops below 2.5 V                                                                                     | 0x1   |
| 12:11 | R/W  | DCDC_IDLE_CLK_D<br>IV     | Idle Clock Divider<br>00 = 2<br>01 = 4<br>10 = 8<br>11 = 16                                                                                                 | 0x1   |
| 10:3  | R/W  | DCDC_PRIORITY             | Charge priority register (4x 2 bit ID)<br>Charge sequence is [1:0] > [3:2] > [5:4] > [7:6]<br>ID[V14] = 00<br>ID[V18] = 01<br>ID[VDD] = 10<br>ID[V18P] = 11 | 0xE4  |

Table 199: DCDC\_CTRL\_0\_REG (0x50000082)

| Bit | Mode | Symbol         | Description                                                                             | Reset |
|-----|------|----------------|-----------------------------------------------------------------------------------------|-------|
| 2   | R/W  | DCDC_FW_ENABLE | Freewheel switch enable                                                                 | 0x1   |
| 1:0 | R/W  | DCDC_MODE      | DCDC converter mode<br>00 = Disabled<br>01 = Active<br>10 = Sleep mode<br>11 = Disabled | 0x0   |

Table 200: DCDC\_CTRL\_1\_REG (0x50000084)

| Bit   | Mode | Symbol                    | Description                                                                                                                                                     | Reset |
|-------|------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | R/W  | DCDC_STARTUP_DELAY        | Delay between turning bias on and converter becoming active<br>0 - 31 us, 1 us step size                                                                        | 0xA   |
| 10:5  | R/W  | DCDC_GLOBAL_MAX_IDLE_TIME | Global maximum idle time<br>The current limit of any output that is idle for this long will be downramped faster than normal<br>0 - 7875 ns, 125 ns step size   | 0x20  |
| 4:0   | R/W  | DCDC_TIMEOUT              | P and N switch timeout, if switch is closed longer than this a timeout is generated and the FSM is forced to the next state<br>0 - 1937.5 ns, 62.5 ns step size | 0x10  |

Table 201: DCDC\_CTRL\_2\_REG (0x50000086)

| Bit   | Mode | Symbol                | Description                                                                              | Reset |
|-------|------|-----------------------|------------------------------------------------------------------------------------------|-------|
| 15:12 | R/W  | DCDC_TIMEOUT_IRQ_TRIG | Number of timeout events before timeout interrupt is generated                           | 0x8   |
| 11:8  | R/W  | DCDC_TIMEOUT_IRQ_RES  | Number of successive non-timed out charge events required to clear timeout event counter | 0x8   |
| 7:6   | R/W  | DCDC_TUNE             | Trim current sensing circuitry<br>00 = +0 %<br>01 = +4 %<br>10 = +8 %<br>11 = +12 %      | 0x0   |
| 5:3   | R/W  | DCDC_LSSUP_TRIM       | Trim low side supply voltage<br>$V = 2\text{ V} + 100\text{ mV} * N$                     | 0x5   |
| 2:0   | R/W  | DCDC_HSGND_TRIM       | Trim high side ground<br>$V = V_{BAT} - (2.2\text{ V} + 200\text{ mV} * N)$              | 0x5   |

Table 202: DCDC\_V14\_0\_REG (0x50000088)

| Bit   | Mode | Symbol                  | Description                                                                                  | Reset |
|-------|------|-------------------------|----------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_V14_FAST_RAMPING   | V14 output fast current ramping (improves response time at the cost of more ripple)          | 0x1   |
| 14:10 | R/W  | DCDC_V14_VOLTAGE        | V14 output voltage<br>$V = 1.2\text{ V} + 25\text{ mV} * N$                                  | 0x8   |
| 9:5   | R/W  | DCDC_V14_CUR_LIM_MAX_HV | V14 output maximum current limit (high battery voltage mode)<br>$I = 30\text{ mA} * (1 + N)$ | 0xD   |
| 4:0   | R/W  | DCDC_V14_CUR_LIM_MIN    | V14 output minimum current limit<br>$I = 30\text{ mA} * (1 + N)$                             | 0x4   |

Table 203: DCDC\_V14\_1\_REG (0x5000008A)

| Bit   | Mode | Symbol                      | Description                                                                                                                                        | Reset |
|-------|------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_V14_ENABL<br>E_HV      | V14 output enable (high battery voltage mode)<br>0 = Disabled<br>1 = Enabled                                                                       | 0x1   |
| 14    | R/W  | DCDC_V14_ENABL<br>E_LV      | V14 output enable (low battery voltage mode)<br>0 = Disabled<br>1 = Enabled                                                                        | 0x1   |
| 13:10 | R/W  | DCDC_V14_CUR_LI<br>M_MAX_LV | V14 output maximum current limit low battery voltage mode)<br>$I = 30 \text{ mA} * (1 + N)$                                                        | 0x6   |
| 9:5   | R/W  | DCDC_V14_IDLE_H<br>YST      | V14 output idle time hysteresis<br>0 - 3875 ns, 125 ns step size<br>IDLE_MAX = IDLE_MIN + IDLE_HYST<br>Maximum idle time before decreasing CUR_LIM | 0x4   |
| 4:0   | R/W  | DCDC_V14_IDLE_MI<br>N       | V14 output minimum idle time<br>0 - 3875 ns, 125 ns step size<br>Minimum idle time, CUR_LIM is increased if this limit is not reached              | 0x10  |

Table 204: DCDC\_V18\_0\_REG (0x5000008C)

| Bit   | Mode | Symbol                      | Description                                                                                   | Reset |
|-------|------|-----------------------------|-----------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_V18_FAST_R<br>AMPING   | V18 output fast current ramping (improves response time at the cost of more ripple)           | 0x1   |
| 14:10 | R/W  | DCDC_V18_VOLTAGE            | V18 output voltage<br>$V = 1.2 \text{ V} + 25 \text{ mV} * N$                                 | 0x18  |
| 9:5   | R/W  | DCDC_V18_CUR_LI<br>M_MAX_HV | V18 output maximum current limit (high battery voltage mode)<br>$I = 30 \text{ mA} * (1 + N)$ | 0x1F  |
| 4:0   | R/W  | DCDC_V18_CUR_LI<br>M_MIN    | V18 output minimum current limit<br>$I = 30 \text{ mA} * (1 + N)$                             | 0x4   |

Table 205: DCDC\_V18\_1\_REG (0x5000008E)

| Bit   | Mode | Symbol                      | Description                                                                                                                                        | Reset |
|-------|------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_V18_ENABL<br>E_HV      | V18 output enable (high battery voltage mode)<br>0 = Disabled<br>1 = Enabled                                                                       | 0x1   |
| 14    | R/W  | DCDC_V18_ENABL<br>E_LV      | V18 output enable (low battery voltage mode)<br>0 = Disabled<br>1 = Enabled                                                                        | 0x0   |
| 13:10 | R/W  | DCDC_V18_CUR_LI<br>M_MAX_LV | V18 output maximum current limit low battery voltage mode)<br>$I = 30 \text{ mA} * (1 + N)$                                                        | 0xF   |
| 9:5   | R/W  | DCDC_V18_IDLE_H<br>YST      | V18 output idle time hysteresis<br>0 - 3875 ns, 125 ns step size<br>IDLE_MAX = IDLE_MIN + IDLE_HYST<br>Maximum idle time before decreasing CUR_LIM | 0x4   |
| 4:0   | R/W  | DCDC_V18_IDLE_MI<br>N       | V18 output minimum idle time<br>0 - 3875 ns, 125 ns step size<br>Minimum idle time, CUR_LIM is increased if this limit is not reached              | 0x10  |



Table 206: DCDC\_VDD\_0\_REG (0x50000090)

| Bit   | Mode | Symbol                  | Description                                                                                   | Reset |
|-------|------|-------------------------|-----------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_VDD_FAST_RAMPING   | VDD output fast current ramping (improves response time at the cost of more ripple)           | 0x1   |
| 14:10 | R/W  | DCDC_VDD_VOLTAGE        | VDD output voltage<br>$V = 0.8 \text{ V} + 25 \text{ mV} * N$                                 | 0x10  |
| 9:5   | R/W  | DCDC_VDD_CUR_LIM_MAX_HV | VDD output maximum current limit (high battery voltage mode)<br>$I = 30 \text{ mA} * (1 + N)$ | 0x18  |
| 4:0   | R/W  | DCDC_VDD_CUR_LIM_MIN    | VDD output minimum current limit<br>$I = 30 \text{ mA} * (1 + N)$                             | 0x4   |

Table 207: DCDC\_VDD\_1\_REG (0x50000092)

| Bit   | Mode | Symbol                  | Description                                                                                                                                             | Reset |
|-------|------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_VDD_ENABLER_HV     | VDD output enable (high battery voltage mode)<br>0 = Disabled<br>1 = Enabled                                                                            | 0x1   |
| 14    | R/W  | DCDC_VDD_ENABLER_LV     | VDD output enable (low battery voltage mode)<br>0 = Disabled<br>1 = Enabled                                                                             | 0x1   |
| 13:10 | R/W  | DCDC_VDD_CUR_LIM_MAX_LV | VDD output maximum current limit low battery voltage mode)<br>$I = 30 \text{ mA} * (1 + N)$                                                             | 0xB   |
| 9:5   | R/W  | DCDC_VDD_IDLE_HYST      | VDD output idle time hysteresis<br>0 - 3875 ns, 125 ns step size<br>$IDLE\_MAX = IDLE\_MIN + IDLE\_HYST$<br>Maximum idle time before decreasing CUR_LIM | 0x4   |
| 4:0   | R/W  | DCDC_VDD_IDLE_MIN       | VDD output minimum idle time<br>0 - 3875 ns, 125 ns step size<br>Minimum idle time, CUR_LIM is increased if this limit is not reached                   | 0x10  |

Table 208: DCDC\_V18P\_0\_REG (0x50000094)

| Bit   | Mode | Symbol                   | Description                                                                                    | Reset |
|-------|------|--------------------------|------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_V18P_FAST_RAMPING   | V18P output fast current ramping (improves response time at the cost of more ripple)           | 0x1   |
| 14:10 | R/W  | DCDC_V18P_VOLTAGE        | V18P output voltage<br>$V = 1.2 \text{ V} + 25 \text{ mV} * N$                                 | 0x18  |
| 9:5   | R/W  | DCDC_V18P_CUR_LIM_MAX_HV | V18P output maximum current limit (high battery voltage mode)<br>$I = 30 \text{ mA} * (1 + N)$ | 0x1F  |
| 4:0   | R/W  | DCDC_V18P_CUR_LIM_MIN    | V18P output minimum current limit<br>$I = 30 \text{ mA} * (1 + N)$                             | 0x4   |

Table 209: DCDC\_V18P\_1\_REG (0x50000096)

| Bit | Mode | Symbol              | Description                                                                   | Reset |
|-----|------|---------------------|-------------------------------------------------------------------------------|-------|
| 15  | R/W  | DCDC_V18P_ENABLE_HV | V18P output enable (high battery voltage mode)<br>0 = Disabled<br>1 = Enabled | 0x1   |

Table 209: DCDC\_V18P\_1\_REG (0x50000096)

| Bit   | Mode | Symbol                       | Description                                                                                                                                         | Reset |
|-------|------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 14    | R/W  | DCDC_V18P_ENAB<br>LE_LV      | V18P output enable (low battery voltage mode)<br>0 = Disabled<br>1 = Enabled                                                                        | 0x0   |
| 13:10 | R/W  | DCDC_V18P_CUR_<br>LIM_MAX_LV | V18P output maximum current limit low battery voltage mode)<br>$I = 30 \text{ mA} * (1 + N)$                                                        | 0xF   |
| 9:5   | R/W  | DCDC_V18P_IDLE_<br>HYST      | V18P output idle time hysteresis<br>0 - 3875 ns, 125 ns step size<br>IDLE_MAX = IDLE_MIN + IDLE_HYST<br>Maximum idle time before decreasing CUR_LIM | 0x4   |
| 4:0   | R/W  | DCDC_V18P_IDLE_<br>MIN       | V18P output minimum idle time<br>0 - 3875 ns, 125 ns step size<br>Minimum idle time, CUR_LIM is increased if this limit is not reached              | 0x10  |

Table 210: DCDC\_RET\_0\_REG (0x50000098)

| Bit   | Mode | Symbol                    | Description                                                           | Reset |
|-------|------|---------------------------|-----------------------------------------------------------------------|-------|
| 15:13 | R/W  | DCDC_V18P_RET_<br>CYCLES  | Charge cycles for V18P output in sleep mode<br>Cycles = $1 + 2 * N$   | 0x5   |
| 12:8  | R/W  | DCDC_V18P_CUR_<br>LIM_RET | V18P output sleep mode current limit<br>$I = 30 \text{ mA} * (1 + N)$ | 0xA   |
| 7:5   | R/W  | DCDC_VDD_RET_C<br>YCLES   | Charge cycles for VDD output in sleep mode<br>Cycles = $1 + 2 * N$    | 0x5   |
| 4:0   | R/W  | DCDC_VDD_CUR_LI<br>M_RET  | VDD output sleep mode current limit<br>$I = 30 \text{ mA} * (1 + N)$  | 0x6   |

Table 211: DCDC\_RET\_1\_REG (0x5000009A)

| Bit   | Mode | Symbol                   | Description                                                          | Reset |
|-------|------|--------------------------|----------------------------------------------------------------------|-------|
| 15:13 | R/W  | DCDC_V18_RET_C<br>YCLES  | Charge cycles for V18 output in sleep mode<br>Cycles = $1 + 2 * N$   | 0x5   |
| 12:8  | R/W  | DCDC_V18_CUR_LI<br>M_RET | V18 output sleep mode current limit<br>$I = 30 \text{ mA} * (1 + N)$ | 0xA   |
| 7:5   | R/W  | DCDC_V14_RET_C<br>YCLES  | Charge cycles for V14 output in sleep mode<br>Cycles = $1 + 2 * N$   | 0x2   |
| 4:0   | R/W  | DCDC_V14_CUR_LI<br>M_RET | V14 output sleep mode current limit<br>$I = 30 \text{ mA} * (1 + N)$ | 0x6   |

Table 212: DCDC\_TRIM\_REG (0x5000009C)

| Bit  | Mode | Symbol                   | Description                                                                                                                                  | Reset |
|------|------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 13   | R/W  | DCDC_P_COMP_M<br>AN_TRIM | Trim mode for P side comparator<br>0 = Automatic<br>1 = Manual                                                                               | 0x0   |
| 12:7 | R/W  | DCDC_P_COMP_TR<br>IM     | Manual trim value for P side comparator<br>Signed magnitude representation<br>011111 = +47 mV<br>000000 = 100000 = +16 mV<br>111111 = -15 mV | 0x0   |

Table 212: DCDC\_TRIM\_REG (0x5000009C)

| Bit | Mode | Symbol                   | Description                                                                                                                                  | Reset |
|-----|------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | DCDC_N_COMP_M<br>AN_TRIM | Trim mode for N side comparator<br>0 = Automatic<br>1 = Manual                                                                               | 0x0   |
| 5:0 | R/W  | DCDC_N_COMP_TR<br>IM     | Manual trim value for N side comparator<br>Signed magnitude representation<br>011111 = +13 mV<br>000000 = 100000 = -22 mV<br>111111 = -56 mV | 0x0   |

Table 213: DCDC\_TEST\_0\_REG (0x5000009E)

| Bit   | Mode | Symbol                  | Description                                                                                                                                                           | Reset |
|-------|------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | DCDC_FORCE_CO<br>MP_CLK | Disables automatic comparator clock, clock lines values<br>based on DCDC_COMP_CLK                                                                                     | 0x0   |
| 14    | R/W  | DCDC_FORCE_CUR<br>RENT  | Force output current setting                                                                                                                                          | 0x0   |
| 13:11 | R/W  | DCDC_OUTPUT_M<br>ONITOR | Output monitor switch (connect to ADC)<br>000 = None<br>001 = V14<br>010 = V18<br>011 = VDD<br>100 = VPA<br>101 = None<br>110 = None<br>111 = None                    | 0x0   |
| 10:8  | R/W  | DCDC_ANA_TEST           | Analog test bus<br>000 = None<br>001 = High side ground<br>010 = Low side supply<br>011 = 1.2 V buffer output<br>100 = None<br>101 = None<br>110 = None<br>111 = None | 0x0   |
| 7     | R/W  | DCDC_FORCE_IDL<br>E     | Force idle mode                                                                                                                                                       | 0x0   |
| 6     | R/W  | DCDC_FORCE_V18<br>P     | Force V18P switch on                                                                                                                                                  | 0x0   |
| 5     | R/W  | DCDC_FORCE_VDD          | Force VDD switch on                                                                                                                                                   | 0x0   |
| 4     | R/W  | DCDC_FORCE_V18          | Force V18 switch on                                                                                                                                                   | 0x0   |
| 3     | R/W  | DCDC_FORCE_V14          | Force V14 switch on                                                                                                                                                   | 0x0   |
| 2     | R/W  | DCDC_FORCE_FW           | Force FW switch on                                                                                                                                                    | 0x0   |
| 1     | R/W  | DCDC_FORCE_NS<br>W      | Force N switch on                                                                                                                                                     | 0x0   |
| 0     | R/W  | DCDC_FORCE_PS<br>W      | Force P switch on                                                                                                                                                     | 0x0   |

Table 214: DCDC\_TEST\_1\_REG (0x500000A0)

| Bit  | Mode | Symbol        | Description                                                                                               | Reset |
|------|------|---------------|-----------------------------------------------------------------------------------------------------------|-------|
| 12:9 | R/W  | DCDC_COMP_CLK | Forced clock values for [COMP_VPA, COMP_VDD,<br>COMP_V18, COMP_V14] (requires<br>DCDC_FORCE_COMP_CLK = 1) | 0x0   |

Table 214: DCDC\_TEST\_1\_REG (0x500000A0)

| Bit | Mode | Symbol                | Description                                                                                                                                                                                                                                                                                       | Reset |
|-----|------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 8:4 | R/W  | DCDC_TEST_CURR<br>ENT | Current limit setting when current limit is forced                                                                                                                                                                                                                                                | 0x0   |
| 3:0 | R/W  | DCDC_TEST_REG         | Determines which register appears on the testbus<br>0x0 = DCDC_NONE<br>0x1 = DCDC_STATUS_0<br>0x2 = DCDC_STATUS_1<br>0x3 = DCDC_STATUS_2<br>0x4 = DCDC_STATUS_3<br>0x5 = DCDC_STATUS_4<br>0x6 = DCDC_TRIM_0<br>0x7 = DCDC_TRIM_1<br>0x8 = DCDC_TRIM_2<br>0x9 = DCDC_TRIM_3<br>0xA-0xF = DCDC_NONE | 0x0   |

Table 215: DCDC\_STATUS\_0\_REG (0x500000A2)

| Bit  | Mode | Symbol                | Description                | Reset |
|------|------|-----------------------|----------------------------|-------|
| 11:9 | R    | DCDC_CHARGE_RE<br>G_3 | Charge register position 3 | 0x0   |
| 8:6  | R    | DCDC_CHARGE_RE<br>G_2 | Charge register position 2 | 0x0   |
| 5:3  | R    | DCDC_CHARGE_RE<br>G_1 | Charge register position 1 | 0x0   |
| 2:0  | R    | DCDC_CHARGE_RE<br>G_0 | Charge register position 0 | 0x0   |

Table 216: DCDC\_STATUS\_1\_REG (0x500000A4)

| Bit | Mode | Symbol                  | Description                                                                                                                           | Reset |
|-----|------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------|-------|
| 11  | R    | DCDC_V18P_AVAIL<br>ABLE | Indicates whether V18P is available<br>Requires that converter is enabled, output is enabled and<br>V_OK and V_NOK have both occurred | 0x0   |
| 10  | R    | DCDC_VDD_AVAILA<br>BLE  | Indicates whether VDD is available<br>Requires that converter is enabled, output is enabled and<br>V_OK and V_NOK have both occurred  | 0x0   |
| 9   | R    | DCDC_V18_AVAILA<br>BLE  | Indicates whether V18 is available<br>Requires that converter is enabled, output is enabled and<br>V_OK and V_NOK have both occurred  | 0x0   |
| 8   | R    | DCDC_V14_AVAILA<br>BLE  | Indicates whether V14 is available<br>Requires that converter is enabled, output is enabled and<br>V_OK and V_NOK have both occurred  | 0x0   |
| 7   | R    | DCDC_V18P_OK            | OK output of V18P comparator                                                                                                          | 0x0   |
| 6   | R    | DCDC_VDD_OK             | OK output of VDD comparator                                                                                                           | 0x0   |
| 5   | R    | DCDC_V18_OK             | OK output of V18 comparator                                                                                                           | 0x0   |
| 4   | R    | DCDC_V14_OK             | OK output of V14 comparator                                                                                                           | 0x0   |
| 3   | R    | DCDC_V18P_NOK           | NOK output of V18P comparator                                                                                                         | 0x0   |
| 2   | R    | DCDC_VDD_NOK            | NOK output of VDD comparator                                                                                                          | 0x0   |
| 1   | R    | DCDC_V18_NOK            | NOK output of V18 comparator                                                                                                          | 0x0   |
| 0   | R    | DCDC_V14_NOK            | NOK output of V14 comparator                                                                                                          | 0x0   |

Table 217: DCDC\_STATUS\_2\_REG (0x500000A6)

| Bit | Mode | Symbol             | Description                                   | Reset |
|-----|------|--------------------|-----------------------------------------------|-------|
| 11  | R    | DCDC_V18P_SW_STATE | DCDC state machine V18P output                | 0x0   |
| 10  | R    | DCDC_VDD_SW_STATE  | DCDC state machine VDD output                 | 0x0   |
| 9   | R    | DCDC_V18_SW_STATE  | DCDC state machine V18 output                 | 0x0   |
| 8   | R    | DCDC_V14_SW_STATE  | DCDC state machine V14 output                 | 0x0   |
| 7   | R    | DCDC_NSW_STATE     | DCDC state machine NSW output                 | 0x0   |
| 6   | R    | DCDC_PSW_STATE     | DCDC state machine PSW output                 | 0x0   |
| 5   | R    | DCDC_P_COMP_P      | DCDC P side dynamic comparator P output       | 0x0   |
| 4   | R    | DCDC_P_COMP_N      | DCDC P side dynamic comparator N output       | 0x0   |
| 3   | R    | DCDC_N_COMP_P      | DCDC N side dynamic comparator P output       | 0x0   |
| 2   | R    | DCDC_N_COMP_N      | DCDC N side dynamic comparator N output       | 0x0   |
| 1   | R    | DCDC_P_COMP        | DCDC P side continuous time comparator output | 0x0   |
| 0   | R    | DCDC_N_COMP        | DCDC N side continuous time comparator output | 0x0   |

Table 218: DCDC\_STATUS\_3\_REG (0x500000A8)

| Bit | Mode | Symbol          | Description                                               | Reset |
|-----|------|-----------------|-----------------------------------------------------------|-------|
| 10  | R    | DCDC_LV_MODE    | Indicates if the converter is in low battery voltage mode | 0x0   |
| 9:5 | R    | DCDC_I_LIM_V18P | Actual V18P current limit                                 | 0x4   |
| 4:0 | R    | DCDC_I_LIM_VDD  | Actual VDD current limit                                  | 0x4   |

Table 219: DCDC\_STATUS\_4\_REG (0x500000AA)

| Bit | Mode | Symbol         | Description              | Reset |
|-----|------|----------------|--------------------------|-------|
| 9:5 | R    | DCDC_I_LIM_V18 | Actual V18 current limit | 0x4   |
| 4:0 | R    | DCDC_I_LIM_V14 | Actual V14 current limit | 0x4   |

Table 220: DCDC\_TRIM\_0\_REG (0x500000AC)

| Bit  | Mode | Symbol          | Description                                                                                                                                     | Reset |
|------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 11:6 | R    | DCDC_V14_TRIM_P | P comparator trim value when V14 is active<br>Signed magnitude representation<br>011111 = +47 mV<br>000000 = 100000 = +16 mV<br>111111 = -15 mV | 0x0   |
| 5:0  | R    | DCDC_V14_TRIM_N | N comparator trim value when V14 is active<br>Signed magnitude representation<br>011111 = +13 mV<br>000000 = 100000 = -22 mV<br>111111 = -56 mV | 0x0   |

Table 221: DCDC\_TRIM\_1\_REG (0x500000AE)

| Bit  | Mode | Symbol          | Description                                                                                                                                     | Reset |
|------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 11:6 | R    | DCDC_V18_TRIM_P | P comparator trim value when V18 is active<br>Signed magnitude representation<br>011111 = +47 mV<br>000000 = 100000 = +16 mV<br>111111 = -15 mV | 0x0   |
| 5:0  | R    | DCDC_V18_TRIM_N | N comparator trim value when V18 is active<br>Signed magnitude representation<br>011111 = +13 mV<br>000000 = 100000 = -22 mV<br>111111 = -56 mV | 0x0   |

Table 222: DCDC\_TRIM\_2\_REG (0x500000B0)

| Bit  | Mode | Symbol          | Description                                                                                                                                     | Reset |
|------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 11:6 | R    | DCDC_VDD_TRIM_P | P comparator trim value when VDD is active<br>Signed magnitude representation<br>011111 = +47 mV<br>000000 = 100000 = +16 mV<br>111111 = -15 mV | 0x0   |
| 5:0  | R    | DCDC_VDD_TRIM_N | N comparator trim value when VDD is active<br>Signed magnitude representation<br>011111 = +13 mV<br>000000 = 100000 = -22 mV<br>111111 = -56 mV | 0x0   |

Table 223: DCDC\_TRIM\_3\_REG (0x500000B2)

| Bit  | Mode | Symbol           | Description                                                                                                                                      | Reset |
|------|------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 11:6 | R    | DCDC_V18P_TRIM_P | P comparator trim value when V18P is active<br>Signed magnitude representation<br>011111 = +47 mV<br>000000 = 100000 = +16 mV<br>111111 = -15 mV | 0x0   |
| 5:0  | R    | DCDC_V18P_TRIM_N | N comparator trim value when V18P is active<br>Signed magnitude representation<br>011111 = +13 mV<br>000000 = 100000 = -22 mV<br>111111 = -56 mV | 0x0   |

Table 224: DCDC\_IRQ\_STATUS\_REG (0x500000B4)

| Bit | Mode | Symbol                       | Description                                                | Reset |
|-----|------|------------------------------|------------------------------------------------------------|-------|
| 4   | R    | DCDC_BROWN_OUT_IRQ_STATUS    | Brown out detector triggered (battery voltage below 2.5 V) | 0x0   |
| 3   | R    | DCDC_V18P_TIMEOUT_IRQ_STATUS | Timeout occurred on V18P output                            | 0x0   |
| 2   | R    | DCDC_VDD_TIMEOUT_IRQ_STATUS  | Timeout occurred on VDD output                             | 0x0   |
| 1   | R    | DCDC_V18_TIMEOUT_IRQ_STATUS  | Timeout occurred on V18 output                             | 0x0   |
| 0   | R    | DCDC_V14_TIMEOUT_IRQ_STATUS  | Timeout occurred on V14 output                             | 0x0   |

Table 225: DCDC\_IRQ\_CLEAR\_REG (0x500000B6)

| Bit | Mode | Symbol                      | Description                  | Reset |
|-----|------|-----------------------------|------------------------------|-------|
| 4   | R0/W | DCDC_BROWN_OUT_IRQ_CLEAR    | Clear brown out interrupt    | 0x0   |
| 3   | R0/W | DCDC_V18P_TIMEOUT_IRQ_CLEAR | Clear V18P timeout interrupt | 0x0   |
| 2   | R0/W | DCDC_VDD_TIMEOUT_IRQ_CLEAR  | Clear VDD timeout interrupt  | 0x0   |
| 1   | R0/W | DCDC_V18_TIMEOUT_IRQ_CLEAR  | Clear V18 timeout interrupt  | 0x0   |
| 0   | R0/W | DCDC_V14_TIMEOUT_IRQ_CLEAR  | Clear V14 timeout interrupt  | 0x0   |

Table 226: DCDC\_IRQ\_MASK\_REG (0x500000B8)

| Bit | Mode | Symbol                     | Description                 | Reset |
|-----|------|----------------------------|-----------------------------|-------|
| 4   | R/W  | DCDC_BROWN_OUT_IRQ_MASK    | Mask brown out interrupt    | 0x0   |
| 3   | R/W  | DCDC_V18P_TIMEOUT_IRQ_MASK | Mask V18P timeout interrupt | 0x0   |
| 2   | R/W  | DCDC_VDD_TIMEOUT_IRQ_MASK  | Mask VDD timeout interrupt  | 0x0   |
| 1   | R/W  | DCDC_V18_TIMEOUT_IRQ_MASK  | Mask V18 timeout interrupt  | 0x0   |
| 0   | R/W  | DCDC_V14_TIMEOUT_IRQ_MASK  | Mask V14 timeout interrupt  | 0x0   |



## 37.8 WAKEUP REGISTER FILE

Table 227: Register map WakeUp

| Address    | Port                | Description                                               |
|------------|---------------------|-----------------------------------------------------------|
| 0x50000100 | WKUP_CTRL_REG       | Control register for the wakeup counter                   |
| 0x50000102 | WKUP_COMPARE_REG    | Number of events before wakeup interrupt                  |
| 0x50000104 | WKUP_RESET_IRQ_REG  | Reset wakeup interrupt                                    |
| 0x50000106 | WKUP_COUNTER_REG    | Actual number of events of the wakeup counter             |
| 0x50000108 | WKUP_RESET_CNTR_REG | Reset the event counter                                   |
| 0x5000010A | WKUP_SELECT_P0_REG  | select which inputs from P0 port can trigger wkup counter |
| 0x5000010C | WKUP_SELECT_P1_REG  | select which inputs from P1 port can trigger wkup counter |
| 0x5000010E | WKUP_SELECT_P2_REG  | select which inputs from P2 port can trigger wkup counter |
| 0x50000110 | WKUP_SELECT_P3_REG  | select which inputs from P3 port can trigger wkup counter |
| 0x50000112 | WKUP_SELECT_P4_REG  | select which inputs from P3 port can trigger wkup counter |
| 0x50000114 | WKUP_POL_P0_REG     | select the sesitivity polarity for each P0 input          |
| 0x50000116 | WKUP_POL_P1_REG     | select the sesitivity polarity for each P1 input          |
| 0x50000118 | WKUP_POL_P2_REG     | select the sesitivity polarity for each P2 input          |
| 0x5000011A | WKUP_POL_P3_REG     | select the sesitivity polarity for each P3 input          |
| 0x5000011C | WKUP_POL_P4_REG     | select the sesitivity polarity for each P3 input          |

Table 228: WKUP\_CTRL\_REG (0x50000100)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                | Reset |
|------|------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -               | Reserved                                                                                                                                                                   | 0x0   |
| 7    | R/W  | WKUP_ENABLE_IRQ | 0: no interrupt will be enabled<br>1: if the event counter reaches the value set by WKUP_COMPARE_REG an IRQ will be generated                                              | 0x0   |
| 6    | R/W  | WKUP_SFT_KEYHIT | 0 = no effect<br><br>1 = emulate key hit. The event counter will increment by 1 (after debouncing if enabled). First make this bit 0 before any new key hit can be sensed. | 0x0   |
| 5:0  | R/W  | WKUP_DEB_VALUE  | Keyboard debounce time. If set to 0, no debouncing will be done.<br>Debounce time: $N \cdot 1 \text{ ms}$ . $N = 1..63$<br>Keyboard Debounce Time                          | 0x0   |

Table 229: WKUP\_COMPARE\_REG (0x50000102)

| Bit  | Mode | Symbol  | Description                                                                            | Reset |
|------|------|---------|----------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -       | Reserved                                                                               | 0x0   |
| 7:0  | R/W  | COMPARE | The number of events that have to be counted before the wakeup interrupt will be given | 0x0   |

Table 230: WKUP\_RESET\_IRQ\_REG (0x50000104)

| Bit  | Mode | Symbol       | Description                                                                             | Reset |
|------|------|--------------|-----------------------------------------------------------------------------------------|-------|
| 15:0 | W    | WKUP_IRQ_RST | writing any value to this register will reset the interrupt. read-ing always returns 0. | 0x0   |

Table 231: WKUP\_COUNTER\_REG (0x50000106)

| Bit  | Mode | Symbol      | Description                                                                                                            | Reset |
|------|------|-------------|------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                               | 0x0   |
| 7:0  | R    | EVENT_VALUE | This value represents the number of events that have been counted so far. It will be reset by resetting the interrupt. | 0x0   |

Table 232: WKUP\_RESET\_CNTR\_REG (0x50000108)

| Bit  | Mode | Symbol        | Description                                                     | Reset |
|------|------|---------------|-----------------------------------------------------------------|-------|
| 15:0 | W    | WKUP_CNTR_RST | writing any value to this register will reset the event counter | 0x0   |

Table 233: WKUP\_SELECT\_P0\_REG (0x5000010A)

| Bit | Mode | Symbol         | Description                                                                                              | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_SELECT_P0 | 0: input P0x is not enabled for wakeup event counter<br>1: input P0x is enabled for wakeup event counter | 0x0   |

Table 234: WKUP\_SELECT\_P1\_REG (0x5000010C)

| Bit | Mode | Symbol         | Description                                                                                              | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_SELECT_P1 | 0: input P1x is not enabled for wakeup event counter<br>1: input P1x is enabled for wakeup event counter | 0x0   |

Table 235: WKUP\_SELECT\_P2\_REG (0x5000010E)

| Bit | Mode | Symbol         | Description                                                                                              | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------|-------|
| 4:0 | R/W  | WKUP_SELECT_P2 | 0: input P2x is not enabled for wakeup event counter<br>1: input P2x is enabled for wakeup event counter | 0x0   |

Table 236: WKUP\_SELECT\_P3\_REG (0x50000110)

| Bit | Mode | Symbol         | Description                                                                                              | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_SELECT_P3 | 0: input P3x is not enabled for wakeup event counter<br>1: input P3x is enabled for wakeup event counter | 0x0   |

Table 237: WKUP\_SELECT\_P4\_REG (0x50000112)

| Bit | Mode | Symbol         | Description                                                                                              | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_SELECT_P4 | 0: input P4x is not enabled for wakeup event counter<br>1: input P4x is enabled for wakeup event counter | 0x0   |

Table 238: WKUP\_POL\_P0\_REG (0x50000114)

| Bit | Mode | Symbol      | Description                                                                                                                                                   | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_POL_P0 | 0: enabled input P0x will increment the event counter if that input goes high<br>1: enabled input P0x will increment the event counter if that input goes low | 0x0   |

Table 239: WKUP\_POL\_P1\_REG (0x50000116)

| Bit | Mode | Symbol      | Description                                                                                                                                                   | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_POL_P1 | 0: enabled input P1x will increment the event counter if that input goes high<br>1: enabled input P1x will increment the event counter if that input goes low | 0x0   |

Table 240: WKUP\_POL\_P2\_REG (0x50000118)

| Bit | Mode | Symbol      | Description                                                                                                                                                   | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4:0 | R/W  | WKUP_POL_P2 | 0: enabled input P2x will increment the event counter if that input goes high<br>1: enabled input P2x will increment the event counter if that input goes low | 0x0   |

Table 241: WKUP\_POL\_P3\_REG (0x5000011A)

| Bit | Mode | Symbol      | Description                                                                                                                                                   | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_POL_P3 | 0: enabled input P3x will increment the event counter if that input goes high<br>1: enabled input P3x will increment the event counter if that input goes low | 0x0   |

Table 242: WKUP\_POL\_P4\_REG (0x5000011C)

| Bit | Mode | Symbol      | Description                                                                                                                                                   | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | WKUP_POL_P4 | 0: enabled input P4x will increment the event counter if that input goes high<br>1: enabled input P4x will increment the event counter if that input goes low | 0x0   |

## 37.9 TIMER1 REGISTER FILE

Table 243: Register map Timer1

| Address    | Port                     | Description                                                                                                             |
|------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 0x50000200 | CAPTIM_CTRL_REG          | Capture Timer control register                                                                                          |
| 0x50000202 | CAPTIM_TIMER_VAL_REG     | Capture Timer counter value                                                                                             |
| 0x50000204 | CAPTIM_STATUS_REG        | Capture Timer status register                                                                                           |
| 0x50000206 | CAPTIM_GPIO1_CONF_REG    | Capture Timer gpio1 selection                                                                                           |
| 0x50000208 | CAPTIM_GPIO2_CONF_REG    | Capture Timer gpio2 selection                                                                                           |
| 0x5000020A | CAPTIM_RELOAD_REG        | Capture Timer reload value and Delay in shot mode                                                                       |
| 0x5000020C | CAPTIM_SHOTWIDTH_REG     | Capture Timer Shot duration in shot mode                                                                                |
| 0x5000020E | CAPTIM_PRESCALER_REG     | Capture Timer prescaler value                                                                                           |
| 0x50000210 | CAPTIM_CAPTURE_GPIO1_REG | Capture Timer value for event on GPIO1                                                                                  |
| 0x50000212 | CAPTIM_CAPTURE_GPIO2_REG | Capture Timer value for event on GPIO2                                                                                  |
| 0x50000214 | CAPTIM_PRESCALER_VAL_REG | Capture Timer interrupt status register                                                                                 |
| 0x50000216 | CAPTIM_PWM_FREQ_REG      | Capture Timer pwm frequency register. PWM5 period is defined by the reference clock frequency multiplied by this value. |
| 0x50000218 | CAPTIM_PWM_DC_REG        | Capture Timer pwm dc register                                                                                           |

Table 244: CAPTIM\_CTRL\_REG (0x50000200)

| Bit | Mode | Symbol                   | Description                                                                                                                          | Reset |
|-----|------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7   | R/W  | CAPTIM_SYS_CLK_EN        | '1' When Capture Timer use the system clock else use the clock 32KHz                                                                 | 0x0   |
| 6   | R/W  | CAPTIM_FREE_RUN_MODE_EN  | Only when timer counts up, if it is '1' timer does not zero when reaches to reload value. it is zero only when it has the max value. | 0x0   |
| 5   | R/W  | CAPTIM_IRQ_EN            | '1' When Capture timer IRQ unmask, '0' masked                                                                                        | 0x0   |
| 4   | R/W  | CAPTIM_IN2_EVENT_FALL_EN | '1' When Input1 event type is falling edge, '0' rising edge                                                                          | 0x0   |
| 3   | R/W  | CAPTIM_IN1_EVENT_FALL_EN | '1' When Input2 event type is falling edge, '0' rising edge                                                                          | 0x0   |
| 2   | R/W  | CAPTIM_COUNT_DOWN_EN     | '1' when timer counts down, '0' count up                                                                                             | 0x0   |
| 1   | R/W  | CAPTIM_ONESHOT_MODE_EN   | '1' Capture Timer in OneShot mode, '0' Capture/Timer mode                                                                            | 0x0   |
| 0   | R/W  | CAPTIM_EN                | '1' Capture Timer enabled, else disabled                                                                                             | 0x0   |

Table 245: CAPTIM\_TIMER\_VAL\_REG (0x50000202)

| Bit  | Mode | Symbol             | Description                   | Reset |
|------|------|--------------------|-------------------------------|-------|
| 15:0 | R    | CAPTIM_TIMER_VALUE | Gives the current timer value | 0x0   |

Table 246: CAPTIM\_STATUS\_REG (0x50000204)

| Bit | Mode | Symbol               | Description                                                         | Reset |
|-----|------|----------------------|---------------------------------------------------------------------|-------|
| 3:2 | R    | CAPTIM_ONESHOT_PHASE | 0 : Wait for event, 1 : Delay phase, 2 : Start Shot, 3 : Shot phase | 0x0   |
| 1   | R    | CAPTIM_IN2_STATE     | Gives the logic level of the IN1                                    | 0x0   |
| 0   | R    | CAPTIM_IN1_STATE     | Gives the logic level of the IN2                                    | 0x0   |

Table 247: CAPTIM\_GPIO1\_CONF\_REG (0x50000206)

| Bit | Mode | Symbol            | Description                                                                                        | Reset |
|-----|------|-------------------|----------------------------------------------------------------------------------------------------|-------|
| 5:0 | R/W  | CAPTIM_GPIO1_CONF | Select one of the 37 GPIOs as IN1, Valid value 0-37. 1 for P00 .. 37 for P47. When 0 Disable input | 0x0   |

Table 248: CAPTIM\_GPIO2\_CONF\_REG (0x50000208)

| Bit | Mode | Symbol            | Description                                                                                        | Reset |
|-----|------|-------------------|----------------------------------------------------------------------------------------------------|-------|
| 5:0 | R/W  | CAPTIM_GPIO2_CONF | Select one of the 37 GPIOs as IN2, Valid value 0-37. 1 for P00 .. 37 for P47. When 0 Disable input | 0x0   |

Table 249: CAPTIM\_RELOAD\_REG (0x5000020A)

| Bit  | Mode | Symbol        | Description                                                                                                                                            | Reset |
|------|------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | CAPTIM_RELOAD | Reload or max value in timer mode, Delay phase duration in oneshot mode. Actual delay is the register value plus synchronization time (3 clock cycles) | 0x0   |

Table 250: CAPTIM\_SHOTWIDTH\_REG (0x5000020C)

| Bit  | Mode | Symbol           | Description                         | Reset |
|------|------|------------------|-------------------------------------|-------|
| 15:0 | R/W  | CAPTIM_SHOTWIDTH | Shot phase duration in oneshot mode | 0x0   |

Table 251: CAPTIM\_PRESCALER\_REG (0x5000020E)

| Bit  | Mode | Symbol           | Description                                                                               | Reset |
|------|------|------------------|-------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | CAPTIM_PRESCALER | Define the timer count frequency. $\text{Freq} = \text{Freq\_clock} / (\text{value} + 1)$ | 0x0   |

Table 252: CAPTIM\_CAPTURE\_GPIO1\_REG (0x50000210)

| Bit  | Mode | Symbol               | Description                               | Reset |
|------|------|----------------------|-------------------------------------------|-------|
| 15:0 | R    | CAPTIM_CAPTURE_GPIO1 | Gives the Capture time for event on GPIO1 | 0x0   |

Table 253: CAPTIM\_CAPTURE\_GPIO2\_REG (0x50000212)

| Bit  | Mode | Symbol               | Description                               | Reset |
|------|------|----------------------|-------------------------------------------|-------|
| 15:0 | R    | CAPTIM_CAPTURE_GPIO2 | Gives the Capture time for event on GPIO2 | 0x0   |

Table 254: CAPTIM\_PRESCALER\_VAL\_REG (0x50000214)

| Bit  | Mode | Symbol               | Description                       | Reset |
|------|------|----------------------|-----------------------------------|-------|
| 15:0 | R    | CAPTIM_PRESCALER_VAL | Gives the current prescaler value | 0x0   |

Table 255: CAPTIM\_PWM\_FREQ\_REG (0x50000216)

| Bit  | Mode | Symbol          | Description                                                 | Reset |
|------|------|-----------------|-------------------------------------------------------------|-------|
| 15:0 | R/W  | CAPTIM_PWM_FREQ | Define the PWM frequency. = prescaler frequency / (value+1) | 0x0   |

Table 256: CAPTIM\_PWM\_DC\_REG (0x50000218)

| Bit  | Mode | Symbol        | Description                                       | Reset |
|------|------|---------------|---------------------------------------------------|-------|
| 15:0 | R/W  | CAPTIM_PWM_DC | Define the PWM duty cycle = pwm_dc / (pwm_freq+1) | 0x0   |

## 37.10 UART REGISTER FILE

Table 257: Register map UART

| Address    | Port                  | Description                                             |
|------------|-----------------------|---------------------------------------------------------|
| 0x50001000 | UART_RBR_THR_DLL_REG  | Receive Buffer Register                                 |
| 0x50001004 | UART_IER_DLH_REG      | Interrupt Enable Register                               |
| 0x50001008 | UART_IIR_FCR_REG      | Interrupt Identification Register                       |
| 0x5000100C | UART_LCR_REG          | Line Control Register                                   |
| 0x50001010 | UART_MCR_REG          | Modem Control Register                                  |
| 0x50001014 | UART_LSR_REG          | Line Status Register                                    |
| 0x5000101C | UART_SCR_REG          | Scratchpad Register                                     |
| 0x5000107C | UART_USR_REG          | UART Status register.                                   |
| 0x50001088 | UART_SRR_REG          | Software Reset Register.                                |
| 0x50001090 | UART_SBCR_REG         | Shadow Break Control Register                           |
| 0x500010A8 | UART_DMA_SA_REG       | DMA Software Acknowledge                                |
| 0x500010C0 | UART_DLF_REG          | Divisor Latch Fraction Register                         |
| 0x500010F4 | UART_CPR_REG          | Component Parameter Register                            |
| 0x500010F8 | UART_UCV_REG          | Component Version                                       |
| 0x500010FC | UART_CTR_REG          | Component Type Register                                 |
| 0x50001100 | UART2_RBR_THR_DLL_REG | Receive Buffer Register                                 |
| 0x50001104 | UART2_IER_DLH_REG     | Interrupt Enable Register                               |
| 0x50001108 | UART2_IIR_FCR_REG     | Interrupt Identification Register/FIFO Control Register |
| 0x5000110C | UART2_LCR_REG         | Line Control Register                                   |
| 0x50001110 | UART2_MCR_REG         | Modem Control Register                                  |
| 0x50001114 | UART2_LSR_REG         | Line Status Register                                    |
| 0x50001118 | UART2_MSR_REG         | Modem Status Register                                   |
| 0x5000111C | UART2_SCR_REG         | Scratchpad Register                                     |
| 0x50001130 | UART2_SRBR_STHR0_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001134 | UART2_SRBR_STHR1_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001138 | UART2_SRBR_STHR2_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x5000113C | UART2_SRBR_STHR3_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001140 | UART2_SRBR_STHR4_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001144 | UART2_SRBR_STHR5_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001148 | UART2_SRBR_STHR6_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x5000114C | UART2_SRBR_STHR7_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001150 | UART2_SRBR_STHR8_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001154 | UART2_SRBR_STHR9_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001158 | UART2_SRBR_STHR10_REG | Shadow Receive/Transmit Buffer Register                 |
| 0x5000115C | UART2_SRBR_STHR11_REG | Shadow Receive/Transmit Buffer Register                 |
| 0x50001160 | UART2_SRBR_STHR12_REG | Shadow Receive/Transmit Buffer Register                 |
| 0x50001164 | UART2_SRBR_STHR13_REG | Shadow Receive/Transmit Buffer Register                 |
| 0x50001168 | UART2_SRBR_STHR14_REG | Shadow Receive/Transmit Buffer Register                 |
| 0x5000116C | UART2_SRBR_STHR15_REG | Shadow Receive/Transmit Buffer Register                 |
| 0x50001170 | UART2_FAR_REG         | FIFO Access Register                                    |
| 0x5000117C | UART2_USR_REG         | UART Status register.                                   |
| 0x50001180 | UART2_TFL_REG         | Transmit FIFO Level                                     |
| 0x50001184 | UART2_RFL_REG         | Receive FIFO Level.                                     |



Table 257: Register map UART

| Address    | Port            | Description                     |
|------------|-----------------|---------------------------------|
| 0x50001188 | UART2_SRR_REG   | Software Reset Register.        |
| 0x5000118C | UART2_SRTS_REG  | Shadow Request to Send          |
| 0x50001190 | UART2_SBCR_REG  | Shadow Break Control Register   |
| 0x50001194 | UART2_SDMAM_REG | Shadow DMA Mode                 |
| 0x50001198 | UART2_SFE_REG   | Shadow FIFO Enable              |
| 0x5000119C | UART2_SRT_REG   | Shadow RCVR Trigger             |
| 0x500011A0 | UART2_STET_REG  | Shadow TX Empty Trigger         |
| 0x500011A4 | UART2_HTX_REG   | Halt TX                         |
| 0x500011A8 | UART2_DMASA_REG | DMA Software Acknowledge        |
| 0x500011C0 | UART2_DLF_REG   | Divisor Latch Fraction Register |
| 0x500011F4 | UART2_CPR_REG   | Component Parameter Register    |
| 0x500011F8 | UART2_UCV_REG   | Component Version               |
| 0x500011FC | UART2_CTR_REG   | Component Type Register         |

Table 258: UART\_RBR\_THR\_DLL\_REG (0x50001000)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 258: UART\_RBR\_THR\_DLL\_REG (0x50001000)

| Bit | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | RBR_THR_DLL | <p>Receive Buffer Register: (RBR).<br/>This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. The data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error.</p> <p>Transmit Holding Register: (THR)<br/>This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. Writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten.</p> <p>Divisor Latch (Low): (DLL)<br/>This register makes up the lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows:<br/> <math display="block">\text{baud rate} = (\text{serial clock freq}) / (16 * \text{divisor})</math> Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLL is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.</p> <p>Divisor Latch (Low): (DLH) (Note: This register is placed in UART_IER_DLH_REG with offset 0x4)<br/>Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock frequency divided by sixteen times the value of the baud rate divisor, as follows:<br/> <math display="block">\text{baud rate} = (\text{serial clock freq}) / (16 * \text{divisor}).</math> Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.</p> | 0x0   |

Table 259: UART\_IER\_DLH\_REG (0x50001004)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                  | Reset |
|------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -          | Reserved                                                                                                                                                                                                                     | 0x0   |
| 7    | R/W  | PTIME_DLH7 | Interrupt Enable Register: PTIME, Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[7] of the 8 bit DLH register. | 0x0   |
| 6:4  | -    | -          | Reserved                                                                                                                                                                                                                     | 0x0   |
| 3    | -    | -          | Reserved                                                                                                                                                                                                                     | 0x0   |

Table 259: UART\_IER\_DLH\_REG (0x50001004)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                       | Reset |
|-----|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2   | R/W  | ELSI_DHL2  | Interrupt Enable Register: ELSI, Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[2] of the 8 bit DLH register.                                 | 0x0   |
| 1   | R/W  | ETBEI_DLH1 | Interrupt Enable Register: ETBEI, Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[1] of the 8 bit DLH register. | 0x0   |
| 0   | R/W  | ERBFI_DLH0 | Interrupt Enable Register: ERBFI, Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt. These are the second highest priority interrupts. 0 = disabled 1 = enabled Divisor Latch (High): Bit[0] of the 8 bit DLH register.                | 0x0   |

Table 260: UART\_IIR\_FCR\_REG (0x50001008)

| Bit  | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                          | Reset |
|------|------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | IIR_FCR | Interrupt Identification Register: Bits[7:6], returns 00. Bits[3:0], Interrupt ID (or IID): This indicates the highest priority pending interrupt which can be one of the following types: 0001 = no interrupt pending. 0010 = THR empty. 0100 = received data available. 0110 = receiver line status. 0111 = busy detect. 1100 = character timeout. | 0x0   |

Table 261: UART\_LCR\_REG (0x5000100C)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 7    | R/W  | UART_DLAB | Divisor Latch Access Bit.<br>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.<br>This bit must be cleared after initial baud rate setup in order to access other registers.                                                                                                                                                                                                                                                  | 0x0   |
| 6    | R/W  | UART_BC   | Break Control Bit.<br>This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the serial line is forced low until the Break bit is cleared. If active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. | 0x0   |
| 5    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 4    | R/W  | UART_EPS  | Even Parity Select. Writeable only when UART is not busy (USR[0] is zero).<br>This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.                                                                                                                                                                                     | 0x0   |

Table 261: UART\_LCR\_REG (0x5000100C)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|-----|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3   | R/W  | UART_PEN  | Parity Enable. Writeable only when UART is not busy (USR[0] is zero).<br>This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively.<br>0 = parity disabled<br>1 = parity enabled                                                                                                                                                                                                                                                                                               | 0x0   |
| 2   | R/W  | UART_STOP | Number of stop bits.<br>This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data.<br>If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit.<br>0 = 1 stop bit<br>1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit | 0x0   |
| 1:0 | R/W  | UART_DLS  | Data Length Select.<br>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:<br>00 = 5 bits<br>01 = 6 bits<br>10 = 7 bits<br>11 = 8 bits                                                                                                                                                                                                                                                                                                     | 0x0   |

Table 262: UART\_MCR\_REG (0x50001010)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|------|------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 6    | R/W  | UART_SIRE | SIR Mode Enable.<br>This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol".<br>0 = IrDA SIR Mode disabled<br>1 = IrDA SIR Mode enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 5    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 4    | R/W  | UART_LB   | LoopBack Bit.<br>This is used to put the UART into a diagnostic mode for test purposes.<br>If operating in UART mode (SIR_MODE not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally.<br>If operating in infrared mode (SIR_MODE active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line. | 0x0   |

Table 262: UART\_MCR\_REG (0x50001010)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3   | R/W  | UART_OUT2 | OUT2.<br>This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is:<br>0 = out2_n de-asserted (logic 1)<br>1 = out2_n asserted (logic 0)<br>Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input. | 0x0   |
| 2   | R/W  | UART_OUT1 | OUT1.<br>This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is:<br>0 = out1_n de-asserted (logic 1)<br>1 = out1_n asserted (logic 0)<br>Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input. | 0x0   |
| 1:0 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |

Table 263: UART\_LSR\_REG (0x50001014)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 6    | R    | UART_TEMT | Transmitter Empty bit.<br>This bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x1   |
| 5    | R    | UART_THRE | Transmit Holding Register Empty bit.<br>If THRE mode is disabled (IER[7] set to zero), this bit indicates that the THR.<br>This bit is set whenever data is transferred from the THR to the transmitter shift register and no new data has been written to the THR. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled.                                                                                                                                                                                                                                                                                                                                                                                                      | 0x1   |
| 4    | R    | UART_BI   | Break Interrupt bit.<br>This is used to indicate the detection of a break sequence on the serial input data.<br>If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits.<br>If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART.<br>Reading the LSR clears the BI bit. The BI indication occurs immediately and persists until the LSR is read. | 0x0   |

Table 263: UART\_LSR\_REG (0x50001014)

| Bit | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|-----|------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3   | R    | UART_FE | <p>Framing Error bit.</p> <p>This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.</p> <p>When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]).</p> <p>0 = no framing error<br/>1 = framing error</p> <p>Reading the LSR clears the FE bit.</p> | 0x0   |
| 2   | R    | UART_PE | <p>Parity Error bit.</p> <p>This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.</p> <p>It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]).</p> <p>0 = no parity error<br/>1 = parity error</p> <p>Reading the LSR clears the PE bit.</p>                                                                                                                                                                                                                                                                                          | 0x0   |
| 1   | R    | UART_OE | <p>Overrun error bit.</p> <p>This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.</p> <p>The OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten.</p> <p>0 = no overrun error<br/>1 = overrun error</p> <p>Reading the LSR clears the OE bit.</p>                                                                                                                                                                                                                                             | 0x0   |
| 0   | R    | UART_DR | <p>Data Ready bit.</p> <p>This is used to indicate that the receiver contains at least one character in the RBR.</p> <p>0 = no data ready<br/>1 = data ready</p> <p>This bit is cleared when the RBR is read.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |

Table 264: UART\_SCR\_REG (0x5000101C)

| Bit  | Mode | Symbol               | Description                                                                                                       | Reset |
|------|------|----------------------|-------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -                    | Reserved                                                                                                          | 0x0   |
| 7:0  | R/W  | UART_SCRATCH_P<br>AD | This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART Ctrl. | 0x0   |

Table 265: UART\_USR\_REG (0x5000107C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 265: UART\_USR\_REG (0x5000107C)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|-----|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R    | UART_BUSY | UART Busy. This indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive. 0 - DW_apb_uart is idle or inactive 1 - DW_apb_uart is busy (actively transferring data) Note that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in the THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled) the assertion of this bit will also be delayed by several cycles of the slower clock. | 0x0   |

Table 266: UART\_SRR\_REG (0x50001088)

| Bit  | Mode | Symbol  | Description                                                                                                                                                              | Reset |
|------|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -       | Reserved                                                                                                                                                                 | 0x0   |
| 0    | W    | UART_UR | UART Reset. This asynchronously resets the UART Ctrl and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset. | 0x0   |

Table 267: UART\_SBCR\_REG (0x50001090)

| Bit  | Mode | Symbol                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|------|------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -                         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 0    | R/W  | UART_SHADOW_BREAK_CONTROL | Shadow Break Control Bit.<br>This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device.<br>If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared.<br>If SIR_MODE active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver. | 0x0   |

Table 268: UART\_DMASA\_REG (0x500010A8)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | W    | DMASA  | This register is use to perform DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This will cause the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. | 0x0   |



Table 269: UART\_DLF\_REG (0x500010C0)

| Bit | Mode | Symbol   | Description                                                                                                                                          | Reset |
|-----|------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3:0 | R/W  | UART_DLF | The fractional value is added to integer value set by DLH, DLL. Fractional value is determined by (Divisor Fraction value)/(2 <sup>DLF_SIZE</sup> ). | 0x0   |

Table 270: UART\_CPR\_REG (0x500010F4)

| Bit  | Mode | Symbol | Description                  | Reset  |
|------|------|--------|------------------------------|--------|
| 15:0 | R    | CPR    | Component Parameter Register | 0x3941 |

Table 271: UART\_UCV\_REG (0x500010F8)

| Bit  | Mode | Symbol | Description       | Reset     |
|------|------|--------|-------------------|-----------|
| 15:0 | R    | UCV    | Component Version | 0x331352A |

Table 272: UART\_CTR\_REG (0x500010FC)

| Bit  | Mode | Symbol | Description             | Reset      |
|------|------|--------|-------------------------|------------|
| 15:0 | R    | CTR    | Component Type Register | 0x44570110 |

Table 273: UART2\_RBR\_THR\_DLL\_REG (0x50001100)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 273: UART2\_RBR\_THR\_DLL\_REG (0x50001100)

| Bit | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-----|------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | RBR_THR_DLL | <p><b>Receive Buffer Register: (RBR).</b><br/>This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p><b>Transmit Holding Register: (THR)</b><br/>This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> <p><b>Divisor Latch (Low): (DLL)</b><br/>This register makes up the lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows:<br/> <math display="block">\text{baud rate} = (\text{serial clock freq}) / (16 * \text{divisor})</math> Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLL is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.</p> <p><b>Divisor Latch (Low): (DLH) (Note: This register is placed in UART_IER_DLH_REG with offset 0x4)</b><br/>Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock frequency divided by sixteen times the value of the baud rate divisor, as follows:<br/> <math display="block">\text{baud rate} = (\text{serial clock freq}) / (16 * \text{divisor}).</math> Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.</p> | 0x0   |

Table 274: UART2\_IER\_DLH\_REG (0x50001104)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -          | Reserved                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 7    | R/W  | PTIME_DLH7 | Interrupt Enable Register: PTIME, Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[7] of the 8 bit DLH register.                                                                                                                                                | 0x0   |
| 6:4  | -    | -          | Reserved                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 3    | -    | -          | Reserved                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 2    | R/W  | ELSI_DHL2  | Interrupt Enable Register: ELSI, Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[2] of the 8 bit DLH register.                                                                                           | 0x0   |
| 1    | R/W  | ETBEI_DLH1 | Interrupt Enable Register: ETBEI, Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[1] of the 8 bit DLH register.                                                           | 0x0   |
| 0    | R/W  | ERBFI_DLH0 | Interrupt Enable Register: ERBFI, Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFO's enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled Divisor Latch (High): Bit[0] of the 8 bit DLH register. | 0x0   |

Table 275: UART2\_IIR\_FCR\_REG (0x50001108)

| Bit  | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IIR_FCR | <p>Interrupt Identification Register, reading this register; FIFO Control Register, writing to this register.</p> <p>Interrupt Identification Register: Bits[7:6], FIFO's Enabled (or FIFOSE): This is used to indicate whether the FIFO's are enabled or disabled. 00 = disabled. 11 = enabled. Bits[3:0], Interrupt ID (or IID): This indicates the highest priority pending interrupt which can be one of the following types: 0001 = no interrupt pending. 0010 = THR empty. 0100 = received data available. 0110 = receiver line status. 0111 = busy detect. 1100 = character timeout.</p> <p>FIFO Control Register Bits[7:6], RCVR Trigger (or RT):. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode it is used to determine when the rts_n signal will be de-asserted. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than full Bits[5:4], TX Empty Trigger (or TET): This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. It also determines when the dma_tx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full Bit[3], DMA Mode (or DMAM): This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. 0 = mode 0 1 = mode 1 Bit[2], XMIT FIFO Reset (or XFIFOR): This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[1], RCVR FIFO Reset (or RFIFOR): This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[0], FIFO Enable (or FIFOE): This enables/disables the transmit (XMIT) and receive (RCVR) FIFO's. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFO's will be reset.</p> | 0x0   |

Table 276: UART2\_LCR\_REG (0x5000110C)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                            | Reset |
|------|------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                               | 0x0   |
| 7    | R/W  | UART_DLAB | <p>Divisor Latch Access Bit.</p> <p>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>This bit must be cleared after initial baud rate setup in order to access other registers.</p> | 0x0   |

Table 276: UART2\_LCR\_REG (0x5000110C)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|-----|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | UART_BC   | Break Control Bit.<br>This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.                                                | 0x0   |
| 5   | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 4   | R/W  | UART_EPS  | Even Parity Select. Writeable only when UART is not busy (USR[0] is zero).<br>This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.                                                                                                                                                                                                                                  | 0x0   |
| 3   | R/W  | UART_PEN  | Parity Enable. Writeable only when UART is not busy (USR[0] is zero)<br>This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively.<br>0 = parity disabled<br>1 = parity enabled                                                                                                                                                                                                                                                                                                | 0x0   |
| 2   | R/W  | UART_STOP | Number of stop bits.<br>This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data.<br>If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit.<br>0 = 1 stop bit<br>1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit | 0x0   |
| 1:0 | R/W  | UART_DLS  | Data Length Select.<br>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:<br>00 = 5 bits<br>01 = 6 bits<br>10 = 7 bits<br>11 = 8 bits                                                                                                                                                                                                                                                                                                     | 0x0   |

Table 277: UART2\_MCR\_REG (0x50001110)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                       | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -         | Reserved                                                                                                                                                                          | 0x0   |
| 6    | R/W  | UART_SIRE | SIR Mode Enable.<br>This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol".<br>0 = IrDA SIR Mode disabled<br>1 = IrDA SIR Mode enabled | 0x0   |

Table 277: UART2\_MCR\_REG (0x50001110)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|-----|------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | UART_AFCE | Auto Flow Control Enable.<br>Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in "Auto Flow Control".<br>0 = Auto Flow Control Mode disabled<br>1 = Auto Flow Control Mode enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 4   | R/W  | UART_LB   | LoopBack Bit.<br>This is used to put the UART into a diagnostic mode for test purposes.<br>If operating in UART mode (SIR_MODE not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally.<br>If operating in infrared mode (SIR_MODE active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.                                                                                                                               | 0x0   |
| 3   | R/W  | UART_OUT2 | OUT2.<br>This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is:<br>0 = out2_n de-asserted (logic 1)<br>1 = out2_n asserted (logic 0)<br>Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input.                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 2   | R/W  | UART_OUT1 | OUT1.<br>This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is:<br>0 = out1_n de-asserted (logic 1)<br>1 = out1_n asserted (logic 0)<br>Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input.                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 1   | R/W  | UART_RTS  | Request to Send.<br>This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data.<br>When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low.<br>Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input. | 0x0   |
| 0   | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |

Table 278: UART2\_LSR\_REG (0x50001114)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 7    | R    | UART_RFE  | Receiver FIFO Error bit.<br>This bit is only relevant when FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO.<br>0 = no error in RX FIFO<br>1 = error in RX FIFO<br>This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 6    | R    | UART_TEMT | Transmitter Empty bit.<br>If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x1   |
| 5    | R    | UART_THRE | Transmit Holding Register Empty bit.<br>If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty.<br>This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.                                                                                                                                                                                                                           | 0x1   |
| 4    | R    | UART_BI   | Break Interrupt bit.<br>This is used to indicate the detection of a break sequence on the serial input data.<br>If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits.<br>If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART.<br>In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO.<br>Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. | 0x0   |



Table 278: UART2\_LSR\_REG (0x50001114)

| Bit | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|-----|------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3   | R    | UART_FE | <p>Framing Error bit.</p> <p>This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.</p> <p>In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO.</p> <p>When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]).</p> <p>0 = no framing error<br/>1 = framing error</p> <p>Reading the LSR clears the FE bit.</p> | 0x0   |
| 2   | R    | UART_PE | <p>Parity Error bit.</p> <p>This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.</p> <p>In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO.</p> <p>It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]).</p> <p>0 = no parity error<br/>1 = parity error</p> <p>Reading the LSR clears the PE bit.</p>                                                                                                                                                                                                                                                                                       | 0x0   |
| 1   | R    | UART_OE | <p>Overrun error bit.</p> <p>This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.</p> <p>In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost.</p> <p>0 = no overrun error<br/>1 = overrun error</p> <p>Reading the LSR clears the OE bit.</p>                                                                                                                                                                                                   | 0x0   |
| 0   | R    | UART_DR | <p>Data Ready bit.</p> <p>This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO.</p> <p>0 = no data ready<br/>1 = data ready</p> <p>This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |

Table 279: UART2\_MSR\_REG (0x50001118)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:5 | -    | -      | Reserved    | 0x0   |

Table 279: UART2\_MSR\_REG (0x50001118)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|-----|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4   | R    | UART_CTS  | Clear to Send.<br>This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the UART Ctrl.<br>0 = cts_n input is de-asserted (logic 1)<br>1 = cts_n input is asserted (logic 0)<br>In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).                                                                                                                          | 0x0   |
| 3:1 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 0   | R    | UART_DCTS | Delta Clear to Send.<br>This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.<br>0 = no change on cts_n since last read of MSR<br>1 = change on cts_n since last read of MSR<br>Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS).<br>Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted. | 0x0   |

Table 280: UART2\_SCR\_REG (0x5000111C)

| Bit  | Mode | Symbol               | Description                                                                                                       | Reset |
|------|------|----------------------|-------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -                    | Reserved                                                                                                          | 0x0   |
| 7:0  | R/W  | UART_SCRATCH_P<br>AD | This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART Ctrl. | 0x0   |

Table 281: UART2\_SRBR\_STHR0\_REG (0x50001130)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 281: UART2\_SRBR\_STHR0\_REG (0x50001130)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 282: UART2\_SRBR\_STHR1\_REG (0x50001134)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 282: UART2\_SRBR\_STHR1\_REG (0x50001134)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 283: UART2\_SRBR\_STHR2\_REG (0x50001138)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 283: UART2\_SRBR\_STHR2\_REG (0x50001138)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 284: UART2\_SRBR\_STHR3\_REG (0x5000113C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 284: UART2\_SRBR\_STHR3\_REG (0x5000113C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 285: UART2\_SRBR\_STHR4\_REG (0x50001140)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 285: UART2\_SRBR\_STHR4\_REG (0x50001140)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sir_in) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 286: UART2\_SRBR\_STHR5\_REG (0x50001144)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |



Table 286: UART2\_SRBR\_STHR5\_REG (0x50001144)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 287: UART2\_SRBR\_STHR6\_REG (0x50001148)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 287: UART2\_SRBR\_STHR6\_REG (0x50001148)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 288: UART2\_SRBR\_STHR7\_REG (0x5000114C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 288: UART2\_SRBR\_STHR7\_REG (0x5000114C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 289: UART2\_SRBR\_STHR8\_REG (0x50001150)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 289: UART2\_SRBR\_STHR8\_REG (0x50001150)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 290: UART2\_SRBR\_STHR9\_REG (0x50001154)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 290: UART2\_SRBR\_STHR9\_REG (0x50001154)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 291: UART2\_SRBR\_STHR10\_REG (0x50001158)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 291: UART2\_SRBR\_STHR10\_REG (0x50001158)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 292: UART2\_SRBR\_STHR11\_REG (0x5000115C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 292: UART2\_SRBR\_STHR11\_REG (0x5000115C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sir_in) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 293: UART2\_SRBR\_STHR12\_REG (0x50001160)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |



Table 293: UART2\_SRBR\_STHR12\_REG (0x50001160)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 294: UART2\_SRBR\_STHR13\_REG (0x50001164)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 294: UART2\_SRBR\_STHR13\_REG (0x50001164)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 295: UART2\_SRBR\_STHR14\_REG (0x50001168)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 295: UART2\_SRBR\_STHR14\_REG (0x50001168)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 296: UART2\_SRBR\_STHR15\_REG (0x5000116C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 296: UART2\_SRBR\_STHR15\_REG (0x5000116C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SRBR_STHRX | <p>Shadow Receive Buffer Register x: 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. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.</p> <p>Shadow Transmit Holding Register 0: 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. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If FIFO's are disabled (FCR[0] set to zero) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If FIFO's are enabled (FCR[0] set to one) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> | 0x0   |

Table 297: UART2\_FAR\_REG (0x50001170)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|-----|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R    | UART_FAR | <p>Description: Writes will have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFO's are implemented and enabled. When FIFO's are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFO's are treated as empty.</p> | 0x0   |

Table 298: UART2\_USR\_REG (0x5000117C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:5 | -    | -      | Reserved    | 0x0   |

Table 298: UART2\_USR\_REG (0x5000117C)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|-----|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4   | R    | UART_RFF  | Receive FIFO Full.<br>This is used to indicate that the receive FIFO is completely full.<br>0 = Receive FIFO not full<br>1 = Receive FIFO Full<br>This bit is cleared when the RX FIFO is no longer full.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 3   | R    | UART_RFNE | Receive FIFO Not Empty.<br>This is used to indicate that the receive FIFO contains one or more entries.<br>0 = Receive FIFO is empty<br>1 = Receive FIFO is not empty<br>This bit is cleared when the RX FIFO is empty.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 2   | R    | UART_TFE  | Transmit FIFO Empty.<br>This is used to indicate that the transmit FIFO is completely empty.<br>0 = Transmit FIFO is not empty<br>1 = Transmit FIFO is empty<br>This bit is cleared when the TX FIFO is no longer empty.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x1   |
| 1   | R    | UART_TFNF | Transmit FIFO Not Full.<br>This is used to indicate that the transmit FIFO is not full.<br>0 = Transmit FIFO is full<br>1 = Transmit FIFO is not full<br>This bit is cleared when the TX FIFO is full.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x1   |
| 0   | R    | UART_BUSY | UART Busy. This indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive. 0 - DW_apb_uart is idle or inactive 1 - DW_apb_uart is busy (actively transferring data) Note that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in the THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled) the assertion of this bit will also be delayed by several cycles of the slower clock. | 0x0   |

Table 299: UART2\_TFL\_REG (0x50001180)

| Bit  | Mode | Symbol                  | Description                                                                             | Reset |
|------|------|-------------------------|-----------------------------------------------------------------------------------------|-------|
| 15:0 | R    | UART_TRANSMIT_FIFOLEVEL | Transmit FIFO Level.<br>This indicates the number of data entries in the transmit FIFO. | 0x0   |

Table 300: UART2\_RFL\_REG (0x50001184)

| Bit  | Mode | Symbol                 | Description                                                                           | Reset |
|------|------|------------------------|---------------------------------------------------------------------------------------|-------|
| 15:0 | R    | UART_RECEIVE_FIFOLEVEL | Receive FIFO Level.<br>This indicates the number of data entries in the receive FIFO. | 0x0   |

Table 301: UART2\_SRR\_REG (0x50001188)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:3 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 2    | W    | UART_XFR | XMIT FIFO Reset.<br>This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit. | 0x0   |
| 1    | W    | UART_RFR | RCVR FIFO Reset.<br>This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO. This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.   | 0x0   |
| 0    | W    | UART_UR  | UART Reset. This asynchronously resets the UART Ctrl and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.                                                                                                                                                                                                                                        | 0x0   |

Table 302: UART2\_SRTS\_REG (0x5000118C)

| Bit  | Mode | Symbol                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|------|------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -                           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 0    | R/W  | UART_SHADOW_REQUEST_TO_SEND | Shadow Request to Send.<br>This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART Ctrl is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high.<br>In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold).<br>Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input. | 0x0   |

Table 303: UART2\_SBCR\_REG (0x50001190)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 303: UART2\_SBCR\_REG (0x50001190)

| Bit | Mode | Symbol                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R/W  | UART_SHADOW_BREAK_CONTROL | Shadow Break Control Bit.<br>This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device.<br>If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared.<br>If SIR_MODE active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver. | 0x0   |

Table 304: UART2\_SDMAM\_REG (0x50001194)

| Bit  | Mode | Symbol               | Description                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|------|------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -                    | Reserved                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 0    | R/W  | UART_SHADOW_DMA_MODE | Shadow DMA Mode.<br>This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals.<br>0 = mode 0<br>1 = mode 1 | 0x0   |

Table 305: UART2\_SFE\_REG (0x50001198)

| Bit  | Mode | Symbol                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|------|------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -                       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 0    | R/W  | UART_SHADOW_FIFO_ENABLE | Shadow FIFO Enable.<br>This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset. | 0x0   |

Table 306: UART2\_SRT\_REG (0x5000119C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:2 | -    | -      | Reserved    | 0x0   |



Table 306: UART2\_SRT\_REG (0x5000119C)

| Bit | Mode | Symbol                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|-----|------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1:0 | R/W  | UART_SHADOW_R_CVR_TRIGGER | <p>Shadow RCVR Trigger.</p> <p>This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated.</p> <p>This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported:</p> <p>00 = 1 character in the FIFO<br/> 01 = FIFO ¼ full<br/> 10 = FIFO ½ full<br/> 11 = FIFO 2 less than full</p> | 0x0   |

Table 307: UART2\_STET\_REG (0x500011A0)

| Bit  | Mode | Symbol                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:2 | -    | -                            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 1:0  | R/W  | UART_SHADOW_TX_EMPTY_TRIGGER | <p>Shadow TX Empty Trigger.</p> <p>This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated.</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>00 = FIFO empty<br/> 01 = 2 characters in the FIFO<br/> 10 = FIFO ¼ full<br/> 11 = FIFO ½ full</p> | 0x0   |

Table 308: UART2\_HTX\_REG (0x500011A4)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                              | Reset |
|------|------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 0    | R/W  | UART_HALT_TX | <p>This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled.</p> <p>0 = Halt TX disabled<br/> 1 = Halt TX enabled</p> <p>Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.</p> | 0x0   |

Table 309: UART2\_DMASA\_REG (0x500011A8)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | W    | DMASA  | This register is use to perform DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This will cause the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. | 0x0   |

Table 310: UART2\_DLF\_REG (0x500011C0)

| Bit | Mode | Symbol   | Description                                                                                                                                                    | Reset |
|-----|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3:0 | R/W  | UART_DLF | The fractional value is added to integer value set by DLH, DLL. Fractional value is determined by $(\text{Divisor Fraction value}) / (2^{\text{DLF\_SIZE}})$ . | 0x0   |

Table 311: UART2\_CPR\_REG (0x500011F4)

| Bit  | Mode | Symbol | Description                  | Reset  |
|------|------|--------|------------------------------|--------|
| 15:0 | R    | CPR    | Component Parameter Register | 0x3D71 |

Table 312: UART2\_UCV\_REG (0x500011F8)

| Bit  | Mode | Symbol | Description       | Reset      |
|------|------|--------|-------------------|------------|
| 15:0 | R    | UCV    | Component Version | 0x3331352A |

Table 313: UART2\_CTR\_REG (0x500011FC)

| Bit  | Mode | Symbol | Description             | Reset      |
|------|------|--------|-------------------------|------------|
| 15:0 | R    | CTR    | Component Type Register | 0x44570110 |

## 37.11 SPI REGISTER FILE

Table 314: Register map SPI

| Address    | Port               | Description                  |
|------------|--------------------|------------------------------|
| 0x50001200 | SPI_CTRL_REG       | SPI control register 0       |
| 0x50001202 | SPI_RX_TX_REG0     | SPI RX/TX register0          |
| 0x50001204 | SPI_RX_TX_REG1     | SPI RX/TX register1          |
| 0x50001206 | SPI_CLEAR_INT_REG  | SPI clear interrupt register |
| 0x50001208 | SPI_CTRL_REG1      | SPI control register 1       |
| 0x50001300 | SPI2_CTRL_REG      | SPI control register 0       |
| 0x50001302 | SPI2_RX_TX_REG0    | SPI RX/TX register0          |
| 0x50001304 | SPI2_RX_TX_REG1    | SPI RX/TX register1          |
| 0x50001306 | SPI2_CLEAR_INT_REG | SPI clear interrupt register |
| 0x50001308 | SPI2_CTRL_REG1     | SPI control register 1       |

Table 315: SPI\_CTRL\_REG (0x50001200)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                         | Reset |
|-----|------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R/W  | SPI_EN_CTRL  | 0 = SPI_EN pin disabled in slave mode. Pin SPI_EN is don't care.<br>1 = SPI_EN pin enabled in slave mode.                                                                                           | 0x0   |
| 14  | R/W  | SPI_MINT     | 0 = Disable SPI_INT_BIT to ICU<br>1 = Enable SPI_INT_BIT to ICU.<br>Note that the SPI_INT interrupt is shared with AD_INT interrupt                                                                 | 0x0   |
| 13  | R    | SPI_INT_BIT  | 0 = RX Register or FIFO is empty.<br>1 = SPI interrupt. Data has been transmitted and received- Must be reset by SW by writing to SPI_CLEAR_INT_REG.                                                | 0x0   |
| 12  | R    | SPI_DI       | Returns the actual value of pin SPI_DIN (delayed with two internal SPI clock cycles)                                                                                                                | 0x0   |
| 11  | R    | SPI_TXH      | 0 = TX-FIFO is not full, data can be written.<br>1 = TX-FIFO is full, data can not be written.                                                                                                      | 0x0   |
| 10  | R/W  | SPI_FORCE_DO | 0 = normal operation<br>1 = Force SPIDO output level to value of SPI_DO.                                                                                                                            | 0x0   |
| 9   | R/W  | SPI_RST      | 0 = normal operation<br>1 = Reset SPI. Same function as SPI_ON except that internal clock remain active.                                                                                            | 0x0   |
| 8:7 | R/W  | SPI_WORD     | 00 = 8 bits mode, only SPI_RX_TX_REG0 used<br>01 = 16 bit mode, only SPI_RX_TX_REG0 used<br>10 = 32 bits mode, SPI_RX_TX_REG0 & SPI_RX_TX_REG1 used<br>11 = 9 bits mode. Only valid in master mode. | 0x0   |
| 6   | R/W  | SPI_SMN      | Master/slave mode<br>0 = Master,<br>1 = Slave(SPI1 only)                                                                                                                                            | 0x0   |
| 5   | R/W  | SPI_DO       | Pin SPI_DO output level when SPI is idle or when SPI_FORCE_DO=1                                                                                                                                     | 0x0   |
| 4:3 | R/W  | SPI_CLK      | Select SPI_CLK clock output frequency in master mode:<br>00 = SPI_CLK / 8<br>01 = SPI_CLK / 4<br>10 = SPI_CLK / 2<br>11 = SPI_CLK / 14                                                              | 0x0   |
| 2   | R/W  | SPI_POL      | Select SPI_CLK polarity.<br>0 = SPI_CLK is initially low.<br>1 = SPI_CLK is initially high.                                                                                                         | 0x0   |

Table 315: SPI\_CTRL\_REG (0x50001200)

| Bit | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                               | Reset |
|-----|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | SPI_PHA | Select SPI_CLK phase. See functional timing diagrams in SPI chapter                                                                                                                                                                                                                                                                                                       | 0x0   |
| 0   | R/W  | SPI_ON  | 0 = SPI Module switched off (power saving). Everything is reset except SPI_CTRL_REG0 and SPI_CTRL_REG1.<br>When this bit is cleared the SPI will remain active in master mode until the shift register and holding register are both empty.<br>1 = SPI Module switched on. Should only be set after all control bits have their desired values. So two writes are needed! | 0x0   |

Table 316: SPI\_RX\_TX\_REG0 (0x50001202)

| Bit  | Mode | Symbol    | Description                                                                                                                                                             | Reset |
|------|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | SPI_DATA0 | Write: SPI_TX_REG0 output register 0 (TX-FIFO)<br>Read: SPI_RX_REG0 input register 0 (RX-FIFO)<br>In 8 or 9 bits mode bits 15 to 8 are not used, they contain old data. | 0x0   |

Table 317: SPI\_RX\_TX\_REG1 (0x50001204)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                   | Reset |
|------|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | SPI_DATA1 | Write: SPI_TX_REG1 output register 1 (MSB's of TX-FIFO)<br>Read: SPI_RX_REG1 input register 1 (MSB's of RX-FIFO)<br>In 8 or 9 or 16 bits mode bits this register is not used. | 0x0   |

Table 318: SPI\_CLEAR\_INT\_REG (0x50001206)

| Bit  | Mode | Symbol        | Description                                                                                       | Reset |
|------|------|---------------|---------------------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | SPI_CLEAR_INT | Writing any value to this register will clear the SPI_CTRL_REG[SPI_INT_BIT]<br>Reading returns 0. | 0x0   |

Table 319: SPI\_CTRL\_REG1 (0x50001208)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                           | Reset |
|------|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -            | Reserved                                                                                                                                                                                                                                                                              | 0x0   |
| 4    | R/W  | SPI_9BIT_VAL | Determines the value of the first bit in 9 bits SPI mode.                                                                                                                                                                                                                             | 0x0   |
| 3    | R    | SPI_BUSY     | 0 = The SPI is not busy with a transfer. This means that either no TX-data is available or that the transfers have been suspended due to a full RX-FIFO. The SPIx_CTRL_REG0[SPI_INT_BIT] can be used to distinguish between these situations.<br>1 = The SPI is busy with a transfer. | 0x0   |
| 2    | R/W  | SPI_PRIORITY | 0 = The SPI has low priority, the DMA request signals are reset after the corresponding acknowledge.<br>1 = The SPI has high priority, DMA request signals remain active until the FIFOS are filled/emptied, so the DMA holds the AHB bus.                                            | 0x0   |

Table 319: SPI\_CTRL\_REG1 (0x50001208)

| Bit | Mode | Symbol        | Description                                                                                                                                                                                                                                             | Reset |
|-----|------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1:0 | R/W  | SPI_FIFO_MODE | 0: TX-FIFO and RX-FIFO used (Bidirectional mode).<br>1: RX-FIFO used (Read Only Mode) TX-FIFO single depth, no flow control<br>2: TX-FIFO used (Write Only Mode), RX-FIFO single depth, no flow control<br>3: No FIFOs used (backwards compatible mode) | 0x3   |

Table 320: SPI2\_CTRL\_REG (0x50001300)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                         | Reset |
|-----|------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R/W  | SPI_EN_CTRL  | 0 = SPI_EN pin disabled in slave mode. Pin SPI_EN is don't care.<br>1 = SPI_EN pin enabled in slave mode.                                                                                           | 0x0   |
| 14  | R/W  | SPI_MINT     | 0 = Disable SPI_INT_BIT to ICU<br>1 = Enable SPI_INT_BIT to ICU.<br>Note that the SPI_INT interrupt is shared with AD_INT interrupt                                                                 | 0x0   |
| 13  | R    | SPI_INT_BIT  | 0 = RX Register or FIFO is empty.<br>1 = SPI interrupt. Data has been transmitted and received- Must be reset by SW by writing to SPI_CLEAR_INT_REG.                                                | 0x0   |
| 12  | R    | SPI_DI       | Returns the actual value of pin SPI_DIN (delayed with two internal SPI clock cycles)                                                                                                                | 0x0   |
| 11  | R    | SPI_TXH      | 0 = TX-FIFO is not full, data can be written.<br>1 = TX-FIFO is full, data can not be written.                                                                                                      | 0x0   |
| 10  | R/W  | SPI_FORCE_DO | 0 = normal operation<br>1 = Force SPIDO output level to value of SPI_DO.                                                                                                                            | 0x0   |
| 9   | R/W  | SPI_RST      | 0 = normal operation<br>1 = Reset SPI. Same function as SPI_ON except that internal clock remain active.                                                                                            | 0x0   |
| 8:7 | R/W  | SPI_WORD     | 00 = 8 bits mode, only SPI_RX_TX_REG0 used<br>01 = 16 bit mode, only SPI_RX_TX_REG0 used<br>10 = 32 bits mode, SPI_RX_TX_REG0 & SPI_RX_TX_REG1 used<br>11 = 9 bits mode. Only valid in master mode. | 0x0   |
| 6   | R/W  | SPI_SMN      | Master/slave mode<br>0 = Master,<br>1 = Slave(SPI1 only)                                                                                                                                            | 0x0   |
| 5   | R/W  | SPI_DO       | Pin SPI_DO output level when SPI is idle or when SPI_FORCE_DO=1                                                                                                                                     | 0x0   |
| 4:3 | R/W  | SPI_CLK      | Select SPI_CLK clock output frequency in master mode:<br>00 = SPI_CLK / 8<br>01 = SPI_CLK / 4<br>10 = SPI_CLK / 2<br>11 = SPI_CLK / 14                                                              | 0x0   |
| 2   | R/W  | SPI_POL      | Select SPI_CLK polarity.<br>0 = SPI_CLK is initially low.<br>1 = SPI_CLK is initially high.                                                                                                         | 0x0   |
| 1   | R/W  | SPI_PHA      | Select SPI_CLK phase. See functional timing diagrams in SPI chapter                                                                                                                                 | 0x0   |

Table 320: SPI2\_CTRL\_REG (0x50001300)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                               | Reset |
|-----|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R/W  | SPI_ON | 0 = SPI Module switched off (power saving). Everything is reset except SPI_CTRL_REG0 and SPI_CTRL_REG1.<br>When this bit is cleared the SPI will remain active in master mode until the shift register and holding register are both empty.<br>1 = SPI Module switched on. Should only be set after all control bits have their desired values. So two writes are needed! | 0x0   |

Table 321: SPI2\_RX\_TX\_REG0 (0x50001302)

| Bit  | Mode | Symbol    | Description                                                                                                                                                             | Reset |
|------|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | SPI_DATA0 | Write: SPI_TX_REG0 output register 0 (TX-FIFO)<br>Read: SPI_RX_REG0 input register 0 (RX-FIFO)<br>In 8 or 9 bits mode bits 15 to 8 are not used, they contain old data. | 0x0   |

Table 322: SPI2\_RX\_TX\_REG1 (0x50001304)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                   | Reset |
|------|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | SPI_DATA1 | Write: SPI_TX_REG1 output register 1 (MSB's of TX-FIFO)<br>Read: SPI_RX_REG1 input register 1 (MSB's of RX-FIFO)<br>In 8 or 9 or 16 bits mode bits this register is not used. | 0x0   |

Table 323: SPI2\_CLEAR\_INT\_REG (0x50001306)

| Bit  | Mode | Symbol        | Description                                                                                       | Reset |
|------|------|---------------|---------------------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | SPI_CLEAR_INT | Writing any value to this register will clear the SPI_CTRL_REG[SPI_INT_BIT]<br>Reading returns 0. | 0x0   |

Table 324: SPI2\_CTRL\_REG1 (0x50001308)

| Bit  | Mode | Symbol        | Description                                                                                                                                                                                                                                                                           | Reset |
|------|------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -             | Reserved                                                                                                                                                                                                                                                                              | 0x0   |
| 4    | R/W  | SPI_9BIT_VAL  | Determines the value of the first bit in 9 bits SPI mode.                                                                                                                                                                                                                             | 0x0   |
| 3    | R    | SPI_BUSY      | 0 = The SPI is not busy with a transfer. This means that either no TX-data is available or that the transfers have been suspended due to a full RX-FIFO. The SPIx_CTRL_REG0[SPI_INT_BIT] can be used to distinguish between these situations.<br>1 = The SPI is busy with a transfer. | 0x0   |
| 2    | R/W  | SPI_PRIORITY  | 0 = The SPI has low priority, the DMA request signals are reset after the corresponding acknowledge.<br>1 = The SPI has high priority, DMA request signals remain active until the FIFOs are filled/emtpied, so the DMA holds the AHB bus.                                            | 0x0   |
| 1:0  | R/W  | SPI_FIFO_MODE | 0: TX-FIFO and RX-FIFO used (Bidirectional mode).<br>1: RX-FIFO used (Read Only Mode) TX-FIFO single depth, no flow control<br>2: TX-FIFO used (Write Only Mode), RX-FIFO single depth, no flow control<br>3: No FIFOs used (backwards compatible mode)                               | 0x3   |





## 37.12 I2C REGISTER FILE

Table 325: Register map I2C

| Address    | Port                     | Description                                      |
|------------|--------------------------|--------------------------------------------------|
| 0x50001400 | I2C_CON_REG              | I2C Control Register                             |
| 0x50001404 | I2C_TAR_REG              | I2C Target Address Register                      |
| 0x50001408 | I2C_SAR_REG              | I2C Slave Address Register                       |
| 0x5000140C | I2C_HS_MADDR_REG         | I2C High Speed Master Mode Code Address Register |
| 0x50001410 | I2C_DATA_CMD_REG         | I2C Rx/Tx Data Buffer and Command Register       |
| 0x50001414 | I2C_SS_SCL_HCNT_REG      | Standard Speed I2C Clock SCL High Count Register |
| 0x50001418 | I2C_SS_SCL_LCNT_REG      | Standard Speed I2C Clock SCL Low Count Register  |
| 0x5000141C | I2C_FS_SCL_HCNT_REG      | Fast Speed I2C Clock SCL High Count Register     |
| 0x50001420 | I2C_FS_SCL_LCNT_REG      | Fast Speed I2C Clock SCL Low Count Register      |
| 0x5000142C | I2C_INTR_STAT_REG        | I2C Interrupt Status Register                    |
| 0x50001430 | I2C_INTR_MASK_REG        | I2C Interrupt Mask Register                      |
| 0x50001434 | I2C_RAW_INTR_STAT_REG    | I2C Raw Interrupt Status Register                |
| 0x50001438 | I2C_RX_TL_REG            | I2C Receive FIFO Threshold Register              |
| 0x5000143C | I2C_TX_TL_REG            | I2C Transmit FIFO Threshold Register             |
| 0x50001440 | I2C_CLR_INTR_REG         | Clear Combined and Individual Interrupt Register |
| 0x50001444 | I2C_CLR_RX_UNDER_REG     | Clear RX_UNDER Interrupt Register                |
| 0x50001448 | I2C_CLR_RX_OVER_REG      | Clear RX_OVER Interrupt Register                 |
| 0x5000144C | I2C_CLR_TX_OVER_REG      | Clear TX_OVER Interrupt Register                 |
| 0x50001450 | I2C_CLR_RD_REQ_REG       | Clear RD_REQ Interrupt Register                  |
| 0x50001454 | I2C_CLR_TX_ABRT_REG      | Clear TX_ABRT Interrupt Register                 |
| 0x50001458 | I2C_CLR_RX_DONE_REG      | Clear RX_DONE Interrupt Register                 |
| 0x5000145C | I2C_CLR_ACTIVITY_REG     | Clear ACTIVITY Interrupt Register                |
| 0x50001460 | I2C_CLR_STOP_DET_REG     | Clear STOP_DET Interrupt Register                |
| 0x50001464 | I2C_CLR_START_DET_REG    | Clear START_DET Interrupt Register               |
| 0x50001468 | I2C_CLR_GEN_CALL_REG     | Clear GEN_CALL Interrupt Register                |
| 0x5000146C | I2C_ENABLE_REG           | I2C Enable Register                              |
| 0x50001470 | I2C_STATUS_REG           | I2C Status Register                              |
| 0x50001474 | I2C_TXFLR_REG            | I2C Transmit FIFO Level Register                 |
| 0x50001478 | I2C_RXFLR_REG            | I2C Receive FIFO Level Register                  |
| 0x5000147C | I2C_SDA_HOLD_REG         | I2C SDA Hold Time Length Register                |
| 0x50001480 | I2C_TX_ABRT_SOURCE_REG   | I2C Transmit Abort Source Register               |
| 0x50001488 | I2C_DMA_CR_REG           | DMA Control Register                             |
| 0x5000148C | I2C_DMA_TDLR_REG         | DMA Transmit Data Level Register                 |
| 0x50001490 | I2C_DMA_RDLR_REG         | I2C Receive Data Level Register                  |
| 0x50001494 | I2C_SDA_SETUP_REG        | I2C SDA Setup Register                           |
| 0x50001498 | I2C_ACK_GENERAL_CALL_REG | I2C ACK General Call Register                    |
| 0x5000149C | I2C_ENABLE_STATUS_REG    | I2C Enable Status Register                       |
| 0x500014A0 | I2C_IC_FS_SPKLEN_REG     | I2C SS and FS spike suppression limit Size       |
| 0x500014F4 | I2C_COMP_PARAM1_REG      | Component Parameter Register                     |
| 0x500014F6 | I2C_COMP_PARAM2_REG      | Component Parameter Register 2                   |
| 0x500014F8 | I2C_COMP_VERSION_REG     | I2C Component Version Register                   |
| 0x500014FA | I2C_COMP2_VERSION        | I2C Component2 Version Register                  |
| 0x500014FC | I2C_COMP_TYPE_REG        | I2C Component Type Register                      |

Table 325: Register map I2C

| Address    | Port                      | Description                                      |
|------------|---------------------------|--------------------------------------------------|
| 0x500014FE | I2C_COMP_TYPE2_REG        | I2C Component2 Type Register                     |
| 0x50001500 | I2C2_CON_REG              | I2C Control Register                             |
| 0x50001504 | I2C2_TAR_REG              | I2C Target Address Register                      |
| 0x50001508 | I2C2_SAR_REG              | I2C Slave Address Register                       |
| 0x5000150C | I2C2_HS_MADDR_REG         | I2C High Speed Master Mode Code Address Register |
| 0x50001510 | I2C2_DATA_CMD_REG         | I2C Rx/Tx Data Buffer and Command Register       |
| 0x50001514 | I2C2_SS_SCL_HCNT_REG      | Standard Speed I2C Clock SCL High Count Register |
| 0x50001518 | I2C2_SS_SCL_LCNT_REG      | Standard Speed I2C Clock SCL Low Count Register  |
| 0x5000151C | I2C2_FS_SCL_HCNT_REG      | Fast Speed I2C Clock SCL High Count Register     |
| 0x50001520 | I2C2_FS_SCL_LCNT_REG      | Fast Speed I2C Clock SCL Low Count Register      |
| 0x5000152C | I2C2_INTR_STAT_REG        | I2C Interrupt Status Register                    |
| 0x50001530 | I2C2_INTR_MASK_REG        | I2C Interrupt Mask Register                      |
| 0x50001534 | I2C2_RAW_INTR_STAT_REG    | I2C Raw Interrupt Status Register                |
| 0x50001538 | I2C2_RX_TL_REG            | I2C Receive FIFO Threshold Register              |
| 0x5000153C | I2C2_TX_TL_REG            | I2C Transmit FIFO Threshold Register             |
| 0x50001540 | I2C2_CLR_INTR_REG         | Clear Combined and Individual Interrupt Register |
| 0x50001544 | I2C2_CLR_RX_UNDER_REG     | Clear RX_UNDER Interrupt Register                |
| 0x50001548 | I2C2_CLR_RX_OVER_REG      | Clear RX_OVER Interrupt Register                 |
| 0x5000154C | I2C2_CLR_TX_OVER_REG      | Clear TX_OVER Interrupt Register                 |
| 0x50001550 | I2C2_CLR_RD_REQ_REG       | Clear RD_REQ Interrupt Register                  |
| 0x50001554 | I2C2_CLR_TX_ABRT_REG      | Clear TX_ABRT Interrupt Register                 |
| 0x50001558 | I2C2_CLR_RX_DONE_REG      | Clear RX_DONE Interrupt Register                 |
| 0x5000155C | I2C2_CLR_ACTIVITY_REG     | Clear ACTIVITY Interrupt Register                |
| 0x50001560 | I2C2_CLR_STOP_DET_REG     | Clear STOP_DET Interrupt Register                |
| 0x50001564 | I2C2_CLR_START_DET_REG    | Clear START_DET Interrupt Register               |
| 0x50001568 | I2C2_CLR_GEN_CALL_REG     | Clear GEN_CALL Interrupt Register                |
| 0x5000156C | I2C2_ENABLE_REG           | I2C Enable Register                              |
| 0x50001570 | I2C2_STATUS_REG           | I2C Status Register                              |
| 0x50001574 | I2C2_TXFLR_REG            | I2C Transmit FIFO Level Register                 |
| 0x50001578 | I2C2_RXFLR_REG            | I2C Receive FIFO Level Register                  |
| 0x5000157C | I2C2_SDA_HOLD_REG         | I2C SDA Hold Time Length Register                |
| 0x50001580 | I2C2_TX_ABRT_SOURCE_REG   | I2C Transmit Abort Source Register               |
| 0x50001588 | I2C2_DMA_CR_REG           | DMA Control Register                             |
| 0x5000158C | I2C2_DMA_TDLR_REG         | DMA Transmit Data Level Register                 |
| 0x50001590 | I2C2_DMA_RDLR_REG         | I2C Receive Data Level Register                  |
| 0x50001594 | I2C2_SDA_SETUP_REG        | I2C SDA Setup Register                           |
| 0x50001598 | I2C2_ACK_GENERAL_CALL_REG | I2C ACK General Call Register                    |
| 0x5000159C | I2C2_ENABLE_STATUS_REG    | I2C Enable Status Register                       |
| 0x500015A0 | I2C2_IC_FS_SPKLEN_REG     | I2C SS and FS spike suppression limit Size       |
| 0x500015F4 | I2C2_COMP_PARAM1_REG      | Component Parameter Register                     |
| 0x500015F6 | I2C2_COMP_PARAM2_REG      | Component Parameter Register 2                   |
| 0x500015F8 | I2C2_COMP_VERSION_REG     | I2C Component Version Register                   |

Table 325: Register map I2C

| Address    | Port                | Description                     |
|------------|---------------------|---------------------------------|
| 0x500015FA | I2C2_COMP2_VERSION  | I2C Component2 Version Register |
| 0x500015FC | I2C2_COMP_TYPE_REG  | I2C Component Type Register     |
| 0x500015FE | I2C2_COMP_TYPE2_REG | I2C Component2 Type Register    |

Table 326: I2C\_CON\_REG (0x50001400)

| Bit  | Mode | Symbol               | Description                                                                                                                                                                                                                                                             | Reset |
|------|------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -                    | Reserved                                                                                                                                                                                                                                                                | 0x0   |
| 6    | R/W  | I2C_SLAVE_DISABLE    | Slave enabled or disabled after reset is applied, which means software does not have to configure the slave.<br>0=slave is enabled<br>1=slave is disabled<br>Software should ensure that if this bit is written with '0', then bit 0 should also be written with a '0'. | 0x1   |
| 5    | R/W  | I2C_RESTART_EN       | Determines whether RESTART conditions may be sent when acting as a master<br>0= disable<br>1=enable                                                                                                                                                                     | 0x1   |
| 4    | R/W  | I2C_10BITADDR_MASTER | Controls whether the controller starts its transfers in 7- or 10-bit addressing mode when acting as a master.<br>0= 7-bit addressing<br>1= 10-bit addressing                                                                                                            | 0x1   |
| 3    | R/W  | I2C_10BITADDR_SLAVE  | When acting as a slave, this bit controls whether the controller responds to 7- or 10-bit addresses.<br>0= 7-bit addressing<br>1= 10-bit addressing                                                                                                                     | 0x1   |
| 2:1  | R/W  | I2C_SPEED            | These bits control at which speed the controller operates.<br>1= standard mode (100 kbit/s)<br>2= fast mode (400 kbit/s)                                                                                                                                                | 0x2   |
| 0    | R/W  | I2C_MASTER_MODE      | This bit controls whether the controller master is enabled.<br>0= master disabled<br>1= master enabled<br>Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'.                                                     | 0x1   |

Table 327: I2C\_TAR\_REG (0x50001404)

| Bit   | Mode | Symbol  | Description                                                                                                                                                                                                  | Reset |
|-------|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -       | Reserved                                                                                                                                                                                                     | 0x0   |
| 11    | R/W  | SPECIAL | This bit indicates whether software performs a General Call or START BYTE command.<br>0: ignore bit 10 GC_OR_START and use IC_TAR normally<br>1: perform special I2C command as specified in GC_OR_START bit | 0x0   |

Table 327: I2C\_TAR\_REG (0x50001404)

| Bit | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10  | R/W  | 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 controller.<br>0: General Call Address - after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The controller remains in General Call mode until the SPECIAL bit value (bit 11) is cleared.<br>1: START BYTE                                                                                                       | 0x0   |
| 9:0 | R/W  | IC_TAR      | This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits.<br>Note: If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave<br>Writes to this register succeed only when IC_ENABLE[0] is set to 0 | 0x55  |

Table 328: I2C\_SAR\_REG (0x50001408)

| Bit   | Mode | Symbol | Description                                                                                                                                                                                                                                                                                      | Reset |
|-------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:10 | -    | -      | Reserved                                                                                                                                                                                                                                                                                         | 0x0   |
| 9:0   | R/W  | 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. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. | 0x55  |

Table 329: I2C\_HS\_MADDR\_REG (0x5000140C)

| Bit | Mode | Symbol     | Description                                                    | Reset |
|-----|------|------------|----------------------------------------------------------------|-------|
| 2:0 | R/W  | IIC_HS_MAR | This bit field holds the value of the I2C HS mode master code. | 0x1   |

Table 330: I2C\_DATA\_CMD\_REG (0x50001410)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:9 | -    | -      | Reserved    | 0x0   |

Table 330: I2C\_DATA\_CMD\_REG (0x50001410)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|-----|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 8   | R/W  | CMD    | <p>This bit controls whether a read or a write is performed. This bit does not control the direction when the I2C Ctrl acts as a slave. It controls only the direction when it acts as a master.</p> <p>1 = Read<br/>0 = Write</p> <p>When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a "don't care" because writes to this register are not required. In slave-transmitter mode, a "0" indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the I2C_RAW_INTR_STAT_REG), unless bit 11 (SPECIAL) in the I2C_TAR register has been cleared.</p> <p>If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs.</p> <p>NOTE: It is possible that while attempting a master I2C read transfer on the controller, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing the controller. In this type of scenario, it ignores the I2C_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt</p> | 0x0   |
| 7:0 | R/W  | DAT    | <p>This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the controller. However, when you read this register, these bits return the value of data received on the controller's interface.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |

Table 331: I2C\_SS\_SCL\_HCNT\_REG (0x50001414)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_SS_SCL_HCNT | <p>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. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.</p> <p>NOTE: This register must not be programmed to a value higher than 65525, because the controller uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of IC_SS_SCL_HCNT + 10.</p> | 0x48  |

Table 332: I2C\_SS\_SCL\_LCNT\_REG (0x50001418)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_SS_SCL_LCNT | 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.<br>This register can be written only when the I2C interface is disabled which corresponds to the I2C_ENABLE register being set to 0. Writes at other times have no effect.<br>The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set. | 0x4F  |

Table 333: I2C\_FS\_SCL\_HCNT\_REG (0x5000141C)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_FS_SCL_HCNT | 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. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. This register can be written only when the I2C interface is disabled, which corresponds to the I2C_ENABLE register being set to 0. Writes at other times have no effect.<br>The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. | 0x8   |

Table 334: I2C\_FS\_SCL\_LCNT\_REG (0x50001420)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|------|------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_FS_SCL_LCNT | 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. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. This register can be written only when the I2C interface is disabled, which corresponds to the I2C_ENABLE register being set to 0. Writes at other times have no effect.<br>The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the controller. The lower byte must be programmed first. Then the upper byte is programmed. | 0x17  |

Table 335: I2C\_INTR\_STAT\_REG (0x5000142C)

| Bit   | Mode | Symbol     | Description                                                                                                                                                                                                                                                      | Reset |
|-------|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -          | Reserved                                                                                                                                                                                                                                                         | 0x0   |
| 11    | R    | R_GEN_CALL | Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling controller or when the CPU reads bit 0 of the I2C_CLR_GEN_CALL register. The controller stores the received data in the Rx buffer. | 0x0   |

Table 335: I2C\_INTR\_STAT\_REG (0x5000142C)

| Bit | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10  | R    | R_START_DET | Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 9   | R    | R_STOP_DET  | Indicates whether a STOP condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 8   | R    | R_ACTIVITY  | This bit captures I2C Ctrl activity and stays set until it is cleared. There are four ways to clear it:<br>=> Disabling the I2C Ctrl<br>=> Reading the IC_CLR_ACTIVITY register<br>=> Reading the IC_CLR_INTR register<br>=> System reset<br>Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the controller module is idle, this bit remains set until cleared, indicating that there was activity on the bus.                                                                                                                                                                                                       | 0x0   |
| 7   | R    | R_RX_DONE   | When the controller is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 6   | R    | R_TX_ABRT   | This bit indicates if the controller, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a "transmit abort".<br>When this bit is set to 1, the I2C_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places.<br>NOTE: The controller flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register I2C_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface. | 0x0   |
| 5   | R    | R_RD_REQ    | This bit is set to 1 when the controller is acting as a slave and another I2C master is attempting to read data from the controller. The controller holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the I2C_DATA_CMD register. This bit is set to 0 just after the processor reads the I2C_CLR_RD_REQ register                                                                                                                            | 0x0   |
| 4   | R    | R_TX_EMPTY  | This bit is set to 1 when the transmit buffer is at or below the threshold value set in the I2C_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.                                                                                                                                                       | 0x0   |
| 3   | R    | R_TX_OVER   | Set during transmit if the transmit buffer is filled to 32 and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared                                                                                                                                                                                                                                                                                                                                         | 0x0   |



Table 335: I2C\_INTR\_STAT\_REG (0x5000142C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2   | R    | R_RX_FULL  | Set when the receive buffer reaches or goes above the RX_TL threshold in the I2C_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (I2C_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the I2C_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues. | 0x0   |
| 1   | R    | R_RX_OVER  | Set if the receive buffer is completely filled to 32 and an additional byte is received from an external I2C device. The controller acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.                              | 0x0   |
| 0   | R    | 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. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.                                                                                                                              | 0x0   |

Table 336: I2C\_INTR\_MASK\_REG (0x50001430)

| Bit   | Mode | Symbol      | Description                                                                              | Reset |
|-------|------|-------------|------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -           | Reserved                                                                                 | 0x0   |
| 11    | R/W  | M_GEN_CALL  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 10    | R/W  | M_START_DET | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x0   |
| 9     | R/W  | M_STOP_DET  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x0   |
| 8     | R/W  | M_ACTIVITY  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x0   |
| 7     | R/W  | M_RX_DONE   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 6     | R/W  | M_TX_ABRT   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 5     | R/W  | M_RD_REQ    | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 4     | R/W  | M_TX_EMPTY  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 3     | R/W  | M_TX_OVER   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 2     | R/W  | M_RX_FULL   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 1     | R/W  | M_RX_OVER   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 0     | R/W  | M_RX_UNDER  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |

Table 337: I2C\_RAW\_INTR\_STAT\_REG (0x50001434)

| Bit   | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 11    | R    | GEN_CALL  | Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling controller or when the CPU reads bit 0 of the I2C_CLR_GEN_CALL register. I2C Ctrl stores the received data in the Rx buffer.                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 10    | R    | START_DET | Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 9     | R    | STOP_DET  | Indicates whether a STOP condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 8     | R    | ACTIVITY  | This bit captures I2C Ctrl activity and stays set until it is cleared. There are four ways to clear it:<br>=> Disabling the I2C Ctrl<br>=> Reading the IC_CLR_ACTIVITY register<br>=> Reading the IC_CLR_INTR register<br>=> System reset<br>Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the controller module is idle, this bit remains set until cleared, indicating that there was activity on the bus.                                                                                                                                                                                                       | 0x0   |
| 7     | R    | RX_DONE   | When the controller is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 6     | R    | TX_ABRT   | This bit indicates if the controller, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a "transmit abort".<br>When this bit is set to 1, the I2C_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places.<br>NOTE: The controller flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register I2C_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface. | 0x0   |
| 5     | R    | RD_REQ    | This bit is set to 1 when I2C Ctrl is acting as a slave and another I2C master is attempting to read data from the controller. The controller holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the I2C_DATA_CMD register. This bit is set to 0 just after the processor reads the I2C_CLR_RD_REQ register                                                                                                                                  | 0x0   |
| 4     | R    | TX_EMPTY  | This bit is set to 1 when the transmit buffer is at or below the threshold value set in the I2C_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.                                                                                                                                                       | 0x0   |

Table 337: I2C\_RAW\_INTR\_STAT\_REG (0x50001434)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3   | R    | TX_OVER  | Set during transmit if the transmit buffer is filled to 32 and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared                                                                                                        | 0x0   |
| 2   | R    | RX_FULL  | Set when the receive buffer reaches or goes above the RX_TL threshold in the I2C_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (I2C_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the I2C_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues. | 0x0   |
| 1   | R    | RX_OVER  | Set if the receive buffer is completely filled to 32 and an additional byte is received from an external I2C device. The controller acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.                              | 0x0   |
| 0   | 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. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.                                                                                                                              | 0x0   |

Table 338: I2C\_RX\_TL\_REG (0x50001438)

| Bit  | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 4:0  | R/W  | RX_TL  | Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in I2C_RAW_INTR_STAT register). The valid range is 0-31, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 31 sets the threshold for 32 entries. | 0x0   |

Table 339: I2C\_TX\_TL\_REG (0x5000143C)

| Bit  | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 4:0  | R/W  | RX_TL  | Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in I2C_RAW_INTR_STAT register). The valid range is 0-31, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 31 sets the threshold for 32 entries.. | 0x0   |

Table 340: I2C\_CLR\_INTR\_REG (0x50001440)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                       | Reset |
|------|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                          | 0x0   |
| 0    | R    | CLR_INTR | Read this register to clear the combined interrupt, all individual interrupts, and the I2C_TX_ABORT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the I2C_TX_ABORT_SOURCE register for an exception to clearing I2C_TX_ABORT_SOURCE | 0x0   |

Table 341: I2C\_CLR\_RX\_UNDER\_REG (0x50001444)

| Bit  | Mode | Symbol       | Description                                                                                   | Reset |
|------|------|--------------|-----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                      | 0x0   |
| 0    | R    | CLR_RX_UNDER | Read this register to clear the RX_UNDER interrupt (bit 0) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 342: I2C\_CLR\_RX\_OVER\_REG (0x50001448)

| Bit  | Mode | Symbol      | Description                                                                                  | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                     | 0x0   |
| 0    | R    | CLR_RX_OVER | Read this register to clear the RX_OVER interrupt (bit 1) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 343: I2C\_CLR\_TX\_OVER\_REG (0x5000144C)

| Bit  | Mode | Symbol      | Description                                                                                  | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                     | 0x0   |
| 0    | R    | CLR_TX_OVER | Read this register to clear the TX_OVER interrupt (bit 3) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 344: I2C\_CLR\_RD\_REQ\_REG (0x50001450)

| Bit  | Mode | Symbol     | Description                                                                                 | Reset |
|------|------|------------|---------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -          | Reserved                                                                                    | 0x0   |
| 0    | R    | CLR_RD_REQ | Read this register to clear the RD_REQ interrupt (bit 5) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 345: I2C\_CLR\_TX\_ABORT\_REG (0x50001454)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 0    | R    | CLR_TX_ABORT | Read this register to clear the TX_ABORT interrupt (bit 6) of the I2C_RAW_INTR_STAT register, and the I2C_TX_ABORT_SOURCE register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the I2C_TX_ABORT_SOURCE register for an exception to clearing I2C_TX_ABORT_SOURCE. | 0x0   |

Table 346: I2C\_CLR\_RX\_DONE\_REG (0x50001458)

| Bit  | Mode | Symbol      | Description                                                                                  | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                     | 0x0   |
| 0    | R    | CLR_RX_DONE | Read this register to clear the RX_DONE interrupt (bit 7) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 347: I2C\_CLR\_ACTIVITY\_REG (0x5000145C)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 0    | R    | 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. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register | 0x0   |

Table 348: I2C\_CLR\_STOP\_DET\_REG (0x50001460)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 0    | R    | 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. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register. | 0x0   |

Table 349: I2C\_CLR\_START\_DET\_REG (0x50001464)

| Bit  | Mode | Symbol        | Description                                                                                    | Reset |
|------|------|---------------|------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -             | Reserved                                                                                       | 0x0   |
| 0    | R    | CLR_START_DET | Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register. | 0x0   |

Table 350: I2C\_CLR\_GEN\_CALL\_REG (0x50001468)

| Bit  | Mode | Symbol       | Description                                                                                | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                   | 0x0   |
| 0    | R    | CLR_GEN_CALL | Read this register to clear the GEN_CALL interrupt (bit 11) of I2C_RAW_INTR_STAT register. | 0x0   |

Table 351: I2C\_ENABLE\_REG (0x5000146C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 351: I2C\_ENABLE\_REG (0x5000146C)

| Bit | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|-----|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R/W  | CTRL_ENABLE | <p>Controls whether the controller is enabled.</p> <p>0: Disables the controller (TX and RX FIFOs are held in an erased state)</p> <p>1: Enables the controller</p> <p>Software can disable the controller while it is active. However, it is important that care be taken to ensure that the controller is disabled properly. When the controller is disabled, the following occurs:</p> <ul style="list-style-type: none"> <li>* The TX FIFO and RX FIFO get flushed.</li> <li>* Status bits in the IC_INTR_STAT register are still active until the controller goes into IDLE state.</li> </ul> <p>If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the controller stops the current transfer at the end of the current byte and does not acknowledge the transfer.</p> <p>There is a two ic_clk delay when enabling or disabling the controller</p> | 0x0   |

Table 352: I2C\_STATUS\_REG (0x50001470)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                | Reset |
|------|------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                   | 0x0   |
| 6    | R    | SLV_ACTIVITY | <p>Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set.</p> <p>0: Slave FSM is in IDLE state so the Slave part of the controller is not Active</p> <p>1: Slave FSM is not in IDLE state so the Slave part of the controller is Active</p>       | 0x0   |
| 5    | R    | MST_ACTIVITY | <p>Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set.</p> <p>0: Master FSM is in IDLE state so the Master part of the controller is not Active</p> <p>1: Master FSM is not in IDLE state so the Master part of the controller is Active</p> | 0x0   |
| 4    | R    | RFF          | <p>Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.</p> <p>0: Receive FIFO is not full</p> <p>1: Receive FIFO is full</p>                                                          | 0x0   |
| 3    | R    | RFNE         | <p>Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty.</p> <p>0: Receive FIFO is empty</p> <p>1: Receive FIFO is not empty</p>                                                                                       | 0x0   |
| 2    | R    | TFE          | <p>Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.</p> <p>0: Transmit FIFO is not empty</p> <p>1: Transmit FIFO is empty</p>                   | 0x1   |
| 1    | R    | TFNF         | <p>Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.</p> <p>0: Transmit FIFO is full</p> <p>1: Transmit FIFO is not full</p>                                                                                                  | 0x1   |

Table 352: I2C\_STATUS\_REG (0x50001470)

| Bit | Mode | Symbol       | Description          | Reset |
|-----|------|--------------|----------------------|-------|
| 0   | R    | I2C_ACTIVITY | I2C Activity Status. | 0x0   |

Table 353: I2C\_TXFLR\_REG (0x50001474)

| Bit  | Mode | Symbol | Description                                                                                                                 | Reset |
|------|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -      | Reserved                                                                                                                    | 0x0   |
| 5:0  | R    | TXFLR  | Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Size is constrained by the TXFLR value | 0x0   |

Table 354: I2C\_RXFLR\_REG (0x50001478)

| Bit  | Mode | Symbol | Description                                                                                                               | Reset |
|------|------|--------|---------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -      | Reserved                                                                                                                  | 0x0   |
| 5:0  | R    | RXFLR  | Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. Size is constrained by the RXFLR value | 0x0   |

Table 355: I2C\_SDA\_HOLD\_REG (0x5000147C)

| Bit  | Mode | Symbol      | Description   | Reset |
|------|------|-------------|---------------|-------|
| 15:0 | R/W  | IC_SDA_HOLD | SDA Hold time | 0x1   |

Table 356: I2C\_TX\_ABRT\_SOURCE\_REG (0x50001480)

| Bit | Mode | Symbol               | Description                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-----|------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R    | ABRT_SLVRD_INTX      | 1: 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 2IC_DATA_CMD register                                                                                                                                                                                                                                                               | 0x0   |
| 14  | R    | ABRT_SLV_ARBLOST     | 1: Slave lost the bus while transmitting data to a remote master. I2C_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never "owns" the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then the controller no longer own the bus. | 0x0   |
| 13  | R    | ABRT_SLVFLUSH_TXFIFO | 1: 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.                                                                                                                                                                                                                                                                                      | 0x0   |
| 12  | R    | ARB_LOST             | 1: Master has lost arbitration, or if I2C_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Note: I2C can be both master and slave at the same time.                                                                                                                                                                                                                                                 | 0x0   |
| 11  | R    | ABRT_MASTER_DIS      | 1: User tries to initiate a Master operation with the Master mode disabled.                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 10  | R    | ABRT_10B_RD_NO RSTR  | 1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode.                                                                                                                                                                                                                                                                                                      | 0x0   |



Table 356: I2C\_TX\_ABRT\_SOURCE\_REG (0x50001480)

| Bit | Mode | Symbol             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 9   | R    | ABRT_SBYTE_NORSTRT | To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (I2C_CON[5]=1), the SPECIAL bit must be cleared (I2C_TAR[11]), or the GC_OR_START bit must be cleared (I2C_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the user is trying to send a START Byte. | 0x0   |
| 8   | R    | ABRT_HS_NORSTR T   | 1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 7   | R    | ABRT_SBYTE_ACKDET  | 1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 6   | R    | ABRT_HS_ACKDET     | 1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 5   | R    | ABRT_GCALL_READ    | 1: the controller 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).                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 4   | R    | ABRT_GCALL_NOACK   | 1: the controller in master mode sent a General Call and no slave on the bus acknowledged the General Call.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 3   | R    | ABRT_TXDATA_NOACK  | 1: 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 acknowledgement from the remote slave(s).                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 2   | R    | ABRT_10ADDR2_NOACK | 1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 1   | R    | ABRT_10ADDR1_NOACK | 1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 0   | R    | ABRT_7B_ADDR_NOACK | 1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |

Table 357: I2C\_DMA\_CR\_REG (0x50001488)

| Bit | Mode | Symbol | Description                                                                                                                        | Reset |
|-----|------|--------|------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | TDMAE  | Transmit DMA Enable. //This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled | 0x0   |
| 0   | R/W  | RDMAE  | Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled       | 0x0   |

Table 358: I2C\_DMA\_TDLR\_REG (0x5000148C)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                   | Reset |
|-----|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4:0 | R/W  | DMATDL | Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. | 0x0   |

Table 359: I2C\_DMA\_RDLR\_REG (0x50001490)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|-----|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4:0 | R/W  | DMARDL | Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO. | 0x0   |

Table 360: I2C\_SDA\_SETUP\_REG (0x50001494)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7:0  | R/W  | SDA_SETUP | SDA Setup.<br>This register controls the amount of time delay (number of I2C clock periods) between the rising edge of SCL and SDA changing by holding SCL low when I2C block services a read request while operating as a slave-transmitter. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus Specification. This register must be programmed with a value equal to or greater than 2.<br>It is recommended that if the required delay is 1000ns, then for an I2C frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11. Writes to this register succeed only when IC_ENABLE[0] = 0. | 0x64  |

Table 361: I2C\_ACK\_GENERAL\_CALL\_REG (0x50001498)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                       | Reset |
|------|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                          | 0x0   |
| 0    | R/W  | ACK_GEN_CALL | ACK General Call. When set to 1, I2C Ctrl responds with a ACK (by asserting ic_data_oe) when it receives a General Call. When set to 0, the controller does not generate General Call interrupts. | 0x0   |

Table 362: I2C\_ENABLE\_STATUS\_REG (0x5000149C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:3 | -    | -      | Reserved    | 0x0   |

Table 362: I2C\_ENABLE\_STATUS\_REG (0x5000149C)

| Bit | Mode | Symbol                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|-----|------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2   | R    | SLV_RX_DATA_LOST        | Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0. When read as 1, the controller is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the controller has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit is also set to 1. When read as 0, the controller is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.                                                                                                                                                                                                                                                                                                    | 0x0   |
| 1   | R    | SLV_DISABLED_WHILE_BUSY | Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while:<br>(a) I2C Ctrl is receiving the address byte of the Slave-Transmitter operation from a remote master; OR,<br>(b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, the controller is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in I2C Ctrl (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the the controller has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, the controller is deemed to have been disabled when there is master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0. | 0x0   |
| 0   | R    | IC_EN                   | ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, the controller is deemed to be in an enabled state. When read as 0, the controller is deemed completely inactive. NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |

Table 363: I2C\_IC\_FS\_SPKLEN\_REG (0x500014A0)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 363: I2C\_IC\_FS\_SPKLEN\_REG (0x500014A0)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|-----|------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | IC_FS_SPKLEN | This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 2; hardware prevents values less than this being written, and if attempted results in 2 being set. | 0x1   |

Table 364: I2C\_COMP\_PARAM1\_REG (0x500014F4)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 365: I2C\_COMP\_PARAM2\_REG (0x500014F6)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 366: I2C\_COMP\_VERSION\_REG (0x500014F8)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 367: I2C\_COMP2\_VERSION (0x500014FA)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 368: I2C\_COMP\_TYPE\_REG (0x500014FC)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 369: I2C\_COMP\_TYPE2\_REG (0x500014FE)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 370: I2C2\_CON\_REG (0x50001500)

| Bit  | Mode | Symbol            | Description                                                                                                                                                                                                                                                             | Reset |
|------|------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -                 | Reserved                                                                                                                                                                                                                                                                | 0x0   |
| 6    | R/W  | I2C_SLAVE_DISABLE | Slave enabled or disabled after reset is applied, which means software does not have to configure the slave.<br>0=slave is enabled<br>1=slave is disabled<br>Software should ensure that if this bit is written with '0', then bit 0 should also be written with a '0'. | 0x1   |

Table 370: I2C2\_CON\_REG (0x50001500)

| Bit | Mode | Symbol               | Description                                                                                                                                                                                                         | Reset |
|-----|------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | I2C_RESTART_EN       | Determines whether RESTART conditions may be sent when acting as a master<br>0= disable<br>1=enable                                                                                                                 | 0x1   |
| 4   | R/W  | I2C_10BITADDR_MASTER | Controls whether the controller starts its transfers in 7- or 10-bit addressing mode when acting as a master.<br>0= 7-bit addressing<br>1= 10-bit addressing                                                        | 0x1   |
| 3   | R/W  | I2C_10BITADDR_SLAVE  | When acting as a slave, this bit controls whether the controller responds to 7- or 10-bit addresses.<br>0= 7-bit addressing<br>1= 10-bit addressing                                                                 | 0x1   |
| 2:1 | R/W  | I2C_SPEED            | These bits control at which speed the controller operates.<br>1= standard mode (100 kbit/s)<br>2= fast mode (400 kbit/s)                                                                                            | 0x2   |
| 0   | R/W  | I2C_MASTER_MODE      | This bit controls whether the controller master is enabled.<br>0= master disabled<br>1= master enabled<br>Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'. | 0x1   |

Table 371: I2C2\_TAR\_REG (0x50001504)

| Bit   | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-------|------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 11    | R/W  | SPECIAL     | This bit indicates whether software performs a General Call or START BYTE command.<br>0: ignore bit 10 GC_OR_START and use IC_TAR normally<br>1: perform special I2C command as specified in GC_OR_START bit                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 10    | R/W  | 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 controller.<br>0: General Call Address - after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The controller remains in General Call mode until the SPECIAL bit value (bit 11) is cleared.<br>1: START BYTE                                                                                                       | 0x0   |
| 9:0   | R/W  | IC_TAR      | This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits.<br>Note: If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave<br>Writes to this register succeed only when IC_ENABLE[0] is set to 0 | 0x55  |

Table 372: I2C2\_SAR\_REG (0x50001508)

| Bit   | Mode | Symbol | Description                                                                                                                                                                                                                                                                                      | Reset |
|-------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:10 | -    | -      | Reserved                                                                                                                                                                                                                                                                                         | 0x0   |
| 9:0   | R/W  | 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. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. | 0x55  |

Table 373: I2C2\_HS\_MADDR\_REG (0x5000150C)

| Bit | Mode | Symbol     | Description                                                    | Reset |
|-----|------|------------|----------------------------------------------------------------|-------|
| 2:0 | R/W  | IIC_HS_MAR | This bit field holds the value of the I2C HS mode master code. | 0x1   |

Table 374: I2C2\_DATA\_CMD\_REG (0x50001510)

| Bit  | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:9 | -    | -      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 8    | R/W  | CMD    | <p>This bit controls whether a read or a write is performed. This bit does not control the direction when the I2C Ctrl acts as a slave. It controls only the direction when it acts as a master.</p> <p>1 = Read<br/>0 = Write</p> <p>When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a "don't care" because writes to this register are not required. In slave-transmitter mode, a "0" indicates that CPU data is to be transmitted and as DAT or IC_DATA_CMD[7:0]. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the I2C_RAW_INTR_STAT_REG), unless bit 11 (SPECIAL) in the I2C_TAR register has been cleared.</p> <p>If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs.</p> <p>NOTE: It is possible that while attempting a master I2C read transfer on the controller, a RD_REQ interrupt may have occurred simultaneously due to a remote I2C master addressing the controller. In this type of scenario, it ignores the I2C_DATA_CMD write, generates a TX_ABRT interrupt, and waits to service the RD_REQ interrupt</p> | 0x0   |
| 7:0  | R/W  | DAT    | This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the controller. However, when you read this register, these bits return the value of data received on the controller's interface.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |

Table 375: I2C2\_SS\_SCL\_HCNT\_REG (0x50001514)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_SS_SCL_HCNT | <p>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. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.</p> <p>NOTE: This register must not be programmed to a value higher than 65525, because the controller uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of IC_SS_SCL_HCNT + 10.</p> | 0x48  |

Table 376: I2C2\_SS\_SCL\_LCNT\_REG (0x50001518)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_SS_SCL_LCNT | <p>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.</p> <p>This register can be written only when the I2C interface is disabled which corresponds to the I2C_ENABLE register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set.</p> | 0x4F  |

Table 377: I2C2\_FS\_SCL\_HCNT\_REG (0x5000151C)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_FS_SCL_HCNT | <p>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. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. This register can be written only when the I2C interface is disabled, which corresponds to the I2C_ENABLE register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.</p> | 0x8   |



Table 378: I2C2\_FS\_SCL\_LCNT\_REG (0x50001520)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|------|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IC_FS_SCL_LCNT | <p>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. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. This register can be written only when the I2C interface is disabled, which corresponds to the I2C_ENABLE register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the controller. The lower byte must be programmed first. Then the upper byte is programmed.</p> | 0x17  |

Table 379: I2C2\_INTR\_STAT\_REG (0x5000152C)

| Bit   | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|-------|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 11    | R    | R_GEN_CALL  | Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling controller or when the CPU reads bit 0 of the I2C_CLR_GEN_CALL register. The controller stores the received data in the Rx buffer.                                                                                                                                                                                                                                                                                      | 0x0   |
| 10    | R    | R_START_DET | Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 9     | R    | R_STOP_DET  | Indicates whether a STOP condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 8     | R    | R_ACTIVITY  | <p>This bit captures I2C Ctrl activity and stays set until it is cleared. There are four ways to clear it:</p> <ul style="list-style-type: none"> <li>=&gt; Disabling the I2C Ctrl</li> <li>=&gt; Reading the IC_CLR_ACTIVITY register</li> <li>=&gt; Reading the IC_CLR_INTR register</li> <li>=&gt; System reset</li> </ul> <p>Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the controller module is idle, this bit remains set until cleared, indicating that there was activity on the bus.</p> | 0x0   |
| 7     | R    | R_RX_DONE   | When the controller is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.                                                                                                                                                                                                                                                                                                                         | 0x0   |

Table 379: I2C2\_INTR\_STAT\_REG (0x5000152C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R    | R_TX_ABRT  | <p>This bit indicates if the controller, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a "transmit abort".</p> <p>When this bit is set to 1, the I2C_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places.</p> <p>NOTE: The controller flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register I2C_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.</p> | 0x0   |
| 5   | R    | R_RD_REQ   | <p>This bit is set to 1 when the controller is acting as a slave and another I2C master is attempting to read data from the controller. The controller holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the I2C_DATA_CMD register. This bit is set to 0 just after the processor reads the I2C_CLR_RD_REQ register</p>                                                                                                                                    | 0x0   |
| 4   | R    | R_TX_EMPTY | <p>This bit is set to 1 when the transmit buffer is at or below the threshold value set in the I2C_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.</p>                                                                                                                                                               | 0x0   |
| 3   | R    | R_TX_OVER  | <p>Set during transmit if the transmit buffer is filled to 32 and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared</p>                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 2   | R    | R_RX_FULL  | <p>Set when the receive buffer reaches or goes above the RX_TL threshold in the I2C_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (I2C_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the I2C_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.</p>                                                                                                                                                                                                                                          | 0x0   |
| 1   | R    | R_RX_OVER  | <p>Set if the receive buffer is completely filled to 32 and an additional byte is received from an external I2C device. The controller acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.</p>                                                                                                                                                                                                                                                                       | 0x0   |
| 0   | R    | R_RX_UNDER | <p>Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.</p>                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |

Table 380: I2C2\_INTR\_MASK\_REG (0x50001530)

| Bit   | Mode | Symbol      | Description                                                                              | Reset |
|-------|------|-------------|------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -           | Reserved                                                                                 | 0x0   |
| 11    | R/W  | M_GEN_CALL  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 10    | R/W  | M_START_DET | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x0   |
| 9     | R/W  | M_STOP_DET  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x0   |
| 8     | R/W  | M_ACTIVITY  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x0   |
| 7     | R/W  | M_RX_DONE   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 6     | R/W  | M_TX_ABRT   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 5     | R/W  | M_RD_REQ    | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 4     | R/W  | M_TX_EMPTY  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 3     | R/W  | M_TX_OVER   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 2     | R/W  | M_RX_FULL   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 1     | R/W  | M_RX_OVER   | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |
| 0     | R/W  | M_RX_UNDER  | These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register. | 0x1   |

Table 381: I2C2\_RAW\_INTR\_STAT\_REG (0x50001534)

| Bit   | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|-------|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 11    | R    | GEN_CALL  | Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling controller or when the CPU reads bit 0 of the I2C_CLR_GEN_CALL register. I2C Ctrl stores the received data in the Rx buffer.                                                                                                                                                                                                    | 0x0   |
| 10    | R    | START_DET | Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                       | 0x0   |
| 9     | R    | STOP_DET  | Indicates whether a STOP condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 8     | R    | ACTIVITY  | This bit captures I2C Ctrl activity and stays set until it is cleared. There are four ways to clear it:<br>=> Disabling the I2C Ctrl<br>=> Reading the IC_CLR_ACTIVITY register<br>=> Reading the IC_CLR_INTR register<br>=> System reset<br>Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the controller module is idle, this bit remains set until cleared, indicating that there was activity on the bus. | 0x0   |

Table 381: I2C2\_RAW\_INTR\_STAT\_REG (0x50001534)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-----|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7   | R    | RX_DONE  | When the controller is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 6   | R    | TX_ABRT  | This bit indicates if the controller, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a "transmit abort".<br>When this bit is set to 1, the I2C_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places.<br>NOTE: The controller flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register I2C_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface. | 0x0   |
| 5   | R    | RD_REQ   | This bit is set to 1 when I2C Ctrl is acting as a slave and another I2C master is attempting to read data from the controller. The controller holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the I2C_DATA_CMD register. This bit is set to 0 just after the processor reads the I2C_CLR_RD_REQ register                                                                                                                                  | 0x0   |
| 4   | R    | TX_EMPTY | This bit is set to 1 when the transmit buffer is at or below the threshold value set in the I2C_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.                                                                                                                                                       | 0x0   |
| 3   | R    | TX_OVER  | Set during transmit if the transmit buffer is filled to 32 and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 2   | R    | RX_FULL  | Set when the receive buffer reaches or goes above the RX_TL threshold in the I2C_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (I2C_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the I2C_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.                                                                                                                                                                                                                                  | 0x0   |
| 1   | R    | RX_OVER  | Set if the receive buffer is completely filled to 32 and an additional byte is received from an external I2C device. The controller acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.                                                                                                                                                                                                                                                               | 0x0   |

Table 381: I2C2\_RAW\_INTR\_STAT\_REG (0x50001534)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                           | Reset |
|-----|------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | 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. If the module is disabled (I2C_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared. | 0x0   |

Table 382: I2C2\_RX\_TL\_REG (0x50001538)

| Bit  | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 4:0  | R/W  | RX_TL  | Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in I2C_RAW_INTR_STAT register). The valid range is 0-31, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 31 sets the threshold for 32 entries. | 0x0   |

Table 383: I2C2\_TX\_TL\_REG (0x5000153C)

| Bit  | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 4:0  | R/W  | RX_TL  | Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in I2C_RAW_INTR_STAT register). The valid range is 0-31, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 31 sets the threshold for 32 entries.. | 0x0   |

Table 384: I2C2\_CLR\_INTR\_REG (0x50001540)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                    | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                       | 0x0   |
| 0    | R    | CLR_INTR | Read this register to clear the combined interrupt, all individual interrupts, and the I2C_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the I2C_TX_ABRT_SOURCE register for an exception to clearing I2C_TX_ABRT_SOURCE | 0x0   |

Table 385: I2C2\_CLR\_RX\_UNDER\_REG (0x50001544)

| Bit  | Mode | Symbol       | Description                                                                                   | Reset |
|------|------|--------------|-----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                      | 0x0   |
| 0    | R    | CLR_RX_UNDER | Read this register to clear the RX_UNDER interrupt (bit 0) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 386: I2C2\_CLR\_RX\_OVER\_REG (0x50001548)

| Bit  | Mode | Symbol      | Description                                                                                  | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                     | 0x0   |
| 0    | R    | CLR_RX_OVER | Read this register to clear the RX_OVER interrupt (bit 1) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 387: I2C2\_CLR\_TX\_OVER\_REG (0x5000154C)

| Bit  | Mode | Symbol      | Description                                                                                  | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                     | 0x0   |
| 0    | R    | CLR_TX_OVER | Read this register to clear the TX_OVER interrupt (bit 3) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 388: I2C2\_CLR\_RD\_REQ\_REG (0x50001550)

| Bit  | Mode | Symbol     | Description                                                                                 | Reset |
|------|------|------------|---------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -          | Reserved                                                                                    | 0x0   |
| 0    | R    | CLR_RD_REQ | Read this register to clear the RD_REQ interrupt (bit 5) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 389: I2C2\_CLR\_TX\_ABRT\_REG (0x50001554)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                          | Reset |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 0    | R    | CLR_TX_ABRT | Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the I2C_TX_ABRT_SOURCE register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the I2C_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE. | 0x0   |

Table 390: I2C2\_CLR\_RX\_DONE\_REG (0x50001558)

| Bit  | Mode | Symbol      | Description                                                                                  | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                     | 0x0   |
| 0    | R    | CLR_RX_DONE | Read this register to clear the RX_DONE interrupt (bit 7) of the I2C_RAW_INTR_STAT register. | 0x0   |

Table 391: I2C2\_CLR\_ACTIVITY\_REG (0x5000155C)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 0    | R    | 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. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register | 0x0   |

Table 392: I2C2\_CLR\_STOP\_DET\_REG (0x50001560)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 0    | R    | 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. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register. | 0x0   |

Table 393: I2C2\_CLR\_START\_DET\_REG (0x50001564)

| Bit  | Mode | Symbol        | Description                                                                                    | Reset |
|------|------|---------------|------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -             | Reserved                                                                                       | 0x0   |
| 0    | R    | CLR_START_DET | Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register. | 0x0   |

Table 394: I2C2\_CLR\_GEN\_CALL\_REG (0x50001568)

| Bit  | Mode | Symbol       | Description                                                                                | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                   | 0x0   |
| 0    | R    | CLR_GEN_CALL | Read this register to clear the GEN_CALL interrupt (bit 11) of I2C_RAW_INTR_STAT register. | 0x0   |

Table 395: I2C2\_ENABLE\_REG (0x5000156C)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 0    | R/W  | CTRL_ENABLE | Controls whether the controller is enabled.<br>0: Disables the controller (TX and RX FIFOs are held in an erased state)<br>1: Enables the controller<br>Software can disable the controller while it is active. However, it is important that care be taken to ensure that the controller is disabled properly. When the controller is disabled, the following occurs:<br>* The TX FIFO and RX FIFO get flushed.<br>* Status bits in the IC_INTR_STAT register are still active until the controller goes into IDLE state.<br>If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the controller stops the current transfer at the end of the current byte and does not acknowledge the transfer.<br>There is a two ic_clk delay when enabling or disabling the controller | 0x0   |

Table 396: I2C2\_STATUS\_REG (0x50001570)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:7 | -    | -      | Reserved    | 0x0   |



Table 396: I2C2\_STATUS\_REG (0x50001570)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                 | Reset |
|-----|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R    | SLV_ACTIVITY | Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set.<br>0: Slave FSM is in IDLE state so the Slave part of the controller is not Active<br>1: Slave FSM is not in IDLE state so the Slave part of the controller is Active       | 0x0   |
| 5   | R    | MST_ACTIVITY | Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set.<br>0: Master FSM is in IDLE state so the Master part of the controller is not Active<br>1: Master FSM is not in IDLE state so the Master part of the controller is Active | 0x0   |
| 4   | R    | RFF          | Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.<br>0: Receive FIFO is not full<br>1: Receive FIFO is full                                                          | 0x0   |
| 3   | R    | RFNE         | Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty.<br>0: Receive FIFO is empty<br>1: Receive FIFO is not empty                                                                                       | 0x0   |
| 2   | R    | TFE          | Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt.<br>0: Transmit FIFO is not empty<br>1: Transmit FIFO is empty                   | 0x1   |
| 1   | R    | TFNF         | Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.<br>0: Transmit FIFO is full<br>1: Transmit FIFO is not full                                                                                                  | 0x1   |
| 0   | R    | I2C_ACTIVITY | I2C Activity Status.                                                                                                                                                                                                                                                                        | 0x0   |

Table 397: I2C2\_TXFLR\_REG (0x50001574)

| Bit  | Mode | Symbol | Description                                                                                                                 | Reset |
|------|------|--------|-----------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -      | Reserved                                                                                                                    | 0x0   |
| 5:0  | R    | TXFLR  | Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Size is constrained by the TXFLR value | 0x0   |

Table 398: I2C2\_RXFLR\_REG (0x50001578)

| Bit  | Mode | Symbol | Description                                                                                                               | Reset |
|------|------|--------|---------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -      | Reserved                                                                                                                  | 0x0   |
| 5:0  | R    | RXFLR  | Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. Size is constrained by the RXFLR value | 0x0   |

Table 399: I2C2\_SDA\_HOLD\_REG (0x5000157C)

| Bit  | Mode | Symbol      | Description   | Reset |
|------|------|-------------|---------------|-------|
| 15:0 | R/W  | IC_SDA_HOLD | SDA Hold time | 0x1   |

Table 400: I2C2\_TX\_ABRT\_SOURCE\_REG (0x50001580)

| Bit | Mode | Symbol               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|-----|------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R    | ABRT_SLVRD_INTX      | 1: 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 2IC_DATA_CMD register                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 14  | R    | ABRT_SLV_ARBLOST     | 1: Slave lost the bus while transmitting data to a remote master.<br>I2C_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never "owns" the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then the controller no longer own the bus.                                                                                                                                                                                  | 0x0   |
| 13  | R    | ABRT_SLVFLUSH_TXFIFO | 1: 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.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 12  | R    | ARB_LOST             | 1: Master has lost arbitration, or if I2C_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Note: I2C can be both master and slave at the same time.                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 11  | R    | ABRT_MASTER_DIS      | 1: User tries to initiate a Master operation with the Master mode disabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 10  | R    | ABRT_10B_RD_NO_RSTRT | 1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 9   | R    | ABRT_SBYTE_NORSTRT   | To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (I2C_CON[5]=1), the SPECIAL bit must be cleared (I2C_TAR[11]), or the GC_OR_START bit must be cleared (I2C_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the user is trying to send a START Byte. | 0x0   |
| 8   | R    | ABRT_HS_NORSTRT      | 1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 7   | R    | ABRT_SBYTE_ACKDET    | 1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 6   | R    | ABRT_HS_ACKDET       | 1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 5   | R    | ABRT_GCALL_READ      | 1: the controller 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).                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 4   | R    | ABRT_GCALL_NOACK     | 1: the controller in master mode sent a General Call and no slave on the bus acknowledged the General Call.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |

Table 400: I2C2\_TX\_ABORT\_SOURCE\_REG (0x50001580)

| Bit | Mode | Symbol             | Description                                                                                                                                                                                                 | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3   | R    | ABRT_TXDATA_NOACK  | 1: 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). | 0x0   |
| 2   | R    | ABRT_10ADDR2_NOACK | 1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.                                                                                    | 0x0   |
| 1   | R    | ABRT_10ADDR1_NOACK | 1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.                                                                                                    | 0x0   |
| 0   | R    | ABRT_7B_ADDR_NOACK | 1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.                                                                                                               | 0x0   |

Table 401: I2C2\_DMA\_CR\_REG (0x50001588)

| Bit | Mode | Symbol | Description                                                                                                                        | Reset |
|-----|------|--------|------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | TDMAE  | Transmit DMA Enable. //This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled | 0x0   |
| 0   | R/W  | RDMAE  | Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled       | 0x0   |

Table 402: I2C2\_DMA\_TDLR\_REG (0x5000158C)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                   | Reset |
|-----|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4:0 | R/W  | DMATDL | Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. | 0x0   |

Table 403: I2C2\_DMA\_RDLR\_REG (0x50001590)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|-----|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4:0 | R/W  | DMARDL | Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE = 1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO. | 0x0   |

Table 404: I2C2\_SDA\_SETUP\_REG (0x50001594)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 404: I2C2\_SDA\_SETUP\_REG (0x50001594)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|-----|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | SDA_SETUP | <p>SDA Setup.</p> <p>This register controls the amount of time delay (number of I2C clock periods) between the rising edge of SCL and SDA changing by holding SCL low when I2C block services a read request while operating as a slave-transmitter. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus Specification. This register must be programmed with a value equal to or greater than 2.</p> <p>It is recommended that if the required delay is 1000ns, then for an I2C frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11. Writes to this register succeed only when IC_ENABLE[0] = 0.</p> | 0x64  |

Table 405: I2C2\_ACK\_GENERAL\_CALL\_REG (0x50001598)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                              | Reset |
|------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -            | Reserved                                                                                                                                                                                                 | 0x0   |
| 0    | R/W  | ACK_GEN_CALL | <p>ACK General Call. When set to 1, I2C Ctrl responds with a ACK (by asserting ic_data_oe) when it receives a General Call. When set to 0, the controller does not generate General Call interrupts.</p> | 0x0   |

Table 406: I2C2\_ENABLE\_STATUS\_REG (0x5000159C)

| Bit  | Mode | Symbol           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset |
|------|------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:3 | -    | -                | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 2    | R    | SLV_RX_DATA_LOST | <p>Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0. When read as 1, the controller is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the controller has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit is also set to 1.</p> <p>When read as 0, the controller is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer.</p> <p>NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.</p> | 0x0   |

Table 406: I2C2\_ENABLE\_STATUS\_REG (0x5000159C)

| Bit | Mode | Symbol                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R    | SLV_DISABLED_WHILE_BUSY | <p>Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while:</p> <p>(a) I2C Ctrl is receiving the address byte of the Slave-Transmitter operation from a remote master; OR,</p> <p>(b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, the controller is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in I2C Ctrl (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect.</p> <p>NOTE: If the remote I2C master terminates the transfer with a STOP condition before the the controller has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1.</p> <p>When read as 0, the controller is deemed to have been disabled when there is master activity, or when the I2C bus is idle.</p> <p>NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.</p> | 0x0   |
| 0   | R    | IC_EN                   | <p>ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, the controller is deemed to be in an enabled state.</p> <p>When read as 0, the controller is deemed completely inactive.</p> <p>NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |

Table 407: I2C2\_IC\_FS\_SPKLEN\_REG (0x500015A0)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7:0  | R/W  | IC_FS_SPKLEN | <p>This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. The minimum valid value is 2; hardware prevents values less than this being written, and if attempted results in 2 being set.</p> | 0x1   |

Table 408: I2C2\_COMP\_PARAM1\_REG (0x500015F4)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 409: I2C2\_COMP\_PARAM2\_REG (0x500015F6)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 410: I2C2\_COMP\_VERSION\_REG (0x500015F8)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 411: I2C2\_COMP2\_VERSION (0x500015FA)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 412: I2C2\_COMP\_TYPE\_REG (0x500015FC)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

Table 413: I2C2\_COMP\_TYPE2\_REG (0x500015FE)

| Bit | Mode | Symbol | Description | Reset |
|-----|------|--------|-------------|-------|
| -   | -    | -      | Undefined   | -     |

## 37.13 KEYBOARD SCAN REGISTER FILE

Table 414: Register map KEYBOARD SCAN

| Address    | Port                  | Description                                          |
|------------|-----------------------|------------------------------------------------------|
| 0x50001600 | KBSCN_CTRL_REG        | Keyboard scanner control register                    |
| 0x50001602 | KBSCN_CTRL2_REG       | Keyboard scanner control 2 register                  |
| 0x50001604 | KBSCN_MATRIX_SIZE_REG | Defines the number of rows and columns of the matrix |
| 0x50001606 | KBSCN_DEBOUNCE_REG    | Defines the debounce time for key press and release  |
| 0x50001608 | KBSCN_STATUS_REG      | keyboard scanner Interrupt status register           |
| 0x5000160A | KBSCN_MESSAGE_KEY_REG | Returns a key message from the message queue         |
| 0x5000160C | KBSCN_P00_MODE_REG    | Defines the keyboard mode for P00                    |
| 0x5000160E | KBSCN_P01_MODE_REG    | Defines the keyboard mode for P01                    |
| 0x50001610 | KBSCN_P02_MODE_REG    | Defines the keyboard mode for P02                    |
| 0x50001612 | KBSCN_P03_MODE_REG    | Defines the keyboard mode for P03                    |
| 0x50001614 | KBSCN_P04_MODE_REG    | Defines the keyboard mode for P04                    |
| 0x50001616 | KBSCN_P05_MODE_REG    | Defines the keyboard mode for P05                    |
| 0x50001618 | KBSCN_P06_MODE_REG    | Defines the keyboard mode for P06                    |
| 0x5000161A | KBSCN_P07_MODE_REG    | Defines the keyboard mode for P07                    |
| 0x5000161C | KBSCN_P10_MODE_REG    | Defines the keyboard mode for P10                    |
| 0x5000161E | KBSCN_P11_MODE_REG    | Defines the keyboard mode for P11                    |
| 0x50001620 | KBSCN_P12_MODE_REG    | Defines the keyboard mode for P12                    |
| 0x50001622 | KBSCN_P13_MODE_REG    | Defines the keyboard mode for P13                    |
| 0x50001624 | KBSCN_P14_MODE_REG    | Defines the keyboard mode for P14                    |
| 0x50001626 | KBSCN_P15_MODE_REG    | Defines the keyboard mode for P15                    |
| 0x50001628 | KBSCN_P16_MODE_REG    | Defines the keyboard mode for P16                    |
| 0x5000162A | KBSCN_P17_MODE_REG    | Defines the keyboard mode for P17                    |
| 0x5000162C | KBSCN_P20_MODE_REG    | Defines the keyboard mode for P20                    |
| 0x5000162E | KBSCN_P21_MODE_REG    | Defines the keyboard mode for P21                    |
| 0x50001630 | KBSCN_P22_MODE_REG    | Defines the keyboard mode for P22                    |
| 0x50001632 | KBSCN_P23_MODE_REG    | Defines the keyboard mode for P23                    |
| 0x50001634 | KBSCN_P24_MODE_REG    | Defines the keyboard mode for P24                    |
| 0x5000163C | KBSCN_P30_MODE_REG    | Defines the keyboard mode for P30                    |
| 0x5000163E | KBSCN_P31_MODE_REG    | Defines the keyboard mode for P31                    |
| 0x50001640 | KBSCN_P32_MODE_REG    | Defines the keyboard mode for P32                    |
| 0x50001642 | KBSCN_P33_MODE_REG    | Defines the keyboard mode for P33                    |
| 0x50001644 | KBSCN_P34_MODE_REG    | Defines the keyboard mode for P34                    |
| 0x50001646 | KBSCN_P35_MODE_REG    | Defines the keyboard mode for P35                    |
| 0x50001648 | KBSCN_P36_MODE_REG    | Defines the keyboard mode for P36                    |
| 0x5000164A | KBSCN_P37_MODE_REG    | Defines the keyboard mode for P37                    |
| 0x5000164C | KBSCN_P40_MODE_REG    | Defines the keyboard mode for P40                    |
| 0x5000164E | KBSCN_P41_MODE_REG    | Defines the keyboard mode for P41                    |
| 0x50001650 | KBSCN_P42_MODE_REG    | Defines the keyboard mode for P42                    |
| 0x50001652 | KBSCN_P43_MODE_REG    | Defines the keyboard mode for P43                    |
| 0x50001654 | KBSCN_P44_MODE_REG    | Defines the keyboard mode for P44                    |
| 0x50001656 | KBSCN_P45_MODE_REG    | Defines the keyboard mode for P45                    |
| 0x50001658 | KBSCN_P46_MODE_REG    | Defines the keyboard mode for P46                    |
| 0x5000165A | KBSCN_P47_MODE_REG    | Defines the keyboard mode for P47                    |



Table 415: KBSCN\_CTRL\_REG (0x50001600)

| Bit   | Mode | Symbol                      | Description                                                                      | Reset |
|-------|------|-----------------------------|----------------------------------------------------------------------------------|-------|
| 14    | R0/W | KBSCN_RESET_FIF<br>O        | '1' reset fifo, read always '0'                                                  | 0x0   |
| 13:12 | R/W  | KBSCN_CLKDIV                | Defines keyboard clk. "00" div/1, "01" div/4, "10" div/16, "11" div/64           | 0x0   |
| 11    | R/W  | KBSCN_INACTIVE_<br>EN       | '1' After inactive time the keyboard scanner stops the key maxtrix scan          | 0x0   |
| 10:4  | R/W  | KBSCN_INACTIVE_<br>TIME     | Defines the inactive time in scan cycles. Value 0 is not allowed                 | 0x0   |
| 3     | R/W  | KBSCN_IRQ_FIFO_<br>MASK     | '1' Enable IRQ for fifo over and under flow                                      | 0x0   |
| 2     | R/W  | KBSCN_IRQ_INACTI<br>VE_MASK | '1' : Enable IRQ for inactive                                                    | 0x0   |
| 1     | R/W  | KBSCN_IRQ_MESS<br>AGE_MASK  | '1' : Enable IRQ for message                                                     | 0x0   |
| 0     | R/W  | KBSCN_EN                    | '1' : Enable keyboard scanner, Auto clear when inactive enable and inactive case | 0x0   |

Table 416: KBSCN\_CTRL2\_REG (0x50001602)

| Bit  | Mode | Symbol                    | Description                                         | Reset |
|------|------|---------------------------|-----------------------------------------------------|-------|
| 15:0 | R/W  | KBSCN_ROW_ACTI<br>VE_TIME | Define the row active time in keyboard clock cycles | 0x0   |

Table 417: KBSCN\_MATRIX\_SIZE\_REG (0x50001604)

| Bit | Mode | Symbol                  | Description                                                                                      | Reset |
|-----|------|-------------------------|--------------------------------------------------------------------------------------------------|-------|
| 8:4 | R/W  | KBSCN_MATRIX_C<br>OLUMN | Defines the number of the columns of the keyboard matrix minus 1. Zero means number of columns 1 | 0x0   |
| 3:0 | R/W  | KBSCN_MATRIX_R<br>OW    | Defines the number of the rows of the keyboard matrix minus 1. Zero means number of rows 1       | 0x0   |

Table 418: KBSCN\_DEBOUNCE\_REG (0x50001606)

| Bit  | Mode | Symbol                          | Description                                                                                            | Reset |
|------|------|---------------------------------|--------------------------------------------------------------------------------------------------------|-------|
| 11:6 | R/W  | KBSCN_DEBOUNCE<br>_PRESS_TIME   | Defines the press debounce time in cycles of full matrix scan. One means no debounce, zero is reserved | 0x2   |
| 5:0  | R/W  | KBSCN_DEBOUNCE<br>_RELEASE_TIME | Defines the press debounce time in cycles of full matrix scan. One means no debounce, zero is reserved | 0x2   |

Table 419: KBSCN\_STATUS\_REG (0x50001608)

| Bit | Mode | Symbol                        | Description                                         | Reset |
|-----|------|-------------------------------|-----------------------------------------------------|-------|
| 8   | R    | KBSCN_FIFO_UNDE<br>RFL        | '1' Fifo Underflow occurred                         | 0x0   |
| 7   | R    | KBSCN_FIFO_OVER<br>FL         | '1' Fifo Overflow occurred                          | 0x0   |
| 6:2 | R    | KBSCN_NUM_MESS<br>AGE         | Defines how many messages there are in the fifo.    | 0x0   |
| 1   | R    | KBSCN_INACTIVE_I<br>RQ_STATUS | There is no keyboard activity for a predefined time | 0x0   |

Table 419: KBSCN\_STATUS\_REG (0x50001608)

| Bit | Mode | Symbol               | Description                                     | Reset |
|-----|------|----------------------|-------------------------------------------------|-------|
| 0   | R    | KBSCN_MES_IRQ_STATUS | There is at least one last message in the fifo. | 0x0   |

Table 420: KBSCN\_MESSAGE\_KEY\_REG (0x5000160A)

| Bit | Mode | Symbol             | Description                                                                                     | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------------|-------|
| 10  | R    | KBSCN_LAST_ENTRY   | '1' : this message is the last of the group message, else '0'.<br>When '1' bits 9:0 are all '1' | 0x0   |
| 9   | R    | KBSCN_KEY_STATE    | '0' : New key state is release<br>'1' : New key state is press                                  | 0x0   |
| 8:4 | R    | KBSCN_KEYID_COLUMN | Defines the column id of key                                                                    | 0x0   |
| 3:0 | R    | KBSCN_KEYID_ROW    | Defines the row id of key                                                                       | 0x0   |

Table 421: KBSCN\_P00\_MODE\_REG (0x5000160C)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 422: KBSCN\_P01\_MODE\_REG (0x5000160E)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 423: KBSCN\_P02\_MODE\_REG (0x50001610)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 424: KBSCN\_P03\_MODE\_REG (0x50001612)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 425: KBSCN\_P04\_MODE\_REG (0x50001614)

| Bit | Mode | Symbol        | Description                          | Reset |
|-----|------|---------------|--------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column  | 0x0   |

Table 425: KBSCN\_P04\_MODE\_REG (0x50001614)

| Bit | Mode | Symbol     | Description                                           | Reset |
|-----|------|------------|-------------------------------------------------------|-------|
| 4:0 | R/W  | KBSCN_MODE | Defines the row/column index that has to be connected | 0x0   |

Table 426: KBSCN\_P05\_MODE\_REG (0x50001616)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 427: KBSCN\_P06\_MODE\_REG (0x50001618)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 428: KBSCN\_P07\_MODE\_REG (0x5000161A)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 429: KBSCN\_P10\_MODE\_REG (0x5000161C)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 430: KBSCN\_P11\_MODE\_REG (0x5000161E)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 431: KBSCN\_P12\_MODE\_REG (0x50001620)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 432: KBSCN\_P13\_MODE\_REG (0x50001622)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 433: KBSCN\_P14\_MODE\_REG (0x50001624)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 434: KBSCN\_P15\_MODE\_REG (0x50001626)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 435: KBSCN\_P16\_MODE\_REG (0x50001628)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 436: KBSCN\_P17\_MODE\_REG (0x5000162A)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 437: KBSCN\_P20\_MODE\_REG (0x5000162C)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 438: KBSCN\_P21\_MODE\_REG (0x5000162E)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 439: KBSCN\_P22\_MODE\_REG (0x50001630)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 440: KBSCN\_P23\_MODE\_REG (0x50001632)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 441: KBSCN\_P24\_MODE\_REG (0x50001634)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 442: KBSCN\_P30\_MODE\_REG (0x5000163C)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 443: KBSCN\_P31\_MODE\_REG (0x5000163E)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 444: KBSCN\_P32\_MODE\_REG (0x50001640)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 445: KBSCN\_P33\_MODE\_REG (0x50001642)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 446: KBSCN\_P34\_MODE\_REG (0x50001644)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 447: KBSCN\_P35\_MODE\_REG (0x50001646)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 448: KBSCN\_P36\_MODE\_REG (0x50001648)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 449: KBSCN\_P37\_MODE\_REG (0x5000164A)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 450: KBSCN\_P40\_MODE\_REG (0x5000164C)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 451: KBSCN\_P41\_MODE\_REG (0x5000164E)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 452: KBSCN\_P42\_MODE\_REG (0x50001650)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 453: KBSCN\_P43\_MODE\_REG (0x50001652)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 454: KBSCN\_P44\_MODE\_REG (0x50001654)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 455: KBSCN\_P45\_MODE\_REG (0x50001656)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 456: KBSCN\_P46\_MODE\_REG (0x50001658)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |

Table 457: KBSCN\_P47\_MODE\_REG (0x5000165A)

| Bit | Mode | Symbol        | Description                                           | Reset |
|-----|------|---------------|-------------------------------------------------------|-------|
| 6   | R/W  | KBSCN_GPIO_EN | '1' GPIO is enable for row or column                  | 0x0   |
| 5   | R/W  | KBSCN_ROW     | '1' GPIO is row, '0' GPIO is column                   | 0x0   |
| 4:0 | R/W  | KBSCN_MODE    | Defines the row/column index that has to be connected | 0x0   |



## 37.14 IR REGISTER FILE

Table 458: Register map IR

| Address    | Port                    | Description                              |
|------------|-------------------------|------------------------------------------|
| 0x50001700 | IR_FREQ_CARRIER_ON_REG  | Defines the carrier signal high duration |
| 0x50001702 | IR_FREQ_CARRIER_OFF_REG | Defines the carrier signal low duration  |
| 0x50001704 | IR_LOGIC_ONE_TIME_REG   | Defines the logic one waveform           |
| 0x50001706 | IR_LOGIC_ZERO_TIME_REG  | Defines the logic zero waveform          |
| 0x50001708 | IR_CTRL_REG             | IR control register                      |
| 0x5000170A | IR_STATUS_REG           | IR status register                       |
| 0x5000170C | IR_REPEAT_TIME_REG      | Defines the repeat time                  |
| 0x5000170E | IR_MAIN_FIFO_REG        | Main fifo write register                 |
| 0x50001710 | IR_REPEAT_FIFO_REG      | Repeat fifo write register               |
| 0x50001712 | IR_IRQ_STATUS_REG       | IR interrupt status register             |

Table 459: IR\_FREQ\_CARRIER\_ON\_REG (0x50001700)

| Bit | Mode | Symbol             | Description                                                                               | Reset |
|-----|------|--------------------|-------------------------------------------------------------------------------------------|-------|
| 9:0 | R/W  | IR_FREQ_CARRIER_ON | Defines the carrier signal high duration in IR_clk cycles. 0x0 is not allowed as a value. | 0x1   |

Table 460: IR\_FREQ\_CARRIER\_OFF\_REG (0x50001702)

| Bit | Mode | Symbol              | Description                                              | Reset |
|-----|------|---------------------|----------------------------------------------------------|-------|
| 9:0 | R/W  | IR_FREQ_CARRIER_OFF | Defines the carrier signal low duration in IR_clk cycles | 0x1   |

Table 461: IR\_LOGIC\_ONE\_TIME\_REG (0x50001704)

| Bit  | Mode | Symbol             | Description                                                    | Reset |
|------|------|--------------------|----------------------------------------------------------------|-------|
| 15:8 | R/W  | IR_LOGIC_ONE_MARK  | Defines the mark duration in carrier clock cycles. Must be >0  | 0x1   |
| 7:0  | R/W  | IR_LOGIC_ONE_SPACE | Defines the space duration in carrier clock cycles. Must be >0 | 0x1   |

Table 462: IR\_LOGIC\_ZERO\_TIME\_REG (0x50001706)

| Bit  | Mode | Symbol              | Description                                                    | Reset |
|------|------|---------------------|----------------------------------------------------------------|-------|
| 15:8 | R/W  | IR_LOGIC_ZERO_MARK  | Defines the mark duration in carrier clock cycles. Must be >0  | 0x1   |
| 7:0  | R/W  | IR_LOGIC_ZERO_SPACE | Defines the space duration in carrier clock cycles. Must be >0 | 0x1   |

Table 463: IR\_CTRL\_REG (0x50001708)

| Bit | Mode | Symbol              | Description                                                                                                          | Reset |
|-----|------|---------------------|----------------------------------------------------------------------------------------------------------------------|-------|
| 8   | R/W  | IR_IRQ_EN           | 1 = Enables the interrupt generation upon TX completion<br>0 = masks out the interrupt generation upon TX completion | 0x0   |
| 7   | R/W  | IR_LOGIC_ONE_FORMAT | 1 = Logic one starts with a Space followed by a Mark<br>0 = Logic one starts with a Mark followed by a Space         | 0x0   |

Table 463: IR\_CTRL\_REG (0x50001708)

| Bit | Mode | Symbol               | Description                                                                                                                                  | Reset |
|-----|------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | IR_LOGIC_ZERO_FORMAT | 1 = Logic zero starts with a Space followed by a Mark<br>0 = Logic zero starts with a Mark followed by a Space                               | 0x0   |
| 5   | R/W  | IR_INVERT_OUTPUT     | 1 = IR output is inverted<br>0 = IR output is not inverted                                                                                   | 0x0   |
| 4   | R/W  | IR_REPEAT_TYPE       | 1 = repeat command is defined at Repeat FIFO<br>0 = repeat command is defined at Code FIFO                                                   | 0x0   |
| 3   | R/W  | IR_TX_START          | 1 = IR transmits a command<br>0 = IR is stopped<br>While this bit is 1 and SW programs it to 0, the code FIFO will be flushed automatically. | 0x0   |
| 2   | R/W  | IR_ENABLE            | 1 = IR block is enabled<br>0 = IR block is disabled and at reset state. This also resets the pointers at the FIFOs                           | 0x0   |
| 1   | R0/W | IR_REP_FIFO_RESET    | 1 = Flush Repeat FIFO (auto clear)                                                                                                           | 0x0   |
| 0   | R0/W | IR_CODE_FIFO_RESET   | 1 = Flush Code FIFO (auto clear)                                                                                                             | 0x0   |

Table 464: IR\_STATUS\_REG (0x5000170A)

| Bit | Mode | Symbol             | Description                                                            | Reset |
|-----|------|--------------------|------------------------------------------------------------------------|-------|
| 10  | R    | IR_BUSY            | 1 = IR generator is busy sending a message<br>0 = IR generator is idle | 0x0   |
| 9:6 | R    | IR_REP_FIFO_WORDS  | Contains the amount of words in Repeat FIFO (updated only on write)    | 0x0   |
| 5:0 | R    | IR_CODE_FIFO_WORDS | Contains the amount of words in Code FIFO (updated only on write)      | 0x0   |

Table 465: IR\_REPEAT\_TIME\_REG (0x5000170C)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                             | Reset |
|------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | IR_REPEAT_TIME | Defines the repeat time in carrier clock cycles. The repeat timer will start counting from the start of the command and will trigger the output of the same command residing in the Code FIFO or the special command residing in the Repeat FIFO as soon as it expires. | 0x0   |

Table 466: IR\_MAIN\_FIFO\_REG (0x5000170E)

| Bit  | Mode | Symbol            | Description               | Reset |
|------|------|-------------------|---------------------------|-------|
| 15:0 | R0/W | IR_CODE_FIFO_DATA | Code FIFO data write port | 0x0   |

Table 467: IR\_REPEAT\_FIFO\_REG (0x50001710)

| Bit  | Mode | Symbol                  | Description                 | Reset |
|------|------|-------------------------|-----------------------------|-------|
| 15:0 | R0/W | IR_REPEAT_FIFO_D<br>ATA | Repeat FIFO data write port | 0x0   |

Table 468: IR\_IRQ\_STATUS\_REG (0x50001712)

| Bit | Mode | Symbol     | Description                         | Reset |
|-----|------|------------|-------------------------------------|-------|
| 0   | R    | IR_IRQ_ACK | When read Interrupt line is cleared | 0x0   |

## 37.15 USB REGISTER FILE

Table 469: Register map USB

| Address    | Port            | Description                                                                   |
|------------|-----------------|-------------------------------------------------------------------------------|
| 0x50001800 | USB_MCTRL_REG   | Main Control Register)                                                        |
| 0x50001802 | USB_XCVDIAG_REG | Transceiver diagnostic Register (for test purpose only)                       |
| 0x50001804 | USB_TCR_REG     | Transceiver configuration Register                                            |
| 0x50001806 | USB_UTR_REG     | USB test Register (for test purpose only)                                     |
| 0x50001808 | USB_FAR_REG     | Function Address Register                                                     |
| 0x5000180A | USB_NFSR_REG    | Node Functional State Register                                                |
| 0x5000180C | USB_MAEV_REG    | Main Event Register                                                           |
| 0x5000180E | USB_MAMSK_REG   | Main Mask Register                                                            |
| 0x50001810 | USB_ALTEV_REG   | Alternate Event Register                                                      |
| 0x50001812 | USB_ALTMSK_REG  | Alternate Mask Register                                                       |
| 0x50001814 | USB_TXEV_REG    | Transmit Event Register                                                       |
| 0x50001816 | USB_TXMSK_REG   | Transmit Mask Register                                                        |
| 0x50001818 | USB_RXEV_REG    | Receive Event Register                                                        |
| 0x5000181A | USB_RXMSK_REG   | Receive Mask Register                                                         |
| 0x5000181C | USB_NAKEV_REG   | NAK Event Register                                                            |
| 0x5000181E | USB_NAKMSK_REG  | NAK Mask Register                                                             |
| 0x50001820 | USB_FWEV_REG    | FIFO Warning Event Register                                                   |
| 0x50001822 | USB_FWMSK_REG   | FIFO Warning Mask Register                                                    |
| 0x50001824 | USB_FNH_REG     | Frame Number High Byte Register                                               |
| 0x50001826 | USB_FNL_REG     | Frame Number Low Byte Register                                                |
| 0x5000183E | USB_UX20CDR_REG | Transceiver 2.0 Configuration and Diagnostics Register(for test purpose only) |
| 0x50001840 | USB_EPC0_REG    | Endpoint Control 0 Register                                                   |
| 0x50001842 | USB_TXD0_REG    | Transmit Data 0 Register                                                      |
| 0x50001844 | USB_TXS0_REG    | Transmit Status 0 Register                                                    |
| 0x50001846 | USB_TXC0_REG    | Transmit command 0 Register                                                   |
| 0x50001848 | USB_EP0_NAK_REG | EP0 INNAK and OUTNAK Register                                                 |
| 0x5000184A | USB_RXD0_REG    | Receive Data 0 Register                                                       |
| 0x5000184C | USB_RXS0_REG    | Receive Status 0 Register                                                     |
| 0x5000184E | USB_RXC0_REG    | Receive Command 0 Register                                                    |
| 0x50001850 | USB_EPC1_REG    | Endpoint Control Register 1                                                   |
| 0x50001852 | USB_TXD1_REG    | Transmit Data Register 1                                                      |
| 0x50001854 | USB_TXS1_REG    | Transmit Status Register 1                                                    |
| 0x50001856 | USB_TXC1_REG    | Transmit Command Register 1                                                   |
| 0x50001858 | USB_EPC2_REG    | Endpoint Control Register 2                                                   |
| 0x5000185A | USB_RXD1_REG    | Receive Data Register,1                                                       |
| 0x5000185C | USB_RXS1_REG    | Receive Status Register 1                                                     |
| 0x5000185E | USB_RXC1_REG    | Receive Command Register 1                                                    |
| 0x50001860 | USB_EPC3_REG    | Endpoint Control Register 3                                                   |
| 0x50001862 | USB_TXD2_REG    | Transmit Data Register 2                                                      |
| 0x50001864 | USB_TXS2_REG    | Transmit Status Register 2                                                    |
| 0x50001866 | USB_TXC2_REG    | Transmit Command Register 2                                                   |
| 0x50001868 | USB_EPC4_REG    | Endpoint Control Register 4                                                   |
| 0x5000186A | USB_RXD2_REG    | Receive Data Register 2                                                       |

Table 469: Register map USB

| Address    | Port                 | Description                  |
|------------|----------------------|------------------------------|
| 0x5000186C | USB_RXS2_REG         | Receive Status Register 2    |
| 0x5000186E | USB_RXC2_REG         | Receive Command Register 2   |
| 0x50001870 | USB_EPC5_REG         | Endpoint Control Register 5  |
| 0x50001872 | USB_TXD3_REG         | Transmit Data Register 3     |
| 0x50001874 | USB_TXS3_REG         | Transmit Status Register 3   |
| 0x50001876 | USB_TXC3_REG         | Transmit Command Register 3  |
| 0x50001878 | USB_EPC6_REG         | Endpoint Control Register 6  |
| 0x5000187A | USB_RXD3_REG         | Receive Data Register 3      |
| 0x5000187C | USB_RXS3_REG         | Receive Status Register 3    |
| 0x5000187E | USB_RXC3_REG         | Receive Command Register 3   |
| 0x500018D0 | USB_DMA_CTRL_REG     | USB DMA control register     |
| 0x500018D4 | USB_CHARGER_CTRL_REG | USB Charger Control Register |
| 0x500018D6 | USB_CHARGER_STAT_REG | USB Charger Status Register  |

Table 470: USB\_MCTRL\_REG (0x50001800)

| Bit  | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|------|------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 4    | R/W  | LSMODE  | Low Speed Mode<br>This bit enables USB 1.5 Mbit/s low speed and swaps D+ and D- pull-up resistors. Changing speed may only be done if USBEN is set to 0.<br>Also D+ and D- rise and fall times are adjusted according to the USB specification.                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 3    | R/W  | USB_NAT | Node Attached<br>This bit indicates that this node is ready to be detected as attached to USB. When cleared to 0 the transceiver forces SE0 on the USB port to prevent the hub (to which this node is connected to) from detecting an attach event. After reset or when the USB node is disabled, this bit is cleared to 0 to give the device time before it must respond to commands. After this bit has been set to 1, the device no longer drives the USB and should be ready to receive Reset signalling from the hub.<br>Note: This bit can only be set if USBEN is '1' | 0x0   |
| 2    | -    | -       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 1    | R/W  | USB_DBG | Debug Mode.<br>When this bit is set, the following registers are writable: Main Event (MAEV), Alternate Event (ALTEV), NAK Event (NAKEV), Transmit Status and Receive Status. Setting the DBG bit forces the node into a locked state. The node states can be read out of the transceiver diagnostic register (XCV-DIAG) at location 0xFF6802 by setting the DIAG bit in the Test Control register (UTR).<br>Note: The operation of CoR bits is not effected by entering Debug mode) Note: This bit can only be set if USBEN is '1'                                          | 0x0   |

Table 470: USB\_MCTRL\_REG (0x50001800)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|-----|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R/W  | USBEN  | <p>USB EnableSetting this bit to 1 enables the Full/Low Speed USB node. If the USBEN bit is cleared to 0, the USB is disabled and the 48 MHz clock within the USB node is stopped. In addition, all USB registers are set to their reset state, except for the mentioned registers in Note 45</p> <p>Note that the transceiver forces SE0 on the bus to prevent the hub to detected the USB node, when it is disabled (not attached).</p> <p>The USBEN bit is cleared to 0 after reset</p> | 0x0   |

Table 471: USB\_XCVDIAG\_REG (0x50001802)

| Bit  | Mode | Symbol       | Description                                                                                                                   | Reset |
|------|------|--------------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -            | Reserved                                                                                                                      | 0x0   |
| 7    | R    | USB_VPIN     | With Bit0 = 1 this bit shows the level of the USB_Dp receive data from transceiver; i.e. D+ <= VSE.                           | 0x0   |
| 6    | R    | USB_VMIN     | With Bit0 = 1 this bit shows the level USB_Dm receive data from transceiver; i.e. D- <= VSE.                                  | 0x0   |
| 5    | R    | USB_RCV      | With Bit0 = 1 this bit shows the differential level of the receive comparator.                                                | 0x0   |
| 4    | -    | -            | Reserved                                                                                                                      | 0x0   |
| 3    | R/W  | USB_XCV_TXEN | With Bit0 = 1, this bit enables test Bits 2,1. Must be kept to '0' for normal operation                                       | 0x0   |
| 2    | R/W  | USB_XCV_TXN  | With Bit3,0 = 1, this bit sets USB_Dm to a high level, independent of LSMODE selection                                        | 0x0   |
| 1    | R/W  | USB_XCV_TXP  | With Bit3,0 = 1, this bit sets USB_Dp to a high level, independent of LSMODE selection                                        | 0x0   |
| 0    | R/W  | USB_XCV_TEST | <p>Enable USB_XCVDIAG_REG</p> <p>0: Normal operation, test bits disabled</p> <p>1: Enable test bits 7,6,5,3,2,1 (Note 48)</p> | 0x0   |

Table 472: USB\_TCR\_REG (0x50001804)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                      | Reset |
|------|------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -        | Reserved                                                                                                                                                                                                         | 0x0   |
| 7:5  | R/W  | USB_VADJ | <p>Reference Voltage/ Threshold voltage AdjustControls the single-ended receiver threshold.</p> <p>Shall not be modified unless instructed by Dialog Semiconductor</p> <p>Only enabled if USB_UTR_REG[7] = 1</p> | 0x4   |
| 4:0  | R/W  | USB_CADJ | <p>Transmitter Current Adjust</p> <p>Controls the driver edge rate control current.</p> <p>Shall not be modified unless instructed by Dialog Semiconductor</p> <p>Only enabled if USB_UTR_REG[7] = 1</p>         | 0x10  |

Table 473: USB\_UTR\_REG (0x50001806)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                  | Reset |
|------|------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                                                                                                                                                                                     | 0x0   |
| 7    | R/W  | USB_DIAG    | Diagnostic enable<br>'0': Normal operational.<br>'1': Access to the USB_XCVDIAG_REG and USB_TCR_REG enabled. For diagnostic purposes only                                                                                                                                    | 0x0   |
| 6    | R/W  | USB_NCRC    | No CRC16<br>When this bit is set to 1, all packets transmitted by the Full/ Low Speed USB node are sent without a trailing CRC16. Receive operations are unaffected. This mode is used to check that CRC errors can be detected by other nodes. For diagnostic purposes only | 0x0   |
| 5    | R/W  | USB_SF      | Short Frame<br>Enables the Frame timer to lock and track, short, non-compliant USB frame sizes. The Short Frame bit should not be set during normal operation. For test purposes only                                                                                        | 0x0   |
| 4:0  | R/W  | USB_UTR_RES | Reserved. Must be kept to '0'                                                                                                                                                                                                                                                | 0x0   |

Table 474: USB\_FAR\_REG (0x50001808)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                  | Reset |
|------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 7    | R/W  | USB_AD_EN | Address Enable<br>When set to 1, USB address field bits 6-0 are used in address comparison (see Address detection on page 218 for a description). When cleared to 0, the device does not respond to any token on the USB bus.<br>Note: If the DEF bit in the Endpoint Control 0 register is set, Endpoint 0 responds to the default address. | 0x0   |
| 6:0  | R/W  | USB_AD    | Address<br>This field holds the 7-bit function address used to transmit and receive all tokens addressed to this device.                                                                                                                                                                                                                     | 0x0   |

Table 475: USB\_NFSR\_REG (0x5000180A)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:2 | -    | -      | Reserved    | 0x0   |



Table 475: USB\_NFSR\_REG (0x5000180A)

| Bit | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-----|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1:0 | R/W  | USB_NFS | <p>The Node Functional State Register reports and controls the current functional state of the USB node.</p> <p>00: NodeReset<br/>This is the USB Reset state. This is entered upon a module reset or by software upon detection of a USB Reset. Upon entry, all endpoint pipes are disabled. DEF in the Endpoint Control 0 (EPC0) register and AD_EN in the Function Address (FAR) register should be cleared by software on entry to this state. On exit, DEF should be reset so the device responds to the default address.</p> <p>01: NodeResume<br/>In this state, resume K signalling is generated. This state should be entered by firmware to initiate a remote wake-up sequence by the device. The node must remain in this state for at least 1 ms and no more than 15 ms.</p> <p>10: NodeOperational<br/>This is the normal operational state. In this state the node is configured for operation on the USB bus.</p> <p>11: NodeSuspend<br/>Suspend state should be entered by firmware on detection of a Suspend event while in Operational state. While in Suspend state, the transceivers operate in their low-power suspend mode. All endpoint controllers and the bits TX_EN, LAST and RX_EN are reset, while all other internal states are frozen. On detection of bus activity, the RESUME bit in the ALTEV register is set. In response, software can cause entry to NodeOperational state.</p> | 0x0   |

Table 476: USB\_MAEV\_REG (0x5000180C)

| Bit   | Mode | Symbol      | Description                                                                                                                                                                                                                                             | Reset |
|-------|------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -           | Reserved                                                                                                                                                                                                                                                | 0x0   |
| 11    | R/W  | USB_CH_EV   | <p>USB Charger event<br/>This bit is set if one of the bits in USB_CHARGER_STAT_REG[2:0] change. This bit is cleared to 0 when if USB_CHARGER_STAT_REG is read.</p>                                                                                     | 0x0   |
| 10    | R/W  | USB_EP0_NAK | <p>Endpoint 0 NAK Event<br/>This bit is an OR of EP0_NAK_REG[EP0_OUTNAK] and EP0_NAK_REG[EP0_INNAK] bits. USB_EP0_NAK is cleared to 0 when EP0_NAK_REG is read.</p>                                                                                     | 0x0   |
| 9     | R/W  | USB_EP0_RX  | <p>Endpoint 0 Receive Event<br/>This bit is a copy of the RXS0[RX_LAST] and is cleared to 0 when this RXS0 register is read.<br/>Note: Since Endpoint 0 implements a store and forward principle, an overrun condition for FIFO0 cannot occur</p>       | 0x0   |
| 8     | R/W  | USB_EP0_TX  | <p>Endpoint 0 Transmit Event<br/>This bit is a copy of the TXS0[TX_DONE] bit and is cleared to 0 when the TXS0 register is read.<br/>Note: Since Endpoint 0 implements a store and forward principle, an underrun condition for FIFO0 cannot occur.</p> | 0x0   |
| 7     | R/W  | USB_INTR    | <p>Master Interrupt Enable<br/>This bit is hardwired to 0 in the Main Event (MAEV) register; bit 7 in the Main Mask (MAMSK) register is the Master Interrupt Enable.</p>                                                                                | 0x0   |

Table 476: USB\_MAEV\_REG (0x5000180C)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                       | Reset |
|-----|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | USB_RX_EV | Receive Event<br>This bit is set to 1 if any of the unmasked bits in the Receive Event (RXEV) register is set to 1. It indicates that a SETUP or OUT transaction has been completed. This bit is cleared to 0 when all of the RX_LAST bits in each Receive Status (RXSn) register and all RXOVRN bits in the RXEV register are cleared to 0.      | 0x0   |
| 5   | R/W  | USB_ULD   | Unlocked/Locked Detected<br>This bit is set to 1, when the frame timer has either entered unlocked condition from a locked condition, or has re-entered a locked condition from an unlocked condition as determined by the UL bit in the Frame Number (FNH or FNL) register. This bit is cleared to 0 when the register is read.                  | 0x0   |
| 4   | R/W  | USB_NAK   | Negative Acknowledge Event<br>This bit indicates that one of the unmasked NAK Event (NAKEV) register bits has been set to 1. This bit is cleared to 0 when the NAKEV register is read.                                                                                                                                                            | 0x0   |
| 3   | R/W  | USB_FRAME | Frame Event<br>This bit is set to 1, if the frame counter is updated with a new value. This can be due to the receipt of a valid SOF packet on the USB or to an artificial update if the frame counter was unlocked or a frame was missed. This bit is cleared to 0 when the register is read.                                                    | 0x0   |
| 2   | R/W  | USB_TX_EV | Transmit Event<br>This bit is set to 1, if any of the unmasked bits in the Transmit Event (TXEV) register (TXFIFOn or TXUNDRNn) is set to 1. Therefore, it indicates that an IN transaction has been completed. This bit is cleared to 0 when all the TX_DONE bits and the TXUNDRN bits in each Transmit Status (TXSn) register are cleared to 0. | 0x0   |
| 1   | R/W  | USB_ALT   | Alternate Event<br>This bit indicates that one of the unmasked ALTEV register bits has been set to 1. This bit is cleared to 0 by reading the ALTEV register.                                                                                                                                                                                     | 0x0   |
| 0   | R/W  | USB_WARN  | Warning Event<br>This bit indicates that one of the unmasked bits in the FIFO Warning Event (FWEV) register has been set to 1. This bit is cleared to 0 by reading the FWEV register.                                                                                                                                                             | 0x0   |

Table 477: USB\_MAMSK\_REG (0x5000180E)

| Bit   | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                   | Reset |
|-------|------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | -    | -             | Reserved                                                                                                                                                                                                                                                                                      | 0x0   |
| 11    | R/W  | USB_M_CH_EV   | The Main Mask Register masks out events reported in the MAEV registers. A bit set to 1, enables the interrupts for the respective event in the MAEV register. If the corresponding bit is cleared to 0, interrupt generation for this event is disabled. Same Bit Definition as MAEV Register | 0x0   |
| 10    | R/W  | USB_M_EP0_NAK | Same Bit Definition as MAEV Register                                                                                                                                                                                                                                                          | 0x0   |
| 9     | R/W  | USB_M_EP0_RX  | Same Bit Definition as MAEV Register                                                                                                                                                                                                                                                          | 0x0   |
| 8     | R/W  | USB_M_EP0_TX  | Same Bit Definition as MAEV Register                                                                                                                                                                                                                                                          | 0x0   |
| 7     | R/W  | USB_M_INTR    | Same Bit Definition as MAEV Register                                                                                                                                                                                                                                                          | 0x0   |
| 6     | R/W  | USB_M_RX_EV   | Same Bit Definition as MAEV Register                                                                                                                                                                                                                                                          | 0x0   |
| 5     | R/W  | USB_M_ULD     | Same Bit Definition as MAEV Register                                                                                                                                                                                                                                                          | 0x0   |

Table 477: USB\_MAMSK\_REG (0x5000180E)

| Bit | Mode | Symbol      | Description                          | Reset |
|-----|------|-------------|--------------------------------------|-------|
| 4   | R/W  | USB_M_NAK   | Same Bit Definition as MAEV Register | 0x0   |
| 3   | R/W  | USB_M_FRAME | Same Bit Definition as MAEV Register | 0x0   |
| 2   | R/W  | USB_M_TX_EV | Same Bit Definition as MAEV Register | 0x0   |
| 1   | R/W  | USB_M_ALT   | Same Bit Definition as MAEV Register | 0x0   |
| 0   | R/W  | USB_M_WARN  | Same Bit Definition as MAEV Register | 0x0   |

Table 478: USB\_ALTEV\_REG (0x50001810)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -          | Reserved                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 7    | R/W  | USB_RESUME | Resume<br>Resume signalling is detected on the USB when the device is in Suspend state (NFS in the NFSR register is set to SUSPEND), and a non IDLE signal is present on the USB, indicating that this device should begin it's wake-up sequence and enter Operational state. This bit is cleared when the register is read.                                          | 0x0   |
| 6    | R/W  | USB_RESET  | Reset<br>This bit is set to 1, when 2.5 $\frac{1}{4}$ s of SEO have been detected on the upstream port. In response, the functional state should be reset (NFS in the NFSR register is set to RESET), where it must remain for at least 100 $\frac{1}{4}$ s. The functional state can then return to Operational state. This bit is cleared when the register is read | 0x0   |
| 5    | R/W  | USB_SD5    | Suspend Detect 5 ms<br>This bit is set to 1 after 5 ms of IDLE have been detected on the upstream port, indicating that this device is permitted to perform a remote wake-up operation. The resume may be initiated under firmware control by writing the resume value to the NFSR register. This bit is cleared when the register is read.                           | 0x0   |
| 4    | R/W  | USB_SD3    | Suspend Detect 3 ms<br>This bit is set to 1 after 3 ms of IDLE have been detected on the upstream port, indicating that the device should be suspended. The suspend occurs under firmware control by writing the suspend value to the Node Functional State (NFSR) register. This bit is cleared when the register is read.                                           | 0x0   |
| 3    | R/W  | USB_EOP    | End of Packet<br>A valid EOP sequence was been detected on the USB. It is used when this device has initiated a Remote wake-up sequence to indicate that the Resume sequence has been acknowledged and completed by the host. This bit is cleared when the register is read.                                                                                          | 0x0   |
| 2    | -    | -          | Reserved                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 1:0  | -    | -          | Reserved                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 479: USB\_ALTMSK\_REG (0x50001812)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 479: USB\_ALTMSK\_REG (0x50001812)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                                                                 | Reset |
|-----|------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7   | R/W  | USB_M_RESUME | A bit set to 1 in this register enables automatic setting of the ALT bit in the MAEV register when the respective event in the ALTEV register occurs. Otherwise, setting MAEV.ALT bit is disabled.<br>Same Bit Definition as ALTEV Register | 0x0   |
| 6   | R/W  | USB_M_RESET  | Same Bit Definition as ALTEV Register                                                                                                                                                                                                       | 0x0   |
| 5   | R/W  | USB_M_SD5    | Same Bit Definition as ALTEV Register                                                                                                                                                                                                       | 0x0   |
| 4   | R/W  | USB_M_SD3    | Same Bit Definition as ALTEV Register                                                                                                                                                                                                       | 0x0   |
| 3   | R/W  | USB_M_EOP    | Same Bit Definition as ALTEV Register                                                                                                                                                                                                       | 0x0   |
| 2   | -    | -            | Reserved                                                                                                                                                                                                                                    | 0x0   |
| 1:0 | -    | -            | Reserved                                                                                                                                                                                                                                    | 0x0   |

Table 480: USB\_TXEV\_REG (0x50001814)

| Bit  | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                      | Reset |
|------|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -             | Reserved                                                                                                                                                                                                                                                                                                         | 0x0   |
| 6:4  | R    | USB_TXUDRRN31 | Transmit Underrun n: 3:1<br>The bit n is a copy of the respective TX_URUN bit from the corresponding Transmit Status register (TXSn). Whenever any of the Transmit FIFOs underflows, the respective TXUDRRN bit is set to 1. These bits are cleared to 0 when the corresponding Transmit Status register is read | 0x0   |
| 3    | -    | -             | Reserved                                                                                                                                                                                                                                                                                                         | 0x0   |
| 2:0  | R    | USB_TXFIFO31  | Transmit FIFO n: 3:1<br>The bit n is a copy of the TX_DONE bit from the corresponding Transmit Status register (TXSn). A bit is set to 1 when the IN transaction for the corresponding transmit endpoint n has been completed. These bits are cleared to 0 when the corresponding TXSn register is read.         | 0x0   |

Table 481: USB\_TXMSK\_REG (0x50001816)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -               | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 6:4  | R/W  | USB_M_TXUDRRN31 | The Transmit Mask Register is used to select the bits of the TXEV registers, which causes the TX_EV bit in the MAEV register to be set to 1. When a bit is set to 1 and the corresponding bit in the TXEV register is set to 1, the TX_EV bit in the MAEV register is set to 1. When cleared to 0, the corresponding bit in the TXEV register does not cause TX_EV to be set to 1. Same Bit Definition as TXEV Register | 0x0   |
| 3    | -    | -               | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 2:0  | R/W  | USB_M_TXFIFO31  | Same Bit Definition as TXEV Register                                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |

Table 482: USB\_RXEV\_REG (0x50001818)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:7 | -    | -      | Reserved    | 0x0   |

Table 482: USB\_RXEV\_REG (0x50001818)

| Bit | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Reset |
|-----|------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6:4 | R    | USB_RXOVRN31 | Receive Overrun n: 3:1<br>The bit n is set to 1 in the event of an overrun condition in the corresponding receive FIFO n. They are cleared to 0 when the register is read. The firmware must check the respective RX_ERR bits that packets received for the other receive endpoints (EP2, EP4 and EP6, ) are not corrupted by errors, as these endpoints support data streaming (packets which are longer than the actual FIFO depth).                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 3   | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 2:0 | R    | USB_RXFIFO31 | Receive FIFO n: 3:1<br>The bit n is set to 1 whenever either RX_ERR or RX_LAST in the respective Receive Status register (RXSn) is set to 1. Reading the corresponding RXSn register automatically clears these bits. The CoR function is disabled, when the Freeze signal is asserted. The USB node discards all packets for Endpoint 0 received with errors. This is necessary in case of retransmission due to media errors, ensuring that a good copy of a SETUP packet is captured. Otherwise, the FIFO may potentially be tied up, holding corrupted data and unable to receive a retransmission of the same packet. If data streaming is used for the receive endpoints (EP2, EP4 and EP6, EP8) the firmware must check the respective RX_ERR bits to ensure the packets received are not corrupted by errors. | 0x0   |

Table 483: USB\_RXMSK\_REG (0x5000181A)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                               | Reset |
|------|------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 6:4  | R/W  | USB_M_RXOVRN31 | The Receive Mask Register is used to select the bits of the RXEV registers, which causes the RX_EV bit in the MAEV register to be set to 1. When set to 1 and the corresponding bit in the RXEV register is set to 1, RX_EV bit in the MAEV register is set to 1. When cleared to 0, the corresponding bit in the RXEV register does not cause RX_EV to be set to 1. Same Bit Definition as RXEV Register | 0x0   |
| 3    | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |
| 2:0  | R/W  | USB_M_RXFIFO31 | Same Bit Definition as RXEV Register                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |

Table 484: USB\_NAKEV\_REG (0x5000181C)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                           | Reset |
|------|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                              | 0x0   |
| 6:4  | R    | USB_OUT31 | OUT n: 3:1<br>The bit n is set to 1 when a NAK handshake is generated for an enabled address/endpoint combination (AD_EN in the FAR register is set to 1 and EP_EN in the EPCx register is set to 1) in response to an OUT token. This bit is not set if NAK is generated as result of an overrun condition. It is cleared when the register is read. | 0x0   |
| 3    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 484: USB\_NAKEV\_REG (0x5000181C)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2:0 | R    | USB_IN31 | IN n: 3:1<br>The bit n is set to 1 when a NAK handshake is generated for an enabled address/endpoint combination (AD_EN in the Function Address, FAR, register is set to 1 and EP_EN in the Endpoint Control, EPCx, register is set to 1) in response to an IN token. This bit is cleared when the register is read. | 0x0   |

Table 485: USB\_NAKMSK\_REG (0x5000181E)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                             | Reset |
|------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -           | Reserved                                                                                                                                                                                                                                | 0x0   |
| 6:4  | R/W  | USB_M_OUT31 | When set and the corresponding bit in the NAKEV register is set, the NAK bit in the MAEV register is set. When cleared, the corresponding bit in the NAKEV register does not cause NAK to be set. Same Bit Definition as NAKEV Register | 0x0   |
| 3    | -    | -           | Reserved                                                                                                                                                                                                                                | 0x0   |
| 2:0  | R/W  | USB_M_IN31  | Same Bit Definition as NAKEV Register                                                                                                                                                                                                   | 0x0   |

Table 486: USB\_FWEV\_REG (0x50001820)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset |
|------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 6:4  | R    | USB_RXWARN31 | Receive Warning n: 3:1<br>The bit n is set to 1 when the respective receive endpoint FIFO reaches the warning limit, as specified by the RFWL bits of the respective EPCx register. This bit is cleared when the warning condition is cleared by either reading data from the FIFO or when the FIFO is flushed.                                                                                                                                                | 0x0   |
| 3    | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 2:0  | R    | USB_TXWARN31 | Transmit Warning n: 3:1<br>The bit n is set to 1 when the respective transmit endpoint FIFO reaches the warning limit, as specified by the TFWL bits of the respective TXCn register, and transmission from the respective endpoint is enabled. This bit is cleared when the warning condition is cleared by either writing new data to the FIFO when the FIFO is flushed, or when transmission is done, as indicated by the TX_DONE bit in the TXSn register. | 0x0   |

Table 487: USB\_FWMSK\_REG (0x50001822)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 6:4  | R/W  | USB_M_RXWARN31 | The FIFO Warning Mask Register selects, which FWEV bits are reported in the MAEV register. A bit set to 1 and the corresponding bit in the FWEV register is set 1, causes the WARN bit in the MAEV register to be set to 1. When cleared to 0, the corresponding bit in the FWEV register does not cause WARN to be set to 1. Same Bit Definition as FWEV Register | 0x0   |
| 3    | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                           | 0x0   |

Table 487: USB\_FWMSK\_REG (0x50001822)

| Bit | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2:0 | R/W  | USB_M_TXWARN31 | The FIFO Warning Mask Register selects, which FWEV bits are reported in the MAEV register. A bit set to 1 and the corresponding bit in the FWEV register is set 1, causes the WARN bit in the MAEV register to be set to 1. When cleared to 0, the corresponding bit in the FWEV register does not cause WARN to be set to 1. Same Bit Definition as FWEV Register | 0x0   |

Table 488: USB\_FNH\_REG (0x50001824)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7    | R    | USB_MF      | Missed SOF Flag<br>This flag is set to 1, when the frame number in a valid received SOF does not match the expected next value, or when an SOF is not received within 12060 bit times. This bit is set by the hardware and is cleared by reading the FNH register.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x1   |
| 6    | R    | USB_UL      | Unlock Flag<br>This bit indicates that at least two frames were received without an expected frame number, or that no valid SOF was received within 12060 bit times. If this bit is set, the frame number from the next valid SOF packet is loaded in FN. This bit is set by the hardware and is cleared by reading the FNH register.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 0x1   |
| 5    | R    | USB_RFC     | Reset Frame Count<br>Writing a 1 to this bit resets the frame number to 00016, after which this bit clears itself to 0 again. This bit always reads 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 4:3  | -    | -           | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 2:0  | R    | USB_FN_10_8 | Frame Number<br>This 3-bit field contains the three most significant bits (MSB) of the current frame number, received in the last SOF packet. If a valid frame number is not received within 12060 bit times (Frame Length Maximum, FLMAX, with tolerance) of the previous change, the frame number is incremented artificially. If two successive frames are missed or are incorrect, the current FN is frozen and loaded with the next frame number from a valid SOF packet.<br>If the frame number low byte was read by firmware before reading the FNH register, the user actually reads the contents of a buffer register which holds the value of the three frame number bits of this register when the low byte was read. Therefore, the correct sequence to read the frame number is: FNL, FNH. Read operations to the FNH register, without first reading the Frame Number Low Byte (FNL) register directly, read the actual value of the three MSBs of the frame number. | 0x0   |

Table 489: USB\_FNL\_REG (0x50001826)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |



Table 489: USB\_FNL\_REG (0x50001826)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                          | Reset |
|-----|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R    | USB_FN | The Frame Number Low Byte Register holds the low byte of the frame number. To ensure consistency, reading this low byte causes the three frame number bits in the FNL register to be locked until this register is read. The correct sequence to read the frame number is: FNL, FNL. | 0x0   |

Table 490: USB\_UX20CDR\_REG (0x5000183E)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                           | Reset |
|------|------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -              | Reserved                                                                                                                                                                                              | 0x0   |
| 7    | R    | RPU_TEST7      | Test bit                                                                                                                                                                                              | 0x0   |
| 6    | R/W  | RPU_TEST_SW2   | 0: Closes SW2 switch to reduced pull-up resistor connected to the USB_Dp and USB_Dm.<br>1: Opens SW2 switch resistor connected to the USB_Dp and USB_Dm (independent of the VBus state).<br>(Note 47) | 0x0   |
| 5    | R/W  | RPU_TEST_SW1   | 0: Enable the pull-up resistor on USB_Dp (SW1 closed)<br>1: Disable the pull-up resistor on USB_Dp (SW1 open)<br>(Independent of the VBus state).<br>(Note 47)                                        | 0x0   |
| 4    | R/W  | RPU_TEST_EN    | Pull-Up Resistor Test Enable<br>0: Normal operation<br>1: Enables the test features controlled by RPU_TEST_SW1, RPU_TEST_SW1DM and RPU_TEST_SW2<br>(Note 47)                                          | 0x0   |
| 3    | -    | -              | Reserved                                                                                                                                                                                              | 0x0   |
| 2    | R/W  | RPU_TEST_SW1DM | 0: Enable the pull-up resistor on USB_Dm (SW1DM closed)<br>1: Disable the pull-up resistor on USB_Dm (SW1DM open)<br>(Independent of the VBus state).<br>(Note 47)                                    | 0x0   |
| 1    | R/W  | RPU_RCDELAY    | Test bit, must be kept 0                                                                                                                                                                              | 0x0   |
| 0    | R/W  | RPU_SSPTEN     | Test bit, must be kept 0                                                                                                                                                                              | 0x0   |

Table 491: USB\_EPC0\_REG (0x50001840)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|------|------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 7    | R/W  | USB_STALL | Stall<br>Setting this bit to 1 causes the chip to generate STALL handshakes under the following conditions:<br>- The transmit FIFO is enabled and an IN token is received.<br>- The receive FIFO is enabled and an OUT token is received.<br>Note: A SETUP token does not cause a STALL handshake to be generated when this bit is set.<br>Upon transmitting the STALL handshake, the RX_LAST and the TX_DONE bits in the respective Receive/Transmit Status registers are set to 1. | 0x0   |

Table 491: USB\_EPC0\_REG (0x50001840)

| Bit | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|-----|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | USB_DEF | <p>Default Address</p> <p>When set to 1, the device responds to the default address regardless of the contents of FAR6-0/EP03-0 fields. When an IN packet is transmitted for the endpoint, the DEF bit is automatically cleared to 0.</p> <p>This bit aids in the transition from default address to assigned address. The transition from the default address 00000000000b to an address assigned during bus enumeration may not occur in the middle of the SET_ADDRESS control sequence. This is necessary to complete the control sequence. However, the address must change immediately after this sequence finishes in order to avoid errors when another control sequence immediately follows the SET_ADDRESS command.</p> <p>On USB reset, the firmware has 10 ms for set-up, and should write 8016 to the FAR register and 0016 to the EPC0 register. On receipt of a SET_ADDRESS command, the firmware must write 4016 to the EPC0 register and (8016 or &lt;assigned_function_address&gt;) to the FAR register. It must then queue a zero length IN packet to complete the status phase of the SET_ADDRESS control sequence.</p> | 0x0   |
| 5:4 | -    | -       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 3:0 | R    | USB_EP  | <p>Endpoint Address</p> <p>This field holds the 4-bit Endpoint address. For Endpoint 0, these bits are hardwired to 0000b. Writing a 1 to any of the EP bits is ignored.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 0x0   |

Table 492: USB\_TXD0\_REG (0x50001842)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                        | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -        | Reserved                                                                                                                                                                                                                                           | 0x0   |
| 7:0  | W    | USB_TXFD | <p>Transmit FIFO Data Byte</p> <p>See Bidirectional Control Endpoint FIFO0 on page 220 for a description of data handling.</p> <p>The firmware is expected to write only the packet payload data. The PID and CRC16 are created automatically.</p> | 0x0   |

Table 493: USB\_TXS0\_REG (0x50001844)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                      | Reset |
|------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 6    | R    | USB_ACK_STAT | <p>Acknowledge Status</p> <p>This bit indicates the status, as received from the host, of the ACK for the packet previously sent. This bit is to be interpreted when TX_DONE is set to 1. It is set to 1, when an ACK is received; otherwise, it remains cleared. This bit is also cleared to 0, when this register is read.</p> | 0x0   |
| 5    | R    | USB_TX_DONE  | <p>Transmission Done</p> <p>When set to 1, this bit indicates that a packet has completed transmission. It is cleared to 0, when this register is read.</p>                                                                                                                                                                      | 0x0   |
| 4:0  | R    | USB_TCOUNT   | <p>Transmission Count</p> <p>This 5-bit field indicates the number of empty bytes available in the FIFO. This field is never larger than 8 for Endpoint 0.</p>                                                                                                                                                                   | 0x8   |

Table 494: USB\_TXC0\_REG (0x50001846)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 4    | R/W  | USB_IGN_IN     | Ignore IN Tokens<br>When this bit is set to 1, the endpoint will ignore any IN tokens directed to its configured address.                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 3    | R/W  | USB_FLUSH      | Flush FIFO<br>Writing a 1 to this bit flushes all data from the control endpoint FIFOs, resets the endpoint to Idle state, clears the FIFO read and write pointer, and then clears itself. If the endpoint is currently using the FIFO0 to transfer data on USB, flushing is delayed until after the transfer is done. It is equivalent to the FLUSH bit in the RXC0 register.                                                                                                                                     | 0x0   |
| 2    | R/W  | USB_TOGGLE_TX0 | Toggle<br>This bit specifies the PID used when transmitting the packet. A value of 0 causes a DATA0 PID to be generated, while a value of 1 causes a DATA1 PID to be generated. This bit is not altered by the hardware.                                                                                                                                                                                                                                                                                           | 0x0   |
| 1    | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 0    | R/W  | USB_TX_EN      | Transmission Enable<br>This bit enables data transmission from the FIFO. It is cleared to 0 by hardware after transmitting a single packet, or a STALL handshake, in response to an IN token. It must be set to 1 by firmware to start packet transmission. The RX_EN bit in the Receive Command 0 (RXC0) register takes precedence over this bit; i.e. if RX_EN is set, TX_EN bit is ignored until RX_EN is reset.<br>Zero length packets are indicated by setting this bit without writing any data to the FIFO. | 0x0   |

Table 495: USB\_EP0\_NAK\_REG (0x50001848)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:2 | -    | -              | Reserved                                                                                                                                                                                                                                                                                                           | 0x0   |
| 1    | R    | USB_EP0_OUTNAK | End point 0 OUT NAK<br>This bit is set to 1 when a NAK handshake is generated for an enabled address/endpoint combination (AD_EN in the FAR register is set to 1) in response to an OUT token. This bit is not set if NAK is generated as result of an overrun condition. It is cleared when the register is read. | 0x0   |
| 0    | R    | USB_EP0_INNAK  | End point 0 IN NAK<br>This bit is set to 1 when a NAK handshake is generated for an enabled address/endpoint combination (AD_EN in the FAR register is set to 1) in response to an IN token. This bit is cleared when the register is read.                                                                        | 0x0   |

Table 496: USB\_RXD0\_REG (0x5000184A)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 496: USB\_RXD0\_REG (0x5000184A)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R    | USB_RXFD | Receive FIFO Data Byte<br>See<br>Bidirectional Control Endpoint FIFO0 on page 220<br>for a description of data handling.<br>The firmware should expect to read only the packet payload<br>data. The PID and CRC16 are removed from the incoming<br>data stream automatically.<br>In TEST mode this register allow read/write access. | 0x0   |

Table 497: USB\_RXS0\_REG (0x5000184C)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                      | Reset |
|------|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 7    | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 6    | R    | USB_SETUP      | Setup<br>This bit indicates that the setup packet has been received.<br>This bit is unchanged for zero length packets. It is cleared to<br>0 when this register is read.                                                                                                                                                         | 0x0   |
| 5    | R    | USB_TOGGLE_RX0 | Toggle<br>This bit specified the PID used when receiving the packet. A<br>value of 0 indicates that the last successfully received packet<br>had a DATA0 PID, while a value of 1 indicates that this<br>packet had a DATA1 PID. This bit is unchanged for zero<br>length packets. It is cleared to 0 when this register is read. | 0x0   |
| 4    | R    | USB_RX_LAST    | Receive Last Bytes<br>This bit indicates that an ACK was sent upon completion of a<br>successful receive operation. This bit is unchanged for zero<br>length packets. It is cleared to 0 when this register is read.                                                                                                             | 0x0   |
| 3:0  | R    | USB_RCOUNT     | Receive Count<br>This 4-bit field contains the number of bytes presently in the<br>RX FIFO. This number is never larger than 8 for Endpoint 0.                                                                                                                                                                                   | 0x0   |

Table 498: USB\_RXC0\_REG (0x5000184E)

| Bit  | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset |
|------|------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 5    | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 4    | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 3    | R/W  | USB_FLUSH     | Flush<br>Writing a 1 to this bit flushes all data from the control end-<br>point FIFOs, resets the endpoint to Idle state, clears the<br>FIFO read and write pointer, and then clears itself. If the end-<br>point is currently using FIFO0 to transfer data on USB, flush-<br>ing is delayed until after the transfer is done. This bit is<br>cleared to 0 on reset. This bit is equivalent to FLUSH in the<br>TXC0 register. | 0x0   |
| 2    | R/W  | USB_IGN_SETUP | Ignore SETUP Tokens<br>When this bit is set to 1, the endpoint ignores any SETUP<br>tokens directed to its configured address.                                                                                                                                                                                                                                                                                                 | 0x0   |
| 1    | R/W  | USB_IGN_OUT   | Ignore OUT Tokens<br>When this bit is set to 1, the endpoint ignores any OUT<br>tokens directed to its configured address.                                                                                                                                                                                                                                                                                                     | 0x0   |

Table 498: USB\_RXC0\_REG (0x5000184E)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-----|------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R/W  | USB_RX_EN | Receive Enable<br>OUT packet reception is disabled after every data packet is received, or when a STALL handshake is returned in response to an OUT token. A 1 must be written to this bit to re-enable data reception. Reception of SETUP packets is always enabled. In the case of back-to-back SETUP packets (for a given endpoint) where a valid SETUP packet is received with no other intervening non-SETUP tokens, the Endpoint Controller discards the new SETUP packet and returns an ACK handshake. If any other reasons prevent the Endpoint Controller from accepting the SETUP packet, it must not generate a handshake. This allows recovery from a condition where the ACK of the first SETUP token was lost by the host. | 0x0   |

Table 499: USB\_EPC1\_REG (0x50001850)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7    | R/W  | USB_STALL | Stall<br>Setting this bit to 1 causes the chip to generate STALL handshakes under the following conditions:<br>The transmit FIFO is enabled and an IN token is received.<br>The receive FIFO is enabled and an OUT token is received.<br>Setting this bit to 1 does not generate a STALL handshake in response to a SETUP token                    | 0x0   |
| 6    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 5    | R/W  | USB_ISO   | Isochronous<br>When this bit is set to 1, the endpoint is isochronous. This implies that no NAK is sent if the endpoint is not ready but enabled; i.e. If an IN token is received and no data is available in the FIFO to transmit, or if an OUT token is received and the FIFO is full since there is no USB handshake for isochronous transfers. | 0x0   |
| 4    | R/W  | USB_EP_EN | Endpoint Enable<br>When this bit is set to 1, the EP[3:0] field is used in address comparison, together with the AD[6:0] field in the FAR register. See Section 36.8 for a description. When cleared to 0, the endpoint does not respond to any token on the USB bus.                                                                              | 0x0   |
| 3:0  | R/W  | USB_EP    | Endpoint Address<br>This 4-bit field holds the endpoint address.                                                                                                                                                                                                                                                                                   | 0x0   |

Table 500: USB\_TXD1\_REG (0x50001852)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 500: USB\_TXD1\_REG (0x50001852)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                            | Reset |
|-----|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | W    | USB_TXFD | Transmit FIFO Data Byte<br>See<br>Transmit Endpoint FIFOs on page 222<br>for a description of endpoint FIFO data handling. The firm-<br>ware is expected to write only the packet payload data. PID<br>and CRC16 are inserted automatically in the transmit data<br>stream.<br>In TEST mode this register allow read/write access via the<br>core bus. | 0x0   |

Table 501: USB\_TXS1\_REG (0x50001854)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7    | R    | USB_TX_URUN  | Transmit FIFO Underrun<br>This bit is set to 1, if the transmit FIFO becomes empty dur-<br>ing a transmission, and no new data is written to the FIFO. If<br>so, the Media Access Controller (MAC) forces a bit stuff error<br>followed by an EOP. This bit is cleared to 0, when this regis-<br>ter is read.                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 6    | R    | USB_ACK_STAT | Acknowledge Status<br>This bit is interpreted when TX_DONE is set. It's function dif-<br>fers depending on whether ISO (ISO in the EPCx register is<br>set) or non-ISO operation (ISO is reset) is used.<br>For non-ISO operation, this bit indicates the acknowledge<br>status (from the host) about the ACK for the previously sent<br>packet. This bit itself is set to 1, when an ACK is received;<br>otherwise, it is cleared to 0.<br>For ISO operation, this bit is set if a frame number LSB<br>match (see IGN_ISOMSK bit in the USB_TXCx_REG)<br>occurs, and data was sent in response to an IN token. Other-<br>wise, this bit is cleared to 0, the FIFO is flushed and<br>TX_DONE is set.<br>This bit is also cleared to 0, when this register is read. | 0x0   |
| 5    | R    | USB_TX_DONE  | Transmission Done<br>When set to 1, this bit indicates that the endpoint responded<br>to a USB packet. Three conditions can cause this bit to be<br>set:<br>A data packet completed transmission in response to an IN<br>token with non-ISO operation.<br>The endpoint sent a STALL handshake in response to an IN<br>token<br>A scheduled ISO frame was transmitted or discarded.<br>This bit is cleared to 0 when this register is read.                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 4:0  | R    | USB_TCOUNT   | Transmission Count<br>This 5-bit field holds the number of empty bytes available in<br>the FIFO. If this number is greater than 31, a value of 31 is<br>actually reported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x1F  |

Table 502: USB\_TXC1\_REG (0x50001856)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 502: USB\_TXC1\_REG (0x50001856)

| Bit | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7   | R/W  | USB_IGN_ISOMSK | Ignore ISO Mask<br>This bit has an effect only if the endpoint is set to be isochronous. If set to 1, this bit disables locking of specific frame numbers with the alternate function of the TOGGLE bit. Thus data is transmitted upon reception of the next IN token. If cleared to 0, data is only transmitted when FNL0 matches TOGGLE. This bit is cleared to 0 after reset.                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 6:5 | R/W  | USB_TFWL       | Transmit FIFO Warning Limit<br>These bits specify how many more bytes can be transmitted from the respective FIFO before an underrun condition occurs. If the number of bytes remaining in the FIFO is equal to or less than the selected warning limit, the TXWARN bit in the FWEV register is set. To avoid interrupts caused by setting this bit while the FIFO is being filled before a transmission begins, TXWARN is only set when transmission from the endpoint is enabled (TX_ENn in the TXCn register is set).<br>TFWL[1:0] :<br>00: TFWL disabled<br>01: Less than 5 bytes remaining in FIFO<br>10: Less than 9 bytes remaining in FIFO<br>11: Less than 17 bytes remaining in FIFO                                                                                                                       | 0x0   |
| 4   | R/W  | USB_RFF        | Refill FIFO<br>Setting the LAST bit to 1 automatically saves the Transmit Read Pointer (TXRP) to a buffer. When the RFF bit is set to 1, the buffered TXRP is reloaded into the TXRP. This allows the user to repeat the last transaction if no ACK was received from the host. If the MAC is currently using the FIFO to transmit, TXRP is reloaded only after the transmission is complete. After reload, this bit is cleared to 0 by hardware.                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 3   | R/W  | USB_FLUSH      | Flush FIFO<br>Writing a 1 to this bit flushes all data from the corresponding transmit FIFO, resets the endpoint to Idle state, and clears both the FIFO read and write pointers. If the MAC is currently using the FIFO to transmit, data is flushed after the transmission is complete. After data flushing, this bit is cleared to 0 by hardware.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 2   | R/W  | USB_TOGGLE_TX  | Toggle<br>The function of this bit differs depending on whether ISO (ISO bit in the EPCn register is set to 1) or non-ISO operation (ISO bit is cleared to 0) is used.<br>For non-ISO operation, it specifies the PID used when transmitting the packet. A value of 0 causes a DATA0 PID to be generated, while a value of 1 causes a DATA1 PID to be generated.<br>For ISO operation, this bit and the LSB of the frame counter (FNL0) act as a mask for the TX_EN bit to allow pre-queuing of packets to specific frame numbers; i.e. transmission is enabled only if bit 0 in the FNL register is set to TOGGLE. If an IN token is not received while this condition is true, the contents of the FIFO are flushed with the next SOF. If the endpoint is set to ISO, data is always transferred with a DATA0 PID. | 0x0   |



Table 502: USB\_TXC1\_REG (0x50001856)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-----|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | USB_LAST  | <p>Last Byte</p> <p>Setting this bit to 1 indicates that the entire packet has been written into the FIFO. This is used especially for streaming data to the FIFO while the actual transmission occurs. If the LAST bit is not set to 1 and the transmit FIFO becomes empty during a transmission, a stuff error followed by an EOP is forced on the bus. Zero length packets are indicated by setting this bit without writing any data to the FIFO. The transmit state machine transmits the payload data, CRC16 and the EOP signal before clearing this bit.</p> | 0x0   |
| 0   | R/W  | USB_TX_EN | <p>Transmission Enable</p> <p>This bit enables data transmission from the FIFO. It is cleared to 0 by hardware after transmitting a single packet or after a STALL handshake in response to an IN token. It must be set to 1 by firmware to start packet transmission.</p>                                                                                                                                                                                                                                                                                          | 0x0   |

Table 503: USB\_EPC2\_REG (0x50001858)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 7    | R/W  | USB_STALL | <p>Stall</p> <p>Setting this bit to 1 causes the chip to generate STALL handshakes under the following conditions:</p> <ul style="list-style-type: none"> <li>The transmit FIFO is enabled and an IN token is received.</li> <li>The receive FIFO is enabled and an OUT token is received.</li> </ul> <p>Setting this bit to 1 does not generate a STALL handshake in response to a SETUP token</p> | 0x0   |
| 6    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 5    | R/W  | USB_ISO   | <p>Isochronous</p> <p>When this bit is set to 1, the endpoint is isochronous. This implies that no NAK is sent if the endpoint is not ready but enabled; i.e. If an IN token is received and no data is available in the FIFO to transmit, or if an OUT token is received and the FIFO is full since there is no USB handshake for isochronous transfers.</p>                                       | 0x0   |
| 4    | R/W  | USB_EP_EN | <p>Endpoint Enable</p> <p>When this bit is set to 1, the EP[3:0] field is used in address comparison, together with the AD[6:0] field in the FAR register. See Section 36.8 for a description. When cleared to 0, the endpoint does not respond to any token on the USB bus.</p>                                                                                                                    | 0x0   |
| 3:0  | R/W  | USB_EP    | <p>Endpoint Address</p> <p>This 4-bit field holds the endpoint address.</p>                                                                                                                                                                                                                                                                                                                         | 0x0   |

Table 504: USB\_RXD1\_REG (0x5000185A)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 504: USB\_RXD1\_REG (0x5000185A)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R    | USB_RXFD | Receive FIFO Data Byte<br>See<br>Receive Endpoint FIFO on page 223<br>for a description of Endpoint FIFO data handling. The firmware should expect to read only the packet payload data. The PID and CRC16 are terminated by the receive state machine.<br>In TEST mode this register allow read/write access via the core bus. | 0x0   |

Table 505: USB\_RXS1\_REG (0x5000185C)

| Bit  | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 7    | R    | USB_RX_ERR    | Receive Error<br>When set to 1, this bit indicates a media error, such as bit-stuffing or CRC. If this bit is set to 1, the firmware must flush the respective FIFO.                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 6    | R    | USB_SETUP     | Setup<br>This bit indicates that the setup packet has been received. It is cleared when this register is read.                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 5    | R    | USB_TOGGLE_RX | Toggle<br>The function of this bit differs depending on whether ISO (ISO in the EPCn register is set) or non-ISO operation (ISO is reset) is used.<br>For non-ISO operation, a value of 0 indicates that the last successfully received packet had a DATA0 PID, while a value of 1 indicates that this packet had a DATA1 PID.<br>For ISO operation, this bit reflects the LSB of the frame number (FNL0) after a packet was successfully received for this endpoint.<br>This bit is reset to 0 by reading the RXSn register. | 0x0   |
| 4    | R    | USB_RX_LAST   | Receive Last<br>This bit indicates that an ACK was sent upon completion of a successful receive operation. This bit is cleared to 0 when this register is read.                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 3:0  | R    | USB_RCOUNT    | Receive Counter<br>This 4-bit field contains the number of bytes presently in the endpoint receive FIFO. If this number is greater than 15, a value of 15 is actually reported.                                                                                                                                                                                                                                                                                                                                               | 0x0   |

Table 506: USB\_RXC1\_REG (0x5000185E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 6:5  | R/W  | USB_RFWL | Receive FIFO Warning Limit<br>These bits specify how many more bytes can be received to the respective FIFO before an overrun condition occurs. If the number of empty bytes remaining in the FIFO is equal to or less than the selected warning limit, the RXWARN bit in the FWEV register is set to 1. RFWL[1:0] :<br>00: RFWL disabled<br>01: Less than 5 bytes remaining in FIFO<br>10: Less than 9 bytes remaining in FIFO<br>11: Less than 17 bytes remaining in FIFO | 0x0   |

Table 506: USB\_RXC1\_REG (0x5000185E)

| Bit | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4   | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 3   | R/W  | USB_FLUSH     | Flush FIFO<br>Writing a 1 to this bit flushes all data from the corresponding receive FIFO, resets the endpoint to Idle state, and resets both the FIFO read and write pointers. If the MAC is currently using the FIFO to receive data, flushing is delayed until after receiving is completed.                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 2   | R/W  | USB_IGN_SETUP | Ignore SETUP Tokens<br>When this bit is set to 1, the endpoint ignores any SETUP tokens directed to its configured address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 1   | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 0   | R/W  | USB_RX_EN     | Receive Enable<br>OUT packet cannot be received after every data packet is received, or when a STALL handshake is returned in response to an OUT token. This bit must be written with a 1 to re-enable data reception. SETUP packets can always be received. In the case of back-to-back SETUP packets (for a given endpoint) where a valid SETUP packet has been received with no other intervening non-SETUP tokens, the receive state machine discards the new SETUP packet and returns an ACK handshake. If, for any other reason, the receive state machine cannot accept the SETUP packet, no HANDSHAKE should be generated. | 0x0   |

Table 507: USB\_EPC3\_REG (0x50001860)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7    | R/W  | USB_STALL | Stall<br>Setting this bit to 1 causes the chip to generate STALL handshakes under the following conditions:<br>The transmit FIFO is enabled and an IN token is received.<br>The receive FIFO is enabled and an OUT token is received.<br>Setting this bit to 1 does not generate a STALL handshake in response to a SETUP token                    | 0x0   |
| 6    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 5    | R/W  | USB_ISO   | Isochronous<br>When this bit is set to 1, the endpoint is isochronous. This implies that no NAK is sent if the endpoint is not ready but enabled; i.e. If an IN token is received and no data is available in the FIFO to transmit, or if an OUT token is received and the FIFO is full since there is no USB handshake for isochronous transfers. | 0x0   |
| 4    | R/W  | USB_EP_EN | Endpoint Enable<br>When this bit is set to 1, the EP[3:0] field is used in address comparison, together with the AD[6:0] field in the FAR register. See Section 36.8 for a description. When cleared to 0, the endpoint does not respond to any token on the USB bus.                                                                              | 0x0   |
| 3:0  | R/W  | USB_EP    | Endpoint Address<br>This 4-bit field holds the endpoint address.                                                                                                                                                                                                                                                                                   | 0x0   |

Table 508: USB\_TXD2\_REG (0x50001862)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 7:0  | W    | USB_TXFD | Transmit FIFO Data Byte<br>See<br>Transmit Endpoint FIFOs on page 222<br>for a description of endpoint FIFO data handling. The firm-<br>ware is expected to write only the packet payload data. PID<br>and CRC16 are inserted automatically in the transmit data<br>stream.<br>In TEST mode this register allow read/write access via the<br>core bus. | 0x0   |

Table 509: USB\_TXS2\_REG (0x50001864)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7    | R    | USB_TX_URUN  | Transmit FIFO Underrun<br>This bit is set to 1, if the transmit FIFO becomes empty dur-<br>ing a transmission, and no new data is written to the FIFO. If<br>so, the Media Access Controller (MAC) forces a bit stuff error<br>followed by an EOP. This bit is cleared to 0, when this regis-<br>ter is read.                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 6    | R    | USB_ACK_STAT | Acknowledge Status<br>This bit is interpreted when TX_DONE is set. It's function dif-<br>fers depending on whether ISO (ISO in the EPCx register is<br>set) or non-ISO operation (ISO is reset) is used.<br>For non-ISO operation, this bit indicates the acknowledge<br>status (from the host) about the ACK for the previously sent<br>packet. This bit itself is set to 1, when an ACK is received;<br>otherwise, it is cleared to 0.<br>For ISO operation, this bit is set if a frame number LSB<br>match (see IGN_ISOMSK bit in the USB_TXCx_REG)<br>occurs, and data was sent in response to an IN token. Other-<br>wise, this bit is cleared to 0, the FIFO is flushed and<br>TX_DONE is set.<br>This bit is also cleared to 0, when this register is read. | 0x0   |
| 5    | R    | USB_TX_DONE  | Transmission Done<br>When set to 1, this bit indicates that the endpoint responded<br>to a USB packet. Three conditions can cause this bit to be<br>set:<br>A data packet completed transmission in response to an IN<br>token with non-ISO operation.<br>The endpoint sent a STALL handshake in response to an IN<br>token<br>A scheduled ISO frame was transmitted or discarded.<br>This bit is cleared to 0 when this register is read.                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 4:0  | R    | USB_TCOUNT   | Transmission Count<br>This 5-bit field holds the number of empty bytes available in<br>the FIFO. If this number is greater than 31, a value of 31 is<br>actually reported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x1F  |

Table 510: USB\_TXC2\_REG (0x50001866)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 510: USB\_TXC2\_REG (0x50001866)

| Bit | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7   | R/W  | USB_IGN_ISOMSK | Ignore ISO Mask<br>This bit has an effect only if the endpoint is set to be isochronous. If set to 1, this bit disables locking of specific frame numbers with the alternate function of the TOGGLE bit. Thus data is transmitted upon reception of the next IN token. If cleared to 0, data is only transmitted when FNL0 matches TOGGLE. This bit is cleared to 0 after reset.                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 6:5 | R/W  | USB_TFWL       | Transmit FIFO Warning Limit<br>These bits specify how many more bytes can be transmitted from the respective FIFO before an underrun condition occurs. If the number of bytes remaining in the FIFO is equal to or less than the selected warning limit, the TXWARN bit in the FWEV register is set. To avoid interrupts caused by setting this bit while the FIFO is being filled before a transmission begins, TXWARN is only set when transmission from the endpoint is enabled (TX_ENn in the TXCn register is set).<br>TFWL[1:0] :<br>00: TFWL disabled<br>01: Less than 5 bytes remaining in FIFO<br>10: Less than 9 bytes remaining in FIFO<br>11: Less than 17 bytes remaining in FIFO                                                                                                                       | 0x0   |
| 4   | R/W  | USB_RFF        | Refill FIFO<br>Setting the LAST bit to 1 automatically saves the Transmit Read Pointer (TXRP) to a buffer. When the RFF bit is set to 1, the buffered TXRP is reloaded into the TXRP. This allows the user to repeat the last transaction if no ACK was received from the host. If the MAC is currently using the FIFO to transmit, TXRP is reloaded only after the transmission is complete. After reload, this bit is cleared to 0 by hardware.                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 3   | R/W  | USB_FLUSH      | Flush FIFO<br>Writing a 1 to this bit flushes all data from the corresponding transmit FIFO, resets the endpoint to Idle state, and clears both the FIFO read and write pointers. If the MAC is currently using the FIFO to transmit, data is flushed after the transmission is complete. After data flushing, this bit is cleared to 0 by hardware.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 2   | R/W  | USB_TOGGLE_TX  | Toggle<br>The function of this bit differs depending on whether ISO (ISO bit in the EPCn register is set to 1) or non-ISO operation (ISO bit is cleared to 0) is used.<br>For non-ISO operation, it specifies the PID used when transmitting the packet. A value of 0 causes a DATA0 PID to be generated, while a value of 1 causes a DATA1 PID to be generated.<br>For ISO operation, this bit and the LSB of the frame counter (FNL0) act as a mask for the TX_EN bit to allow pre-queuing of packets to specific frame numbers; i.e. transmission is enabled only if bit 0 in the FNL register is set to TOGGLE. If an IN token is not received while this condition is true, the contents of the FIFO are flushed with the next SOF. If the endpoint is set to ISO, data is always transferred with a DATA0 PID. | 0x0   |

Table 510: USB\_TXC2\_REG (0x50001866)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-----|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | USB_LAST  | <p>Last Byte</p> <p>Setting this bit to 1 indicates that the entire packet has been written into the FIFO. This is used especially for streaming data to the FIFO while the actual transmission occurs. If the LAST bit is not set to 1 and the transmit FIFO becomes empty during a transmission, a stuff error followed by an EOP is forced on the bus. Zero length packets are indicated by setting this bit without writing any data to the FIFO. The transmit state machine transmits the payload data, CRC16 and the EOP signal before clearing this bit.</p> | 0x0   |
| 0   | R/W  | USB_TX_EN | <p>Transmission Enable</p> <p>This bit enables data transmission from the FIFO. It is cleared to 0 by hardware after transmitting a single packet or after a STALL handshake in response to an IN token. It must be set to 1 by firmware to start packet transmission.</p>                                                                                                                                                                                                                                                                                          | 0x0   |

Table 511: USB\_EPC4\_REG (0x50001868)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 7    | R/W  | USB_STALL | <p>Stall</p> <p>Setting this bit to 1 causes the chip to generate STALL handshakes under the following conditions:</p> <ul style="list-style-type: none"> <li>The transmit FIFO is enabled and an IN token is received.</li> <li>The receive FIFO is enabled and an OUT token is received.</li> </ul> <p>Setting this bit to 1 does not generate a STALL handshake in response to a SETUP token</p> | 0x0   |
| 6    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 5    | R/W  | USB_ISO   | <p>Isochronous</p> <p>When this bit is set to 1, the endpoint is isochronous. This implies that no NAK is sent if the endpoint is not ready but enabled; i.e. If an IN token is received and no data is available in the FIFO to transmit, or if an OUT token is received and the FIFO is full since there is no USB handshake for isochronous transfers.</p>                                       | 0x0   |
| 4    | R/W  | USB_EP_EN | <p>Endpoint Enable</p> <p>When this bit is set to 1, the EP[3:0] field is used in address comparison, together with the AD[6:0] field in the FAR register. See Section 36.8 for a description. When cleared to 0, the endpoint does not respond to any token on the USB bus.</p>                                                                                                                    | 0x0   |
| 3:0  | R/W  | USB_EP    | <p>Endpoint Address</p> <p>This 4-bit field holds the endpoint address.</p>                                                                                                                                                                                                                                                                                                                         | 0x0   |

Table 512: USB\_RXD2\_REG (0x5000186A)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 512: USB\_RXD2\_REG (0x5000186A)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                   | Reset |
|-----|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R    | USB_RXFD | Receive FIFO Data Byte<br>See<br>Receive Endpoint FIFO on page 223<br>for a description of Endpoint FIFO data handling. The firm-<br>ware should expect to read only the packet payload data.<br>The PID and CRC16 are terminated by the receive state<br>machine.<br>In TEST mode this register allow read/write access via the<br>core bus. | 0x0   |

Table 513: USB\_RXS2\_REG (0x5000186C)

| Bit  | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|------|------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 7    | R    | USB_RX_ERR    | Receive Error<br>When set to 1, this bit indicates a media error, such as bit-<br>stuffing or CRC. If this bit is set to 1, the firmware must flush<br>the respective FIFO.                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 6    | R    | USB_SETUP     | Setup<br>This bit indicates that the setup packet has been received. It<br>is cleared when this register is read.                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 5    | R    | USB_TOGGLE_RX | Toggle<br>The function of this bit differs depending on whether ISO<br>(ISO in the EPCn register is set) or non-ISO operation (ISO<br>is reset) is used.<br>For non-ISO operation, a value of 0 indicates that the last<br>successfully received packet had a DATA0 PID, while a<br>value of 1 indicates that this packet had a DATA1 PID.<br>For ISO operation, this bit reflects the LSB of the frame num-<br>ber (FNL0) after a packet was successfully received for this<br>endpoint.<br>This bit is reset to 0 by reading the RXSn register. | 0x0   |
| 4    | R    | USB_RX_LAST   | Receive Last<br>This bit indicates that an ACK was sent upon completion of a<br>successful receive operation. This bit is cleared to 0 when<br>this register is read.                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 3:0  | R    | USB_RCOUNT    | Receive Counter<br>This 4-bit field contains the number of bytes presently in the<br>endpoint receive FIFO. If this number is greater than 15, a<br>value of 15 is actually reported.                                                                                                                                                                                                                                                                                                                                                             | 0x0   |

Table 514: USB\_RXC2\_REG (0x5000186E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 6:5  | R/W  | USB_RFWL | Receive FIFO Warning Limit<br>These bits specify how many more bytes can be received to<br>the respective FIFO before an overrun condition occurs. If<br>the number of empty bytes remaining in the FIFO is equal to<br>or less than the selected warning limit, the RXWARN bit in<br>the FWEV register is set to 1. RFWL[1:0] :<br>00: RFWL disabled<br>01: Less than 5 bytes remaining in FIFO<br>10: Less than 9 bytes remaining in FIFO<br>11: Less than 17 bytes remaining in FIFO | 0x0   |



Table 514: USB\_RXC2\_REG (0x5000186E)

| Bit | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4   | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 3   | R/W  | USB_FLUSH     | Flush FIFO<br>Writing a 1 to this bit flushes all data from the corresponding receive FIFO, resets the endpoint to Idle state, and resets both the FIFO read and write pointers. If the MAC is currently using the FIFO to receive data, flushing is delayed until after receiving is completed.                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 2   | R/W  | USB_IGN_SETUP | Ignore SETUP Tokens<br>When this bit is set to 1, the endpoint ignores any SETUP tokens directed to its configured address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 1   | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 0   | R/W  | USB_RX_EN     | Receive Enable<br>OUT packet cannot be received after every data packet is received, or when a STALL handshake is returned in response to an OUT token. This bit must be written with a 1 to re-enable data reception. SETUP packets can always be received. In the case of back-to-back SETUP packets (for a given endpoint) where a valid SETUP packet has been received with no other intervening non-SETUP tokens, the receive state machine discards the new SETUP packet and returns an ACK handshake. If, for any other reason, the receive state machine cannot accept the SETUP packet, no HANDSHAKE should be generated. | 0x0   |

Table 515: USB\_EPC5\_REG (0x50001870)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7    | R/W  | USB_STALL | Stall<br>Setting this bit to 1 causes the chip to generate STALL handshakes under the following conditions:<br>The transmit FIFO is enabled and an IN token is received.<br>The receive FIFO is enabled and an OUT token is received.<br>Setting this bit to 1 does not generate a STALL handshake in response to a SETUP token                    | 0x0   |
| 6    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 5    | R/W  | USB_ISO   | Isochronous<br>When this bit is set to 1, the endpoint is isochronous. This implies that no NAK is sent if the endpoint is not ready but enabled; i.e. If an IN token is received and no data is available in the FIFO to transmit, or if an OUT token is received and the FIFO is full since there is no USB handshake for isochronous transfers. | 0x0   |
| 4    | R/W  | USB_EP_EN | Endpoint Enable<br>When this bit is set to 1, the EP[3:0] field is used in address comparison, together with the AD[6:0] field in the FAR register. See Section 36.8 for a description. When cleared to 0, the endpoint does not respond to any token on the USB bus.                                                                              | 0x0   |
| 3:0  | R/W  | USB_EP    | Endpoint Address<br>This 4-bit field holds the endpoint address.                                                                                                                                                                                                                                                                                   | 0x0   |

Table 516: USB\_TXD3\_REG (0x50001872)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 7:0  | W    | USB_TXFD | Transmit FIFO Data Byte<br>See<br>Transmit Endpoint FIFOs on page 222<br>for a description of endpoint FIFO data handling. The firm-<br>ware is expected to write only the packet payload data. PID<br>and CRC16 are inserted automatically in the transmit data<br>stream.<br>In TEST mode this register allow read/write access via the<br>core bus. | 0x0   |

Table 517: USB\_TXS3\_REG (0x50001874)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7    | R    | USB_TX_URUN  | Transmit FIFO Underrun<br>This bit is set to 1, if the transmit FIFO becomes empty dur-<br>ing a transmission, and no new data is written to the FIFO. If<br>so, the Media Access Controller (MAC) forces a bit stuff error<br>followed by an EOP. This bit is cleared to 0, when this regis-<br>ter is read.                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 6    | R    | USB_ACK_STAT | Acknowledge Status<br>This bit is interpreted when TX_DONE is set. It's function dif-<br>fers depending on whether ISO (ISO in the EPCx register is<br>set) or non-ISO operation (ISO is reset) is used.<br>For non-ISO operation, this bit indicates the acknowledge<br>status (from the host) about the ACK for the previously sent<br>packet. This bit itself is set to 1, when an ACK is received;<br>otherwise, it is cleared to 0.<br>For ISO operation, this bit is set if a frame number LSB<br>match (see IGN_ISOMSK bit in the USB_TXCx_REG)<br>occurs, and data was sent in response to an IN token. Other-<br>wise, this bit is cleared to 0, the FIFO is flushed and<br>TX_DONE is set.<br>This bit is also cleared to 0, when this register is read. | 0x0   |
| 5    | R    | USB_TX_DONE  | Transmission Done<br>When set to 1, this bit indicates that the endpoint responded<br>to a USB packet. Three conditions can cause this bit to be<br>set:<br>A data packet completed transmission in response to an IN<br>token with non-ISO operation.<br>The endpoint sent a STALL handshake in response to an IN<br>token<br>A scheduled ISO frame was transmitted or discarded.<br>This bit is cleared to 0 when this register is read.                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 4:0  | R    | USB_TCOUNT   | Transmission Count<br>This 5-bit field holds the number of empty bytes available in<br>the FIFO. If this number is greater than 31, a value of 31 is<br>actually reported.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x1F  |

Table 518: USB\_TXC3\_REG (0x50001876)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 518: USB\_TXC3\_REG (0x50001876)

| Bit | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7   | R/W  | USB_IGN_ISOMSK | Ignore ISO Mask<br>This bit has an effect only if the endpoint is set to be isochronous. If set to 1, this bit disables locking of specific frame numbers with the alternate function of the TOGGLE bit. Thus data is transmitted upon reception of the next IN token. If cleared to 0, data is only transmitted when FNL0 matches TOGGLE. This bit is cleared to 0 after reset.                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |
| 6:5 | R/W  | USB_TFWL       | Transmit FIFO Warning Limit<br>These bits specify how many more bytes can be transmitted from the respective FIFO before an underrun condition occurs. If the number of bytes remaining in the FIFO is equal to or less than the selected warning limit, the TXWARN bit in the FWEV register is set. To avoid interrupts caused by setting this bit while the FIFO is being filled before a transmission begins, TXWARN is only set when transmission from the endpoint is enabled (TX_ENn in the TXCn register is set).<br>TFWL[1:0] :<br>00: TFWL disabled<br>01: Less than 5 bytes remaining in FIFO<br>10: Less than 9 bytes remaining in FIFO<br>11: Less than 17 bytes remaining in FIFO                                                                                                                       | 0x0   |
| 4   | R/W  | USB_RFF        | Refill FIFO<br>Setting the LAST bit to 1 automatically saves the Transmit Read Pointer (TXRP) to a buffer. When the RFF bit is set to 1, the buffered TXRP is reloaded into the TXRP. This allows the user to repeat the last transaction if no ACK was received from the host. If the MAC is currently using the FIFO to transmit, TXRP is reloaded only after the transmission is complete. After reload, this bit is cleared to 0 by hardware.                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 3   | R/W  | USB_FLUSH      | Flush FIFO<br>Writing a 1 to this bit flushes all data from the corresponding transmit FIFO, resets the endpoint to Idle state, and clears both the FIFO read and write pointers. If the MAC is currently using the FIFO to transmit, data is flushed after the transmission is complete. After data flushing, this bit is cleared to 0 by hardware.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 2   | R/W  | USB_TOGGLE_TX  | Toggle<br>The function of this bit differs depending on whether ISO (ISO bit in the EPCn register is set to 1) or non-ISO operation (ISO bit is cleared to 0) is used.<br>For non-ISO operation, it specifies the PID used when transmitting the packet. A value of 0 causes a DATA0 PID to be generated, while a value of 1 causes a DATA1 PID to be generated.<br>For ISO operation, this bit and the LSB of the frame counter (FNL0) act as a mask for the TX_EN bit to allow pre-queuing of packets to specific frame numbers; i.e. transmission is enabled only if bit 0 in the FNL register is set to TOGGLE. If an IN token is not received while this condition is true, the contents of the FIFO are flushed with the next SOF. If the endpoint is set to ISO, data is always transferred with a DATA0 PID. | 0x0   |

Table 518: USB\_TXC3\_REG (0x50001876)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-----|------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 1   | R/W  | USB_LAST  | <p>Last Byte</p> <p>Setting this bit to 1 indicates that the entire packet has been written into the FIFO. This is used especially for streaming data to the FIFO while the actual transmission occurs. If the LAST bit is not set to 1 and the transmit FIFO becomes empty during a transmission, a stuff error followed by an EOP is forced on the bus. Zero length packets are indicated by setting this bit without writing any data to the FIFO. The transmit state machine transmits the payload data, CRC16 and the EOP signal before clearing this bit.</p> | 0x0   |
| 0   | R/W  | USB_TX_EN | <p>Transmission Enable</p> <p>This bit enables data transmission from the FIFO. It is cleared to 0 by hardware after transmitting a single packet or after a STALL handshake in response to an IN token. It must be set to 1 by firmware to start packet transmission.</p>                                                                                                                                                                                                                                                                                          | 0x0   |

Table 519: USB\_EPC6\_REG (0x50001878)

| Bit  | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 7    | R/W  | USB_STALL | <p>Stall</p> <p>Setting this bit to 1 causes the chip to generate STALL handshakes under the following conditions:</p> <ul style="list-style-type: none"> <li>The transmit FIFO is enabled and an IN token is received.</li> <li>The receive FIFO is enabled and an OUT token is received.</li> </ul> <p>Setting this bit to 1 does not generate a STALL handshake in response to a SETUP token</p> | 0x0   |
| 6    | -    | -         | Reserved                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 5    | R/W  | USB_ISO   | <p>Isochronous</p> <p>When this bit is set to 1, the endpoint is isochronous. This implies that no NAK is sent if the endpoint is not ready but enabled; i.e. If an IN token is received and no data is available in the FIFO to transmit, or if an OUT token is received and the FIFO is full since there is no USB handshake for isochronous transfers.</p>                                       | 0x0   |
| 4    | R/W  | USB_EP_EN | <p>Endpoint Enable</p> <p>When this bit is set to 1, the EP[3:0] field is used in address comparison, together with the AD[6:0] field in the FAR register. See Section 36.8 for a description. When cleared to 0, the endpoint does not respond to any token on the USB bus.</p>                                                                                                                    | 0x0   |
| 3:0  | R/W  | USB_EP    | <p>Endpoint Address</p> <p>This 4-bit field holds the endpoint address.</p>                                                                                                                                                                                                                                                                                                                         | 0x0   |

Table 520: USB\_RXD3\_REG (0x5000187A)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 520: USB\_RXD3\_REG (0x5000187A)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:0 | R    | USB_RXFD | Receive FIFO Data Byte<br>See<br>Receive Endpoint FIFO on page 223<br>for a description of Endpoint FIFO data handling. The firmware should expect to read only the packet payload data. The PID and CRC16 are terminated by the receive state machine.<br>In TEST mode this register allow read/write access via the core bus. | 0x0   |

Table 521: USB\_RXS3\_REG (0x5000187C)

| Bit  | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 7    | R    | USB_RX_ERR    | Receive Error<br>When set to 1, this bit indicates a media error, such as bit-stuffing or CRC. If this bit is set to 1, the firmware must flush the respective FIFO.                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 6    | R    | USB_SETUP     | Setup<br>This bit indicates that the setup packet has been received. It is cleared when this register is read.                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 5    | R    | USB_TOGGLE_RX | Toggle<br>The function of this bit differs depending on whether ISO (ISO in the EPCn register is set) or non-ISO operation (ISO is reset) is used.<br>For non-ISO operation, a value of 0 indicates that the last successfully received packet had a DATA0 PID, while a value of 1 indicates that this packet had a DATA1 PID.<br>For ISO operation, this bit reflects the LSB of the frame number (FNL0) after a packet was successfully received for this endpoint.<br>This bit is reset to 0 by reading the RXSn register. | 0x0   |
| 4    | R    | USB_RX_LAST   | Receive Last<br>This bit indicates that an ACK was sent upon completion of a successful receive operation. This bit is cleared to 0 when this register is read.                                                                                                                                                                                                                                                                                                                                                               | 0x0   |
| 3:0  | R    | USB_RCOUNT    | Receive Counter<br>This 4-bit field contains the number of bytes presently in the endpoint receive FIFO. If this number is greater than 15, a value of 15 is actually reported.                                                                                                                                                                                                                                                                                                                                               | 0x0   |

Table 522: USB\_RXC3\_REG (0x5000187E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:7 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 0x0   |
| 6:5  | R/W  | USB_RFWL | Receive FIFO Warning Limit<br>These bits specify how many more bytes can be received to the respective FIFO before an overrun condition occurs. If the number of empty bytes remaining in the FIFO is equal to or less than the selected warning limit, the RXWARN bit in the FWEV register is set to 1. RFWL[1:0] :<br>00: RFWL disabled<br>01: Less than 5 bytes remaining in FIFO<br>10: Less than 9 bytes remaining in FIFO<br>11: Less than 17 bytes remaining in FIFO | 0x0   |

Table 522: USB\_RXC3\_REG (0x5000187E)

| Bit | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4   | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 3   | R/W  | USB_FLUSH     | Flush FIFO<br>Writing a 1 to this bit flushes all data from the corresponding receive FIFO, resets the endpoint to Idle state, and resets both the FIFO read and write pointers. If the MAC is currently using the FIFO to receive data, flushing is delayed until after receiving is completed.                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 2   | R/W  | USB_IGN_SETUP | Ignore SETUP Tokens<br>When this bit is set to 1, the endpoint ignores any SETUP tokens directed to its configured address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 1   | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 0   | R/W  | USB_RX_EN     | Receive Enable<br>OUT packet cannot be received after every data packet is received, or when a STALL handshake is returned in response to an OUT token. This bit must be written with a 1 to re-enable data reception. SETUP packets can always be received. In the case of back-to-back SETUP packets (for a given endpoint) where a valid SETUP packet has been received with no other intervening non-SETUP tokens, the receive state machine discards the new SETUP packet and returns an ACK handshake. If, for any other reason, the receive state machine cannot accept the SETUP packet, no HANDSHAKE should be generated. | 0x0   |

Table 523: USB\_DMA\_CTRL\_REG (0x500018D0)

| Bit | Mode | Symbol     | Description                                                                                                                                                                              | Reset |
|-----|------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | USB_DMA_EN | 0 = USB DMA control off. (Normal operation)<br>1 = USB_DMA on. DMA channels 0 and 1 are connected by USB Endpoint according bits USB_DMA_TX and USB_DMA_RX                               | 0x0   |
| 5:3 | R/W  | USB_DMA_TX | 000 = DMA channels 1 is connected Tx USB Endpoint 1<br>001 = DMA channels 1 is connected Tx USB Endpoint 3<br>010 = DMA channels 1 is connected Tx USB Endpoint 5<br>100, 1xx = Reserved | 0x0   |
| 2:0 | R/W  | USB_DMA_RX | 000 = DMA channels 0 is connected Rx USB Endpoint 2<br>001 = DMA channels 0 is connected Rx USB Endpoint 4<br>010 = DMA channels 0 is connected Rx USB Endpoint 6<br>100, 1xx = Reserved | 0x0   |

Table 524: USB\_CHARGER\_CTRL\_REG (0x500018D4)

| Bit  | Mode | Symbol      | Description                                                           | Reset |
|------|------|-------------|-----------------------------------------------------------------------|-------|
| 15:6 | -    | -           | Reserved                                                              | 0x0   |
| 5    | R/W  | IDM_SINK_ON | 0 = Disable<br>1 = Enable the Idm_sink to USBm                        | 0x0   |
| 4    | R/W  | IDP_SINK_ON | 0 = Disable<br>1 = Enable the Idp_sink to USBp                        | 0x0   |
| 3    | R/W  | VDM_SRC_ON  | 0 = Disable<br>1 = Enable Vdm_src to USBm and USB_DCP_DET status bit. | 0x0   |
| 2    | R/W  | VDP_SRC_ON  | 0 = Disable<br>1 = Enable the Vdp_src to USB_CHG_DET status bit.      | 0x0   |

Table 524: USB\_CHARGER\_CTRL\_REG (0x500018D4)

| Bit | Mode | Symbol        | Description                                                                       | Reset |
|-----|------|---------------|-----------------------------------------------------------------------------------|-------|
| 1   | R/W  | IDP_SRC_ON    | 0 = Disable<br>1 = Enable the Idp_src and Rdm_dwn.                                | 0x0   |
| 0   | R/W  | USB_CHARGE_ON | 0 = Disable USB charger detect circuit.<br>1 = Enable USB charger detect circuit. | 0x0   |

Table 525: USB\_CHARGER\_STAT\_REG (0x500018D6)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                        | Reset |
|------|------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -           | Reserved                                                                                                                                                                                           | 0x0   |
| 5    | R    | USB_DM_VAL2 | 0 = USBm < 2.3V<br>1 = USBm > 2.5V                                                                                                                                                                 | 0x0   |
| 4    | R    | USB_DP_VAL2 | 0: USBp < 2.3V<br>1: USBp > 2.5V                                                                                                                                                                   | 0x0   |
| 3    | R    | USB_DM_VAL  | 0 = USBm < 0.8V<br>1 = USBm > 1.5V (PS2 or Proprietary Charger)                                                                                                                                    | 0x0   |
| 2    | R    | USB_DP_VAL  | 0 = USBp < 0.8V<br>1 = USBp > 1.5V                                                                                                                                                                 | 0x0   |
| 1    | R    | USB_CHG_DET | 0 = Standard downstream or nothing connected.<br>1 = Charging Downstream Port (CDP) or Dedicated Charging.                                                                                         | 0x0   |
| 0    | R    | USB_DCP_DET | 0 = Charging downstream port is detected.<br>1 = Dedicated charger is detected.<br>Control bit VDM_SRC_ON must be set to validate this status bit.<br>Note: This register shows the actual status. | 0x0   |



## 37.16 GPADC REGISTER FILE

Table 526: Register map GPADC

| Address    | Port                 | Description                                  |
|------------|----------------------|----------------------------------------------|
| 0x50001900 | GP_ADC_CTRL_REG      | General Purpose ADC Control Register         |
| 0x50001902 | GP_ADC_CTRL2_REG     | General Purpose ADC Second Control Register  |
| 0x50001904 | GP_ADC_CTRL3_REG     | General Purpose ADC Third Control Register   |
| 0x50001906 | GP_ADC_OFFP_REG      | General Purpose ADC Positive Offset Register |
| 0x50001908 | GP_ADC_OFFN_REG      | General Purpose ADC Negative Offset Register |
| 0x5000190A | GP_ADC_CLEAR_INT_REG | General Purpose ADC Clear Interrupt Register |
| 0x5000190C | GP_ADC_RESULT_REG    | General Purpose ADC Result Register          |

Table 527: GP\_ADC\_CTRL\_REG (0x50001900)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reset |
|------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15   | R/W  | GP_ADC_LDO_ZERO | 1: Samples and disconnects VREF, should be refreshed frequently. Note that the LDO consumes power when bit is set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x0   |
| 14   | R/W  | GP_ADC_CHOP     | 0: Chopper mode off<br>1: Chopper mode enabled. Takes two samples with opposite GP_ADC_SIGN to cancel the internal offset voltage of the ADC; Highly recommended for DC-measurements.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 13   | R/W  | GP_ADC_SIGN     | 0: Default<br>1: Conversion with opposite sign at input and output to cancel out the internal offset of the ADC and low-frequency                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 12:8 | R/W  | GP_ADC_SEL      | ADC input selection.<br>If GP_ADC_SE = 1 (single ended mode):<br>0: P1[2]<br>1: P1[4]<br>2: P1[3]<br>3: P0[7]<br>4: AVS<br>5: Internal VDD_REF (used for offset calibration)<br>6: VDCDC (see DCDC_TEST_0_REG.DCDC_OUTPUT_MONITOR for more information; GP_ADC_ATT3X scaler automatically selected)<br>7: V33 (GP_ADC_ATT3X scaler automatically selected)<br>8: V33 (GP_ADC_ATT3X scaler automatically selected)<br>9: VBAT (5V to 1.2V scaler selected)<br>16: P0[6]<br>17: P1[0]<br>18: P1[5]<br>19: P2[4]<br>All other combinations are reserved.<br>If GP_ADC_SE = 0 (differential mode):<br>0: P1[2] vs P1[4]<br>All other combinations are P1[3] vs P0[7]. | 0x0   |
| 7    | R/W  | GP_ADC_MUTE     | 0: Normal operation<br>1: Mute ADC input. Takes sample at mid-scale (to determine the internal offset and/or noise of the ADC with regards to VDD_REF which is also sampled by the ADC).                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 6    | R/W  | GP_ADC_SE       | 0: Differential mode<br>1: Single ended mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 5    | R/W  | GP_ADC_MINT     | 0: Disable (mask) GP_ADC_INT.<br>1: Enable GP_ADC_INT to ICU.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 0x0   |

Table 527: GP\_ADC\_CTRL\_REG (0x50001900)

| Bit | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                      | Reset |
|-----|------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 4   | R    | GP_ADC_INT     | 1: AD conversion ready and has generated an interrupt. Must be cleared by writing any value to GP_ADC_CLEAR_INT_REG.                                                                                                                                                                                             | 0x0   |
| 3   | R/W  | GP_ADC_CLK_SEL | 0: Internal high-speed ADC clock used (recommended).<br>1: Digital clock used (ADC_CLK).                                                                                                                                                                                                                         | 0x0   |
| 2   | R/W  | GP_ADC_CONT    | 0: Manual ADC mode, a single result will be generated after setting the GP_ADC_START bit.<br>1: Continuous ADC mode, new ADC results will be constantly stored in GP_ADC_RESULT_REG. Still GP_ADC_START has to be set to start the execution. The time between conversions is configurable with GP_ADC_INTERVAL. | 0x0   |
| 1   | R/W  | GP_ADC_START   | 0: ADC conversion ready.<br>1: If a 1 is written, the ADC starts a conversion. After the conversion this bit will be set to 0 and the GP_ADC_INT bit will be set. It is not allowed to write this bit while it is not (yet) zero.                                                                                | 0x0   |
| 0   | R/W  | GP_ADC_EN      | 0: LDO is off and ADC is disabled..<br>1: LDO is turned on and afterwards the ADC is enabled.                                                                                                                                                                                                                    | 0x0   |

Table 528: GP\_ADC\_CTRL2\_REG (0x50001902)

| Bit   | Mode | Symbol             | Description                                                                                                                                                                                                                  | Reset |
|-------|------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | R/W  | GP_ADC_STORE_DELAY | 0: Data is stored after handshake synchronisation<br>1: Data is stored two ADC_CLK cycles after internal start trigger<br>15: Data is stored sixteen ADC_CLK cycles after internal start trigger                             | 0x0   |
| 11:8  | R/W  | GP_ADC_SMPL_TIME   | 0: The sample time (switch is closed) is one ADC_CLK cycle<br>1: The sample time is 1*32 ADC_CLK cycles<br>2: The sample time is 2*32 ADC_CLK cycles<br>15: The sample time is 15*32 ADC_CLK cycles                          | 0x0   |
| 7:5   | R/W  | GP_ADC_CONV_NS     | 0: 1 sample is taken or 2 in case ADC_CHOP is active.<br>1: 2 samples are taken.<br>2: 4 samples are taken.<br>7: 128 samples are taken.                                                                                     | 0x0   |
| 4     | -    | -                  | Reserved                                                                                                                                                                                                                     | 0x0   |
| 3     | R/W  | GP_ADC_DMA_EN      | 0: DMA functionality disabled<br>1: DMA functionality enabled                                                                                                                                                                | 0x0   |
| 2     | R/W  | GP_ADC_I20U        | 1: Adds 20uA constant load current at the ADC LDO to minimize ripple on the reference voltage of the ADC.                                                                                                                    | 0x0   |
| 1     | R/W  | GP_ADC_IDYN        | 1: Enables dynamic load current at the ADC LDO to minimize ripple on the reference voltage of the ADC.                                                                                                                       | 0x0   |
| 0     | R/W  | GP_ADC_ATTN3X      | 0: Input voltages up to 1.2V allowed.<br>1: Input voltages up to 3.6V allowed by enabling 3x attenuator. (if ADC_SEL=7 or 8, this bit is automatically set to 1)<br>Enabling the attenuator requires a longer sampling time. | 0x0   |

Table 529: GP\_ADC\_CTRL3\_REG (0x50001904)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                                                                                                                   | Reset |
|------|------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | R/W  | GP_ADC_INTERVAL | Defines the interval between two ADC conversions in case GP_ADC_CONT is set.<br>0: No extra delay between two conversions.<br>1: 1.024ms interval between two conversions.<br>2: 2.048ms interval between two conversions.<br>255: 261.12ms interval between two conversions. | 0x0   |
| 7:0  | R/W  | GP_ADC_EN_DEL   | Defines the delay for enabling the ADC after enabling the LDO.<br>0: Not allowed<br>1: 32x ADC_CLK period.<br>n: n*32x ADC_CLK period.                                                                                                                                        | 0x40  |

Table 530: GP\_ADC\_OFFP\_REG (0x50001906)

| Bit   | Mode | Symbol      | Description                                                                                                       | Reset |
|-------|------|-------------|-------------------------------------------------------------------------------------------------------------------|-------|
| 15:10 | -    | -           | Reserved                                                                                                          | 0x0   |
| 9:0   | R/W  | GP_ADC_OFFP | Offset adjust of 'positive' array of ADC-network (effective if "GP_ADC_SE=0", or "GP_ADC_SE=1 AND GP_ADC_SIGN=0") | 0x200 |

Table 531: GP\_ADC\_OFFN\_REG (0x50001908)

| Bit   | Mode | Symbol      | Description                                                                                                       | Reset |
|-------|------|-------------|-------------------------------------------------------------------------------------------------------------------|-------|
| 15:10 | -    | -           | Reserved                                                                                                          | 0x0   |
| 9:0   | R/W  | GP_ADC_OFFN | Offset adjust of 'negative' array of ADC-network (effective if "GP_ADC_SE=0", or "GP_ADC_SE=1 AND GP_ADC_SIGN=1") | 0x200 |

Table 532: GP\_ADC\_CLEAR\_INT\_REG (0x5000190A)

| Bit  | Mode | Symbol         | Description                                                                             | Reset |
|------|------|----------------|-----------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | GP_ADC_CLR_INT | Writing any value to this register will clear the ADC_INT interrupt. Reading returns 0. | 0x0   |

Table 533: GP\_ADC\_RESULT\_REG (0x5000190C)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                                                    | Reset |
|------|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | GP_ADC_VAL | Returns the 10 up to 16 bits linear value of the last AD conversion. The upper 10 bits are always valid, the lower 6 bits are only valid in case oversampling has been applied. Two samples results in one extra bit and 64 samples results in six extra bits. | 0x0   |

## 37.17 QUADRATURE DECODER REGISTER FILE

Table 534: Register map Quadrature Decoder

| Address    | Port              | Description                         |
|------------|-------------------|-------------------------------------|
| 0x50001A00 | QDEC_CTRL_REG     | Quad decoder control register       |
| 0x50001A02 | QDEC_XCNT_REG     | Counter value of the X Axis         |
| 0x50001A04 | QDEC_YCNT_REG     | Counter value of the Y Axis         |
| 0x50001A06 | QDEC_ZCNT_REG     | Counter value of the Z Axis         |
| 0x50001A08 | QDEC_CLOCKDIV_REG | Quad decoder clock divider register |

Table 535: QDEC\_CTRL\_REG (0x50001A00)

| Bit   | Mode | Symbol        | Description                                                                                                                                                           | Reset |
|-------|------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -             | Reserved                                                                                                                                                              | 0x0   |
| 12    | R/W  | CHZ_PORT_EN   | '1': Enable channel                                                                                                                                                   | 0x0   |
| 11    | R/W  | CHY_PORT_EN   | '1': Enable channel                                                                                                                                                   | 0x0   |
| 10    | R/W  | CHX_PORT_EN   | '1': Enable channel                                                                                                                                                   | 0x0   |
| 9:3   | R/W  | QD_IRQ_THRES  | The number of events on either counter (X or Y or Z) that need to be reached before an interrupt is generated. If 0 is written, then threshold is considered to be 1. | 0x2   |
| 2     | R    | QD_IRQ_STATUS | Interrupt Status. If 1 an interrupt has occurred.                                                                                                                     | 0x0   |
| 1     | R/W  | QD_IRQ_CLR    | Writing 1 to this bit clears the interrupt. This bit is auto-cleared                                                                                                  | 0x0   |
| 0     | R/W  | QD_IRQ_MASK   | 0: interrupt is masked<br>1: interrupt is enabled                                                                                                                     | 0x0   |

Table 536: QDEC\_XCNT\_REG (0x50001A02)

| Bit  | Mode | Symbol    | Description                                                          | Reset |
|------|------|-----------|----------------------------------------------------------------------|-------|
| 15:0 | R    | X_COUNTER | Contains a signed value of the events. Zero when channel is disabled | 0x0   |

Table 537: QDEC\_YCNT\_REG (0x50001A04)

| Bit  | Mode | Symbol    | Description                                                          | Reset |
|------|------|-----------|----------------------------------------------------------------------|-------|
| 15:0 | R    | Y_COUNTER | Contains a signed value of the events. Zero when channel is disabled | 0x0   |

Table 538: QDEC\_ZCNT\_REG (0x50001A06)

| Bit  | Mode | Symbol    | Description                                                          | Reset |
|------|------|-----------|----------------------------------------------------------------------|-------|
| 15:0 | R    | Z_COUNTER | Contains a signed value of the events. Zero when channel is disabled | 0     |

Table 539: QDEC\_CLOCKDIV\_REG (0x50001A08)

| Bit | Mode | Symbol        | Description                                                                                                   | Reset |
|-----|------|---------------|---------------------------------------------------------------------------------------------------------------|-------|
| 9:0 | R/W  | CLOCK_DIVIDER | Contains the number of the input clock cycles minus one, that are required to generate one logic clock cycle. | 0x0   |

## 37.18 ANAMISC REGISTER FILE

Table 540: Register map ANAMISC

| Address    | Port                 | Description                                                  |
|------------|----------------------|--------------------------------------------------------------|
| 0x50001B08 | CHARGER_CTRL1_REG    | Charger control register 1                                   |
| 0x50001B0A | CHARGER_CTRL2_REG    | Charger control register 2                                   |
| 0x50001B0C | CHARGER_STATUS_REG   | Charger status and trimming register                         |
| 0x50001B40 | SOC_CTRL1_REG        | Fuel Gauge Control register 1                                |
| 0x50001B42 | SOC_CTRL2_REG        | Fuel Gauge Control register 2                                |
| 0x50001B44 | SOC_CTRL3_REG        | Fuel Gauge Control register 3                                |
| 0x50001B46 | SOC_ADD2CH_REG       | Fuel Gauge manually add extra charge to SOC_CHARGE_CNTRx_REG |
| 0x50001B48 | SOC_CHARGE_CNTR1_REG | Fuel Gauge Charge counter bits 15-0                          |
| 0x50001B4A | SOC_CHARGE_CNTR2_REG | Fuel Gauge Charge counter bits 31-16                         |
| 0x50001B4C | SOC_CHARGE_CNTR3_REG | Fuel Gauge Charge counter bits 39-32                         |
| 0x50001B50 | SOC_CHARGE_AVG_REG   | Fuel Gauge Average charge counter                            |
| 0x50001B52 | SOC_STATUS_REG       | Fuel Gauge Status register                                   |
| 0x50001B54 | SOC_EXT_IN_REG       | Fuel Gauge input test register                               |
| 0x50001B56 | SOC_EXT_OUT_REG      | Fuel Gauge output test register                              |
| 0x50001B60 | CLK_REF_SEL_REG      | Select clock for oscillator calibration                      |
| 0x50001B62 | CLK_REF_CNT_REG      | Count value for oscillator calibration                       |
| 0x50001B64 | CLK_REF_VAL_L_REG    | DIVN reference cycles, lower 16 bits                         |
| 0x50001B66 | CLK_REF_VAL_H_REG    | DIVN reference cycles, upper 16 bits                         |

Table 541: CHARGER\_CTRL1\_REG (0x50001B08)

| Bit   | Mode | Symbol           | Description                                                                                                                                                                                                                                                                                   | Reset |
|-------|------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | -    | -                | Reserved                                                                                                                                                                                                                                                                                      | 0x0   |
| 14    | R/W  | DIE_TEMP_DISABLE | 0: Die temperature protection enabled: charger will be disabled when die temp exceeds value set in DIE_TEMP_SET<br>1: Die temperature protection disabled: testmode, use only in agreement with Dialog                                                                                        | 0x0   |
| 13:12 | R/W  | DIE_TEMP_SET     | Die temperature protection level.<br>Charging will be automatically disabled if set level is exceeded and resumed when temperature has dropped few degrees below set level.<br>00: 0°C (do not use, for test only)<br>01: 50°C (do not use, for test only)<br>10: 80°C (default)<br>11: 100°C | 0x2   |

Table 541: CHARGER\_CTRL1\_REG (0x50001B08)

| Bit  | Mode | Symbol          | Description                                                                                                                                                                                                                                                                                                                                                                                                            | Reset |
|------|------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 11:8 | R/W  | CHARGE_CUR      | Constant Current levels (typical values)<br>0000: 5 mA<br>0001: 10 mA<br>0010: 30 mA<br>0011: 45 mA<br>0100: 60 mA<br>0101: 90 mA<br>0110: 120 mA<br>0111: 150 mA<br>1000: 180 mA<br>1001: 210 mA<br>1010: 270 mA<br>1011: 300 mA<br>1100: 350 mA<br>1101: 400 mA                                                                                                                                                      | 0x0   |
| 7    | R/W  | NTC_LOW_DISABLE | 0: Normal operation: voltage level higher than 7/8 VDD_USB will disable the charger<br>1: NTC low temp limit disabled: use if trickle charging below the minimum temperature is required                                                                                                                                                                                                                               | 0x0   |
| 6    | R/W  | NTC_DISABLE     | 0: Charger NTC protection enabled<br>1: Charger NTC protection disable                                                                                                                                                                                                                                                                                                                                                 | 0x0   |
| 5    | R/W  | CHARGE_ON       | 0: Charger in powerdown<br>1: Charger enabled                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 4:0  | R/W  | CHARGE_LEVEL    | Constant Voltage Levels<br>00000: 3.00V (reset)<br>00001: 3.40V (e.g. 2xNiMH)<br>00010: 3.50V<br>00011: 3.60V (e.g. Li-phosphate)<br>00100: 3.74V<br>00101: 3.86V<br>00110: 4.00V<br>00111: 4.05V<br>01000: 4.10V<br>01001: 4.15V<br>01010: 4.20V (e.g. Li-Co, Li-Mn, NMC)<br>01011: 4.25V<br>01100: 4.30V<br>01101: 4.35V<br>01110: 4.40V<br>01111: 4.50V<br>10000: 4.60V<br>10001: 4.90V e.g. 3xNiMH<br>10010: 5.00V | 0x0   |

Table 542: CHARGER\_CTRL2\_REG (0x50001B0A)

| Bit   | Mode | Symbol              | Description                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-------|------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | R/W  | CHARGER_TEST        | Signals are mapped on SPDIF pin.<br>Also set ANA_TEST_REG[ANA_TESTBUS_TO_ADCPIN] = 1<br>000: normal mode (no test selected)<br>001: Vptat (temperature sensor) [1.4V max]<br>010: Vbat_sense after divider [1.2V]<br>011: Current loop output [0 to vsupply]<br>100: Voltage loop output [0 to vsupply]<br>101: I meas or Iref/10<br>110: Icharge reduced by 26.6<br>111: reserved | 0x0   |
| 12:8  | R/W  | CURRENT_OFFSET_TRIM | do not change, for test purpose only                                                                                                                                                                                                                                                                                                                                               | 0xF   |
| 7:4   | R/W  | CHARGER_VFLOAT_ADJ  | Independent adjustment for the charge level. Adjust range is +/- 1.8%. The 4 bits adjustment is in two's complement.                                                                                                                                                                                                                                                               | 0x0   |
| 3:0   | R/W  | CURRENT_GAIN_TRIM   | do not change, for test purpose only                                                                                                                                                                                                                                                                                                                                               | 0x7   |

Table 543: CHARGER\_STATUS\_REG (0x50001B0C)

| Bit  | Mode | Symbol               | Description                                                                                                                                                                                              | Reset |
|------|------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -                    | Reserved                                                                                                                                                                                                 | 0x0   |
| 6    | R    | CHARGER_TMODE_PROT   | 0: Dietemp below DIE_TEMP_SET level. Normal operation<br>1: Dietemp above DIE_TEMP_SET level. Charging is disabled                                                                                       | 0x0   |
| 5    | R    | CHARGER_BATTEMP_HIGH | 0: Battery pack temperature 'ok' or 'too low' (voltage level on NTC pin above 1/2 VDD_USB)<br>1: Battery pack temperature 'too high' (voltage level on NTC pin below 1/2 VDD_USB)                        | 0x0   |
| 4    | R    | CHARGER_BATTEMP_OK   | 0: Battery pack temperature 'too low' or 'too high' (voltage level on NTC pin below 1/2 or above 7/8 VDD_USB)<br>1: Battery pack temperature 'ok' (voltage level on NTC pin between 1/2 and 7/8 VDD_USB) | 0x0   |
| 3    | R    | CHARGER_BATTEMP_LOW  | 0: Battery pack temperature 'ok' or 'too high' (voltage level on NTC pin below 7/8 VDD_USB)<br>1: Battery pack temperature 'too low' (voltage level on NTC pin above than 7/8 VDD_USB)                   | 0x0   |
| 2    | R    | END_OF_CHARGE        | 0: Actual charge current is between 10...100% of set CHARGE_CUR (or CHARGE_ON=0)<br>1: Actual charge current <10% of set CHARGE_CUR                                                                      | 0x0   |
| 1    | R    | CHARGER_CV_MODE      | 0: voltage loop not in regulation (or charger is off)<br>1: constant voltage mode active, voltage loop in regulation.                                                                                    | 0x0   |
| 0    | R    | CHARGER_CC_MODE      | 0: current loop not in regulation (or charger is off)<br>1: constant current mode active, current loop in regulation.                                                                                    | 0x0   |

Table 544: SOC\_CTRL1\_REG (0x50001B40)

| Bit   | Mode | Symbol   | Description                                                                                                    | Reset |
|-------|------|----------|----------------------------------------------------------------------------------------------------------------|-------|
| 15:14 | R/W  | SOC_CINT | Integrator capacitor scaler<br>0: Cint = 1 pF<br>1: Cint = 2 pF<br>2: Cint = 4 pF<br>3: Cint = 8 pF (=default) | 0x3   |



Table 544: SOC\_CTRL1\_REG (0x50001B40)

| Bit   | Mode | Symbol           | Description                                                                                                                                                                                                                                            | Reset |
|-------|------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:12 | R/W  | SOC_BIAS         | Current DAC scaler<br>0: I <sub>bias</sub> = 2 uA<br>1: I <sub>bias</sub> = 1 uA (=default)<br>2: I <sub>bias</sub> = 0.5 uA<br>3: I <sub>bias</sub> = 0.25 uA                                                                                         | 0x1   |
| 11:9  | R/W  | SOC_CLK          | SOC Sample frequency<br>0: automatic mode<br>1: fs = 18 kHz<br>2: fs = 36 kHz<br>3: fs = 72 kHz<br>4: fs = 144 kHz (=default)<br>5: fs = 288 kHz<br>6: fs = 576 kHz<br>7: fs = 1152 kHz                                                                | 0x4   |
| 8     | R/W  | SOC_LPF          | 0: low-pass filter at integrator inputs disabled<br>1: Enables a low-pass filter at the integrator inputs                                                                                                                                              | 0x0   |
| 7:6   | R/W  | SOC_IDAC         | Scales the current DAC (I <sub>bias</sub> : default=1uA)<br>0: I <sub>dac</sub> =0.25*I <sub>bias</sub><br>1: I <sub>dac</sub> =0.5*I <sub>bias</sub><br>2: I <sub>dac</sub> =I <sub>bias</sub> (=default)<br>3: I <sub>dac</sub> =2*I <sub>bias</sub> | 0x2   |
| 5     | R/W  | SOC_SIGN         | Defines the sign of the charge converter input and output to perform a chopper function to eliminate offset voltage (see also SOC_CHOP and 'sign' on output pin)<br>0: non-inverted inputs and outputs<br>1: inverted inputs and outputs               | 0x0   |
| 4     | R/W  | SOC_GPIO         | Reserved (not yet implemented): switches the SOC-inputs to the GPIO pins                                                                                                                                                                               | 0x0   |
| 3     | R/W  | SOC_MUTE         | 0: Normal operation<br>1: Connect the input voltage to 0V                                                                                                                                                                                              | 0x0   |
| 2     | R/W  | SOC_RESET_AVG    | 1: Reset the SOC_CHARGE_AVG_REG to the last value of SOC_CHARGE_CNTRx_REG                                                                                                                                                                              | 0x0   |
| 1     | R/W  | SOC_RESET_CHARGE | 1: Reset CHARGE_CNTR_REG                                                                                                                                                                                                                               | 0x0   |
| 0     | R/W  | SOC_ENABLE       | 0: SOC analog circuits off. CHARGE_CNTRx_REG can still be written for a manual update.<br>See SOC_ADD2CH_REG<br>1: SOC analog circuits enabled                                                                                                         | 0x0   |

Table 545: SOC\_CTRL2\_REG (0x50001B42)

| Bit   | Mode | Symbol            | Description                                                                                                                                                    | Reset |
|-------|------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | SOC_DYNAVG        | if HIGH then 'weight' of Moving Average is forced to 1 if the converter detects significant input change (if dcharge > 4*delta_c, or high_limit, or low_limit) | 0x0   |
| 14:12 | R/W  | SOC_MAW           | Moving Average Weight factor<br>charge_avg(n) = (weight*charge_avg(n-1) + charge(n)) / (weight+1)<br>where: weight = 2^(soc_maw)                               | 0x7   |
| 11    | R/W  | SOC_CMIREG_ENABLE | SOC_CMIREG enable                                                                                                                                              | 0x0   |

Table 545: SOC\_CTRL2\_REG (0x50001B42)

| Bit  | Mode | Symbol     | Description                                                                                                                                                                                                                                           | Reset |
|------|------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10:8 | R/W  | SOC_CHOP   | Chopping control<br>0: 'external' chopping control with 'soc_sign'-input<br>1: chop each $2^1$ *scycle fs-periods<br>2: chop each $2^2$ *scycle fs-periods<br>..<br>7: chop each $2^7$ *scycle fs-periods.                                            | 0x7   |
| 7:6  | R/W  | SOC_ICM    | adds a common-mode current to Idac to increase the common-mode input-level of the integrator.<br>The common-mode input level is equal to $(Idac+Icm)*Rvi$ ;<br>0: $Icm=0$ ;<br>1: $Icm=1*Ibias$ (=default);<br>2: $Icm=2*Ibias$ ;<br>3: $Icm=4*Ibias$ | 0x1   |
| 5    | R/W  | SOC_DCYCLE | Cycle the current divider segments of Idac<br>0: no cycling<br>1: cycle each scycle fs-periods                                                                                                                                                        | 0x1   |
| 4:2  | R/W  | SOC_SCYCLE | Cycle current segments (8 segments) of Idac<br>0: no cycling<br>1: cycle each fs-period<br>2: cycle each 2 fs-periods<br>..<br>7: cycle each 7 fs-periods                                                                                             | 0x2   |
| 1:0  | R/W  | SOC_RVI    | Voltage-to-current resistor scaler<br>0: $Rvi = 25\text{ k}$<br>1: $Rvi = 50\text{ k}$<br>2: $Rvi = 100\text{ k}$ (= default)<br>3: $Rvi = 200\text{ k}$                                                                                              | 0x2   |

Table 546: SOC\_CTRL3\_REG (0x50001B44)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                             | Reset |
|------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -           | Reserved                                                                                                                                                                                                                                                                                | 0x0   |
| 5:4  | R/W  | SOC_VCM1    | Common Input Voltage target of regulator (see SOC_CMIREG_ENABLE)<br>0: 50 mV<br>1: 100 mV<br>2: 150 mV<br>3: 200 mV                                                                                                                                                                     | 0x1   |
| 3    | R/W  | SOC_DYNHYS  | Reserved. (To be implemented)<br>Hysteresis of the comparator which detects if the integrator voltage is rising or falling                                                                                                                                                              | 0x0   |
| 2    | R/W  | SOC_DYNTARG | Reserved. (To be implemented)<br>0: $V_{int\_target} = 0V$<br>1: $V_{int\_target}$ tracks the 2 MSB's of the charge register)                                                                                                                                                           | 0x0   |
| 1:0  | R/W  | SOC_VSAT    | Trigger level of the high-limit and low-limit comparators.<br>0: $low\_limit = -50mV$ ; $high\_limit = +50mV$<br>1: $low\_limit = -100mV$ ; $high\_limit = +100mV$ (=default)<br>2: $low\_limit = -200mV$ ; $high\_limit = +200mV$<br>3: $low\_limit = -400mV$ ; $high\_limit = +400mV$ | 0x1   |

Table 547: SOC\_ADD2CH\_REG (0x50001B46)

| Bit  | Mode | Symbol     | Description                                                                                               | Reset |
|------|------|------------|-----------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | SOC_ADD2CH | Extra charge to be added to the SOC_CHARGE_CNTRx_REG per sample period (9-bit + sign + 6 fractional bits) | 0x0   |

Table 548: SOC\_CHARGE\_CNTR1\_REG (0x50001B48)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                                                                                                                                                                                   | Reset |
|------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | CHARGE_CNT1 | Sum of the charge-values per sampling period; (bits15:0)<br>The absolute full-scale charge value is 6-bits, At full scale charge current it takes $2^{26}$ sampling periods until overflow of the charge_cnt register after a reset_charge event.<br>At fs=144kHz (=default) this will happen after 33 hours<br>At fs=1.152MHz After 10 hours | 0x0   |

Table 549: SOC\_CHARGE\_CNTR2\_REG (0x50001B4A)

| Bit  | Mode | Symbol      | Description                                               | Reset |
|------|------|-------------|-----------------------------------------------------------|-------|
| 15:0 | R    | CHARGE_CNT2 | Sum of the charge-values per sampling period; (bits23:16) | 0x0   |

Table 550: SOC\_CHARGE\_CNTR3\_REG (0x50001B4C)

| Bit  | Mode | Symbol      | Description                                               | Reset |
|------|------|-------------|-----------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                  | 0x0   |
| 7:0  | R    | CHARGE_CNT3 | Sum of the charge-values per sampling period; (bits39:24) | 0x0   |

Table 551: SOC\_CHARGE\_AVG\_REG (0x50001B50)

| Bit  | Mode | Symbol     | Description                                                      | Reset |
|------|------|------------|------------------------------------------------------------------|-------|
| 15:0 | R    | CHARGE_AVG | Average of 'charge' current (9-bit + sign and 6 fractional bits) | 0x0   |

Table 552: SOC\_STATUS\_REG (0x50001B52)

| Bit  | Mode | Symbol           | Description                                                                                                                                                                                                | Reset |
|------|------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:2 | -    | -                | Reserved                                                                                                                                                                                                   | 0x0   |
| 1    | R    | SOC_INT_LOCKED   | 0: Normal Operation<br>1: Integrator is pushed over high or low limit.<br>Returns to '0' if the converter runs for more than 2 sequential sampling periods in a 'safe' region ( $dcharge < 2 * \Delta_c$ ) | 0x0   |
| 0    | R    | SOC_INT_OVERLOAD | 0: Normal Operation<br>1: Integrator exceeds high or low limit with full-scale IDAC (charge) for more than 3 sequential sampling periods                                                                   | 0x0   |

Table 553: SOC\_EXT\_IN\_REG (0x50001B54)

| Bit   | Mode | Symbol            | Description                            | Reset |
|-------|------|-------------------|----------------------------------------|-------|
| 15    | R/W  | SOC_EXT_IDAC_EN   | 1: Enable 'external' control of Idac   | 0x0   |
| 14    | R/W  | SOC_EXT_SCYCLE_EN | 1: Enable 'external' control of scycle | 0x0   |
| 13:11 | R/W  | SOC_NR_SCYCLE     | Number of the scycle                   | 0x0   |

Table 553: SOC\_EXT\_IN\_REG (0x50001B54)

| Bit | Mode | Symbol        | Description                                                                                                                                   | Reset |
|-----|------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10  | R/W  | SOC_RDAC_DIS  | 0: Disables the resistor divider DAC. The Idac has 6-bits (plus sign)<br>1: Enables the resistor divider DAC. The Idac has 9-bits (plus sign) | 0x0   |
| 9   | R/W  | SOC_IDAC_SIGN | 0: SOC_IDAC_VAL is positive<br>1: SOC_IDAC_VAL is negative                                                                                    | 0x0   |
| 8:0 | R/W  | SOC_IDAC_VAL  | Controls the current for the DAC.<br>0: $0/512 \cdot \text{SOC\_IDAC}$<br>N: $N/512 \cdot \text{SOC\_IDAC}$                                   | 0x0   |

Table 554: SOC\_EXT\_OUT\_REG (0x50001B56)

| Bit  | Mode | Symbol          | Description                  | Reset |
|------|------|-----------------|------------------------------|-------|
| 15:9 | -    | -               | Reserved                     | 0x0   |
| 8    | R    | SOC_CTRL_EVENT  | Controller event             | 0x0   |
| 7:4  | R    | SOC_STATE       | Controller state             | 0x0   |
| 3    | R    | SOC_RISING_COMP | Rising comparator output     | 0x0   |
| 2    | R    | SOC_POS_COMP    | Positive comparator output   | 0x0   |
| 1    | R    | SOC_LOWLIM_COMP | Low_limit comparator output  | 0x0   |
| 0    | R    | SOC_HIGH_LIM    | High_limit comparator output | 0x0   |

Table 555: CLK\_REF\_SEL\_REG (0x50001B60)

| Bit  | Mode | Symbol        | Description                                                                                                                                                               | Reset |
|------|------|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:3 | -    | -             | Reserved                                                                                                                                                                  | 0x0   |
| 2    | R/W  | REF_CAL_START | Writing a '1' starts a calibration. This bit is cleared when calibration is finished, and CLK_REF_VAL is ready.                                                           | 0x0   |
| 1:0  | R/W  | REF_CLK_SEL   | Select clock input for calibration:<br>0x0 : 32KHz RC oscillator clock<br>0x1 : 16MHz RC oscillator clock<br>0x2 : 32KHz XTAL clock<br>0x3 : 11.7KHz RCX oscillator clock | 0x0   |

Table 556: CLK\_REF\_CNT\_REG (0x50001B62)

| Bit  | Mode | Symbol      | Description                                                    | Reset |
|------|------|-------------|----------------------------------------------------------------|-------|
| 15:0 | R/W  | REF_CNT_VAL | Indicates the calibration time, with a decrement counter to 1. | 0x0   |

Table 557: CLK\_REF\_VAL\_L\_REG (0x50001B64)

| Bit  | Mode | Symbol       | Description                                                                                                  | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | XTAL_CNT_VAL | Returns the lower 16 bits of DIVN clock cycles counted during the calibration time, defined with REF_CNT_VAL | 0x0   |

Table 558: CLK\_REF\_VAL\_H\_REG (0x50001B66)

| Bit  | Mode | Symbol       | Description                                                                                                  | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | XTAL_CNT_VAL | Returns the upper 16 bits of DIVN clock cycles counted during the calibration time, defined with REF_CNT_VAL | 0x0   |

## 37.19 CRG REGISTER FILE

Table 559: Register map CRG

| Address    | Port         | Description                      |
|------------|--------------|----------------------------------|
| 0x50001C04 | CLK_PER_REG  | Peripheral divider register      |
| 0x50001C40 | PCM_DIV_REG  | PCM divider and enables          |
| 0x50001C42 | PCM_FDIV_REG | PCM fractional division register |
| 0x50001C44 | PDM_DIV_REG  | PDM divider and enables          |
| 0x50001C46 | SRC_DIV_REG  | SRC divider and enables          |
| 0x50001C4A | USBPAD_REG   | USB pads control register        |

Table 560: CLK\_PER\_REG (0x50001C04)

| Bit | Mode | Symbol         | Description                                                  | Reset |
|-----|------|----------------|--------------------------------------------------------------|-------|
| 11  | R/W  | ADC_CLK_SEL    | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock | 0x0   |
| 10  | R/W  | KBSCAN_CLK_SEL | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock | 0x0   |
| 9   | R/W  | I2C_CLK_SEL    | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock | 0x0   |
| 8   | R/W  | SPI_CLK_SEL    | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock | 0x0   |
| 7:6 | -    | -              | Reserved                                                     | 0x0   |
| 5   | R/W  | KBSCAN_ENABLE  | Enables the clock                                            | 0x0   |
| 4   | R/W  | IR_CLK_ENABLE  | Enables the clock                                            | 0x0   |
| 3   | R/W  | QUAD_ENABLE    | Enables the clock                                            | 0x0   |
| 2   | R/W  | I2C_ENABLE     | Enables the clock                                            | 0x0   |
| 1   | R/W  | SPI_ENABLE     | Enables the clock                                            | 0x0   |
| 0   | R/W  | UART_ENABLE    | Enables the clock                                            | 0x0   |

Table 561: PCM\_DIV\_REG (0x50001C40)

| Bit  | Mode | Symbol      | Description                                                                                                  | Reset |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------|-------|
| 13   | R/W  | PCM_SRC_SEL | Selects the clock source<br>1 = DIV1 clock<br>0 = DIVN clock                                                 | 0x0   |
| 12   | R/W  | CLK_PCM_EN  | Enable for the internally generated PCM clock<br>The PCM_DIV must be set before or together with CLK_PCM_EN. | 0x0   |
| 11:0 | R/W  | PCM_DIV     | PCM clock divider                                                                                            | 0x0   |

Table 562: PCM\_FDIV\_REG (0x50001C42)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | PCM_FDIV | These bits define the fractional division part of the PCM clock. The left most '1' defines the denominator, the number of '1' bits define the numerator. E.g.<br>0x0110 means 2/9, with a distribution of 1.0001.0000<br>0xfeee means 13/16, with a distribution of 1111.1110.1110.1110 | 0x0   |

Table 563: PDM\_DIV\_REG (0x50001C44)

| Bit | Mode | Symbol          | Description                                                                                                  | Reset |
|-----|------|-----------------|--------------------------------------------------------------------------------------------------------------|-------|
| 9   | R/W  | PDM_MASTER_MODE | Master mode selection<br>0: slave mode<br>1: master mode                                                     | 0x0   |
| 8   | R/W  | CLK_PDM_EN      | Enable for the internally generated PDM clock<br>The PDM_DIV must be set before or together with CLK_PDM_EN. | 0x0   |
| 7:0 | R/W  | PDM_DIV         | PDM clock divider                                                                                            | 0x0   |

Table 564: SRC\_DIV\_REG (0x50001C46)

| Bit | Mode | Symbol     | Description                                                                                                  | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------|-------|
| 8   | R/W  | CLK_SRC_EN | Enable for the internally generated SRC clock<br>The SRC_DIV must be set before or together with CLK_SRC_EN. | 0x0   |
| 7:0 | R/W  | SRC_DIV    | SRC clock divider                                                                                            | 0x0   |

Table 565: USBPAD\_REG (0x50001C4A)

| Bit | Mode | Symbol               | Description                                                                                                                                                      | Reset |
|-----|------|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 2   | R/W  | USBPHY_FORCE_SW2_ON  | 0: Pull up resistor SW2 is controlled by the USB controller. It is off when the USB is not enabled.<br>1: Force the pull up resistor on USBP to be 2.3Kohm       | 0x0   |
| 1   | R/W  | USBPHY_FORCE_SW1_OFF | 0: Pull up resistor SW1 is controlled by the USB controller. It is off when the USB is not enabled.<br>1: Force the pull up resistor on USBP to be switched off. | 0x0   |
| 0   | R/W  | USBPAD_EN            | 0: The power for the USB PHY and USB pads is switched on when the USB is enabled.<br>1: The power for the USB PHY and USB pads is forced on.                     | 0x0   |



## 37.20 COEX REGISTER FILE

Table 566: Register map COEX

| Address    | Port              | Description                      |
|------------|-------------------|----------------------------------|
| 0x50002F00 | COEX_CTRL_REG     | COEX Control Register            |
| 0x50002F02 | COEX_STAT_REG     | COEX Status Register             |
| 0x50002F04 | COEX_INT_MASK_REG | COEX Interrupt Mask Register     |
| 0x50002F06 | COEX_INT_STAT_REG | COEX Interrupt Status Register   |
| 0x50002F08 | COEX_BLE_PTI_REG  | COEX BLE PTI Control Register    |
| 0x50002F0C | COEX_DIAG_REG     | COEX Diagnostic Monitor Register |
| 0x50002F12 | COEX_PRI1_REG     | COEX Priority Register           |
| 0x50002F14 | COEX_PRI2_REG     | COEX Priority Register           |
| 0x50002F16 | COEX_PRI3_REG     | COEX Priority Register           |
| 0x50002F18 | COEX_PRI4_REG     | COEX Priority Register           |
| 0x50002F1A | COEX_PRI5_REG     | COEX Priority Register           |
| 0x50002F1C | COEX_PRI6_REG     | COEX Priority Register           |
| 0x50002F1E | COEX_PRI7_REG     | COEX Priority Register           |
| 0x50002F20 | COEX_PRI8_REG     | COEX Priority Register           |
| 0x50002F22 | COEX_PRI9_REG     | COEX Priority Register           |
| 0x50002F24 | COEX_PRI10_REG    | COEX Priority Register           |
| 0x50002F26 | COEX_PRI11_REG    | COEX Priority Register           |
| 0x50002F28 | COEX_PRI12_REG    | COEX Priority Register           |
| 0x50002F2A | COEX_PRI13_REG    | COEX Priority Register           |
| 0x50002F2C | COEX_PRI14_REG    | COEX Priority Register           |
| 0x50002F2E | COEX_PRI15_REG    | COEX Priority Register           |
| 0x50002F30 | COEX_PRI16_REG    | COEX Priority Register           |
| 0x50002F32 | COEX_PRI17_REG    | COEX Priority Register           |

Table 567: COEX\_CTRL\_REG (0x50002F00)

| Bit   | Mode | Symbol                 | Description                                                                                                                                                                                                                                                                                           | Reset |
|-------|------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | IGNORE_BLE             | If set to "1" then all BLE requests are ignored by masking the internal "ble_active" signal. Refer also to IGNORE_BLE_STAT.                                                                                                                                                                           | 0x0   |
| 14    | -    | -                      | Reserved                                                                                                                                                                                                                                                                                              | 0x0   |
| 13    | R/W  | IGNORE_EXT             | If set to "1" then all EXT requests are ignored by masking the internal "ext_act" signal ("ext_act" is the logical OR of "ext_act0" and "ext_act1"). Refer also to IGNORE_EXT_STAT.                                                                                                                   | 0x0   |
| 12:11 | R/W  | SEL_BLE_RADIO_B<br>USY | Select the logic driving the BLE core input "ble.radio_busy":<br>0: (decision==BLE) AND rfcu.radio_busy.<br>1: Hold to "0".<br>2: (decision==EXT) OR rfcu.radio_busy.<br>3: (decision==EXT).<br>Selection "0" is the default, while selection "2" is the recommended value if the BLE SW supports it. | 0x0   |
| 10    | R/W  | SEL_BLE_WLAN_TX<br>_RX | If set to "1" then the COEX block will drive the WLAN_TX and WLAN_RX inputs of the BLE core. Otherwise both BLE inputs will be forced to "0".                                                                                                                                                         | 0x0   |
| 9     | -    | -                      | Reserved                                                                                                                                                                                                                                                                                              | 0x0   |
| 8     | -    | -                      | Reserved                                                                                                                                                                                                                                                                                              | 0x0   |
| 7     | -    | -                      | Reserved                                                                                                                                                                                                                                                                                              | 0x0   |

Table 567: COEX\_CTRL\_REG (0x50002F00)

| Bit | Mode | Symbol         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6:5 | R/W  | SEL_COEX_DIAG  | The COEX block can provide internal diagnostic signals by overwriting the BLE diagnostic bus, which is forwarded to GPIO multiplexing. There is no need to program the BLE registers, but only this field and the GPIO PID fields. The encoding of this bitfield is:<br>0: Don't overwrite any BLE diagnostic signal.<br>1: Overwrite the BLE Diagnostic bits 2 down to 0:<br>P2[2]: "closing" sub-state<br>P2[1:0]: "decision" state<br>2: Overwrite the BLE Diagnostic bits 5 down to 3:<br>P1[2]: "closing" sub-state<br>P1[1:0]: "decision" state<br>3: Reserved..                                          | 0x0   |
| 4   | R/W  | SMART_ACT_IMPL | Controls the behavior of the SMART_ACT (and SMART_PRI as a consequence).<br>If SMART_ACT_IMPL="0" then if any BLE MAC request is active then SMART_ACT will be asserted. SMART_ACT will actually be the "ble_active" signal. SMART_ACT will be asserted regardless the decision of the Arbiter to allow or disallow the access to the on-chip radio from the active MAC(s).<br>if SMART_ACT_IMPL="1" then if the Arbiter's decision is to allow EXTERNAL MAC, then keep SMART_ACT to "0", otherwise follow the implementation of SMART_ACT_IMPL="0".                                                            | 0x0   |
| 3   | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 2:1 | -    | -              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 0x0   |
| 0   | R/W  | PRGING_ARBITER | If set to "1" then the current BLE transaction will complete normally and after that no further decision will be taken by the arbiter. The SW must keep this bit to "1" as long as it performs write operations on the COEX_PRI*_REG registers. As soon as the update on the priority registers will be completed, the SW should clear this bit.<br>Note: This bit is updated with the COEX_CLK, so depending on the relationship between the PCLK and COEX_CLK periods a write operation to this bit may be effective in more than one PCLK clock cycles, e.g. when the COEX_CLK rate is slower than the PCLK. | 0x0   |

Table 568: COEX\_STAT\_REG (0x50002F02)

| Bit | Mode | Symbol          | Description                                                                                                                                                                                                                     | Reset |
|-----|------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R    | IGNORE_BLE_STAT | If set to "1" then all BLE requests are ignored by masking immediately the request signal from the BLE.<br>In more detail, the internal signal "ble_active" is the logical AND of this bitfield and the "ble.event_in_process". | 0x0   |
| 14  | -    | -               | Reserved                                                                                                                                                                                                                        | 0x0   |
| 13  | R    | IGNORE_EXT_STAT | If set to "1" then all EXT requests are ignored by masking immediately the request signal from the external MAC.<br>In more detail, the internal signal "ext_active" is the logical AND of this bitfield and the "ext_act".     | 0x0   |
| 12  | R    | COEX_RADIO_BUSY | Current state of RADIO_BUSY signal generated from RFCU, which is the logical OR among all Radio DCFs.<br>Note that the arbiter will process this value with one COEX clock cycle delay.                                         | 0x0   |
| 11  | R    | EXT_ACT1        | Current state of the pin.                                                                                                                                                                                                       | 0x0   |

Table 568: COEX\_STAT\_REG (0x50002F02)

| Bit | Mode | Symbol                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset |
|-----|------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10  | R    | EXT_ACT0              | Current state of the pin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 9   | R    | SMART_PRI             | Current state of the pin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 8   | R    | SMART_ACT             | Current state of the pin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 7   | R    | COEX_DECISION_CLOSING | Provides the value of the "CLOSING" substate.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 6:5 | R    | COEX_DECISION         | Decision values:<br>0: Decision is NONE.<br>1: Decision is BLE.<br>2: Reserved<br>3: Decision is EXT.<br>Note: If "0" (i.e. decision is NONE) then no MAC will have access to the on-chip radio. As a consequence, the SMART_PRI signal will stay low, since no on-chip (SMART) MAC will have priority.<br>Note: The decision NONE will be held as long as there is no "*_active" internal signal from BLE or EXT. Also, if in programming state and the last transaction has been finished, then the decision will also be NONE.<br>Note: While in programming mode, the COEX_PRIx_REGS are considered as invalid, which means that no new decision can be taken. | 0x0   |
| 4:0 | R    | COEX_DECISION_PTR     | Provides the number "x" of the COEX_PRIx_REG that win the last arbitration cycle. If "0" then it is a null pointer, pointing to no COEX_PRIx_REG.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0x0   |

Table 569: COEX\_INT\_MASK\_REG (0x50002F04)

| Bit | Mode | Symbol                   | Description                                                                                                      | Reset |
|-----|------|--------------------------|------------------------------------------------------------------------------------------------------------------|-------|
| 15  | R/W  | COEX_IRQ_ON_DECISION_SW  | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_DECISION_SW] will cause COEX_IRQ_STAT to be set also to "1".  | 0x0   |
| 14  | R/W  | COEX_IRQ_ON_START_MID    | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_START_MID] will cause COEX_IRQ_STAT to be set also to "1".    | 0x0   |
| 13  | R/W  | COEX_IRQ_ON_CLOSING_BRK  | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_CLOSING_BRK] will cause COEX_IRQ_STAT to be set also to "1".  | 0x0   |
| 12  | R/W  | COEX_IRQ_ON_RADIO_BUSY_F | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_RADIO_BUSY_F] will cause COEX_IRQ_STAT to be set also to "1". | 0x0   |
| 11  | R/W  | COEX_IRQ_ON_RADIO_BUSY_R | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_RADIO_BUSY_R] will cause COEX_IRQ_STAT to be set also to "1". | 0x0   |
| 10  | R/W  | COEX_IRQ_ON_BLE_ACTIVE_F | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_BLE_ACTIVE_F] will cause COEX_IRQ_STAT to be set also to "1". | 0x0   |
| 9   | R/W  | COEX_IRQ_ON_BLE_ACTIVE_R | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_BLE_ACTIVE_R] will cause COEX_IRQ_STAT to be set also to "1". | 0x0   |
| 8   | -    | -                        | Reserved                                                                                                         | 0x0   |
| 7   | -    | -                        | Reserved                                                                                                         | 0x0   |

Table 569: COEX\_INT\_MASK\_REG (0x50002F04)

| Bit | Mode | Symbol                  | Description                                                                                                                                           | Reset |
|-----|------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6   | R/W  | COEX_IRQ_ON_EXT_ACT_F   | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_EXT_ACT_F] will cause COEX_IRQ_STAT to be set also to "1".                                         | 0x0   |
| 5   | R/W  | COEX_IRQ_ON_EXT_ACT_R   | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_EXT_ACT_R] will cause COEX_IRQ_STAT to be set also to "1".                                         | 0x0   |
| 4   | R/W  | COEX_IRQ_ON_SMART_PRI_F | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_SMART_PRI_F] will cause COEX_IRQ_STAT to be set also to "1".                                       | 0x0   |
| 3   | R/W  | COEX_IRQ_ON_SMART_PRI_R | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_SMART_PRI_R] will cause COEX_IRQ_STAT to be set also to "1".                                       | 0x0   |
| 2   | R/W  | COEX_IRQ_ON_SMART_ACT_F | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_SMART_ACT_F] will cause COEX_IRQ_STAT to be set also to "1".                                       | 0x0   |
| 1   | R/W  | COEX_IRQ_ON_SMART_ACT_R | If "1" then a "1" on COEX_INT_STAT_REG[COEX_IRQ_ON_SMART_ACT_R] will cause COEX_IRQ_STAT to be set also to "1".                                       | 0x0   |
| 0   | R/W  | COEX_IRQ_MASK           | If set to "1" then sent an COEX_IRQ event to CPU as long as COEX_INT_STAT_REG[COEX_IRQ_STAT] is "1". If cleared then don't sent any IRQ event to CPU. | 0x0   |

Table 570: COEX\_INT\_STAT\_REG (0x50002F06)

| Bit | Mode | Symbol                   | Description                                                                                                                                                                                                                       | Reset |
|-----|------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15  | RC   | COEX_IRQ_ON_DECISION_SW  | IRQ event when the decision switches to a new MAC. It ignores the intermediate transitions to DECISION==NONE. Note that after a Radio domain reset, the first transition of the decision to any MAC will also trigger this event. | 0x0   |
| 14  | RC   | COEX_IRQ_ON_START_MID    | IRQ event when the decision switches to a MAC, while the TX_EN or RX_EN of this MAC are high. This event signals a potential break of a transmission or reception.                                                                | 0x0   |
| 13  | RC   | COEX_IRQ_ON_CLOSING_BRK  | IRQ if while entering into "closing" sub-state, the TX_EN or RX_EN are active. This event signals a potential break of a transmission or reception.                                                                               | 0x0   |
| 12  | RC   | COEX_IRQ_ON_RADIO_BUSY_F | IRQ event on falling edge of RADIO_BUSY.                                                                                                                                                                                          | 0x0   |
| 11  | RC   | COEX_IRQ_ON_RADIO_BUSY_R | IRQ event on rising edge of RADIO_BUSY.                                                                                                                                                                                           | 0x0   |
| 10  | RC   | COEX_IRQ_ON_BLE_ACTIVE_F | IRQ event on falling edge of BLE_ACTIVE internal signal.                                                                                                                                                                          | 0x0   |
| 9   | RC   | COEX_IRQ_ON_BLE_ACTIVE_R | IRQ event on rising edge of BLE_ACTIVE internal signal.                                                                                                                                                                           | 0x0   |
| 8   | -    | -                        | Reserved                                                                                                                                                                                                                          | 0x0   |
| 7   | -    | -                        | Reserved                                                                                                                                                                                                                          | 0x0   |
| 6   | RC   | COEX_IRQ_ON_EXT_ACT_F    | RQ event on falling edge of EXT_ACT.                                                                                                                                                                                              | 0x0   |
| 5   | RC   | COEX_IRQ_ON_EXT_ACT_R    | IRQ event on rising edge of EXT_ACT.                                                                                                                                                                                              | 0x0   |
| 4   | RC   | COEX_IRQ_ON_SMART_PRI_F  | IRQ event on falling edge of SMART_PRI.                                                                                                                                                                                           | 0x0   |

Table 570: COEX\_INT\_STAT\_REG (0x50002F06)

| Bit | Mode | Symbol                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Reset |
|-----|------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3   | RC   | COEX_IRQ_ON_SMART_PRI_R | IRQ event on rising edge of SMART_PRI.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 2   | RC   | COEX_IRQ_ON_SMART_ACT_F | IRQ event on falling edge of SMART_ACT.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 1   | RC   | COEX_IRQ_ON_SMART_ACT_R | IRQ event on rising edge of SMART_ACT.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 0x0   |
| 0   | RC   | COEX_IRQ_STAT           | For each COEX_IRQ_ON_* bitfield of COEX_INT_STAT_REG the corresponding mask is applied and afterwards all the intermediate results are combined with a logical OR in order to produce the COEX_IRQ_STAT bitfield. If furthermore the COEX_INT_MASK_REG[COEX_IRQ_MASK] is set to "1", then a COEX_IRQ signal will be forwarded to the CPU.<br>Note: Each COEX_IRQ_ON_* bitfield of COEX_INT_STAT_REG will be set to "1" on the detection of the corresponding event and will be cleared to "0" on the read of COEX_INT_STAT_REG. The automated clear may delay a couple of PCLK cycles, depending on the relationship between PCLK and COEX_CLK. | 0x0   |

Table 571: COEX\_BLE\_PTI\_REG (0x50002F08)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                            | Reset |
|------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:3 | -    | -            | Reserved                                                                                                                                                                                                                               | 0x0   |
| 2:0  | R/W  | COEX_BLE_PTI | This value specifies the PTI value that characterizes the next BLE transaction that will be initiated on the following "ble_active" positive edge. The value should remain constant during the high period of the "ble_active" signal. | 0x0   |

Table 572: COEX\_DIAG\_REG (0x50002F0C)

| Bit  | Mode | Symbol        | Description                                                                                                                                     | Reset |
|------|------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | RC   | COEX_DIAG_MON | provides the current value of the diagnostic bus forwarded to the GPIO multiplexing (named PPA). Refer to the Pxy_MODE_REG[PID] value BLE_DIAG. | 0x0   |

Table 573: COEX\_PRI1\_REG (0x50002F12)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                          | Reset |
|------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -            | Reserved                                                                                                                                                                                                             | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Specifies the MAC that has been assigned with the specific priority level. The MAC encoding follows the COEX_DECISION bitfield encoding.                                                                             | 0x3   |
| 2:0  | R/W  | COEX_PRI_PTI | The priority level specified by this register will be applied to the packets coming from the MAC specified by the COEX_PRI_MAC bitfield and characterized with the PTI value specified by the COEX_PRI_PTI bitfield. | 0x0   |

Table 574: COEX\_PRI2\_REG (0x50002F14)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:5 | -    | -      | Reserved    | 0x0   |

Table 574: COEX\_PRI2\_REG (0x50002F14)

| Bit | Mode | Symbol       | Description             | Reset |
|-----|------|--------------|-------------------------|-------|
| 4:3 | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x1   |
| 2:0 | R/W  | COEX_PRI_PT1 | Refer to COEX_PRI1_REG. | 0x0   |

Table 575: COEX\_PRI3\_REG (0x50002F16)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x2   |
| 2:0  | R/W  | COEX_PRI_PT1 | Refer to COEX_PRI1_REG. | 0x0   |

Table 576: COEX\_PRI4\_REG (0x50002F18)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PT1 | Refer to COEX_PRI1_REG. | 0x0   |

Table 577: COEX\_PRI5\_REG (0x50002F1A)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PT1 | Refer to COEX_PRI1_REG. | 0x0   |

Table 578: COEX\_PRI6\_REG (0x50002F1C)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PT1 | Refer to COEX_PRI1_REG. | 0x0   |

Table 579: COEX\_PRI7\_REG (0x50002F1E)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PT1 | Refer to COEX_PRI1_REG. | 0x0   |

Table 580: COEX\_PRI8\_REG (0x50002F20)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PT1 | Refer to COEX_PRI1_REG. | 0x0   |

Table 581: COEX\_PRI9\_REG (0x50002F22)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

Table 582: COEX\_PRI10\_REG (0x50002F24)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

Table 583: COEX\_PRI11\_REG (0x50002F26)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

Table 584: COEX\_PRI12\_REG (0x50002F28)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

Table 585: COEX\_PRI13\_REG (0x50002F2A)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

Table 586: COEX\_PRI14\_REG (0x50002F2C)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

Table 587: COEX\_PRI15\_REG (0x50002F2E)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |



Table 588: COEX\_PRI16\_REG (0x50002F30)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

Table 589: COEX\_PRI17\_REG (0x50002F32)

| Bit  | Mode | Symbol       | Description             | Reset |
|------|------|--------------|-------------------------|-------|
| 15:5 | -    | -            | Reserved                | 0x0   |
| 4:3  | R/W  | COEX_PRI_MAC | Refer to COEX_PRI1_REG. | 0x0   |
| 2:0  | R/W  | COEX_PRI_PTI | Refer to COEX_PRI1_REG. | 0x0   |

## 37.21 GPIO REGISTER FILE

Table 590: Register map GPIO

| Address    | Port              | Description                     |
|------------|-------------------|---------------------------------|
| 0x50003000 | P0_DATA_REG       | P0 Data input / output Register |
| 0x50003002 | P1_DATA_REG       | P1 Data input / output Register |
| 0x50003004 | P2_DATA_REG       | P2 Data input / output Register |
| 0x50003006 | P3_DATA_REG       | P3 Data input / output Register |
| 0x50003008 | P4_DATA_REG       | P4 Data input / output Register |
| 0x5000300A | P0_SET_DATA_REG   | P0 Set port pins Register       |
| 0x5000300C | P1_SET_DATA_REG   | P1 Set port pins Register       |
| 0x5000300E | P2_SET_DATA_REG   | P2 Set port pins Register       |
| 0x50003010 | P3_SET_DATA_REG   | P3 Set port pins Register       |
| 0x50003012 | P4_SET_DATA_REG   | P4 Set port pins Register       |
| 0x50003014 | P0_RESET_DATA_REG | P0 Reset port pins Register     |
| 0x50003016 | P1_RESET_DATA_REG | P1 Reset port pins Register     |
| 0x50003018 | P2_RESET_DATA_REG | P2 Reset port pins Register     |
| 0x5000301A | P3_RESET_DATA_REG | P3 Reset port pins Register     |
| 0x5000301C | P4_RESET_DATA_REG | P4 Reset port pins Register     |
| 0x5000301E | P00_MODE_REG      | P00 Mode Register               |
| 0x50003020 | P01_MODE_REG      | P01 Mode Register               |
| 0x50003022 | P02_MODE_REG      | P02 Mode Register               |
| 0x50003024 | P03_MODE_REG      | P03 Mode Register               |
| 0x50003026 | P04_MODE_REG      | P04 Mode Register               |
| 0x50003028 | P05_MODE_REG      | P05 Mode Register               |
| 0x5000302A | P06_MODE_REG      | P06 Mode Register               |
| 0x5000302C | P07_MODE_REG      | P07 Mode Register               |
| 0x5000302E | P10_MODE_REG      | P10 Mode Register               |
| 0x50003030 | P11_MODE_REG      | P11 Mode Register               |
| 0x50003032 | P12_MODE_REG      | P12 Mode Register               |
| 0x50003034 | P13_MODE_REG      | P13 Mode Register               |
| 0x50003036 | P14_MODE_REG      | P14 Mode Register               |
| 0x50003038 | P15_MODE_REG      | P15 Mode Register               |
| 0x5000303A | P16_MODE_REG      | P24 Mode Register               |
| 0x5000303C | P17_MODE_REG      | P25 Mode Register               |
| 0x5000303E | P20_MODE_REG      | P20 Mode Register               |
| 0x50003040 | P21_MODE_REG      | P21 Mode Register               |
| 0x50003042 | P22_MODE_REG      | P22 Mode Register               |
| 0x50003044 | P23_MODE_REG      | P23 Mode Register               |
| 0x50003046 | P24_MODE_REG      | P24 Mode Register               |
| 0x5000304E | P30_MODE_REG      | P30 Mode Register               |
| 0x50003050 | P31_MODE_REG      | P31 Mode Register               |
| 0x50003052 | P32_MODE_REG      | P32 Mode Register               |
| 0x50003054 | P33_MODE_REG      | P33 Mode Register               |
| 0x50003056 | P34_MODE_REG      | P34 Mode Register               |
| 0x50003058 | P35_MODE_REG      | P35 Mode Register               |
| 0x5000305A | P36_MODE_REG      | P36 Mode Register               |

Table 590: Register map GPIO

| Address    | Port               | Description                              |
|------------|--------------------|------------------------------------------|
| 0x5000305C | P37_MODE_REG       | P37 Mode Register                        |
| 0x5000305E | P40_MODE_REG       | P40 Mode Register                        |
| 0x50003060 | P41_MODE_REG       | P41 Mode Register                        |
| 0x50003062 | P42_MODE_REG       | P42 Mode Register                        |
| 0x50003064 | P43_MODE_REG       | P43 Mode Register                        |
| 0x50003066 | P44_MODE_REG       | P44 Mode Register                        |
| 0x50003068 | P45_MODE_REG       | P45 Mode Register                        |
| 0x5000306A | P46_MODE_REG       | P46 Mode Register                        |
| 0x5000306C | P47_MODE_REG       | P47 Mode Register                        |
| 0x500030C0 | P0_PADPWR_CTRL_REG | P0 Output Power Control Register         |
| 0x500030C2 | P1_PADPWR_CTRL_REG | P1 Output Power Control Register         |
| 0x500030C4 | P2_PADPWR_CTRL_REG | P2 Output Power Control Register         |
| 0x500030C6 | P3_PADPWR_CTRL_REG | P3 Output Power Control Register         |
| 0x500030C8 | P4_PADPWR_CTRL_REG | P4 Output Power Control Register         |
| 0x500030D0 | GPIO_CLK_SEL       | Select which clock to map on port in PPA |

Table 591: P0\_DATA\_REG (0x50003000)

| Bit  | Mode | Symbol  | Description                                                                 | Reset |
|------|------|---------|-----------------------------------------------------------------------------|-------|
| 15:8 | -    | -       | Reserved                                                                    | 0x0   |
| 7:0  | R/W  | P0_DATA | Set P0 output register when written; Returns the value of P0 port when read | 0x0   |

Table 592: P1\_DATA\_REG (0x50003002)

| Bit  | Mode | Symbol  | Description                                                                 | Reset |
|------|------|---------|-----------------------------------------------------------------------------|-------|
| 15:8 | -    | -       | Reserved                                                                    | 0x0   |
| 7:0  | R/W  | P1_DATA | Set P1 output register when written; Returns the value of P1 port when read | 0x0   |

Table 593: P2\_DATA\_REG (0x50003004)

| Bit  | Mode | Symbol  | Description                                                                 | Reset |
|------|------|---------|-----------------------------------------------------------------------------|-------|
| 15:5 | -    | -       | Reserved                                                                    | 0x0   |
| 4:0  | R/W  | P2_DATA | Set P2 output register when written; Returns the value of P2 port when read | 0x0   |

Table 594: P3\_DATA\_REG (0x50003006)

| Bit  | Mode | Symbol  | Description                                                                 | Reset |
|------|------|---------|-----------------------------------------------------------------------------|-------|
| 15:8 | -    | -       | Reserved                                                                    | 0x0   |
| 7:0  | R/W  | P3_DATA | Set P3 output register when written; Returns the value of P3 port when read | 0x0   |

Table 595: P4\_DATA\_REG (0x50003008)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 595: P4\_DATA\_REG (0x50003008)

| Bit | Mode | Symbol  | Description                                                                 | Reset |
|-----|------|---------|-----------------------------------------------------------------------------|-------|
| 7:0 | R/W  | P4_DATA | Set P4 output register when written; Returns the value of P4 port when read | 0x0   |

Table 596: P0\_SET\_DATA\_REG (0x5000300A)

| Bit  | Mode | Symbol | Description                                                                     | Reset |
|------|------|--------|---------------------------------------------------------------------------------|-------|
| 15:8 | -    | -      | Reserved                                                                        | 0x0   |
| 7:0  | R/W  | P0_SET | Writing a 1 to P0[y] sets P0[y] to 1. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 597: P1\_SET\_DATA\_REG (0x5000300C)

| Bit  | Mode | Symbol | Description                                                                     | Reset |
|------|------|--------|---------------------------------------------------------------------------------|-------|
| 15:8 | -    | -      | Reserved                                                                        | 0x0   |
| 7:0  | R/W  | P1_SET | Writing a 1 to P1[y] sets P1[y] to 1. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 598: P2\_SET\_DATA\_REG (0x5000300E)

| Bit  | Mode | Symbol | Description                                                                     | Reset |
|------|------|--------|---------------------------------------------------------------------------------|-------|
| 15:5 | -    | -      | Reserved                                                                        | 0x0   |
| 4:0  | R/W  | P2_SET | Writing a 1 to P2[y] sets P2[y] to 1. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 599: P3\_SET\_DATA\_REG (0x50003010)

| Bit  | Mode | Symbol | Description                                                                     | Reset |
|------|------|--------|---------------------------------------------------------------------------------|-------|
| 15:8 | -    | -      | Reserved                                                                        | 0x0   |
| 7:0  | R0/W | P3_SET | Writing a 1 to P3[y] sets P3[y] to 1. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 600: P4\_SET\_DATA\_REG (0x50003012)

| Bit  | Mode | Symbol | Description                                                                     | Reset |
|------|------|--------|---------------------------------------------------------------------------------|-------|
| 15:8 | -    | -      | Reserved                                                                        | 0x0   |
| 7:0  | R0/W | P4_SET | Writing a 1 to P4[y] sets P4[y] to 1. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 601: P0\_RESET\_DATA\_REG (0x50003014)

| Bit  | Mode | Symbol   | Description                                                                     | Reset |
|------|------|----------|---------------------------------------------------------------------------------|-------|
| 15:8 | -    | -        | Reserved                                                                        | 0x0   |
| 7:0  | R/W  | P0_RESET | Writing a 1 to P0[y] sets P0[y] to 0. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 602: P1\_RESET\_DATA\_REG (0x50003016)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

**Table 602: P1\_RESET\_DATA\_REG (0x50003016)**

| Bit | Mode | Symbol   | Description                                                                        | Reset |
|-----|------|----------|------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | P1_RESET | Writing a 1 to P1[y] sets P1[y] to 0. Writing 0 is discarded;<br>Reading returns 0 | 0x0   |

**Table 603: P2\_RESET\_DATA\_REG (0x50003018)**

| Bit  | Mode | Symbol   | Description                                                                        | Reset |
|------|------|----------|------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -        | Reserved                                                                           | 0x0   |
| 4:0  | R/W  | P2_RESET | Writing a 1 to P2[y] sets P2[y] to 0. Writing 0 is discarded;<br>Reading returns 0 | 0x0   |

**Table 604: P3\_RESET\_DATA\_REG (0x5000301A)**

| Bit  | Mode | Symbol   | Description                                                                        | Reset |
|------|------|----------|------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -        | Reserved                                                                           | 0x0   |
| 7:0  | R0/W | P3_RESET | Writing a 1 to P3[y] sets P3[y] to 0. Writing 0 is discarded;<br>Reading returns 0 | 0x0   |

**Table 605: P4\_RESET\_DATA\_REG (0x5000301C)**

| Bit  | Mode | Symbol   | Description                                                                        | Reset |
|------|------|----------|------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -        | Reserved                                                                           | 0x0   |
| 7:0  | R0/W | P4_RESET | Writing a 1 to P4[y] sets P4[y] to 0. Writing 0 is discarded;<br>Reading returns 0 | 0x0   |

**Table 606: P00\_MODE\_REG (0x5000301E)**

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |

Table 606: P00\_MODE\_REG (0x5000301E)

| Bit | Mode | Symbol | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Reset |
|-----|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5:0 | R/W  | PID    | Function of port:<br>0: GPIO, PUPD (see above)<br>1: UART_RX<br>2: UART_TX<br>3: UART_IRDA_RX<br>4: UART_IRDA_TX<br>5: UART2_RX<br>6: UART2_TX<br>7: UART2_IRDA_RX<br>8: UART2_IRDA_TX<br>9: UART2_CTSN<br>10: UART2_RTSN<br>11: SPI_DI<br>12: SPI_DO<br>13: SPI_CLK<br>14: SPI_EN<br>15: SPI2_DI<br>16: SPI2_DO<br>17: SPI2_CLK<br>18: SPI2_EN<br>19: I2C_SCL<br>20: I2C_SDA<br>21: I2C2_SCL<br>22: I2C2_SDA<br>23: PWM0<br>24: PWM1<br>25: PWM2<br>26: PWM3<br>27: PWM4<br>28: BLE_DIAG (ble_diag_0: P2_0, ble_diag_1: P2_1,<br>ble_diag_2: P2_2, ble_diag_3: P1_0, ble_diag_4: P1_1,<br>ble_diag_5: P1_2, ble_diag_6: P1_3, ble_diag_7: P2_3)<br>29: Reserved<br>30: PCM_DI<br>31: PCM_DO<br>32: PCM_FSC<br>33: PCM_CLK<br>34: PDM_DI<br>35: PDM_DO<br>36: PDM_CLK<br>37: USB_SOF<br>38: ADC (only for P0[7:6], P1[5:2,0] and P2[4])<br>38: USB (only for P2[2] and P1[1])<br>38: XTAL32 (only for P2[1:0])<br>39: QD_CHA_X<br>40: QD_CHB_X<br>41: QD_CHA_Y<br>42: QD_CHB_Y<br>43: QD_CHA_Z<br>44: QD_CHB_Z<br>45: IR_OUT<br>46: BREATH<br>47: KB_ROW<br>48: COEX_EXT_ACT0<br>49: COEX_EXT_ACT1<br>50: COEX_SMART_ACT<br>(continued on next page) | 0x0   |

**Table 606: P00\_MODE\_REG (0x5000301E)**

| Bit | Mode | Symbol             | Description                                                                                                                                                                                                                           | Reset |
|-----|------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5:0 | R/W  | PID<br>(continued) | 51: COEX_SMART_PRI<br>52: CLOCK<br>53: ONESHOT<br>54: PWM5<br>55: PORT0_DCF<br>56: PORT1_DCF<br>57: PORT2_DCF<br>58: PORT3_DCF<br>59: PORT4_DCF<br>60: RF_ANT_TRIM[0]<br>61: RF_ANT_TRIM[1]<br>62: RF_ANT_TRIM[2]<br><b>(Note 21)</b> | 0x0   |

**Note 21:** Note: when a certain input function (like SPI\_DI) is selected on more than 1 port pin, the port with the lowest index has the highest priority and P0 has higher priority than P1.

**Table 607: P01\_MODE\_REG (0x50003020)**

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

**Table 608: P02\_MODE\_REG (0x50003022)**

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

**Table 609: P03\_MODE\_REG (0x50003024)**

| Bit   | Mode | Symbol | Description                   | Reset |
|-------|------|--------|-------------------------------|-------|
| 15:11 | -    | -      | Reserved                      | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain | 0x0   |



Table 609: P03\_MODE\_REG (0x50003024)

| Bit | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-----|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 9:8 | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0 | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 610: P04\_MODE\_REG (0x50003026)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 611: P05\_MODE\_REG (0x50003028)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x1   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 612: P06\_MODE\_REG (0x5000302A)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 613: P07\_MODE\_REG (0x5000302C)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 614: P10\_MODE\_REG (0x5000302E)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 615: P11\_MODE\_REG (0x50003030)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 616: P12\_MODE\_REG (0x50003032)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |

**Table 616: P12\_MODE\_REG (0x50003032)**

| Bit | Mode | Symbol | Description           | Reset |
|-----|------|--------|-----------------------|-------|
| 7:6 | -    | -      | Reserved              | 0x0   |
| 5:0 | R/W  | PID    | See P00_MODE_REG[PID] | 0x0   |

**Table 617: P13\_MODE\_REG (0x50003034)**

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

**Table 618: P14\_MODE\_REG (0x50003036)**

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

**Table 619: P15\_MODE\_REG (0x50003038)**

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x1   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

**Table 620: P16\_MODE\_REG (0x5000303A)**

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:11 | -    | -      | Reserved    | 0x0   |

Table 620: P16\_MODE\_REG (0x5000303A)

| Bit | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-----|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10  | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8 | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x1   |
| 7:6 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0 | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 621: P17\_MODE\_REG (0x5000303C)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 622: P20\_MODE\_REG (0x5000303E)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 623: P21\_MODE\_REG (0x50003040)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 624: P22\_MODE\_REG (0x50003042)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 625: P23\_MODE\_REG (0x50003044)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 626: P24\_MODE\_REG (0x50003046)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 627: P30\_MODE\_REG (0x5000304E)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |

Table 627: P30\_MODE\_REG (0x5000304E)

| Bit | Mode | Symbol | Description           | Reset |
|-----|------|--------|-----------------------|-------|
| 7:6 | -    | -      | Reserved              | 0x0   |
| 5:0 | R/W  | PID    | See P00_MODE_REG[PID] | 0x0   |

Table 628: P31\_MODE\_REG (0x50003050)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 629: P32\_MODE\_REG (0x50003052)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 630: P33\_MODE\_REG (0x50003054)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 631: P34\_MODE\_REG (0x50003056)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:11 | -    | -      | Reserved    | 0x0   |

Table 631: P34\_MODE\_REG (0x50003056)

| Bit | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-----|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10  | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8 | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0 | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 632: P35\_MODE\_REG (0x50003058)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 633: P36\_MODE\_REG (0x5000305A)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 634: P37\_MODE\_REG (0x5000305C)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |



Table 635: P40\_MODE\_REG (0x5000305E)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 636: P41\_MODE\_REG (0x50003060)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 637: P42\_MODE\_REG (0x50003062)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 638: P43\_MODE\_REG (0x50003064)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |

Table 638: P43\_MODE\_REG (0x50003064)

| Bit | Mode | Symbol | Description           | Reset |
|-----|------|--------|-----------------------|-------|
| 7:6 | -    | -      | Reserved              | 0x0   |
| 5:0 | R/W  | PID    | See P00_MODE_REG[PID] | 0x0   |

Table 639: P44\_MODE\_REG (0x50003066)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 640: P45\_MODE\_REG (0x50003068)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 641: P46\_MODE\_REG (0x5000306A)

| Bit   | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:11 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 10    | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8   | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6   | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0   | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 642: P47\_MODE\_REG (0x5000306C)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:11 | -    | -      | Reserved    | 0x0   |

Table 642: P47\_MODE\_REG (0x5000306C)

| Bit | Mode | Symbol | Description                                                                                                                                                                         | Reset |
|-----|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 10  | R/W  | PPOD   | 0: Push pull<br>1: Open drain                                                                                                                                                       | 0x0   |
| 9:8 | R/W  | PUPD   | 00 = Input, no resistors selected<br>01 = Input, pull-up selected<br>10 = Input, pull-down selected<br>11 = Output, no resistors selected<br>In ADC mode, these bits are don't care | 0x2   |
| 7:6 | -    | -      | Reserved                                                                                                                                                                            | 0x0   |
| 5:0 | R/W  | PID    | See P00_MODE_REG[PID]                                                                                                                                                               | 0x0   |

Table 643: P0\_PADPWR\_CTRL\_REG (0x500030C0)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                             | Reset |
|------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                                                                                                | 0x0   |
| 7:6  | R/W  | P0_OUT_CTRL | 1 = P0_x port output is powered by VDD1V8P rail<br>0 = P0_x port output is powered by V33 rail<br>bit 6 controls the power supply of P0[6],<br>bit 7 controls the power supply of P0[7] | 0x0   |
| 5:0  | -    | -           | Reserved                                                                                                                                                                                | 0x0   |

Table 644: P1\_PADPWR\_CTRL\_REG (0x500030C2)

| Bit  | Mode | Symbol      | Description                                                                                                                                | Reset |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                                                   | 0x0   |
| 7:0  | R/W  | P1_OUT_CTRL | 1 = P1_x port output is powered by VDD1V8P rail<br>0 = P1_x port output is powered by V33 rail<br>bit x controls the power supply of P1[x] | 0x0   |

Table 645: P2\_PADPWR\_CTRL\_REG (0x500030C4)

| Bit  | Mode | Symbol      | Description                                                                                                                                | Reset |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:5 | -    | -           | Reserved                                                                                                                                   | 0x0   |
| 4:0  | R/W  | P2_OUT_CTRL | 1 = P2_x port output is powered by VDD1V8P rail<br>0 = P2_x port output is powered by V33 rail<br>bit x controls the power supply of P2[x] | 0x0   |

Table 646: P3\_PADPWR\_CTRL\_REG (0x500030C6)

| Bit  | Mode | Symbol      | Description                                                                                                                                | Reset |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                                                   | 0x0   |
| 7:0  | R/W  | P3_OUT_CTRL | 1 = P3_x port output is powered by VDD1V8P rail<br>0 = P3_x port output is powered by V33 rail<br>bit x controls the power supply of P3[x] | 0x0   |

Table 647: P4\_PADPWR\_CTRL\_REG (0x500030C8)

| Bit  | Mode | Symbol      | Description                                                                                                                                | Reset |
|------|------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                                                   | 0x0   |
| 7:0  | R/W  | P4_OUT_CTRL | 1 = P4_x port output is powered by VDD1V8P rail<br>0 = P4_x port output is powered by V33 rail<br>bit x controls the power supply of P4[x] | 0x0   |

Table 648: GPIO\_CLK\_SEL (0x500030D0)

| Bit  | Mode | Symbol         | Description                                                                                                                                                                | Reset |
|------|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:3 | -    | -              | Reserved                                                                                                                                                                   | 0x0   |
| 2:0  | R/W  | FUNC_CLOCK_SEL | Select which clock to map when PID = FUNC_CLOCK.<br>0x0: XTAL32K<br>0x1: RC32K<br>0x2: RCX32K<br>0x3: XTAL16M<br>0x4: RC16M<br>0x5: DIVN<br>0x6: Reserved<br>0x7: Reserved | 0x0   |

## 37.22 WDOG REGISTER FILE

Table 649: Register map WDOG

| Address    | Port              | Description                |
|------------|-------------------|----------------------------|
| 0x50003100 | WATCHDOG_REG      | Watchdog timer register.   |
| 0x50003102 | WATCHDOG_CTRL_REG | Watchdog control register. |

Table 650: WATCHDOG\_REG (0x50003100)

| Bit  | Mode | Symbol       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:9 | R0/W | WDOG_WEN     | 0000.000 = Write enable for Watchdog timer else Write disable. This filter prevents unintentional presetting the watchdog with a SW run-away.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 8    | R/W  | WDOG_VAL_NEG | 0 = Watchdog timer value is positive.<br>1 = Watchdog timer value is negative.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 7:0  | R/W  | WDOG_VAL     | <p><u>Write</u>: Watchdog timer reload value. Note that all bits 15-9 must be 0 to reload this register.</p> <p><u>Read</u>: Actual Watchdog timer value. Decrement by 1 every 10.24 msec. Bit 8 indicates a negative counter value. 2, 1, 0, 1FF<sub>16</sub>, 1FE<sub>16</sub> etc. An NMI or WDOG (SYS) reset is generated under the following conditions:</p> <p>If WATCHDOG_CTRL_REG[NMI_RST] = 0 then</p> <p style="padding-left: 20px;">If WDOG_VAL = 0 -&gt; NMI (Non Maskable Interrupt)</p> <p style="padding-left: 20px;">if WDOG_VAL = 1F0<sub>16</sub> -&gt; WDOG reset -&gt; reload FF<sub>16</sub></p> <p>If WATCHDOG_CTRL_REG[NMI_RST] = 1 then</p> <p style="padding-left: 20px;">if WDOG_VAL &lt;= 0 -&gt; WDOG reset -&gt; reload FF<sub>16</sub></p> | 0xFF  |

Table 651: WATCHDOG\_CTRL\_REG (0x50003102)

| Bit  | Mode | Symbol  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reset |
|------|------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:2 | -    | -       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 1    | -    | -       | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 0x0   |
| 0    | R/W  | NMI_RST | <p>0 = Watchdog timer generates NMI at value 0, and WDOG (SYS) reset at &lt;=16. Timer can be frozen /resumed using SET_FREEZE_REG[FRZ_WDOG]/ RESET_FREEZE_REG[FRZ_WDOG].</p> <p>1 = Watchdog timer generates a WDOG (SYS) reset at value 0 and can not be frozen by Software.</p> <p>Note that this bit can only be set to 1 by SW and only be reset with a WDOG (SYS) reset or SW reset.</p> <p>The watchdog is always frozen when the Cortex-M0 is halted in DEBUG State.</p> | 0x0   |

## 37.23 VERSION REGISTER FILE

Table 652: Register map Version

| Address    | Port              | Description                      |
|------------|-------------------|----------------------------------|
| 0x50003200 | CHIP_ID1_REG      | Chip identification register 1.  |
| 0x50003201 | CHIP_ID2_REG      | Chip identification register 2.  |
| 0x50003202 | CHIP_ID3_REG      | Chip identification register 3.  |
| 0x50003203 | CHIP_SWC_REG      | Software compatibility register. |
| 0x50003204 | CHIP_REVISION_REG | Chip revision register.          |

Table 653: CHIP\_ID1\_REG (0x50003200)

| Bit | Mode | Symbol   | Description                                    | Reset |
|-----|------|----------|------------------------------------------------|-------|
| 7:0 | R    | CHIP_ID1 | First character of device type "680" in ASCII. | 0x36  |

Table 654: CHIP\_ID2\_REG (0x50003201)

| Bit | Mode | Symbol   | Description                                     | Reset |
|-----|------|----------|-------------------------------------------------|-------|
| 7:0 | R    | CHIP_ID2 | Second character of device type "680" in ASCII. | 0x38  |

Table 655: CHIP\_ID3\_REG (0x50003202)

| Bit | Mode | Symbol   | Description                                    | Reset |
|-----|------|----------|------------------------------------------------|-------|
| 7:0 | R    | CHIP_ID3 | Third character of device type "680" in ASCII. | 0x30  |

Table 656: CHIP\_SWC\_REG (0x50003203)

| Bit | Mode | Symbol   | Description                                                                                                                                                                                                           | Reset |
|-----|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7:4 | -    | -        | Reserved                                                                                                                                                                                                              | 0x0   |
| 3:0 | R    | CHIP_SWC | <b>SoftWare</b> Compatibility code.<br>Integer (default = 0) which is incremented if a silicon change has impact on the CPU Firmware.<br>Can be used by software developers to write silicon revision dependent code. | 0x0   |

Table 657: CHIP\_REVISION\_REG (0x50003204)

| Bit | Mode | Symbol      | Description                                                                    | Reset |
|-----|------|-------------|--------------------------------------------------------------------------------|-------|
| 7:0 | R    | REVISION_ID | Chip version, corresponds with type number in ASCII.<br>0x41 = 'A', 0x42 = 'B' | 0x41  |

## 37.24 GPREG REGISTER FILE

Table 658: Register map GPREG

| Address    | Port               | Description                                                         |
|------------|--------------------|---------------------------------------------------------------------|
| 0x50003300 | SET_FREEZE_REG     | Controls freezing of various timers/counters (incl. DMA and USB).   |
| 0x50003302 | RESET_FREEZE_REG   | Controls unfreezing of various timers/counters (incl. DMA and USB). |
| 0x50003304 | DEBUG_REG          | Various debug information register.                                 |
| 0x50003306 | GP_STATUS_REG      | General purpose system status register.                             |
| 0x50003308 | GP_CONTROL_REG     | General purpose system control register.                            |
| 0x5000330A | ECC_BASE_ADDR_REG  | Base address of the ECC Crypto memory register.                     |
| 0x5000330C | LED_CONTROL_REG    | Controls muxing and enabling of the LEDs.                           |
| 0x50003310 | PLL_SYS_CTRL1_REG  | System PLL control register 1.                                      |
| 0x50003312 | PLL_SYS_CTRL2_REG  | System PLL control register 2.                                      |
| 0x50003314 | PLL_SYS_CTRL3_REG  | System PLL control register 3.                                      |
| 0x50003316 | PLL_SYS_STATUS_REG | System PLL status register.                                         |

Table 659: SET\_FREEZE\_REG (0x50003300)

| Bit  | Mode | Symbol      | Description                                                                                                                  | Reset |
|------|------|-------------|------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                                                                                     | 0x0   |
| 7    | R/W  | FRZ_SWTIM2  | If '1', the SW Timer (TIMER2) is frozen, '0' is discarded.                                                                   | 0x0   |
| 6    | R/W  | FRZ_SWTIM1  | If '1', the SW Timer (TIMER1) is frozen, '0' is discarded.                                                                   | 0x0   |
| 5    | R/W  | FRZ_DMA     | If '1', the DMA is frozen, '0' is discarded.                                                                                 | 0x0   |
| 4    | R/W  | FRZ_USB     | If '1', the USB is frozen, '0' is discarded.                                                                                 | 0x0   |
| 3    | R/W  | FRZ_WDOG    | If '1', the watchdog timer is frozen, '0' is discarded. WATCHDOG_CTRL_REG[NMI_RST] must be '0' to allow the freeze function. | 0x0   |
| 2    | R/W  | FRZ_BLETIM  | If '1', the BLE master clock is frozen, '0' is discarded.                                                                    | 0x0   |
| 1    | R/W  | FRZ_SWTIM0  | If '1', the SW Timer (TIMER0) is frozen, '0' is discarded.                                                                   | 0x0   |
| 0    | R/W  | FRZ_WKUPTIM | If '1', the Wake Up Timer is frozen, '0' is discarded.                                                                       | 0x0   |

Table 660: RESET\_FREEZE\_REG (0x50003302)

| Bit  | Mode | Symbol      | Description                                                | Reset |
|------|------|-------------|------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                   | 0x0   |
| 7    | R/W  | FRZ_SWTIM2  | If '1', the SW Timer (TIMER2) continues, '0' is discarded. | 0x0   |
| 6    | R/W  | FRZ_SWTIM1  | If '1', the SW Timer (TIMER1) continues, '0' is discarded. | 0x0   |
| 5    | R/W  | FRZ_DMA     | If '1', the DMA continues, '0' is discarded.               | 0x0   |
| 4    | R/W  | FRZ_USB     | If '1', the USB continues, '0' is discarded.               | 0x0   |
| 3    | R/W  | FRZ_WDOG    | If '1', the watchdog timer continues, '0' is discarded.    | 0x0   |
| 2    | R/W  | FRZ_BLETIM  | If '1', the BLE master clock continues, '0' is discarded.  | 0x0   |
| 1    | R/W  | FRZ_SWTIM0  | If '1', the SW Timer (TIMER0) continues, '0' is discarded. | 0x0   |
| 0    | R/W  | FRZ_WKUPTIM | If '1', the Wake Up Timer continues, '0' is discarded.     | 0x0   |

Table 661: DEBUG\_REG (0x50003304)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |



Table 661: DEBUG\_REG (0x50003304)

| Bit | Mode | Symbol           | Description                                                                                                                                                                                                                                                                                                                                                            | Reset |
|-----|------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R/W  | DEBUGS_FREEZE_EN | Default '1', freezing of the on-chip timers is enabled when the Cortex-M0 is halted in DEBUG State.<br>If '0', freezing of the on-chip timers is depending on FREEZE_REG when the Cortex-M0 is halted in DEBUG State <u>except</u> the watchdog timer. The watchdog timer is always frozen when the Cortex-M0 is halted in DEBUG State.<br>Note: This bit is retained. | 0x1   |

Table 662: GP\_STATUS\_REG (0x50003306)

| Bit  | Mode | Symbol    | Description                                                                                                                             | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:1 | -    | -         | Reserved                                                                                                                                | 0x0   |
| 0    | R/W  | CAL_PHASE | If '1', it designates that the chip is in Calibration Phase i.e. the OTP has been initially programmed but no Calibration has occurred. | 0x0   |

Table 663: GP\_CONTROL\_REG (0x50003308)

| Bit  | Mode | Symbol                | Description                                                                                                                                                                                                                               | Reset |
|------|------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | R    | BLE_DEEPSLDUR_MONITOR | The 8 LSBs of the current value of the BLE Timer DEEPSLDUR. The value has been sampled by using the CPU clock.                                                                                                                            | 0x0   |
| 7:3  | -    | -                     | Reserved                                                                                                                                                                                                                                  | 0x0   |
| 2    | R    | BLE_WAKEUP_LP_IRQ     | The current value of the BLE_WAKEUP_LP_IRQ interrupt request.                                                                                                                                                                             | 0x0   |
| 1    | R/W  | BLE_H2H_BRIDGE_BYPASS | If '1', the AHB-to-AHB bridge is bypassed, needed to access the BLE Register file, only when the system clock source is the XTAL and both hclk and ble_hclk are running at 16MHz, i.e. at the XTAL clock rate.                            | 0x0   |
| 0    | R/W  | BLE_WAKEUP_REQ        | If '1', the BLE wakes up. Must be kept high at least for 1 low power clock period.<br>If the BLE is in deep sleep state, then by setting this bit it will cause the wakeup LP IRQ to be asserted with a delay of 3 to 4 low power cycles. | 0x0   |

Table 664: ECC\_BASE\_ADDR\_REG (0x5000330A)

| Bit  | Mode | Symbol        | Description                                                                                                                                                                                                                                                                                                                                          | Reset |
|------|------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 7    | -    | -             | Reserved                                                                                                                                                                                                                                                                                                                                             | 0x0   |
| 6:0  | R/W  | ECC_BASE_ADDR | Contains the base address of the ECC Crypto memory. Memory allocation is in pages of 1KB and up to 127KB. Since the ECC has an address range of 2KB and the total addressable memory range is 128KB, the maximum value of 0x7F (127KB offset) will result in 1KB at the top of the memory range and the other 1KB at the bottom of the memory range. | 0x0   |

Table 665: LED\_CONTROL\_REG (0x5000330C)

| Bit  | Mode | Symbol       | Description                                                                                                                      | Reset |
|------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:6 | -    | -            | Reserved                                                                                                                         | 0x0   |
| 5    | R/W  | LED3_EN      | 0: LED3 disabled,<br>1: LED3 enabled.                                                                                            | 0x0   |
| 4    | R/W  | LED2_EN      | 0: LED2 disabled,<br>1: LED2 enabled.                                                                                            | 0x0   |
| 3    | R/W  | LED1_EN      | 0: LED1 disabled,<br>1: LED1 enabled.                                                                                            | 0x0   |
| 2    | R/W  | LED3_SRC_SEL | 0: LED3 = PWM4,<br>1: LED3 = Breathing Timer.                                                                                    | 0x0   |
| 1    | R/W  | LED2_SRC_SEL | 0: LED2 = PWM3,<br>1: LED2 = Breathing Timer.                                                                                    | 0x0   |
| 0    | R/W  | LED1_SRC_SEL | 0: LED1 = PWM2,<br>1: LED1 = Breathing Timer.<br>Note: The PWM2/3/4 can also be routed to GPIOs using PID 25/26/27 respectively. | 0x0   |

Table 666: PLL\_SYS\_CTRL1\_REG (0x50003310)

| Bit  | Mode | Symbol            | Description                                                                                            | Reset |
|------|------|-------------------|--------------------------------------------------------------------------------------------------------|-------|
| 15   | -    | -                 | Reserved                                                                                               | 0x0   |
| 14:8 | R/W  | PLL_R_DIV         | PLL Output divider R (x means divide by x, 0 means divide by 1)                                        | 0x1   |
| 7:3  | -    | -                 | Reserved                                                                                               | 0x0   |
| 2    | R/W  | LDO_PLL_VREF_HOLD | 0: indicates that the reference input is tracked,<br>1: indicates that the reference input is sampled. | 0x0   |
| 1    | R/W  | LDO_PLL_ENABLE    | 0: LDO PLL off,<br>1: LDO PLL on.                                                                      | 0x0   |
| 0    | R/W  | PLL_EN            | 0: Power down<br>1: PLL on                                                                             | 0x0   |

Table 667: PLL\_SYS\_CTRL2\_REG (0x50003312)

| Bit   | Mode | Symbol              | Description                                                                                                          | Reset |
|-------|------|---------------------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15    | -    | -                   | Reserved                                                                                                             | 0x0   |
| 14    | R/W  | PLL_SEL_MIN_CUR_INT | 0: VCO current read from min_current <5:0>,<br>1: VCO current is internally determined with a calibration algorithm. | 0x0   |
| 13:12 | R/W  | PLL_DEL_SEL         | PLL manual delay value for Phase Frequency Detector.<br>0: 0.493<br>1: 0.814<br>2: 1.13 ns <- default<br>3: 1.44 ns  | 0x2   |
| 11:8  | -    | -                   | Reserved                                                                                                             | 0x0   |
| 7     | -    | -                   | Reserved                                                                                                             | 0x0   |
| 6:0   | R/W  | PLL_N_DIV           | PLL Loop divider N (x means divide by x, 0 means divide by 1)                                                        | 0x6   |

Table 668: PLL\_SYS\_CTRL3\_REG (0x50003314)

| Bit   | Mode | Symbol        | Description                                                                                                                                                                    | Reset |
|-------|------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15    | R/W  | PLL_RECALIB   | Recalibrate                                                                                                                                                                    | 0x0   |
| 14:10 | R/W  | PLL_START_DEL | Programmable delay time for the loop filter voltage preset value. After PLL_EN is set, the loopfilter precharge resistors are disabled after this delay time. One LSB is 48 ns | 0xF   |
| 9:5   | -    | -             | Reserved                                                                                                                                                                       | 0x0   |
| 4:0   | R/W  | PLL_ICP_SEL   | PLL charge pump current select<br>One LSB is 5uA.                                                                                                                              | 0x9   |

Table 669: PLL\_SYS\_STATUS\_REG (0x50003316)

| Bit   | Mode | Symbol               | Description                                 | Reset |
|-------|------|----------------------|---------------------------------------------|-------|
| 15:12 | -    | -                    | Reserved                                    | 0x0   |
| 11    | R    | PLL_CALIBR_END       | Indicates that calibration has finished.    | 0x0   |
| 10:5  | R    | PLL_PLL_BEST_MIN_CUR | Calibrated VCO frequency band.              | 0x0   |
| 4:2   | -    | -                    | Reserved                                    | 0x0   |
| 1     | R    | LDO_PLL_OK           | 1: Indicates that LDO PLL is in regulation. | 0x0   |
| 0     | R    | PLL_LOCK_FINE        | 1: PLL locked                               | 0x0   |

## 37.25 TIMER0/2 AND BREATH REGISTER FILE

Table 670: Register map Timer0/2 and Breath

| Address    | Port                 | Description                     |
|------------|----------------------|---------------------------------|
| 0x50003400 | TIMER0_CTRL_REG      | Timer0 control register         |
| 0x50003402 | TIMER0_ON_REG        | Timer0 on control register      |
| 0x50003404 | TIMER0_RELOAD_M_REG  | 16 bits reload value for Timer0 |
| 0x50003406 | TIMER0_RELOAD_N_REG  | 16 bits reload value for Timer0 |
| 0x50003408 | PWM2_START_CYCLE     | Defines start Cycle for PWM2    |
| 0x5000340A | PWM3_START_CYCLE     | Defines start Cycle for PWM3    |
| 0x5000340C | PWM4_START_CYCLE     | Defines start Cycle for PWM4    |
| 0x5000340E | PWM2_END_CYCLE       | Defines end Cycle for PWM2      |
| 0x50003410 | PWM3_END_CYCLE       | Defines end Cycle for PWM3      |
| 0x50003412 | PWM4_END_CYCLE       | Defines end Cycle for PWM4      |
| 0x50003414 | TRIPLE_PWM_FREQUENCY | Defines the PMW2,3,4 frequency  |
| 0x50003416 | TRIPLE_PWM_CTRL_REG  | PWM 2 3 4 Control register      |
| 0x50003418 | BREATH_CFG_REG       | Breath configuration register   |
| 0x5000341A | BREATH_DUTY_MAX_REG  | Breath max duty cycle register  |
| 0x5000341C | BREATH_DUTY_MIN_REG  | Breath min duty cycle register  |
| 0x5000341E | BREATH_CTRL_REG      | Breath control register         |

Table 671: TIMER0\_CTRL\_REG (0x50003400)

| Bit  | Mode | Symbol       | Description                                                                                                                                                  | Reset |
|------|------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:4 | -    | -            | Reserved                                                                                                                                                     | 0x0   |
| 3    | R/W  | PWM_MODE     | 0 = PWM signals are '1' during high time.<br>1 = PWM signals send out the (fast) clock divided by 2 during high time.                                        | 0x0   |
| 2    | R/W  | TIM0_CLK_DIV | 1 = Timer0 uses selected clock frequency as is.<br>0 = Timer0 uses selected clock frequency divided by 10.<br>Note that this applies only to the ON-counter. | 0x0   |
| 1    | R/W  | TIM0_CLK_SEL | 1 = Timer0 uses fast clock frequency.<br>0 = Timer0 uses 32 kHz (slow) clock frequency.                                                                      | 0x0   |
| 0    | R/W  | TIM0_CTRL    | 0 = Timer0 is off and in reset state.<br>1 = Timer0 is running.                                                                                              | 0x0   |

Table 672: TIMER0\_ON\_REG (0x50003402)

| Bit  | Mode | Symbol  | Description                                                                      | Reset |
|------|------|---------|----------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | TIM0_ON | Timer0 On reload value:<br>If read the actual counter value ON_CNTER is returned | 0x0   |

Table 673: TIMER0\_RELOAD\_M\_REG (0x50003404)

| Bit  | Mode | Symbol | Description                                                                         | Reset |
|------|------|--------|-------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | TIM0_M | Timer0 'high' reload value<br>If read the actual counter value T0_CNTER is returned | 0x0   |

Table 674: TIMER0\_RELOAD\_N\_REG (0x50003406)

| Bit  | Mode | Symbol | Description                                                                         | Reset |
|------|------|--------|-------------------------------------------------------------------------------------|-------|
| 15:0 | R0/W | TIM0_N | Timer0 'low' reload value:<br>If read the actual counter value T0_CNTER is returned | 0x0   |

Table 675: PWM2\_START\_CYCLE (0x50003408)

| Bit  | Mode | Symbol      | Description                                                                                                                     | Reset |
|------|------|-------------|---------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:0 | R/W  | START_CYCLE | Define the cycle in which the PWM becomes high. if start_cycle larger than freq or end and start are equal, pwm out is always 0 | 0x0   |

Table 676: PWM3\_START\_CYCLE (0x5000340A)

| Bit  | Mode | Symbol      | Description                                                                                                                     | Reset |
|------|------|-------------|---------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:0 | R/W  | START_CYCLE | Define the cycle in which the PWM becomes high. if start_cycle larger than freq or end and start are equal, pwm out is always 0 | 0x0   |

Table 677: PWM4\_START\_CYCLE (0x5000340C)

| Bit  | Mode | Symbol      | Description                                                                                                                     | Reset |
|------|------|-------------|---------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:0 | R/W  | START_CYCLE | Define the cycle in which the PWM becomes high. if start_cycle larger than freq or end and start are equal, pwm out is always 0 | 0x0   |

Table 678: PWM2\_END\_CYCLE (0x5000340E)

| Bit  | Mode | Symbol    | Description                                                                                                                                   | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:0 | R/W  | END_CYCLE | Define the cycle in which the PWM becomes low. If end cycle larger then freq pwm and start cycle not larger then freq pwm, output is always 1 | 0x0   |

Table 679: PWM3\_END\_CYCLE (0x50003410)

| Bit  | Mode | Symbol    | Description                                                                                                                                   | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:0 | R/W  | END_CYCLE | Define the cycle in which the PWM becomes low. If end cycle larger then freq pwm and start cycle not larger then freq pwm, output is always 1 | 0x0   |

Table 680: PWM4\_END\_CYCLE (0x50003412)

| Bit  | Mode | Symbol    | Description                                                                                                                                   | Reset |
|------|------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 13:0 | R/W  | END_CYCLE | Define the cycle in which the PWM becomes low. If end cycle larger then freq pwm and start cycle not larger then freq pwm, output is always 1 | 0x0   |

Table 681: TRIPLE\_PWM\_FREQUENCY (0x50003414)

| Bit  | Mode | Symbol | Description                                        | Reset |
|------|------|--------|----------------------------------------------------|-------|
| 13:0 | R/W  | FREQ   | Freq for PWM 2 3 4, period = timer_clk * ( FREQ+1) | 0x0   |

Table 682: TRIPLE\_PWM\_CTRL\_REG (0x50003416)

| Bit | Mode | Symbol             | Description                  | Reset |
|-----|------|--------------------|------------------------------|-------|
| 2   | R/W  | HW_PAUSE_EN        | '1' = HW can pause PWM 2,3,4 | 0x1   |
| 1   | R/W  | SW_PAUSE_EN        | '1' = PWM 2 3 4 is paused    | 0x0   |
| 0   | R/W  | TRIPLE_PWM_ENA BLE | '1' = PWM 2 3 4 is enabled   | 0x0   |

Table 683: BREATH\_CFG\_REG (0x50003418)

| Bit  | Mode | Symbol    | Description                                                                                           | Reset |
|------|------|-----------|-------------------------------------------------------------------------------------------------------|-------|
| 15:8 | R/W  | BRTH_STEP | Defines the number of PWM periods minus 1, duty cycle will be changed                                 | 0x1   |
| 7:0  | R/W  | BRTH_DIV  | Defines the division factor of the system clock to get to the PWM frequency.( = Sys Clock / (value+1) | 0xFF  |

Table 684: BREATH\_DUTY\_MAX\_REG (0x5000341A)

| Bit | Mode | Symbol        | Description                                                                                  | Reset |
|-----|------|---------------|----------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | BRTH_DUTY_MAX | Defines the maximum duty cycle of the PWM breath function. Duty cycle = value / (brth_div+1) | 0xA   |

Table 685: BREATH\_DUTY\_MIN\_REG (0x5000341C)

| Bit | Mode | Symbol        | Description                                                                                  | Reset |
|-----|------|---------------|----------------------------------------------------------------------------------------------|-------|
| 7:0 | R/W  | BRTH_DUTY_MIN | Defines the minimum duty cycle of the PWM breath function. Duty cycle = value / (brth_div+1) | 0x1   |

Table 686: BREATH\_CTRL\_REG (0x5000341E)

| Bit | Mode | Symbol       | Description                     | Reset |
|-----|------|--------------|---------------------------------|-------|
| 1   | R/W  | BRTH_PWM_POL | Define the output polarity.     | 0x0   |
| 0   | R/W  | BRTH_EN      | '1' enable the Breath operation | 0x0   |

## 37.26 DMA REGISTER FILE

Table 687: Register map DMA

| Address    | Port              | Description                              |
|------------|-------------------|------------------------------------------|
| 0x50003500 | DMA0_A_STARTL_REG | Start address Low A of DMA channel 0     |
| 0x50003502 | DMA0_A_STARTH_REG | Start address High A of DMA channel 0    |
| 0x50003504 | DMA0_B_STARTL_REG | Start address Low B of DMA channel 0     |
| 0x50003506 | DMA0_B_STARTH_REG | Start address High B of DMA channel 0    |
| 0x50003508 | DMA0_INT_REG      | DMA receive interrupt register channel 0 |
| 0x5000350A | DMA0_LEN_REG      | DMA receive length register channel 0    |
| 0x5000350C | DMA0_CTRL_REG     | Control register for the DMA channel 0   |
| 0x5000350E | DMA0_IDX_REG      | Index value of DMA channel 0             |
| 0x50003510 | DMA1_A_STARTL_REG | Start address Low A of DMA channel 1     |
| 0x50003512 | DMA1_A_STARTH_REG | Start address High A of DMA channel 1    |
| 0x50003514 | DMA1_B_STARTL_REG | Start address Low B of DMA channel 1     |
| 0x50003516 | DMA1_B_STARTH_REG | Start address High B of DMA channel 1    |
| 0x50003518 | DMA1_INT_REG      | DMA receive interrupt register channel 1 |
| 0x5000351A | DMA1_LEN_REG      | DMA receive length register channel 1    |
| 0x5000351C | DMA1_CTRL_REG     | Control register for the DMA channel 1   |
| 0x5000351E | DMA1_IDX_REG      | Index value of DMA channel 1             |
| 0x50003520 | DMA2_A_STARTL_REG | Start address Low A of DMA channel 2     |
| 0x50003522 | DMA2_A_STARTH_REG | Start address High A of DMA channel 2    |
| 0x50003524 | DMA2_B_STARTL_REG | Start address Low B of DMA channel 2     |
| 0x50003526 | DMA2_B_STARTH_REG | Start address High B of DMA channel 2    |
| 0x50003528 | DMA2_INT_REG      | DMA receive interrupt register channel 2 |
| 0x5000352A | DMA2_LEN_REG      | DMA receive length register channel 2    |
| 0x5000352C | DMA2_CTRL_REG     | Control register for the DMA channel 2   |
| 0x5000352E | DMA2_IDX_REG      | Index value of DMA channel 2             |
| 0x50003530 | DMA3_A_STARTL_REG | Start address Low A of DMA channel 3     |
| 0x50003532 | DMA3_A_STARTH_REG | Start address High A of DMA channel 3    |
| 0x50003534 | DMA3_B_STARTL_REG | Start address Low B of DMA channel 3     |
| 0x50003536 | DMA3_B_STARTH_REG | Start address High B of DMA channel 3    |
| 0x50003538 | DMA3_INT_REG      | DMA receive interrupt register channel 3 |
| 0x5000353A | DMA3_LEN_REG      | DMA receive length register channel 3    |
| 0x5000353C | DMA3_CTRL_REG     | Control register for the DMA channel 3   |
| 0x5000353E | DMA3_IDX_REG      | Index value of DMA channel 3             |
| 0x50003540 | DMA4_A_STARTL_REG | Start address Low A of DMA channel 4     |
| 0x50003542 | DMA4_A_STARTH_REG | Start address High A of DMA channel 4    |
| 0x50003544 | DMA4_B_STARTL_REG | Start address Low B of DMA channel 4     |
| 0x50003546 | DMA4_B_STARTH_REG | Start address High B of DMA channel 4    |
| 0x50003548 | DMA4_INT_REG      | DMA receive interrupt register channel 4 |
| 0x5000354A | DMA4_LEN_REG      | DMA receive length register channel 4    |
| 0x5000354C | DMA4_CTRL_REG     | Control register for the DMA channel 4   |
| 0x5000354E | DMA4_IDX_REG      | Index value of DMA channel 4             |
| 0x50003550 | DMA5_A_STARTL_REG | Start address Low A of DMA channel 5     |
| 0x50003552 | DMA5_A_STARTH_REG | Start address High A of DMA channel 5    |
| 0x50003554 | DMA5_B_STARTL_REG | Start address Low B of DMA channel 5     |

Table 687: Register map DMA

| Address    | Port               | Description                              |
|------------|--------------------|------------------------------------------|
| 0x50003556 | DMA5_B_STARTH_REG  | Start address High B of DMA channel 5    |
| 0x50003558 | DMA5_INT_REG       | DMA receive interrupt register channel 5 |
| 0x5000355A | DMA5_LEN_REG       | DMA receive length register channel 5    |
| 0x5000355C | DMA5_CTRL_REG      | Control register for the DMA channel 5   |
| 0x5000355E | DMA5_IDX_REG       | Index value of DMA channel 5             |
| 0x50003560 | DMA6_A_STARTL_REG  | Start address Low A of DMA channel 6     |
| 0x50003562 | DMA6_A_STARTH_REG  | Start address High A of DMA channel 6    |
| 0x50003564 | DMA6_B_STARTL_REG  | Start address Low B of DMA channel 6     |
| 0x50003566 | DMA6_B_STARTH_REG  | Start address High B of DMA channel 6    |
| 0x50003568 | DMA6_INT_REG       | DMA receive interrupt register channel 6 |
| 0x5000356A | DMA6_LEN_REG       | DMA receive length register channel 6    |
| 0x5000356C | DMA6_CTRL_REG      | Control register for the DMA channel 6   |
| 0x5000356E | DMA6_IDX_REG       | Index value of DMA channel 6             |
| 0x50003570 | DMA7_A_STARTL_REG  | Start address Low A of DMA channel 7     |
| 0x50003572 | DMA7_A_STARTH_REG  | Start address High A of DMA channel 7    |
| 0x50003574 | DMA7_B_STARTL_REG  | Start address Low B of DMA channel 7     |
| 0x50003576 | DMA7_B_STARTH_REG  | Start address High B of DMA channel 7    |
| 0x50003578 | DMA7_INT_REG       | DMA receive interrupt register channel 7 |
| 0x5000357A | DMA7_LEN_REG       | DMA receive length register channel 7    |
| 0x5000357C | DMA7_CTRL_REG      | Control register for the DMA channel 7   |
| 0x5000357E | DMA7_IDX_REG       | Index value of DMA channel 7             |
| 0x50003580 | DMA_REQ_MUX_REG    | DMA channel assignments                  |
| 0x50003582 | DMA_INT_STATUS_REG | DMA interrupt status register            |
| 0x50003584 | DMA_CLEAR_INT_REG  | DMA clear interrupt register             |

Table 688: DMA0\_A\_STARTL\_REG (0x50003500)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA0_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 689: DMA0\_A\_STARTH\_REG (0x50003502)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA0_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 690: DMA0\_B\_STARTL\_REG (0x50003504)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA0_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 691: DMA0\_B\_STARTH\_REG (0x50003506)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA0_B_STARTH | Destination start address, upper 16 bits | 0x0   |



Table 692: DMA0\_INT\_REG (0x50003508)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA0_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 693: DMA0\_LEN\_REG (0x5000350A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA0_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 694: DMA0\_CTRL\_REG (0x5000350C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 694: DMA0\_CTRL\_REG (0x5000350C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 695: DMA0\_IDX\_REG (0x5000350E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA0_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 696: DMA1\_A\_STARTL\_REG (0x50003510)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA1_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 697: DMA1\_A\_STARTH\_REG (0x50003512)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA1_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 698: DMA1\_B\_STARTL\_REG (0x50003514)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA1_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 699: DMA1\_B\_STARTH\_REG (0x50003516)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA1_B_STARTH | Destination start address, upper 16 bits | 0x0   |

Table 700: DMA1\_INT\_REG (0x50003518)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA1_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 701: DMA1\_LEN\_REG (0x5000351A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA1_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 702: DMA1\_CTRL\_REG (0x5000351C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 702: DMA1\_CTRL\_REG (0x5000351C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 703: DMA1\_IDX\_REG (0x5000351E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA1_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 704: DMA2\_A\_STARTL\_REG (0x50003520)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA2_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 705: DMA2\_A\_STARTH\_REG (0x50003522)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA2_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 706: DMA2\_B\_STARTL\_REG (0x50003524)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA2_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 707: DMA2\_B\_STARTH\_REG (0x50003526)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA2_B_STARTH | Destination start address, upper 16 bits | 0x0   |

Table 708: DMA2\_INT\_REG (0x50003528)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA2_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 709: DMA2\_LEN\_REG (0x5000352A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA2_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 710: DMA2\_CTRL\_REG (0x5000352C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 710: DMA2\_CTRL\_REG (0x5000352C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 711: DMA2\_IDX\_REG (0x5000352E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA2_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 712: DMA3\_A\_STARTL\_REG (0x50003530)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA3_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 713: DMA3\_A\_STARTH\_REG (0x50003532)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA3_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 714: DMA3\_B\_STARTL\_REG (0x50003534)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA3_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 715: DMA3\_B\_STARTH\_REG (0x50003536)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA3_B_STARTH | Destination start address, upper 16 bits | 0x0   |

Table 716: DMA3\_INT\_REG (0x50003538)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA3_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 717: DMA3\_LEN\_REG (0x5000353A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA3_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 718: DMA3\_CTRL\_REG (0x5000353C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                              | 0x0   |



Table 718: DMA3\_CTRL\_REG (0x5000353C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 719: DMA3\_IDX\_REG (0x5000353E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA3_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 720: DMA4\_A\_STARTL\_REG (0x50003540)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA4_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 721: DMA4\_A\_STARTH\_REG (0x50003542)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA4_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 722: DMA4\_B\_STARTL\_REG (0x50003544)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA4_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 723: DMA4\_B\_STARTH\_REG (0x50003546)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA4_B_STARTH | Destination start address, upper 16 bits | 0x0   |



Table 724: DMA4\_INT\_REG (0x50003548)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA4_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 725: DMA4\_LEN\_REG (0x5000354A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA4_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 726: DMA4\_CTRL\_REG (0x5000354C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 726: DMA4\_CTRL\_REG (0x5000354C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 727: DMA4\_IDX\_REG (0x5000354E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA4_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 728: DMA5\_A\_STARTL\_REG (0x50003550)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA5_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 729: DMA5\_A\_STARTH\_REG (0x50003552)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA5_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 730: DMA5\_B\_STARTL\_REG (0x50003554)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA5_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 731: DMA5\_B\_STARTH\_REG (0x50003556)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA5_B_STARTH | Destination start address, upper 16 bits | 0x0   |

Table 732: DMA5\_INT\_REG (0x50003558)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA5_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 733: DMA5\_LEN\_REG (0x5000355A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA5_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 734: DMA5\_CTRL\_REG (0x5000355C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 734: DMA5\_CTRL\_REG (0x5000355C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 735: DMA5\_IDX\_REG (0x5000355E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA5_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 736: DMA6\_A\_STARTL\_REG (0x50003560)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA6_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 737: DMA6\_A\_STARTH\_REG (0x50003562)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA6_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 738: DMA6\_B\_STARTL\_REG (0x50003564)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA6_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 739: DMA6\_B\_STARTH\_REG (0x50003566)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA6_B_STARTH | Destination start address, upper 16 bits | 0x0   |

Table 740: DMA6\_INT\_REG (0x50003568)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA6_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 741: DMA6\_LEN\_REG (0x5000356A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA6_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 742: DMA6\_CTRL\_REG (0x5000356C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                              | 0x0   |

Table 742: DMA6\_CTRL\_REG (0x5000356C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 743: DMA6\_IDX\_REG (0x5000356E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA6_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 744: DMA7\_A\_STARTL\_REG (0x50003570)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA7_A_STARTL | Source start address, lower 16 bits | 0x0   |

Table 745: DMA7\_A\_STARTH\_REG (0x50003572)

| Bit  | Mode | Symbol        | Description                         | Reset |
|------|------|---------------|-------------------------------------|-------|
| 15:0 | R/W  | DMA7_A_STARTH | Source start address, upper 16 bits | 0x0   |

Table 746: DMA7\_B\_STARTL\_REG (0x50003574)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA7_B_STARTL | Destination start address, lower 16 bits | 0x0   |

Table 747: DMA7\_B\_STARTH\_REG (0x50003576)

| Bit  | Mode | Symbol        | Description                              | Reset |
|------|------|---------------|------------------------------------------|-------|
| 15:0 | R/W  | DMA7_B_STARTH | Destination start address, upper 16 bits | 0x0   |

Table 748: DMA7\_INT\_REG (0x50003578)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                   | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA7_INT | Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt. | 0x0   |

Table 749: DMA7\_LEN\_REG (0x5000357A)

| Bit  | Mode | Symbol   | Description                                                                                                          | Reset |
|------|------|----------|----------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R/W  | DMA7_LEN | DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ... | 0x0   |

Table 750: DMA7\_CTRL\_REG (0x5000357C)

| Bit   | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset |
|-------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:13 | -    | -        | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0x0   |
| 12    | R/W  | DMA_INIT | 0 = DMA performs copy A1 to B1, A2 to B2, etc ...<br>1 = DMA performs copy of A1 to B1, B2, etc ...<br>This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.                                                                                                                                                                                                    | 0x0   |
| 11    | R/W  | DMA_IDLE | 0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority.<br>1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.                                                                                                                                                                                                                                                                                     | 0x0   |
| 10:8  | R/W  | DMA_PRIO | The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific:<br>000 = lowest priority<br>111 = highest priority<br>If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus. | 0x0   |
| 7     | R/W  | CIRCULAR | 0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed.<br>1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.                                                                                                                                                                                                                                    | 0x0   |
| 6     | R/W  | AINC     | Enable increment of source address.<br>0 = do not increment (source address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00"; by 2, when BW="01"; by 4, when BW="10")                                                                                                                                                                                                                                                                                                                                                                | 0x0   |



Table 750: DMA7\_CTRL\_REG (0x5000357C)

| Bit | Mode | Symbol     | Description                                                                                                                                                                                                                                      | Reset |
|-----|------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R/W  | BINC       | Enable increment of destination address.<br>0 = do not increment (destination address stays the same during the transfer)<br>1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10") | 0x0   |
| 4   | R/W  | DREQ_MODE  | 0 = DMA channel starts immediately<br>1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)                                                                                                  | 0x0   |
| 3   | R/W  | IRQ_ENABLE | 0 = disables interrupts on this channel<br>1 = enables interrupts on this channel                                                                                                                                                                | 0x0   |
| 2:1 | R/W  | BW         | Bus transfer width:<br>00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI)<br>01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI)<br>10 = 4 Bytes (suggested for Memory-to-Memory transfers)<br>11 = Reserved           | 0x0   |
| 0   | R/W  | DMA_ON     | 0 = DMA channel is off, clocks are disabled<br>1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.                     | 0x0   |

Table 751: DMA7\_IDX\_REG (0x5000357E)

| Bit  | Mode | Symbol   | Description                                                                                                                                                                                                                                                                                                                             | Reset |
|------|------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:0 | R    | DMA7_IDX | This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0.<br>The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW. | 0x0   |

Table 752: DMA\_REQ\_MUX\_REG (0x50003580)

| Bit   | Mode | Symbol    | Description                                                                                                                                                                                                                                                    | Reset |
|-------|------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:12 | R/W  | DMA67_SEL | Select which combination of peripherals are mapped on the DMA channels. The peripherals are mapped as pairs on two channels.<br>Here, the first DMA request is mapped on channel 6 and the second on channel 7.<br>See DMA01_SEL for the peripherals' mapping. | 0x0   |
| 11:8  | R/W  | DMA45_SEL | Select which combination of peripherals are mapped on the DMA channels. The peripherals are mapped as pairs on two channels.<br>Here, the first DMA request is mapped on channel 4 and the second on channel 5.<br>See DMA01_SEL for the peripherals' mapping. | 0x0   |
| 7:4   | R/W  | DMA23_SEL | Select which combination of peripherals are mapped on the DMA channels. The peripherals are mapped as pairs on two channels.<br>Here, the first DMA request is mapped on channel 2 and the second on channel 3.<br>See DMA01_SEL for the peripherals' mapping. | 0x0   |



Table 752: DMA\_REQ\_MUX\_REG (0x50003580)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reset |
|-----|------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 3:0 | R/W  | DMA01_SEL | <p>Select which combination of peripherals are mapped on the DMA channels. The peripherals are mapped as pairs on two channels.</p> <p>Here, the first DMA request is mapped on channel 0 and the second on channel 1.</p> <p>0x0: SPI_rx / SPI_tx<br/> 0x1: SPI2_rx / SPI2_tx<br/> 0x2: UART_rx / UART_tx<br/> 0x3: UART2_rx / UART2_tx<br/> 0x4: I2C_rx / I2C_tx<br/> 0x5: I2C2_rx / I2C2_tx<br/> 0x6: USB_rx / USB_tx<br/> 0x7: Reserved<br/> 0x8: PCM_rx / PCM_tx<br/> 0x9: SRC_rx / SRC_tx (for all the supported conversions)<br/> 0xA: Reserved<br/> 0xB: Reserved<br/> 0xC: ADC / -<br/> 0xD: Reserved<br/> 0xE: Reserved<br/> 0xF: Reserved</p> <p>Note: If any of the four available peripheral selector fields (DMA01_SEL, DMA23_SEL, DMA45_SEL, DMA67_SEL) have the same value, the lesser significant selector has higher priority and will control the dma acknowledge. Hence, if DMA01_SEL = DMA23_SEL, the channels 0 and 1 will generate the DMA acknowledge signals for the selected peripheral. Consequently, it is suggested to assign the intended peripheral value to a unique selector field.</p> | 0x0   |

Table 753: DMA\_INT\_STATUS\_REG (0x50003582)

| Bit  | Mode | Symbol      | Description                                                    | Reset |
|------|------|-------------|----------------------------------------------------------------|-------|
| 15:8 | -    | -           | Reserved                                                       | 0x0   |
| 7    | R    | DMA_IRQ_CH7 | 0 = IRQ on channel 7 is not set<br>1 = IRQ on channel 7 is set | 0x0   |
| 6    | R    | DMA_IRQ_CH6 | 0 = IRQ on channel 6 is not set<br>1 = IRQ on channel 6 is set | 0x0   |
| 5    | R    | DMA_IRQ_CH5 | 0 = IRQ on channel 5 is not set<br>1 = IRQ on channel 5 is set | 0x0   |
| 4    | R    | DMA_IRQ_CH4 | 0 = IRQ on channel 4 is not set<br>1 = IRQ on channel 4 is set | 0x0   |
| 3    | R    | DMA_IRQ_CH3 | 0 = IRQ on channel 3 is not set<br>1 = IRQ on channel 3 is set | 0x0   |
| 2    | R    | DMA_IRQ_CH2 | 0 = IRQ on channel 2 is not set<br>1 = IRQ on channel 2 is set | 0x0   |
| 1    | R    | DMA_IRQ_CH1 | 0 = IRQ on channel 1 is not set<br>1 = IRQ on channel 1 is set | 0x0   |
| 0    | R    | DMA_IRQ_CH0 | 0 = IRQ on channel 0 is not set<br>1 = IRQ on channel 0 is set | 0x0   |

Table 754: DMA\_CLEAR\_INT\_REG (0x50003584)

| Bit  | Mode | Symbol          | Description                                                                    | Reset |
|------|------|-----------------|--------------------------------------------------------------------------------|-------|
| 15:8 | -    | -               | Reserved                                                                       | 0x0   |
| 7    | R0/W | DMA_RST_IRQ_CH7 | Writing a 1 will reset the IRQ of channel 7 ; writing a 0 will have no effect. | 0x0   |
| 6    | R0/W | DMA_RST_IRQ_CH6 | Writing a 1 will reset the IRQ of channel 6 ; writing a 0 will have no effect. | 0x0   |
| 5    | R0/W | DMA_RST_IRQ_CH5 | Writing a 1 will reset the IRQ of channel 5 ; writing a 0 will have no effect. | 0x0   |
| 4    | R0/W | DMA_RST_IRQ_CH4 | Writing a 1 will reset the IRQ of channel 4 ; writing a 0 will have no effect. | 0x0   |
| 3    | R0/W | DMA_RST_IRQ_CH3 | Writing a 1 will reset the IRQ of channel 3 ; writing a 0 will have no effect. | 0x0   |
| 2    | R0/W | DMA_RST_IRQ_CH2 | Writing a 1 will reset the IRQ of channel 2 ; writing a 0 will have no effect. | 0x0   |
| 1    | R0/W | DMA_RST_IRQ_CH1 | Writing a 1 will reset the IRQ of channel 1 ; writing a 0 will have no effect. | 0x0   |
| 0    | R0/W | DMA_RST_IRQ_CH0 | Writing a 1 will reset the IRQ of channel 0 ; writing a 0 will have no effect. | 0x0   |

## 37.27 APU REGISTER FILE

Table 755: Register map APU

| Address    | Port            | Description               |
|------------|-----------------|---------------------------|
| 0x50004000 | SRC1_CTRL_REG   | SRC1 control register     |
| 0x50004004 | SRC1_IN_FS_REG  | SRC1 Sample input rate    |
| 0x50004008 | SRC1_OUT_FS_REG | SRC1 Sample output rate   |
| 0x5000400C | SRC1_IN1_REG    | SRC1 data in 1            |
| 0x50004010 | SRC1_IN2_REG    | SRC1 data in 2            |
| 0x50004014 | SRC1_OUT1_REG   | SRC1 data out 1           |
| 0x50004018 | SRC1_OUT2_REG   | SRC1 data out 2           |
| 0x5000401C | APU_MUX_REG     | APU mux register          |
| 0x50004020 | COEF10_SET1_REG | SRC coefficient 1,0 set 1 |
| 0x50004024 | COEF32_SET1_REG | SRC coefficient 3,2 set 1 |
| 0x50004028 | COEF54_SET1_REG | SRC coefficient 5,4 set 1 |
| 0x5000402C | COEF76_SET1_REG | SRC coefficient 7,6 set 1 |
| 0x50004030 | COEF98_SET1_REG | SRC coefficient 9,8 set 1 |
| 0x50004034 | COEF0A_SET1_REG | SRC coefficient 10 set 1  |
| 0x50004100 | PCM1_CTRL_REG   | PCM1 Control register     |
| 0x50004104 | PCM1_IN1_REG    | PCM1 data in 1            |
| 0x50004108 | PCM1_IN2_REG    | PCM1 data in 2            |
| 0x5000410C | PCM1_OUT1_REG   | PCM1 data out 1           |
| 0x50004110 | PCM1_OUT2_REG   | PCM1 data out 2           |

Table 756: SRC1\_CTRL\_REG (0x50004000)

| Bit   | Mode | Symbol          | Description                                                                                                                                                                     | Reset |
|-------|------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:30 | -    | -               | Reserved                                                                                                                                                                        | 0     |
| 29:28 | R/W  | SRC_PDM_MODE    | PDM Output mode selection on PDM_DO1<br>00: No output<br>01: Right channel (falling edge of PDM_CLK)<br>10: Left channel (rising edge of PDM_CLK)<br>11: Left and Right channel | 0     |
| 27:26 | -    | -               | Reserved                                                                                                                                                                        | 0     |
| 25    | W    | SRC_OUT_FLOWCLR | Writing a 1 clears the SRC1_OUT Overflow/underflow bits 23-22. No more over/underflow indications while bit is 1. Keep 1 until the over/under flow bit is cleared               | 0     |
| 24    | W    | SRC_IN_FLOWCLR  | Writing a 1 clears the SRC1_IN Overflow/underflow bits 21-20. No more over/underflow indications while bit is 1. Keep 1 until the over/under flow bit is cleared                | 0     |
| 23    | R    | SRC_OUT_UNFLOW  | 1 = SRC1_OUT Underflow occurred                                                                                                                                                 | 0     |
| 22    | R    | SRC_OUT_OVFLOW  | 1 = SRC1_OUT Overflow occurred                                                                                                                                                  | 0     |
| 21    | R    | SRC_IN_UNFLOW   | 1 = SRC1_IN Underflow occurred                                                                                                                                                  | 0     |
| 20    | R    | SRC_IN_OVFLOW   | 1 = SRC1_IN Overflow occurred                                                                                                                                                   | 0     |
| 19    | -    | -               | Reserved                                                                                                                                                                        | 0     |
| 18    | R    | SRC_OUT_OK      | SRC1_OUT Status<br>0: acquisition in progress<br>1: acquisition ready (In manual mode this bit is always 1)                                                                     | 0     |

Table 756: SRC1\_CTRL\_REG (0x50004000)

| Bit   | Mode | Symbol                 | Description                                                                                                                                                    | Reset |
|-------|------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 17:16 | R/W  | SRC_OUT_US             | SRC1_OUT UpSampling IIR filters setting<br>00: for sample rates up-to 48kHz<br>01: for sample rates of 96kHz<br>10: reserved<br>11: for sample rates of 192kHz | 0     |
| 15    | -    | -                      | Reserved                                                                                                                                                       | 0     |
| 14    | R/W  | SRC_OUT_CAL_BY<br>PASS | SRC1_OUT1 upsampling filter bypass<br>0: Do not bypass<br>1: Bypass filter                                                                                     | 0     |
| 13    | R/W  | SRC_OUT_AMODE          | SRC1_OUT1 Automatic Conversion mode<br>0: Manual mode<br>1: Automatic mode                                                                                     | 0     |
| 12:10 | -    | -                      | Reserved                                                                                                                                                       | 0     |
| 9:8   | -    | -                      | Reserved                                                                                                                                                       | 0     |
| 7     | R/W  | SRC_DITHER_DISA<br>BLE | Dithering feature<br>0: Enable<br>1: Disable                                                                                                                   | 0     |
| 6     | R    | SRC_IN_OK              | SRC1_IN status<br>0: Acquisition in progress<br>1: Acquisition ready                                                                                           | 0     |
| 5:4   | R/W  | SRC_IN_DS              | SRC1_IN UpSampling IIR filters setting<br>00: for sample rates up-to 48kHz<br>01: for sample rates of 96kHz<br>10: reserved<br>11: for sample rates of 192kHz  | 0     |
| 3     | -    | -                      | Reserved                                                                                                                                                       | 0     |
| 2     | R/W  | SRC_IN_CAL_BYPA<br>SS  | SRC1_IN upsampling filter bypass<br>0: Do not bypass<br>1: Bypass filter                                                                                       | 0     |
| 1     | R/W  | SRC_IN_AMODE           | SRC1_IN Automatic conversion mode<br>0: Manual mode<br>1: Automatic mode                                                                                       | 0     |
| 0     | R/W  | SRC_EN                 | SRC1_IN and SRC1_OUT enable<br>0: disabled<br>1: enabled                                                                                                       | 0     |

Table 757: SRC1\_IN\_FS\_REG (0x50004004)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 31:24 | -    | -      | Reserved    | 0     |

Table 757: SRC1\_IN\_FS\_REG (0x50004004)

| Bit         | Mode      | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Reset       |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
|-------------|-----------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-----------|-----------|----------|---------|---------|---|---------|----------|----------|---|---------|----------|----------|---|---------|----------|----------|---|----------|----------|----------|---|----------|----------|----------|---|----------|----------|----------|---|----------|----------|----------|---|----------|-----------|----------|---|----------|---|
| 23:0        | R/W       | SRC_IN_FS | <p>SRC_IN Sample rate<br/>SRC_IN_FS = 8192*Sample_rate/100<br/>Sample_rate upper limit is 192 kHz. For 96 kHz and 192 kHz SRC_CTRLx_REG[SRC_IN_DS] must be set as shown below:</p> <table><tr><th>Sample_rate</th><th>SRC_IN_FS</th><th>SRC_IN_DS</th><th>Audio BW</th></tr><tr><td>8000 Hz</td><td>0xA0000</td><td>0</td><td>4000 Hz</td></tr><tr><td>11025 Hz</td><td>0x0DC800</td><td>0</td><td>5512 Hz</td></tr><tr><td>16000 Hz</td><td>0x140000</td><td>0</td><td>8000 Hz</td></tr><tr><td>22050 Hz</td><td>0x1B9000</td><td>0</td><td>11025 Hz</td></tr><tr><td>32000 Hz</td><td>0x280000</td><td>0</td><td>16000 Hz</td></tr><tr><td>44100 Hz</td><td>0x372000</td><td>0</td><td>22050 Hz</td></tr><tr><td>48000 Hz</td><td>0x3C0000</td><td>0</td><td>24000 Hz</td></tr><tr><td>96000 Hz</td><td>0x3C0000</td><td>1</td><td>24000 Hz</td></tr><tr><td>192000 Hz</td><td>0x3C0000</td><td>3</td><td>24000 Hz</td></tr></table> <p>In manual SRC mode, SRC_IN_FS can be set and adjusted to the desired sample rate at any time.<br/>In automatic mode the SRC returns the final sample rate as soon as SRC_IN_OK. Note that SRC_DS is not calculated in automatic mode and must be set manually automatic mode with Sample_rate of 96 kHz and 192 kHz.</p> | Sample_rate | SRC_IN_FS | SRC_IN_DS | Audio BW | 8000 Hz | 0xA0000 | 0 | 4000 Hz | 11025 Hz | 0x0DC800 | 0 | 5512 Hz | 16000 Hz | 0x140000 | 0 | 8000 Hz | 22050 Hz | 0x1B9000 | 0 | 11025 Hz | 32000 Hz | 0x280000 | 0 | 16000 Hz | 44100 Hz | 0x372000 | 0 | 22050 Hz | 48000 Hz | 0x3C0000 | 0 | 24000 Hz | 96000 Hz | 0x3C0000 | 1 | 24000 Hz | 192000 Hz | 0x3C0000 | 3 | 24000 Hz | 0 |
| Sample_rate | SRC_IN_FS | SRC_IN_DS | Audio BW                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 8000 Hz     | 0xA0000   | 0         | 4000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 11025 Hz    | 0x0DC800  | 0         | 5512 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 16000 Hz    | 0x140000  | 0         | 8000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 22050 Hz    | 0x1B9000  | 0         | 11025 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 32000 Hz    | 0x280000  | 0         | 16000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 44100 Hz    | 0x372000  | 0         | 22050 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 48000 Hz    | 0x3C0000  | 0         | 24000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 96000 Hz    | 0x3C0000  | 1         | 24000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 192000 Hz   | 0x3C0000  | 3         | 24000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |           |           |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |

Table 758: SRC1\_OUT\_FS\_REG (0x50004008)

| Bit         | Mode       | Symbol     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Reset       |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
|-------------|------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|------------|------------|----------|---------|---------|---|---------|----------|----------|---|---------|----------|----------|---|---------|----------|----------|---|----------|----------|----------|---|----------|----------|----------|---|----------|----------|----------|---|----------|----------|----------|---|----------|-----------|----------|---|----------|---|
| 31:24       | -          | -          | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 0           |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 23:0        | R/W        | SRC_OUT_FS | <p>SRC_OUT Sample rate<br/>SRC_OUT_FS = 8192*Sample_rate/100<br/>Sample_rate upper limit is 192 kHz. For 96 kHz and 192 kHz SRC_CTRLx_REG[SRC_DS] must be set as shown below:</p> <table><thead><tr><th>Sample_rate</th><th>SRC_OUT_FS</th><th>SRC_OUT_DS</th><th>Audio BW</th></tr></thead><tbody><tr><td>8000 Hz</td><td>0xA0000</td><td>0</td><td>4000 Hz</td></tr><tr><td>11025 Hz</td><td>0x0DC800</td><td>0</td><td>5512 Hz</td></tr><tr><td>16000 Hz</td><td>0x140000</td><td>0</td><td>8000 Hz</td></tr><tr><td>22050 Hz</td><td>0x1B9000</td><td>0</td><td>11025 Hz</td></tr><tr><td>32000 Hz</td><td>0x280000</td><td>0</td><td>16000 Hz</td></tr><tr><td>44100 Hz</td><td>0x372000</td><td>0</td><td>22050 Hz</td></tr><tr><td>48000 Hz</td><td>0x3C0000</td><td>0</td><td>24000 Hz</td></tr><tr><td>96000 Hz</td><td>0x3C0000</td><td>1</td><td>24000 Hz</td></tr><tr><td>192000 Hz</td><td>0x3C0000</td><td>3</td><td>24000 Hz</td></tr></tbody></table> <p>In manual SRC mode, SRC_OUT_FS can be set and adjusted to the desired sample rate at any time.<br/>In automatic mode the SRC returns the final sample rate as soon as SRC_OUT_OK. Note that SRC_DS is not calculated in automatic mode and must be set manually automatic mode with Sample_rate of 96 kHz and 192 kHz.</p> | Sample_rate | SRC_OUT_FS | SRC_OUT_DS | Audio BW | 8000 Hz | 0xA0000 | 0 | 4000 Hz | 11025 Hz | 0x0DC800 | 0 | 5512 Hz | 16000 Hz | 0x140000 | 0 | 8000 Hz | 22050 Hz | 0x1B9000 | 0 | 11025 Hz | 32000 Hz | 0x280000 | 0 | 16000 Hz | 44100 Hz | 0x372000 | 0 | 22050 Hz | 48000 Hz | 0x3C0000 | 0 | 24000 Hz | 96000 Hz | 0x3C0000 | 1 | 24000 Hz | 192000 Hz | 0x3C0000 | 3 | 24000 Hz | 0 |
| Sample_rate | SRC_OUT_FS | SRC_OUT_DS | Audio BW                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 8000 Hz     | 0xA0000    | 0          | 4000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 11025 Hz    | 0x0DC800   | 0          | 5512 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 16000 Hz    | 0x140000   | 0          | 8000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 22050 Hz    | 0x1B9000   | 0          | 11025 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 32000 Hz    | 0x280000   | 0          | 16000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 44100 Hz    | 0x372000   | 0          | 22050 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 48000 Hz    | 0x3C0000   | 0          | 24000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 96000 Hz    | 0x3C0000   | 1          | 24000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |
| 192000 Hz   | 0x3C0000   | 3          | 24000 Hz                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |            |            |          |         |         |   |         |          |          |   |         |          |          |   |         |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |          |          |   |          |           |          |   |          |   |

Table 759: SRC1\_IN1\_REG (0x5000400C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 31:8 | R/W  | SRC_IN | SRC1_IN1    | 0     |

Table 760: SRC1\_IN2\_REG (0x50004010)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 31:8 | R/W  | SRC_IN | SRC1_IN2    | 0     |

Table 761: SRC1\_OUT1\_REG (0x50004014)

| Bit  | Mode | Symbol  | Description | Reset |
|------|------|---------|-------------|-------|
| 31:8 | R    | SRC_OUT | SRC1_OUT1   | 0     |

Table 762: SRC1\_OUT2\_REG (0x50004018)

| Bit  | Mode | Symbol  | Description | Reset |
|------|------|---------|-------------|-------|
| 31:8 | R    | SRC_OUT | SRC1_OUT2   | 0     |

Table 763: APU\_MUX\_REG (0x5000401C)

| Bit | Mode | Symbol      | Description                                                              | Reset |
|-----|------|-------------|--------------------------------------------------------------------------|-------|
| 6   | R/W  | PDM1_MUX_IN | PDM1 input mux<br>0 = SRC1_MUX_IN<br>1 = PDM input                       | 0x0   |
| 5:3 | R/W  | PCM1_MUX_IN | PCM1 input mux<br>0 = off<br>1 = SRC1 output<br>2 = PCM output registers | 0x0   |
| 2:0 | R/W  | SRC1_MUX_IN | SRC1 input mux<br>0 = off<br>1 = PCM output<br>2 = SRC1 input registers  | 0x0   |

Table 764: COEF10\_SET1\_REG (0x50004020)

| Bit   | Mode | Symbol    | Description   | Reset  |
|-------|------|-----------|---------------|--------|
| 31:16 | R/W  | SRC_COEF1 | coefficient 1 | 0x79A9 |
| 15:0  | R/W  | SRC_COEF0 | coefficient 0 | 0x9278 |

Table 765: COEF32\_SET1\_REG (0x50004024)

| Bit   | Mode | Symbol    | Description   | Reset  |
|-------|------|-----------|---------------|--------|
| 31:16 | R/W  | SRC_COEF3 | coefficient 3 | 0x6D56 |
| 15:0  | R/W  | SRC_COEF2 | coefficient 2 | 0x8B41 |

Table 766: COEF54\_SET1\_REG (0x50004028)

| Bit   | Mode | Symbol    | Description   | Reset  |
|-------|------|-----------|---------------|--------|
| 31:16 | R/W  | SRC_COEF5 | coefficient 5 | 0x9BC5 |
| 15:0  | R/W  | SRC_COEF4 | coefficient 4 | 0xBE15 |

Table 767: COEF76\_SET1\_REG (0x5000402C)

| Bit   | Mode | Symbol    | Description   | Reset  |
|-------|------|-----------|---------------|--------|
| 31:16 | R/W  | SRC_COEF7 | coefficient 7 | 0x8C28 |
| 15:0  | R/W  | SRC_COEF6 | coefficient 6 | 0x7E1A |

Table 768: COEF98\_SET1\_REG (0x50004030)

| Bit   | Mode | Symbol    | Description   | Reset  |
|-------|------|-----------|---------------|--------|
| 31:16 | R/W  | SRC_COEF9 | coefficient 9 | 0x92D7 |
| 15:0  | R/W  | SRC_COEF8 | coefficient 8 | 0x75E6 |

Table 769: COEF0A\_SET1\_REG (0x50004034)

| Bit  | Mode | Symbol     | Description    | Reset  |
|------|------|------------|----------------|--------|
| 15:0 | R/W  | SRC_COEF10 | coefficient 10 | 0x41F2 |

Table 770: PCM1\_CTRL\_REG (0x50004100)

| Bit   | Mode | Symbol       | Description                                                                                                                                                          | Reset |
|-------|------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:20 | R/W  | PCM_FSC_DIV  | PCM Framesync divider, Values 7-0xFFFF. To divide by N, write N-1. (Minimum value N-1=7 for 8 bits PCM_FSC)<br>Note if PCM_CLK_BIT=1, N must always be even          | 0x0   |
| 19:17 | -    | -            | Reserved                                                                                                                                                             | 0x0   |
| 16    | R/W  | PCM_FSC_EDGE | 0: shift channels 1, 2, 3, 4, 5, 6, 7, 8 after PCM_FSC edge<br>1: shift channels 1, 2, 3, 4 after PCM_FSC edge shift channels 5, 6, 7, 8 after opposite PCM_FSC edge | 0x0   |
| 15:11 | R/W  | PCM_CH_DEL   | Channel delay in multiples of 8 bits                                                                                                                                 | 0x0   |
| 10    | R/W  | PCM_CLK_BIT  | 0:One clock cycle per data bit<br>1:Two clock cycles per data bit                                                                                                    | 0x0   |
| 9     | R/W  | PCM_FSCINV   | 0: PCM FSC<br>1: PCM FSC inverted                                                                                                                                    | 0x0   |
| 8     | R/W  | PCM_CLKINV   | 0:PCM CLK<br>1:PCM CLK inverted                                                                                                                                      | 0x0   |
| 7     | R/W  | PCM_PPOD     | 0:PCM DO push pull<br>1:PCM DO open drain                                                                                                                            | 0x0   |
| 6     | R/W  | PCM_FSCDEL   | 0:PCM FSC starts one cycle before MSB bit<br>1:PCM FSC starts at the same time as MSB bit                                                                            | 0x0   |
| 5:2   | R/W  | PCM_FSCLLEN  | 0:PCM FSC length equal to 1 data bit<br>N:PCM FSC length equal to N*8                                                                                                | 0x0   |
| 1     | R/W  | PCM_MASTER   | 0:PCM interface in slave mode<br>1:PCM interface in master mode                                                                                                      | 0x0   |
| 0     | R/W  | PCM_EN       | 0:PCM interface disabled<br>1:PCM interface enabled                                                                                                                  | 0x0   |

Table 771: PCM1\_IN1\_REG (0x50004104)

| Bit  | Mode | Symbol | Description        | Reset             |
|------|------|--------|--------------------|-------------------|
| 31:0 | R    | PCM_IN | PCM1_IN1 bits 31-0 | 0xFFFFFFFF<br>FFF |

Table 772: PCM1\_IN2\_REG (0x50004108)

| Bit  | Mode | Symbol | Description        | Reset             |
|------|------|--------|--------------------|-------------------|
| 31:0 | R    | PCM_IN | PCM1_IN2 bits 31-0 | 0xFFFFFFFF<br>FFF |

Table 773: PCM1\_OUT1\_REG (0x5000410C)

| Bit  | Mode | Symbol  | Description         | Reset             |
|------|------|---------|---------------------|-------------------|
| 31:0 | R/W  | PCM_OUT | PCM1_OUT1 bits 31-0 | 0xFFFFFFFF<br>FFF |

Table 774: PCM1\_OUT2\_REG (0x50004110)

| Bit  | Mode | Symbol  | Description         | Reset             |
|------|------|---------|---------------------|-------------------|
| 31:0 | R/W  | PCM_OUT | PCM1_OUT2 bits 31-0 | 0xFFFFFFFF<br>FFF |



## 37.28 TRNG REGISTER FILE

Table 775: Register map TRNG

| Address    | Port             | Description              |
|------------|------------------|--------------------------|
| 0x50005000 | TRNG_CTRL_REG    | TRNG control register    |
| 0x50005004 | TRNG_FIFOLVL_REG | TRNG FIFO level register |
| 0x50005008 | TRNG_VER_REG     | TRNG Version register    |

Table 776: TRNG\_CTRL\_REG (0x50005000)

| Bit  | Mode | Symbol      | Description                                                                                                                                                                 | Reset |
|------|------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:2 | -    | -           | Reserved                                                                                                                                                                    | 0x0   |
| 1    | R/W  | TRNG_MODE   | 0: select the TRNG with asynchronous free running oscillators (default)<br>1: select the pseudo-random generator with synchronous oscillators (for simulation purpose only) | 0x0   |
| 0    | R/W  | TRNG_ENABLE | 0: Disable the TRNG<br>1: Enable the TRNG this signal is ignored when the FIFO is full                                                                                      | 0x0   |

Table 777: TRNG\_FIFOLVL\_REG (0x50005004)

| Bit  | Mode | Symbol        | Description                                                                                                                                        | Reset |
|------|------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31:6 | -    | -             | Reserved                                                                                                                                           | 0x0   |
| 5    | R    | TRNG_FIFOFULL | 1:FIFO full indication. This bit is cleared if the FIFO is read.                                                                                   | 0x0   |
| 4:0  | R    | TRNG_FIFOLVL  | Number of 32 bit words of random data in the FIFO (max 31) until the FIFO is full. When it is 0 and TRNG_FIFOFULL is 1, it means the FIFO is full. | 0x0   |

Table 778: TRNG\_VER\_REG (0x50005008)

| Bit   | Mode | Symbol   | Description          | Reset |
|-------|------|----------|----------------------|-------|
| 31:24 | R    | TRNG_MAJ | Major version number | 0x0   |
| 23:16 | R    | TRNG_MIN | Minor version number | 0x0   |
| 15:0  | R    | TRNG_SVN | SVN revision number  | 0x103 |

## 37.29 ELLIPTIC CURVE CONTROLLER REGISTER FILE

Table 779: Register map Elliptic Curve Controller

| Address    | Port            | Description            |
|------------|-----------------|------------------------|
| 0x50006000 | ECC_CONFIG_REG  | Configuration register |
| 0x50006004 | ECC_COMMAND_REG | Command register       |
| 0x50006008 | ECC_CONTROL_REG | Control register       |
| 0x5000600C | ECC_STATUS_REG  | Status register        |
| 0x50006010 | ECC_VERSION_REG | Version register       |

Table 780: ECC\_CONFIG\_REG (0x50006000)

| Bit   | Mode | Symbol     | Description                                                                                                                  | Reset |
|-------|------|------------|------------------------------------------------------------------------------------------------------------------------------|-------|
| 20:16 | R/W  | ECC_OPPTRC | When executing primitive arithmetic operations, this pointer defines the location where the result will be stored in Memory. | 0x0   |
| 15:13 | -    | -          | Reserved                                                                                                                     | 0x0   |
| 12:8  | R/W  | ECC_OPPTRB | When executing primitive arithmetic operations, this Pointer defines where operand B is located in memory.                   | 0x0   |
| 7:5   | -    | -          | Reserved                                                                                                                     | 0x0   |
| 4:0   | R/W  | ECC_OPPTRA | When executing primitive arithmetic operations, this Pointer defines where operand A is located in memory.                   | 0x0   |

Table 781: ECC\_COMMAND\_REG (0x50006004)

| Bit   | Mode | Symbol             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reset |
|-------|------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 31    | R/W  | ECC_CALCR2         | This bit indicates if the IP has to calculate $R \bmod N$ for the next operation. This bit must be set to 1 when a new prime number has been programmed. This bit is automatically cleared when $R \bmod N$ has been calculated.<br>'0': no effect<br>'1': forces the IP to re-calculate $R \bmod N$                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x0   |
| 30    | R/W  | ECC_SIGNB          | Sign of parameter B in equation $y^2 = x^3 + Ax + B$<br>'0': B is positive<br>'1': B is negative                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 29    | R/W  | ECC_SIGNA          | Sign of parameter A in equation $y^2 = x^3 + Ax + B$<br>'0': A is positive<br>'1': A is negative                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0x0   |
| 28:16 | -    | -                  | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 0x0   |
| 15:8  | R/W  | ECC_SIZEOFOPERANDS | This field defines the size (= number of 64-bit double words) of the operands for the current operation. Possible values are limited by the generic parameter g_Log2MaxDataSize that defines the max space allocated or reserved to each operand.<br>Arbitrary Data/Key size from 128 up to 2566 are supported:<br>0x02 (02d) -> 128-bit Data/Key size<br>0x03 (02d) -> 256-bit Data/Key size<br><b>ECC-ECDSA - Prime Field <math>F(p)</math></b><br>0x03 -> 192-bit (Curve P-192)<br>0x04 -> 256-bit (Curves P-224 & P-256)<br><b>ECC-ECDSA - Binary Field <math>F(2^m)</math></b><br>0x03 -> 192-bit (Curve K-163)<br>0x04 -> 256-bit (Curve K-233)<br>- 4 Xers: 0x01, 0x02, 0x4, 0x6 -> 64, 128 & multiples of 128 bits | 0x0   |

Table 781: ECC\_COMMAND\_REG (0x50006004)

| Bit | Mode | Symbol            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset |
|-----|------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 7   | R/W  | ECC_FIELD         | '0': Field is F(p)<br>'1': Field is F(2m)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x0   |
| 6:0 | R/W  | ECC_TYPEOPERATION | <b>Primitive Arithmetic Operations F(p) &amp; F(2m)</b><br>[6:4] = 0x0<br>[3:0] =<br>0x0 -> Reserved<br>0x1 -> Modular Addition<br>0x2 -> Modular Subtraction<br>0x3 -> Modular Multiplication (Odd N)<br>0x4 -> Modular Reduction (Odd N)<br>0x5 -> Modular Division (Odd N)<br>0x6 -> Modular Inversion (Odd N)<br>0x7 -> Reserved<br>0x8 -> Multiplication<br>0x9 -> Modular Inversion (Even N)<br>0xA -> Modular Reduction (Even N)<br>others -> Reserved<br>$C = A + B \bmod N$<br>$C = A - B \bmod N$<br>$C = A * B \bmod N$<br>$C = B \bmod N$<br>$C = A/B \bmod N$<br>$C = 1/B \bmod N$<br>$C = A * B$<br>$C = 1/B \bmod N$<br>$C = B \bmod N$<br>High-level RSA, CRT & DSA Operations - F(p) only<br>([7] forced to 0)<br>[6:4] = 0x1<br>[3:0] =<br>0x0 -> MulModN<br>0x1 -> MulAddN<br>0x2 -> ECMQV (part1)<br>others -> Reserved<br><b>Primitive ECC &amp; Check Point Operations F(p) &amp; F(2m)</b><br>[6:4] = 0x2<br>[3:0] =<br>0x0 -> Point Doubling (Projective Coord.)<br>0x1 -> ptAdd3<br>0x2 -> GenSessionKey<br>0x3 -> Check_AB (ECDSA)<br>0x4 -> Check_n (ECDSA)<br>0x5 -> Check single value less than N<br>0x6 -> Check_Point_On_Curve<br>0x7 -> Reserved<br>0x8 -> Curve25519 point multiplication<br>0x9 -> Ed25519 Check point on curve<br>0xA -> Ed25519 ScalarMult<br>0xB -> Ed25519 CheckValid<br>others -> Reserved<br>(continued on next page) | 0x0   |

Table 781: ECC\_COMMAND\_REG (0x50006004)

| Bit | Mode | Symbol                           | Description                                                                                                                                                                                                                                | Reset |
|-----|------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 6:0 | R/W  | ECC_TYPEOPERATION<br>(continued) | <b>High-level ECC ECDSA Operations F(p) &amp; F(2m)</b><br>[6:4] = 0x3<br>[3:0] =<br>0x0 -> ECMQV (part 2)<br>0x1 -> Verify ZKP<br>0x2 -> ECDSA Domain Parameters Validation<br>others -> Reserved<br>[6:4]=0x4, 0x5, 0x6, 0x7 -> Reserved | 0x0   |

Table 782: ECC\_CONTROL\_REG (0x50006008)

| Bit | Mode | Symbol    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reset |
|-----|------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 0   | R/W  | ECC_START | The Start signal is activated when all data and key inputs have been loaded in the external crypto memory and are available for processing. This signal is active high and is sampled on the rising edge of Clk.<br>When this signal goes high, the PK Command present in the PK_CommandReg[] is initiated and executed. The PK_Start signal is ignored when the core is already processing data and is automatically cleared when the operation is finished | 0x0   |

Table 783: ECC\_STATUS\_REG (0x5000600C)

| Bit   | Mode | Symbol                  | Description                                                                                                                                                                                                                                              | Reset |
|-------|------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| 16    | R    | ECC_BUSY                | This Status Signal indicates that the core is processing data. This signal is active high and goes low when the selected algorithm is finished.                                                                                                          | 0x0   |
| 15:13 | -    | -                       | Reserved                                                                                                                                                                                                                                                 | 0x0   |
| 12    | R    | ECC_PRIMALITYTESTRESULT | After the Miller-Rabin Primality test, this flag is:<br>- set to 0 when the random number under test is probably prime<br>- cleared to 1 when the random number under test is composite                                                                  | 0x0   |
| 11    | R    | ECC_NOTINVERTIBLE       | This flag is set to 1 when executing a modular inversion (PK_CommandReg[3:0] = 0x6 or 0x9) if the operand is not invertible.                                                                                                                             | 0x0   |
| 10    | R    | ECC_PARAM_AB_NOTVALID   | Status signal set to 1 when parameters A and B are not valid, i.e. $4A + 27B = 0$ . This flag is updated after execution of the command Check_AB.                                                                                                        | 0x0   |
| 9     | R    | ECC_SIGNATURE_NOTVALID  | This flag indicates if the signature can be accepted or must be rejected.<br>This flag is set to 1 when the signature is not valid and is updated after execution of the command ECDSA_Generation, ECDSA_Verification, DSA_Generation, DSA_Verification. | 0x0   |
| 8     | -    | -                       | Reserved                                                                                                                                                                                                                                                 | 0x0   |
| 7     | R    | ECC_PARAM_N_NOTVALID    | Status signal set to 1 when Parameter n is not valid. This flag is updated after execution of the command Check_n.                                                                                                                                       | 0x0   |
| 6     | R    | ECC_COUPLE_NOTVALID     | Status signal set to 1 when couple x, y is not valid (i.e. not smaller than the prime). This flag is updated after execution of the command Check_Couple_Less_Prime.                                                                                     | 0x0   |

Table 783: ECC\_STATUS\_REG (0x5000600C)

| Bit | Mode | Symbol                    | Description                                                                                                                             | Reset |
|-----|------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-------|
| 5   | R    | ECC_POINT_PX_AT_INFINITY  | Status signal set to 1 when Point Px is at the infinity. This flag is updated after execution of an ECC operation.                      | 0x0   |
| 4   | R    | ECC_POINT_PX_NOT_ON_CURVE | Status signal set to 1 when Point Px is not on the defined EC. This flag is updated after execution of the command Check_Point_OnCurve. | 0x0   |
| 3:0 | R    | ECC_FAIL_ADDRESSES        | Address of the last Point detected as Not On Curve, Not Valid or at the infinity.                                                       | 0x0   |

Table 784: ECC\_VERSION\_REG (0x50006010)

| Bit  | Mode | Symbol  | Description                                                                                                                           | Reset |
|------|------|---------|---------------------------------------------------------------------------------------------------------------------------------------|-------|
| 15:8 | R    | ECC_HVN | Version of IP to be read via CPU interface.                                                                                           | 0x4   |
| 7:0  | R    | ECC_SVN | Version of Crypto code to be read via CPU interface. Note that this should be read before ECC is used since it corrupts its contents. | 0x0   |

## 38 Specifications

All MIN/MAX specification limits are guaranteed by design, production testing and/or statistical characterization, and are valid over the full operating temperature range and power supply range, unless otherwise noted. Typical values are based on characterization results at default measurement conditions and are informative only.

Default measurement conditions (unless otherwise specified):  $V_{BAT1} = V_{BAT2} = 3.0\text{ V}$ ,  $T_A = 25\text{ }^{\circ}\text{C}$ . All radio measurements are performed with standard RF measurement equipment providing a source/load impedance of  $50\text{ }\Omega$ .

Table 785: Absolute maximum ratings

| Parameter                  | Description                                            | Conditions                          | Min  | Typ | Max  | Unit |
|----------------------------|--------------------------------------------------------|-------------------------------------|------|-----|------|------|
| V <sub>PIN_LIM_DEF</sub>   | limiting voltage on a pin                              | default, unless otherwise specified | -0.1 |     | 3.6  | V    |
| V <sub>BAT_LIM</sub>       | limiting battery supply voltage                        | pin VBAT                            | 0    |     | 6    | V    |
| V <sub>BUS_LIM</sub>       | limiting bus supply voltage                            | pin VBUS                            | 0    |     | 6.5  | V    |
| t <sub>R_SUP</sub>         | power supply rise time                                 |                                     |      |     | 30   | ms   |
| V <sub>PIN_LIM_3V3</sub>   | limiting voltage on a pin                              | 3.3 V I/O pins                      | 0    |     | 3.45 | V    |
| V <sub>PIN_LIM_1V8</sub>   | limiting voltage on a pin                              | 1.8 V I/O pins                      | 0    |     | 1.98 | V    |
| V <sub>ESD_HBM_QFN60</sub> | electrostatic discharge voltage (Human Body Model)     | QFN60 package                       |      |     | 2000 | V    |
| V <sub>ESD_CDM_QFN60</sub> | electrostatic discharge voltage (Charged Device Model) | QFN60 package                       |      |     | 500  | V    |
| T <sub>STG</sub>           | storage temperature                                    |                                     | -50  |     | 150  | °C   |

Table 786: Recommended operating conditions

| Parameter            | Description             | Conditions                                                                                        | Min  | Typ | Max  | Unit |
|----------------------|-------------------------|---------------------------------------------------------------------------------------------------|------|-----|------|------|
| V <sub>BAT</sub>     | battery supply voltage  | pin VBAT1 and VBAT2                                                                               | 1.7  |     | 4.75 | V    |
| V <sub>BAT_OTP</sub> | V33 rail supply voltage | Voltage range for OTP programming. Required temperature for programming is between -20°C and 85°C | 2.25 |     | 3.6  | V    |
| V <sub>BUS</sub>     | bus supply voltage      | pin VBUS                                                                                          | 4.2  |     | 5.75 | V    |
| V <sub>PIN_3V3</sub> | voltage on a pin        | 3.3 V I/O pins                                                                                    | 0    |     | 3.3  | V    |
| V <sub>PIN_1V8</sub> | voltage on a pin        | 1.8 V I/O pins                                                                                    | 0    |     | 1.8  | V    |
| T <sub>A</sub>       | ambient temperature     |                                                                                                   | -40  |     | 85   | °C   |

Table 787: DC characteristics

| Parameter                  | Description            | Conditions                                                                                                                           | Min | Typ | Max | Unit |
|----------------------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|------|
| I <sub>BAT_IDLE</sub>      | battery supply current | CPU is idle (Wait for Interrupt - WFI); sys_clk = 16 MHz; pclk = 2 MHz; DC-DC on; FLASH off; peripherals on; V <sub>BAT</sub> = 3 V. |     | 0.6 |     | mA   |
| I <sub>BAT_RUN_16MHz</sub> | battery supply current | CPU is executing code from 32 kB RAM; sys_clk = 16 MHz; pclk=2 MHz; DC-DC on; FLASH off; peripherals on; V <sub>BAT</sub> = 3 V.     |     | 1.4 |     | mA   |

Table 787: DC characteristics

| Parameter                           | Description            | Conditions                                                                                                                                                             | Min | Typ | Max | Unit |
|-------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|------|
| I <sub>BAT_RUN_48MHz</sub>          | battery supply current | CPU is executing code from 32 kB RAM; sys_clk = 48 MHz; pclk = 2 MHz; DC-DC on; FLASH off; PLL on; peripherals on; V <sub>BAT</sub> = 3 V.                             |     | 3.9 |     | mA   |
| I <sub>BAT_RUN_96MHz</sub>          | battery supply current | CPU is executing code from 32 kB RAM; sys_clk = 96 MHz; pclk = 2 MHz; DC-DC on; FLASH off; PLL on; peripherals on; V <sub>BAT</sub> = 3 V.                             |     | 6.2 |     | mA   |
| I <sub>BAT_HIBERN</sub>             | battery supply current | Hibernation mode; no RAM retained; RCX clock on; VBAT_RET on; DC-DC off; V <sub>DD_RET</sub> = 0.9 V; FLASH off; V <sub>BAT</sub> = 3 V.                               |     | 1.3 |     | μA   |
| I <sub>BAT_DP_SLP_8K</sub>          | battery supply current | Deep Sleep mode; 8 kB RAM retained; RCX clock on; VBAT_RET on; DC-DC off; V <sub>DD_RET</sub> = 0.9 V; FLASH off; V <sub>BAT</sub> = 3 V.                              |     | 1.5 |     | μA   |
| I <sub>BAT_DP_SLP_24K</sub>         | battery supply current | Deep Sleep mode; 24 kB RAM retained; aRCX clock on; VBAT_RET on; DC-DC off; V <sub>DD_RET</sub> = 0.9 V; FLASH off; V <sub>BAT</sub> = 3 V.                            |     | 1.9 |     | μA   |
| I <sub>BAT_EX_SLP_16K_32K_FP</sub>  | battery supply current | Extended Sleep mode; 16 kB (code) and 32 kB (data) RAM retained; XTAL32K on; DC-DC off; V <sub>DD_RET</sub> = 0.9 V; FLASH in Power Down mode; V <sub>BAT</sub> = 3 V. |     | 2.6 |     | μA   |
| I <sub>BAT_EX_SLP_16K_128K_FP</sub> | battery supply current | Extended Sleep mode; 16 kB cache and 128 kB (data) RAM retained; XTAL32K on; DC-DC off; V <sub>DD_RET</sub> = 0.9 V; FLASH in Power Down mode; V <sub>BAT</sub> = 3 V. |     | 4.6 |     | μA   |
| I <sub>BAT_BLE_RX_16M_FS</sub>      | battery supply current | BLE receive mode; f <sub>CLK</sub> = 16 MHz; DC-DC on; FLASH in Standby mode; V <sub>BAT</sub> = 3 V.                                                                  |     | 5.8 |     | mA   |
| I <sub>BAT_BLE_TX_16M_FS</sub>      | battery supply current | BLE transmit mode; f <sub>CLK</sub> = 16 MHz; DC-DC on; FLASH in Standby mode; V <sub>BAT</sub> = 3 V.                                                                 |     | 5.9 |     | mA   |



Table 787: DC characteristics

| Parameter                         | Description            | Conditions                                                                               | Min | Typ | Max | Unit |
|-----------------------------------|------------------------|------------------------------------------------------------------------------------------|-----|-----|-----|------|
| $I_{BAT\_BLE\_RX\_96}$<br>$M\_FS$ | battery supply current | BLE receive mode; $f_{CLK}$ = 96 MHz; DC-DC on; FLASH in Standby mode; $V_{BAT}$ = 3 V.  |     | 8.5 |     | mA   |
| $I_{BAT\_BLE\_TX\_96}$<br>$M\_FS$ | battery supply current | BLE transmit mode; $f_{CLK}$ = 96 MHz; DC-DC on; FLASH in Standby mode; $V_{BAT}$ = 3 V. |     | 8.1 |     | mA   |

Table 788: Timing characteristics

| Parameter       | Description           | Conditions                                                                 | Min | Typ | Max  | Unit    |
|-----------------|-----------------------|----------------------------------------------------------------------------|-----|-----|------|---------|
| $t_{STA\_SUP}$  | supply startup time   | time from power-on reset or wake-up to BootROM/application execution start |     |     | 2000 | $\mu$ s |
| $t_{STA\_BOOT}$ | booter startup time   | BootROM code execution time                                                |     |     | 20   | ms      |
| $t_{CLF\_OTP}$  | cache line fetch time | from OTP; line size = 8 B                                                  |     |     | 6    | clock   |
| $t_{CLF\_FLA}$  | cache line fetch time | from FLASH; line size = 8 B                                                |     |     | 40   | clock   |

Table 789: Thermal characteristics

| Parameter           | Description                                   | Conditions                                                                    | Min | Typ | Max  | Unit           |
|---------------------|-----------------------------------------------|-------------------------------------------------------------------------------|-----|-----|------|----------------|
| $R_{TH\_J-A\_SPCB}$ | Device thermal resistance junction to ambient | aQFN60 package; Non-standard PCB; Zero number of thermal vias (worst case);   |     |     | 56.9 | $^{\circ}$ C/W |
| $R_{TH\_J-A\_PCB}$  | Device thermal resistance junction to ambient | aQFN60 package; JEDEC standard PCB; Zero number of thermal vias (worst case); |     |     | 40   | $^{\circ}$ C/W |

Table 790: 16 MHz Crystal Oscillator: Recommended operating conditions

| Parameter              | Description                  | Conditions                                                                  | Min | Typ | Max | Unit     |
|------------------------|------------------------------|-----------------------------------------------------------------------------|-----|-----|-----|----------|
| $f_{XTAL}(16M)$        | crystal oscillator frequency |                                                                             |     | 16  |     | MHz      |
| $ESR(16M)$             | equivalent series resistance |                                                                             |     |     | 100 | $\Omega$ |
| $C_L(16M)$             | load capacitance             | No external capacitors are required.                                        | 6   | 10  | 12  | pF       |
| $C_0(16M)$             | shunt capacitance            | No external capacitors are required.                                        |     |     | 5   | pF       |
| $\Delta f_{XTAL}(16M)$ | crystal frequency tolerance  | After optional trimming; including aging and temperature drift<br>(Note 22) | -20 |     | 20  | ppm      |

Table 790: 16 MHz Crystal Oscillator: Recommended operating conditions

| Parameter                   | Description                 | Conditions                                                                                       | Min | Typ | Max  | Unit    |
|-----------------------------|-----------------------------|--------------------------------------------------------------------------------------------------|-----|-----|------|---------|
| $\Delta f_{X\_TAL(16M)UNT}$ | crystal frequency tolerance | Untrimmed; including aging and temperature drift<br>(Note 23)                                    | -40 |     | 40   | ppm     |
| $P_{DRV(MAX)(16M)}$         | maximum drive power         | (Note 24)                                                                                        | 100 |     |      | $\mu W$ |
| $V_{CLK(EXT)(16M)}$         | external clock voltage      | In case of external clock source on XTAL16Mp (XTAL16Mm floating or connected to mid-level 0.6 V) | 1   | 1.2 | 1.26 | V       |
| $\phi_N(EXT-ER-NAL)16M$     | phase noise                 | $f_C = 50$ kHz; in case of external clock source                                                 |     |     | -130 | dBc/Hz  |

**Note 22:** Using the internal varicaps a wide range of crystals can be trimmed to the required tolerance.

**Note 23:** Maximum allowed frequency tolerance for compensation by the internal varicap trimming mechanism.

**Note 24:** Select a crystal which can handle a drive level of at least this specification.

Table 791: 16 MHz Crystal Oscillator: Timing characteristics

| Parameter            | Description                     | Conditions              | Min | Typ | Max | Unit |
|----------------------|---------------------------------|-------------------------|-----|-----|-----|------|
| $t_{STA(XTAL)(16M)}$ | crystal oscillator startup time | ESR=25 Ohm; $C_0=3$ pF; | 0.5 | 2   | 3   | ms   |

Table 792: 32 kHz Crystal Oscillator: Recommended operating conditions

| Parameter              | Description                                   | Conditions                                                                           | Min  | Typ    | Max  | Unit      |
|------------------------|-----------------------------------------------|--------------------------------------------------------------------------------------|------|--------|------|-----------|
| $f_{CLK\_EXT\_32K}$    | external clock frequency                      | at pin XTAL32KP/P2_0 in GPIO mode                                                    | 31   |        | 33   | kHz       |
| $V_{CLK\_EXT\_32K}$    | external clock voltage                        | at pin XTAL32KP/P2_0 in GPIO mode                                                    | 1    | 1.2    | 1.26 | V         |
| $f_{XTAL\_32K}$        | crystal oscillator frequency                  |                                                                                      |      | 32.768 |      | kHz       |
| $ESR_{32K}$            | equivalent series resistance                  |                                                                                      |      |        | 100  | $k\Omega$ |
| $C_{L\_32K}$           | load capacitance                              | No external capacitors are required for a 6 pF or 7 pF crystal.                      | 6    | 7      | 9    | pF        |
| $C_{0\_32K}$           | shunt capacitance                             |                                                                                      |      | 1      | 2    | pF        |
| $\Delta f_{XTAL\_32K}$ | crystal frequency tolerance (including aging) | Timing accuracy is dominated by crystal accuracy. A much smaller value is preferred. | -250 |        | 250  | ppm       |
| $P_{DRV\_MAX\_32K}$    | maximum drive power                           | (Note 25)                                                                            | 0.1  |        |      | $\mu W$   |

**Note 25:** Select a crystal that can handle a drive level of at least this specification.

Table 793: 32 kHz Crystal Oscillator: Timing characteristics

| Parameter            | Description                     | Conditions                                                | Min | Typ | Max | Unit |
|----------------------|---------------------------------|-----------------------------------------------------------|-----|-----|-----|------|
| $t_{STA\_XTAL\_32K}$ | crystal oscillator startup time | Typical application, time until 1000 clocks are detected. |     | 400 |     | ms   |

Table 794: 16 MHz RC Oscillator: AC characteristics

| Parameter | Description         | Conditions        | Min    | Typ | Max   | Unit |
|-----------|---------------------|-------------------|--------|-----|-------|------|
| $f_{CLK}$ | clock frequency     |                   | 6      |     | 19    | MHz  |
| $f_{TOL}$ | frequency tolerance | after calibration | -93750 |     | 93750 | ppm  |

Table 795: 16 MHz RC Oscillator: Timing characteristics

| Parameter | Description  | Conditions | Min | Typ | Max | Unit    |
|-----------|--------------|------------|-----|-----|-----|---------|
| $t_{STA}$ | startup time |            |     | 5   |     | $\mu$ s |

Table 796: Stable low frequency RCX Oscillator: Timing characteristics

| Parameter                      | Description                                           | Conditions                 | Min  | Typ  | Max | Unit    |
|--------------------------------|-------------------------------------------------------|----------------------------|------|------|-----|---------|
| $\Delta f_{RC}/\Delta T_{RCX}$ | RCX oscillator frequency variation versus temperature | preferred settings applied | -100 |      | 100 | ppm/deg |
| $f_{RC\_RCX}$                  | RCX oscillator frequency                              | preferred settings applied | 8    | 11.7 | 15  | kHz     |

Table 797: Low Power PLL: AC characteristics

| Parameter      | Description      | Conditions | Min | Typ | Max | Unit |
|----------------|------------------|------------|-----|-----|-----|------|
| $f_{O\_LPPLL}$ | output frequency |            |     | 96  |     | MHz  |

Table 798: Low Power PLL: Timing characteristics

| Parameter            | Description             | Conditions                         | Min | Typ | Max | Unit    |
|----------------------|-------------------------|------------------------------------|-----|-----|-----|---------|
| $t_{LOCK\_LPPLL\_1}$ | frequency settling time | 1 % accuracy after VCO calibration |     |     | 20  | $\mu$ s |
| $t_{LOCK\_LPPLL\_2}$ | frequency settling time | 200 ppm accuracy                   |     | 33  | 100 | $\mu$ s |

Table 799: LDO\_VBAT: Recommended operating conditions

| Parameter             | Description                  | Conditions | Min | Typ | Max | Unit       |
|-----------------------|------------------------------|------------|-----|-----|-----|------------|
| $I_{L\_LDO\_VBAT}$    | load current                 |            |     |     | 100 | mA         |
| $C_{L\_LDO\_VBAT}$    | load capacitance             |            | 1   |     | 100 | $\mu$ F    |
| $ESR_{CL\_LDO\_VBAT}$ | equivalent series resistance |            | 0   |     | 100 | m $\Omega$ |

Table 800: LDO\_VBAT: DC characteristics

| Parameter              | Description           | Conditions                     | Min  | Typ | Max  | Unit |
|------------------------|-----------------------|--------------------------------|------|-----|------|------|
| $I_{LK\_LDO\_VBAT}$    | clamping load current | output short-circuited         |      | 160 |      | mA   |
| $V_{LDO\_VBAT\_START}$ | LDO output voltage    | during startup                 |      |     | 3.3  | V    |
| $V_{LDO\_VBAT\_0}$     | LDO output voltage    | LDO_3V3_LEVEL = 0x0<br>no load | 2.28 | 2.4 | 2.52 | V    |
| $V_{LDO\_VBAT\_1}$     | LDO output voltage    | LDO_3V3_LEVEL = 0x1<br>no load | 3.13 | 3.3 | 3.46 | V    |

Table 800: LDO\_VBAT: DC characteristics

| Parameter                              | Description             | Conditions                         | Min  | Typ  | Max   | Unit |
|----------------------------------------|-------------------------|------------------------------------|------|------|-------|------|
| V <sub>LDO_VBAT_2</sub>                | LDO output voltage      | LDO_3V3_LEVEL = 0x2<br>no load     | 3.27 | 3.45 | 3.62  | V    |
| V <sub>LDO_VBAT_3</sub>                | LDO output voltage      | LDO_3V3_LEVEL = 0x3<br>no load     | 3.42 | 3.6  | 3.78  | V    |
| $\Delta V_O / \Delta I_{L\_LDO\_VBAT}$ | line regulation         | $V_i \geq (V_o + 200 \text{ mV})$  |      | 0.06 | 0.2   | %/V  |
| $\Delta V_O / \Delta I_{L\_LDO\_VBAT}$ | load regulation         | 10 mA < I <sub>load</sub> < 100 mA |      |      | 0.022 | %/mA |
| V <sub>DROP_VBAT</sub>                 | dropout voltage         | I <sub>L</sub> = 1 mA              |      |      | 100   | mV   |
| V <sub>DROP_VBAT_MAX</sub>             | maximum dropout voltage | I <sub>L</sub> = 110 mA            |      |      | 200   | mV   |

Table 801: LDO\_IO: Recommended operating conditions

| Parameter                | Description                  | Conditions | Min | Typ | Max | Unit |
|--------------------------|------------------------------|------------|-----|-----|-----|------|
| I <sub>L_LDO_IO</sub>    | load current                 |            |     |     | 75  | mA   |
| C <sub>L_LDO_IO</sub>    | load capacitance             |            | 1   | 10  | 100 | μF   |
| ESR <sub>CL_LDO_IO</sub> | equivalent series resistance |            | 0   |     | 100 | mΩ   |

Table 802: LDO\_IO: DC characteristics

| Parameter                            | Description             | Conditions                               | Min   | Typ | Max  | Unit |
|--------------------------------------|-------------------------|------------------------------------------|-------|-----|------|------|
| V <sub>LDO_IO</sub>                  | LDO output voltage      |                                          | 1.745 | 1.8 | 1.86 | V    |
| $\Delta V_O / \Delta I_{L\_LDO\_IO}$ | load regulation         | 7.5 mA ≤ I <sub>L</sub> ≤ 75 mA          |       |     | 0.03 | %/mA |
| $\Delta V_O / \Delta I_{L\_LDO\_IO}$ | line regulation         | V <sub>i</sub> > (V <sub>o</sub> + 0.2V) |       |     | 0.5  | %/V  |
| V <sub>DROP_IO</sub>                 | dropout voltage         | I <sub>L</sub> = 1 mA                    |       |     | 100  | mV   |
| V <sub>DROP_IO_MAX</sub>             | maximum dropout voltage | I <sub>L</sub> = 75 mA                   |       |     | 200  | mV   |

Table 803: LDO\_VBAT\_RET: Recommended operating conditions

| Parameter                   | Description  | Conditions | Min | Typ | Max | Unit |
|-----------------------------|--------------|------------|-----|-----|-----|------|
| I <sub>L_LDO_VBAT_RET</sub> | load current |            |     |     | 3   | mA   |

Table 804: LDO\_VBAT\_RET: DC characteristics

| Parameter                   | Description        | Conditions          | Min  | Typ  | Max  | Unit |
|-----------------------------|--------------------|---------------------|------|------|------|------|
| V <sub>LDO_VBAT_RET_0</sub> | LDO output voltage | LDO_3V3_LEVEL = 0x0 | 2.28 | 2.4  | 2.52 | V    |
| V <sub>LDO_VBAT_RET_1</sub> | LDO output voltage | LDO_3V3_LEVEL = 0x1 | 3.13 | 3.3  | 3.46 | V    |
| V <sub>LDO_VBAT_RET_2</sub> | LDO output voltage | LDO_3V3_LEVEL = 0x2 | 3.27 | 3.45 | 3.62 | V    |

Table 804: LDO\_VBAT\_RET: DC characteristics

| Parameter                                   | Description        | Conditions                        | Min  | Typ | Max  | Unit |
|---------------------------------------------|--------------------|-----------------------------------|------|-----|------|------|
| V <sub>LDO_VBAT_RET_3</sub>                 | LDO output voltage | LDO_3V3_LEVEL = 0x3               | 3.42 | 3.6 | 3.78 | V    |
| $\Delta V_O / \Delta I_{L\_LDO\_VBAT\_RET}$ | load regulation    | 0.3 mA < I <sub>load</sub> < 3 mA |      |     | 0.5  | %/mA |

Table 805: LDO\_IO\_RET: Recommended operating conditions

| Parameter                 | Description  | Conditions | Min | Typ | Max | Unit |
|---------------------------|--------------|------------|-----|-----|-----|------|
| I <sub>L_LDO_IO_RET</sub> | load current |            |     |     | 3   | mA   |

Table 806: LDO\_IO\_RET: DC characteristics

| Parameter                            | Description        | Conditions                        | Min  | Typ | Max  | Unit |
|--------------------------------------|--------------------|-----------------------------------|------|-----|------|------|
| V <sub>LDO_VBAT_IO_RET_0</sub>       | LDO output voltage | LDO_3V3_LEVEL = 0x0               | 1.71 | 1.8 | 1.89 | V    |
| $\Delta V_O / \Delta I_{L\_LDO\_IO}$ | load regulation    | 0.3 mA < I <sub>load</sub> < 3 mA |      |     | 1    | %/mA |

Table 807: LDO\_IO\_RET2: DC characteristics

| Parameter                            | Description        | Conditions                        | Min  | Typ | Max  | Unit |
|--------------------------------------|--------------------|-----------------------------------|------|-----|------|------|
| V <sub>LDO_VBAT_IO_RET_0</sub>       | LDO output voltage |                                   | 1.71 | 1.8 | 1.89 | V    |
| $\Delta V_O / \Delta I_{L\_LDO\_IO}$ | load regulation    | 0.3 mA < I <sub>load</sub> < 3 mA |      |     | 1    | %/mA |

Table 808: LDO\_USB: Recommended operating conditions

| Parameter                 | Description                  | Conditions | Min | Typ | Max | Unit |
|---------------------------|------------------------------|------------|-----|-----|-----|------|
| I <sub>L_LDO_USB</sub>    | load current                 |            |     |     | 100 | mA   |
| C <sub>L_LDO_USB</sub>    | load capacitance             |            | 1   |     | 100 | μF   |
| ESR <sub>CL_LDO_USB</sub> | equivalent series resistance |            | 0   |     | 100 | mΩ   |

Table 809: LDO\_USB: DC characteristics

| Parameter                             | Description           | Conditions                              | Min  | Typ  | Max   | Unit |
|---------------------------------------|-----------------------|-----------------------------------------|------|------|-------|------|
| I <sub>LK_LDO_USB</sub>               | clamping load current | output short-circuited                  |      | 160  |       | mA   |
| V <sub>LDO_USB_0</sub>                | LDO output voltage    | LDO_USB_LEVEL = 0x0                     | 2.28 | 2.4  | 2.52  | V    |
| V <sub>LDO_USB_1</sub>                | LDO output voltage    | LDO_USB_LEVEL = 0x1                     | 3.13 | 3.3  | 3.46  | V    |
| V <sub>LDO_USB_2</sub>                | LDO output voltage    | LDO_USB_LEVEL = 0x2                     | 3.27 | 3.45 | 3.62  | V    |
| V <sub>LDO_USB_3</sub>                | LDO output voltage    | LDO_USB_LEVEL = 0x3                     | 3.42 | 3.6  | 3.78  | V    |
| $\Delta V_O / \Delta I_{L\_LDO\_USB}$ | load regulation       | 10 mA ≤ I <sub>L</sub> ≤ 100 mA         |      |      | 0.022 | %/mA |
| $\Delta V_O / \Delta V_{I\_LDO\_USB}$ | line regulation       | V <sub>i</sub> ≥ V <sub>o</sub> + 400mV |      |      | 0.2   | %/V  |

Table 810: SIMO DC-DC converter: Recommended operating conditions

| Parameter           | Description  | Conditions | Min | Typ | Max | Unit |
|---------------------|--------------|------------|-----|-----|-----|------|
| I <sub>L_1V8</sub>  | load current |            |     |     | 75  | mA   |
| I <sub>L_1V8P</sub> | load current |            |     |     | 75  | mA   |

Table 811: SIMO DC-DC converter: DC characteristics

| Parameter             | Description      | Conditions                                               | Min | Typ | Max | Unit |
|-----------------------|------------------|----------------------------------------------------------|-----|-----|-----|------|
| V <sub>O_1V8</sub>    | output voltage   | pin VDD1V8                                               | 1.7 | 1.8 | 1.9 | V    |
| V <sub>RPL_1V8</sub>  | ripple voltage   | peak-to-peak value.<br>Using 10 uF, 16 V, X5R capacitors |     | 60  | 110 | mV   |
| V <sub>ACC_V18</sub>  | voltage accuracy | maximum error on average level                           |     |     | 6   | %    |
| V <sub>O_1V8P</sub>   | output voltage   | pin VDD1V8P                                              | 1.7 | 1.8 | 1.9 | V    |
| V <sub>RPL_1V8P</sub> | ripple voltage   | peak-to-peak value.<br>Using 10 uF, 16 V, X5R capacitors |     | 60  | 110 | mV   |
| V <sub>ACC_1V8P</sub> | voltage accuracy | maximum error on average level                           |     |     | 6   | %    |

Table 812: Brownout Detection: DC characteristics

| Parameter              | Description                                                                | Conditions                 | Min   | Typ  | Max   | Unit |
|------------------------|----------------------------------------------------------------------------|----------------------------|-------|------|-------|------|
| V <sub>RST_1V2_1</sub> | reset voltage                                                              | pin V12; BOD_VDD_LVL = 0x1 | 0.65  | 0.7  | 0.75  | V    |
| V <sub>RST_1V2_3</sub> | reset voltage                                                              | pin V12; BOD_VDD_LVL = 0x3 | 0.75  | 0.8  | 0.85  | V    |
| V <sub>RST_1V2_7</sub> | reset voltage                                                              | pin V12; BOD_VDD_LVL = 0x7 | 1.005 | 1.05 | 1.095 | V    |
| V <sub>RST_1V8</sub>   | reset voltage                                                              | pins VDD1V8, VDD1V8P       | 1.585 | 1.65 | 1.715 | V    |
| V <sub>RST_3V3</sub>   | reset voltage                                                              | pin V33                    | 2.6   | 2.7  | 2.8   | V    |
| V <sub>TH_LDO</sub>    | threshold voltage below which DC-DC converter is stopped and LDO is active | pin VBAT1                  | 2.35  | 2.45 | 2.55  | V    |

Table 813: Charger: DC characteristics

| Parameter             | Description    | Conditions                                           | Min  | Typ | Max   | Unit |
|-----------------------|----------------|------------------------------------------------------|------|-----|-------|------|
| I <sub>CHARGE_0</sub> | charge current | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x0<br>(Note 26) |      | 5   | 5.75  | mA   |
| I <sub>CHARGE_1</sub> | charge current | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x1<br>(Note 26) |      | 10  | 11.5  | mA   |
| I <sub>CHARGE_2</sub> | charge current | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x2<br>(Note 26) | 27   | 30  | 34.5  | mA   |
| I <sub>CHARGE_3</sub> | charge current | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x3<br>(Note 26) | 40.5 | 45  | 51.75 | mA   |

Table 813: Charger: DC characteristics

| Parameter               | Description                                 | Conditions                                              | Min   | Typ  | Max   | Unit |
|-------------------------|---------------------------------------------|---------------------------------------------------------|-------|------|-------|------|
| I <sub>CHARGE_4</sub>   | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x4<br>(Note 26)    | 54    | 60   | 69    | mA   |
| I <sub>CHARGE_5</sub>   | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x5<br>(Note 26)    | 81    | 90   | 103.5 | mA   |
| I <sub>CHARGE_6</sub>   | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x6<br>(Note 26)    | 108   | 120  | 138   | mA   |
| I <sub>CHARGE_7</sub>   | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x7<br>(Note 26)    | 135   | 150  | 172.5 | mA   |
| I <sub>CHARGE_8</sub>   | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x8<br>(Note 26)    | 162   | 180  | 207   | mA   |
| I <sub>CHARGE_9</sub>   | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] = 0x9<br>(Note 26)    | 189   | 210  | 241.5 | mA   |
| I <sub>CHARGE_10</sub>  | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] =<br>0xA<br>(Note 26) | 243   | 270  | 310.5 | mA   |
| I <sub>CHARGE_11</sub>  | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] =<br>0xB<br>(Note 26) | 270   | 300  | 345   | mA   |
| I <sub>CHARGE_12</sub>  | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] =<br>0xC<br>(Note 26) | 315   | 350  | 402.5 | mA   |
| I <sub>CHARGE_13</sub>  | charge current                              | CHARGER_CTRL1_RE<br>G[CHARGE_CUR] =<br>0xD<br>(Note 26) | 360   | 400  | 460   | mA   |
| NTCR <sub>TH_COLD</sub> | NTC to VDD_USB volt-<br>age ratio threshold |                                                         | 83    | 87.5 | 92    | %    |
| NTCR <sub>TH_HOT</sub>  | NTC to VDD_USB volt-<br>age ratio threshold |                                                         | 45    | 50   | 55    | %    |
| V <sub>CHARGE_0</sub>   | charge voltage 00000b                       | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x00           | 2.91  | 3    | 3.09  | V    |
| V <sub>CHARGE_1</sub>   | charge voltage 00001b                       | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x01           | 3.298 | 3.4  | 3.502 | V    |
| V <sub>CHARGE_2</sub>   | charge voltage 00010b                       | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x02           | 3.395 | 3.5  | 3.605 | V    |
| V <sub>CHARGE_3</sub>   | charge voltage 00011b                       | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x03           | 3.492 | 3.6  | 3.708 | V    |
| V <sub>CHARGE_4</sub>   | charge voltage 00100b                       | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x04           | 3.628 | 3.74 | 3.852 | V    |

Table 813: Charger: DC characteristics

| Parameter              | Description           | Conditions                                    | Min   | Typ  | Max   | Unit |
|------------------------|-----------------------|-----------------------------------------------|-------|------|-------|------|
| V <sub>CHARGE_5</sub>  | charge voltage 00101b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x05 | 3.744 | 3.86 | 3.976 | V    |
| V <sub>CHARGE_6</sub>  | charge voltage 00110b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x06 | 3.88  | 4    | 4.12  | V    |
| V <sub>CHARGE_7</sub>  | charge voltage 00111b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x07 | 3.929 | 4.05 | 4.172 | V    |
| V <sub>CHARGE_8</sub>  | charge voltage 01000b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x08 | 3.977 | 4.1  | 4.223 | V    |
| V <sub>CHARGE_9</sub>  | charge voltage 01001b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x09 | 4.026 | 4.15 | 4.275 | V    |
| V <sub>CHARGE_10</sub> | charge voltage 01010b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x0A | 4.074 | 4.2  | 4.326 | V    |
| V <sub>CHARGE_11</sub> | charge voltage 01011b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x0B | 4.123 | 4.25 | 4.378 | V    |
| V <sub>CHARGE_12</sub> | charge voltage 01100b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x0C | 4.171 | 4.3  | 4.429 | V    |
| V <sub>CHARGE_13</sub> | charge voltage 01101b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x0D | 4.22  | 4.35 | 4.481 | V    |
| V <sub>CHARGE_14</sub> | charge voltage 01110b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x0E | 4.268 | 4.4  | 4.532 | V    |
| V <sub>CHARGE_15</sub> | charge voltage 01111b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x0F | 4.365 | 4.5  | 4.635 | V    |
| V <sub>CHARGE_16</sub> | charge voltage 10000b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x10 | 4.462 | 4.6  | 4.738 | V    |
| V <sub>CHARGE_17</sub> | charge voltage 10001b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x11 | 4.753 | 4.9  | 5.047 | V    |
| V <sub>CHARGE_18</sub> | charge voltage 10010b | CHARGER_CTRL1_RE<br>G[CHARGE_LEVEL] =<br>0x12 | 4.85  | 5    | 5.15  | V    |

**Note 26:** minimum allowed charging temperature is -5 °C.

Table 814: Temperature Sensor: DC characteristics

| Parameter              | Description                     | Conditions                          | Min | Typ | Max | Unit |
|------------------------|---------------------------------|-------------------------------------|-----|-----|-----|------|
| T <sub>SENSE</sub>     | sense temperature               | Measured with on chip<br>ADC        | -20 |     | 110 | °C   |
| T <sub>SENSE_ACC</sub> | sense temperature accu-<br>racy | Single point calibration at<br>25°C |     |     | 8.8 | °C   |



Table 815: Fuel Gauge (SOC): DC characteristics

| Parameter   | Description                            | Conditions                                                                    | Min | Typ  | Max | Unit |
|-------------|----------------------------------------|-------------------------------------------------------------------------------|-----|------|-----|------|
| $V_{I\_FS}$ | full-scale input voltage               |                                                                               |     |      | 75  | mV   |
| $SOC_{INL}$ | state of charge integral non-linearity | $R_{SENSE} = 0.1 \Omega$ ; $1 \text{ mA} \leq I_{SENSE} \leq 1000 \text{ mA}$ |     | 0.05 |     | %    |

Table 816: General Purpose ADC: Recommended operating conditions

| Parameter   | Description              | Conditions                                                                                                                                                    | Min | Typ  | Max | Unit |
|-------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------|-----|------|
| ENOB(AVG16) | Effective Number of Bits | Averaging 16 times;<br>GP_ADC_CTRL_REG=0x4C01;<br>GP_ADC_CTRL2_REG=0x0008;<br>GP_ADC_OFFP_REG=0x0200;<br>GP_ADC_OFFN_REG=0x0200;<br>GP_ADC_DELAY2_REG=0xC000; |     | 10.5 |     | bits |
| ENOB(AVG32) | Effective Number of Bits | Averaging 32 times;<br>GP_ADC_CTRL_REG=0x4C01;<br>GP_ADC_CTRL2_REG=0x0008;<br>GP_ADC_OFFP_REG=0x0200;<br>GP_ADC_OFFN_REG=0x0200;<br>GP_ADC_DELAY2_REG=0xC000; |     | 10.8 |     | bits |
| ENOB(AVG64) | Effective Number of Bits | Averaging 64 times;<br>GP_ADC_CTRL_REG=0x4C01;<br>GP_ADC_CTRL2_REG=0x0008;<br>GP_ADC_OFFP_REG=0x0200;<br>GP_ADC_OFFN_REG=0x0200;<br>GP_ADC_DELAY2_REG=0xC000; |     | 11.1 |     | bits |

Table 817: General Purpose ADC: DC characteristics

| Parameter    | Description                       | Conditions                                                                   | Min  | Typ   | Max  | Unit |
|--------------|-----------------------------------|------------------------------------------------------------------------------|------|-------|------|------|
| $V_{I(ZS)}$  | zero-scale input voltage          | single-ended, calibrated at zero input offset calibrated and trimmed bandgap | -6   |       | 6    | mV   |
| $V_{I(FS)}$  | full-scale input voltage          | single-ended, calibrated at zero input offset calibrated and trimmed bandgap | 1150 | 1180  | 1250 | mV   |
| $V_{I(FSN)}$ | negative full-scale input voltage | differential, calibrated at zero input                                       |      | -1180 |      | mV   |

Table 817: General Purpose ADC: DC characteristics

| Parameter    | Description                       | Conditions                                                                                                                             | Min | Typ  | Max | Unit |
|--------------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|-----|------|-----|------|
| $V_{I(FSP)}$ | positive full-scale input voltage | differential, calibrated at zero input                                                                                                 |     | 1180 |     | mV   |
| INL          | integral non-linearity            | GP_ADC_CTRL_REG=0x4C01;<br>GP_ADC_CTRL2_REG=0x0008;<br>GP_ADC_OFFP_REG=0x0200;<br>GP_ADC_OFFN_REG=0x0200;<br>GP_ADC_DELAY2_REG=0xC000; | -2  |      | 2   | LSB  |
| DNL          | differential non-linearity        |                                                                                                                                        | -2  |      | 2   | LSB  |
| $E_{OFS}$    | offset error                      | differential, uncalibrated.<br>GP_ADC_IDYN=1 and GP_ADC_I20U=1                                                                         | -4  |      | 4   | LSB  |
| $E_G$        | gain error                        | GP_ADC_IDYN=1 and GP_ADC_I20U=1                                                                                                        | -32 |      | 32  | LSB  |

Table 818: General Purpose ADC: Timing characteristics

| Parameter       | Description     | Conditions                                                                                                                                                                                                       | Min | Typ  | Max | Unit    |
|-----------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------|-----|---------|
| $t_{CONV(ADC)}$ | conversion time | Excluding initial settling time of the LDO and the 3x-attenuation (if used):<br><br>LDO settling time is 20 $\mu$ s (max), 3x-attenuation settling time = 1 $\mu$ s (max)<br>Using internal ADC-clock (~200 MHz) |     | 0.25 | 0.4 | $\mu$ s |

Table 819: LED Driver: DC characteristics

| Parameter          | Description            | Conditions                                                                          | Min | Typ | Max | Unit |
|--------------------|------------------------|-------------------------------------------------------------------------------------|-----|-----|-----|------|
| $I_{MATCH\_LED}$   | current matching       | relative to average LED output current                                              |     | 0.2 | 2   | %    |
| $I_{O\_MAX\_LED0}$ | maximum output current | 200 mV saturation voltage and 100% duty cycle                                       | 14  | 19  | 24  | mA   |
| $I_{O\_MAX\_LED1}$ | maximum output current | 200 mV saturation voltage and 100% duty cycle                                       | 14  | 19  | 24  | mA   |
| $I_{O\_MAX\_LED2}$ | maximum output current | 200 mV saturation voltage and 100% duty cycle                                       | 14  | 19  | 24  | mA   |
| $I_{ACC\_LED0}$    | current accuracy       | PWM accuracy at 10% duty cycle and above relative to $I_{O\_MAX}$ (100% duty cycle) | -2  |     | 2   | %    |
| $I_{ACC\_LED1}$    | current accuracy       | PWM accuracy at 10% duty cycle and above relative to $I_{O\_MAX}$ (100% duty cycle) | -2  |     | 2   | %    |

Table 819: LED Driver: DC characteristics

| Parameter             | Description       | Conditions                                                                                | Min | Typ | Max | Unit |
|-----------------------|-------------------|-------------------------------------------------------------------------------------------|-----|-----|-----|------|
| I <sub>ACC_LED2</sub> | current accuracy  | PWM accuracy at 10% duty cycle and above relative to I <sub>o_max</sub> (100% duty cycle) | -2  |     | 2   | %    |
| I <sub>OFF_LED0</sub> | off-state current | driver disabled                                                                           |     |     | 0.2 | μA   |
| I <sub>OFF_LED1</sub> | off-state current | driver disabled                                                                           |     |     | 0.2 | μA   |
| I <sub>OFF_LED2</sub> | off-state current | driver disabled                                                                           |     |     | 0.2 | μA   |

Table 820: LED Driver: AC characteristics

| Parameter            | Description   | Conditions | Min | Typ | Max | Unit |
|----------------------|---------------|------------|-----|-----|-----|------|
| f <sub>PWM_LED</sub> | PWM frequency |            | 256 |     | 512 | Hz   |

Table 821: USB charger detect: DC characteristics

| Parameter               | Description              | Conditions                           | Min   | Typ | Max  | Unit |
|-------------------------|--------------------------|--------------------------------------|-------|-----|------|------|
| V <sub>IH_CHG_DET</sub> | HIGH level input voltage | USB_CHARGER_CTRL_REG[VDP_SRC_ON]=1   | 0.4   |     |      | V    |
| V <sub>IL_CHG_DET</sub> | LOW level input voltage  | USB_CHARGER_CTRL_REG[VDP_SRC_ON]=1   |       |     | 0.25 | V    |
| V <sub>IH_DCP_DET</sub> | HIGH level input voltage | USB_CHARGER_CTRL_REG[VDM_SRC_ON]=1   | 0.8   |     |      | V    |
| V <sub>IL_DCP_DET</sub> | LOW level input voltage  | USB_CHARGER_CTRL_REG[VDM_SRC_ON]=1   |       |     | 0.25 | V    |
| V <sub>IH_DM_VAL</sub>  | HIGH level input voltage |                                      | 1.5   |     |      | V    |
| V <sub>IL_DM_VAL</sub>  | LOW level input voltage  |                                      |       |     | 0.8  | V    |
| V <sub>IH_DP_VAL</sub>  | HIGH level input voltage |                                      | 1.5   |     |      | V    |
| V <sub>IL_DP_VAL</sub>  | LOW level input voltage  |                                      |       |     | 0.8  | V    |
| V <sub>IH_DM_VAL2</sub> | HIGH level input voltage |                                      | 2.5   |     |      | V    |
| V <sub>IL_DM_VAL2</sub> | LOW level input voltage  |                                      |       |     | 2.3  | V    |
| V <sub>IH_DP_VAL2</sub> | HIGH level input voltage |                                      | 2.5   |     |      | V    |
| V <sub>IL_DP_VAL2</sub> | LOW level input voltage  |                                      |       |     | 2.3  | V    |
| V <sub>O_DM_SRC</sub>   | output voltage           |                                      | 0.5   |     | 0.7  | V    |
| V <sub>O_DP_SRC</sub>   | output voltage           |                                      | 0.5   |     | 0.7  | V    |
| I <sub>DM_SINK</sub>    | D- sink current          |                                      | 25    |     | 175  | μA   |
| I <sub>DP_SINK</sub>    | D+ sink current          |                                      | 25    |     | 175  | μA   |
| I <sub>DP_SRC</sub>     | D+ source current        |                                      | 5     |     | 13   | μA   |
| R <sub>DM_DWN</sub>     | D- resistance to ground  | USB_CHARGER_CTRL_REG[IDP_SRC_ON] = 1 | 14.25 |     | 24.8 | kΩ   |

Table 822: Digital I/O Pad: DC characteristics

| Parameter              | Description               | Conditions                                                          | Min  | Typ  | Max  | Unit |
|------------------------|---------------------------|---------------------------------------------------------------------|------|------|------|------|
| V <sub>IH</sub>        | HIGH level input voltage  |                                                                     | 0.84 |      |      | V    |
| V <sub>IL</sub>        | LOW level input voltage   |                                                                     |      |      | 0.36 | V    |
| V <sub>OH_1V8</sub>    | HIGH level output voltage | I <sub>O</sub> = -4.8 mA, VDD1V8 = 1.80 V                           | 1.44 |      |      | V    |
| V <sub>OL_1V8</sub>    | LOW level output voltage  | I <sub>O</sub> = 4.8 mA, VDD1V8 = 1.80 V                            |      |      | 0.36 | V    |
| V <sub>OH_3V3</sub>    | HIGH level output voltage | I <sub>O</sub> = -4.8 mA, V33 = 2.35 V                              | 1.88 |      |      | V    |
| V <sub>OL_3V3</sub>    | LOW level output voltage  | I <sub>O</sub> = 4.8 mA, V33 = 2.35 V                               |      |      | 0.47 | V    |
| I <sub>IH</sub>        | HIGH level input current  | V <sub>I</sub> = V33 = 3.3 V, excluding USB pads (P11, P22) and P21 | -10  | 5    | 10   | μA   |
| I <sub>IL</sub>        | LOW level input current   | V <sub>I</sub> = VSS = 0 V, excluding USB pads (P11, P22) and P21   | -10  | 4    | 10   | μA   |
| I <sub>IH_PD_3V3</sub> | HIGH level input current  | V <sub>I</sub> = V33 = 3.3 V                                        | 65   |      | 200  | μA   |
| I <sub>IL_PU_1V8</sub> | LOW level input current   | V <sub>I</sub> = VSS = 0 V, VDD1V8 = 1.8 V                          | -110 |      | -35  | μA   |
| I <sub>IL_PU_3V3</sub> | LOW level input current   | V <sub>I</sub> = VSS = 0 V, V33 = 3.3 V                             | -200 |      | -65  | μA   |
| SR <sub>R</sub>        | rising slew rate          | C <sub>L</sub> = 15 pF; I <sub>L</sub> = 4.8 mA;                    | 0.4  |      | 3.2  | V/ns |
| SR <sub>F</sub>        | falling slew rate         | C <sub>L</sub> = 15 pF; I <sub>L</sub> = 4.8 mA;                    | 0.4  |      | 3.3  | V/ns |
| C <sub>IN</sub>        | input capacitance         |                                                                     |      | 0.75 |      | pF   |

Table 823: Quad SPI Digital I/O Pad: DC characteristics

| Parameter                | Description               | Conditions                            | Min  | Typ | Max  | Unit |
|--------------------------|---------------------------|---------------------------------------|------|-----|------|------|
| V <sub>IH</sub>          | HIGH level input voltage  | VDDIO = 3.45 V                        | 2.42 |     |      | V    |
| V <sub>IL</sub>          | LOW level input voltage   | VDDIO = 3.45 V                        |      |     | 1.04 | V    |
| V <sub>IH_IO_1V8</sub>   | HIGH level input voltage  | VDDIO = 1.6 V                         | 1.12 |     |      | V    |
| V <sub>IL_IO_1V8</sub>   | LOW level input voltage   | VDDIO = 1.6 V                         |      |     | 0.48 | V    |
| V <sub>OH_4mA</sub>      | HIGH level output voltage | I <sub>O</sub> = 4 mA, VDDIO = 2.8 V  | 2.24 |     |      | V    |
| V <sub>OH_8mA</sub>      | HIGH level output voltage | I <sub>O</sub> = 8 mA, VDDIO = 2.8 V  | 2.24 |     |      | V    |
| V <sub>OH_12mA</sub>     | HIGH level output voltage | I <sub>O</sub> = 12 mA, VDDIO = 2.8 V | 2.24 |     |      | V    |
| V <sub>OH_16mA</sub>     | HIGH level output voltage | I <sub>O</sub> = 16 mA, VDDIO = 2.8 V | 2.24 |     |      | V    |
| V <sub>OH_1V6_4mA</sub>  | HIGH level output voltage | I <sub>O</sub> = 3 mA, VDDIO = 1.65 V | 1.32 |     |      | V    |
| V <sub>OH_1V6_8mA</sub>  | HIGH level output voltage | I <sub>O</sub> = 6 mA, VDDIO = 1.65 V | 1.32 |     |      | V    |
| V <sub>OH_1V6_12mA</sub> | HIGH level output voltage | I <sub>O</sub> = 9 mA, VDDIO = 1.65 V | 1.32 |     |      | V    |

Table 823: Quad SPI Digital I/O Pad: DC characteristics

| Parameter                 | Description                             | Conditions                                   | Min  | Typ | Max  | Unit    |
|---------------------------|-----------------------------------------|----------------------------------------------|------|-----|------|---------|
| V <sub>OH_1V6_16mA</sub>  | HIGH level output voltage               | I <sub>O</sub> = 12 mA, VDDIO = 1.65 V       | 1.32 |     |      | V       |
| V <sub>OH_3V3_100uA</sub> | HIGH level output voltage               | I <sub>O</sub> = 100 $\mu$ A, VDDIO = 3.45 V | 3.35 |     |      | V       |
| V <sub>OL_4mA</sub>       | LOW level output voltage                | I <sub>O</sub> = 4 mA, VDDIO = 2.8 V         |      |     | 0.56 | V       |
| V <sub>OL_8mA</sub>       | LOW level output voltage                | I <sub>O</sub> = 8 mA, VDDIO = 2.8 V         |      |     | 0.56 | V       |
| V <sub>OL_12mA</sub>      | LOW level output voltage                | I <sub>O</sub> = 12 mA, VDDIO = 2.8 V        |      |     | 0.56 | V       |
| V <sub>OL_16mA</sub>      | LOW level output voltage                | I <sub>O</sub> = 16 mA, VDDIO = 2.8 V        |      |     | 0.56 | V       |
| V <sub>OL_1V6_4mA</sub>   | LOW level output voltage                | I <sub>O</sub> = 3 mA, VDDIO = 1.65 V        |      |     | 0.33 | V       |
| V <sub>OL_1V6_8mA</sub>   | LOW level output voltage                | I <sub>O</sub> = 6 mA, VDDIO = 1.65 V        |      |     | 0.33 | V       |
| V <sub>OL_1V6_12mA</sub>  | LOW level output voltage                | I <sub>O</sub> = 9 mA, VDDIO = 1.65 V        |      |     | 0.33 | V       |
| V <sub>OL_1V6_16mA</sub>  | LOW level output voltage                | I <sub>O</sub> = 12 mA, VDDIO = 1.65 V       |      |     | 0.33 | V       |
| V <sub>OL_3V3_100uA</sub> | LOW level output voltage                | I <sub>O</sub> = 100 $\mu$ A, VDDIO = 3.45 V |      |     | 0.1  | V       |
| I <sub>IH</sub>           | HIGH level input current                | V <sub>I</sub> = VDDIO, VDDIO = 3.45 V       | -10  |     | 10   | $\mu$ A |
| I <sub>IH_PD</sub>        | HIGH level input current with pull-down | V <sub>I</sub> = 1.8 V                       | 25   | 45  | 75   | $\mu$ A |
| I <sub>IL</sub>           | LOW level input current                 | V <sub>I</sub> = VSS, VDDIO = 3.45 V         | -10  |     | 10   | $\mu$ A |
| I <sub>IL_PU</sub>        | LOW level input current with pull-up    | V <sub>I</sub> = VSS, VDDIO = 1.8 V          | -75  | -45 | -25  | $\mu$ A |

Table 824: Reset Pad: DC characteristics

| Parameter           | Description              | Conditions                      | Min  | Typ | Max  | Unit    |
|---------------------|--------------------------|---------------------------------|------|-----|------|---------|
| V <sub>IH_RST</sub> | HIGH level input voltage | pin RST                         | 0.84 |     |      | V       |
| V <sub>IL_RST</sub> | LOW level input voltage  | pin RST                         |      |     | 0.36 | V       |
| I <sub>IH_RST</sub> | HIGH level input current | pin RST, V <sub>I</sub> = 1.2 V | 25   |     | 75   | $\mu$ A |

Table 825: Radio - BLE mode: Recommended operating conditions

| Parameter         | Description         | Conditions  | Min  | Typ      | Max    | Unit |
|-------------------|---------------------|-------------|------|----------|--------|------|
| f <sub>OPER</sub> | operating frequency |             | 2400 |          | 2483.5 | MHz  |
| N <sub>CH</sub>   | number of channels  |             |      | 40       |        | 1    |
| f <sub>CH</sub>   | channel frequency   | K = 0 to 39 |      | 2402+K*2 |        | MHz  |

Table 826: Radio - BLE mode: DC characteristics

| Parameter              | Description            | Conditions                                                                                           | Min | Typ | Max | Unit |
|------------------------|------------------------|------------------------------------------------------------------------------------------------------|-----|-----|-----|------|
| I <sub>BAT_RF_RX</sub> | battery supply current | radio receiver and synthesizer active; ideal DC-DC converter; T <sub>A</sub> = 25 °C<br>(Note 27)    |     | 3.1 |     | mA   |
| I <sub>BAT_RF_TX</sub> | battery supply current | radio transmitter and synthesizer active; ideal DC-DC converter; T <sub>A</sub> = 25 °C<br>(Note 27) |     | 3.4 |     | mA   |

**Note 27:** The DC-DC converter efficiency is assumed to be 100 % to enable benchmarking of the radio currents at battery supply domain.

Table 827: Radio - BLE mode: AC characteristics

| Parameter               | Description                                       | Conditions                                                                                                                                                                                                                                                      | Min | Typ   | Max | Unit |
|-------------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------|-----|------|
| P <sub>SENS_CLEAN</sub> | sensitivity level                                 | Dirty Transmitter disabled; DC-DC converter disabled; PER = 30.8 %;<br>(Note 28)                                                                                                                                                                                |     | -94   |     | dBm  |
| P <sub>SENS</sub>       | sensitivity level                                 | Normal Operating Conditions; DC-DC converter disabled; PER = 30.8 %;<br>(Note 28)                                                                                                                                                                               |     | -93.5 |     | dBm  |
| P <sub>INT_IMD</sub>    | intermodulation distortion interferer power level | worst-case interferer level @ f <sub>1</sub> , f <sub>2</sub> with 2*f <sub>1</sub> - f <sub>2</sub> = f <sub>0</sub> ,  f <sub>1</sub> - f <sub>2</sub>   = n MHz and n = 3, 4, 5; P <sub>WANTED</sub> = -64 dBm @ f <sub>0</sub> ; PER = 30.8 %;<br>(Note 29) | -35 | -31   |     | dBm  |
| CIR <sub>0</sub>        | carrier to interferer ratio                       | n = 0; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz;<br>(Note 30)                                                                                                                                                                                     |     | 7     | 13  | dB   |
| CIR <sub>1</sub>        | carrier to interferer ratio                       | n = ±1; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz;<br>(Note 30)                                                                                                                                                                                    |     | -3    | 2   | dB   |
| CIR <sub>p2</sub>       | carrier to interferer ratio                       | n = +2 (image frequency); interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz;<br>(Note 30)                                                                                                                                                                  |     | -20   | -12 | dB   |
| CIR <sub>M2</sub>       | carrier to interferer ratio                       | n = -2; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz;<br>(Note 30)                                                                                                                                                                                    |     | -30   | -26 | dB   |
| CIR <sub>p3</sub>       | carrier to interferer ratio                       | n = +3 (image frequency + 1 MHz); interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz;<br>(Note 30)                                                                                                                                                          |     | -30   | -19 | dB   |
| CIR <sub>M3</sub>       | carrier to interferer ratio                       | n = -3; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz;<br>(Note 30)                                                                                                                                                                                    |     | -35   | -33 | dB   |

Table 827: Radio - BLE mode: AC characteristics

| Parameter             | Description                          | Conditions                                                                                                                         | Min  | Typ  | Max  | Unit   |
|-----------------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|------|------|------|--------|
| CIR <sub>4</sub>      | carrier to interferer ratio          | $ n  \geq 4$ (any other BLE channel); interferer @ $f_1 = f_0 + n \cdot 1 \text{ MHz}$ ; <b>(Note 30)</b>                          |      | -37  | -33  | dB     |
| P <sub>BL_I</sub>     | blocker power level                  | $30 \text{ MHz} \leq f_{\text{BL}} \leq 2000 \text{ MHz}$ ; P <sub>WANTED</sub> = -67 dBm; <b>(Note 31)</b>                        | -5   |      |      | dBm    |
| P <sub>BL_II</sub>    | blocker power level <b>(Note 32)</b> | $2003 \text{ MHz} \leq f_{\text{BL}} \leq 2399 \text{ MHz}$ ; P <sub>WANTED</sub> = -67 dBm; <b>(Note 31)</b>                      | -5   |      |      | dBm    |
| P <sub>BL_III</sub>   | blocker power level                  | $2484 \text{ MHz} \leq f_{\text{BL}} \leq 2997 \text{ MHz}$ ; P <sub>WANTED</sub> = -67 dBm; <b>(Note 31)</b>                      | -5   |      |      | dBm    |
| P <sub>BL_IV</sub>    | blocker power level                  | $3000 \text{ MHz} \leq f_{\text{BL}} \leq 12.75 \text{ GHz}$ ; P <sub>WANTED</sub> = -67 dBm; <b>(Note 31)</b>                     | -5   |      |      | dBm    |
| P <sub>RSSI_MIN</sub> | RSSI power level                     | absolute power level for RXRSSI[7:0] = 0;                                                                                          | -116 | -114 | -112 | dBm    |
| P <sub>RSSI_MAX</sub> | RSSI power level                     | upper limit of monotonous range;                                                                                                   | -24  | -22  | -20  | dBm    |
| L <sub>ACC_RSSI</sub> | level accuracy                       | tolerance at 5 % to 95 % confidence interval of P <sub>RF</sub> : when RXRSSI[7:0] = X, $50 < X < 175$ ; burst mode, 1500 packets; |      | 0    | 2    | dB     |
| L <sub>RES_RSSI</sub> | level resolution                     | gradient of monotonous range for RXRSSI[7:0] = X, $50 < X < 175$ ; burst mode, 1500 packets;                                       | 0.47 | 0.48 | 0.49 | dB/LSB |
| ACP <sub>2M</sub>     | adjacent channel power level         | $f_{\text{OFS}} \geq 2 \text{ MHz}$ ; <b>(Note 33)</b>                                                                             |      | -53  |      | dBm    |
| ACP <sub>3M</sub>     | adjacent channel power level         | $f_{\text{OFS}} \geq 3 \text{ MHz}$ ; <b>(Note 33)</b>                                                                             |      | -57  |      | dBm    |
| P <sub>O</sub>        | output power level                   | maximum gain;                                                                                                                      | -1   | 0    | 1    | dBm    |
| P <sub>O_H2</sub>     | output power level (second harmonic) | maximum gain;                                                                                                                      |      |      | -41  | dBm    |
| P <sub>O_H3</sub>     | output power level (third harmonic)  | maximum gain;                                                                                                                      |      |      | -41  | dBm    |
| P <sub>O_H4</sub>     | output power level (fourth harmonic) | maximum gain;                                                                                                                      |      |      | -41  | dBm    |
| P <sub>O_H5</sub>     | output power level (fifth harmonic)  | maximum gain;                                                                                                                      |      |      | -41  | dBm    |
| P <sub>O_NFM</sub>    | output power level (Near Field Mode) | maximum gain; <b>(Note 34)</b>                                                                                                     | -25  | -20  | -15  | dBm    |

**Note 28:** Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.4.1.

**Note 29:** Measured according to Bluetooth® Core Technical Specification document, version 4.0, volume 6, section 4.4. Published value is for  $n = \text{IXIT} = 4$ .  $\text{IXIT} = 5$  gives the same results,  $\text{IXIT} = 3$  gives results that are 5 dB lower.

**Note 30:** Measured according to Bluetooth® Core Technical Specification document, version 4.0, volume 6, section 4.2.

**Note 31:** Measured according to Bluetooth® Core Technical Specification document, version 4.0, volume 6, section 4.3. Due to limitations of the measurement equipment, levels of -5 dBm should be interpreted as > -5 dBm.

**Note 32:** Frequencies close to the ISM band can show slightly worse performance

**Note 33:** Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.2.3.

**Note 34:** To activate the "Near Field Mode", program address 0x50002230, bits 9:5 with the value 0x0. To stop NFC restore previous bitfield value.



## 39 Package information

### 39.1 MOISTURE SENSITIVITY LEVEL (MSL)

The MSL is an indicator for the maximum allowable time period (floor life time) in which a moisture sensitive plastic device, once removed from the dry bag, can be exposed to an environment with a maximum temperature of 30 °C and a maximum relative humidity of 60 % RH. before the solder reflow process.

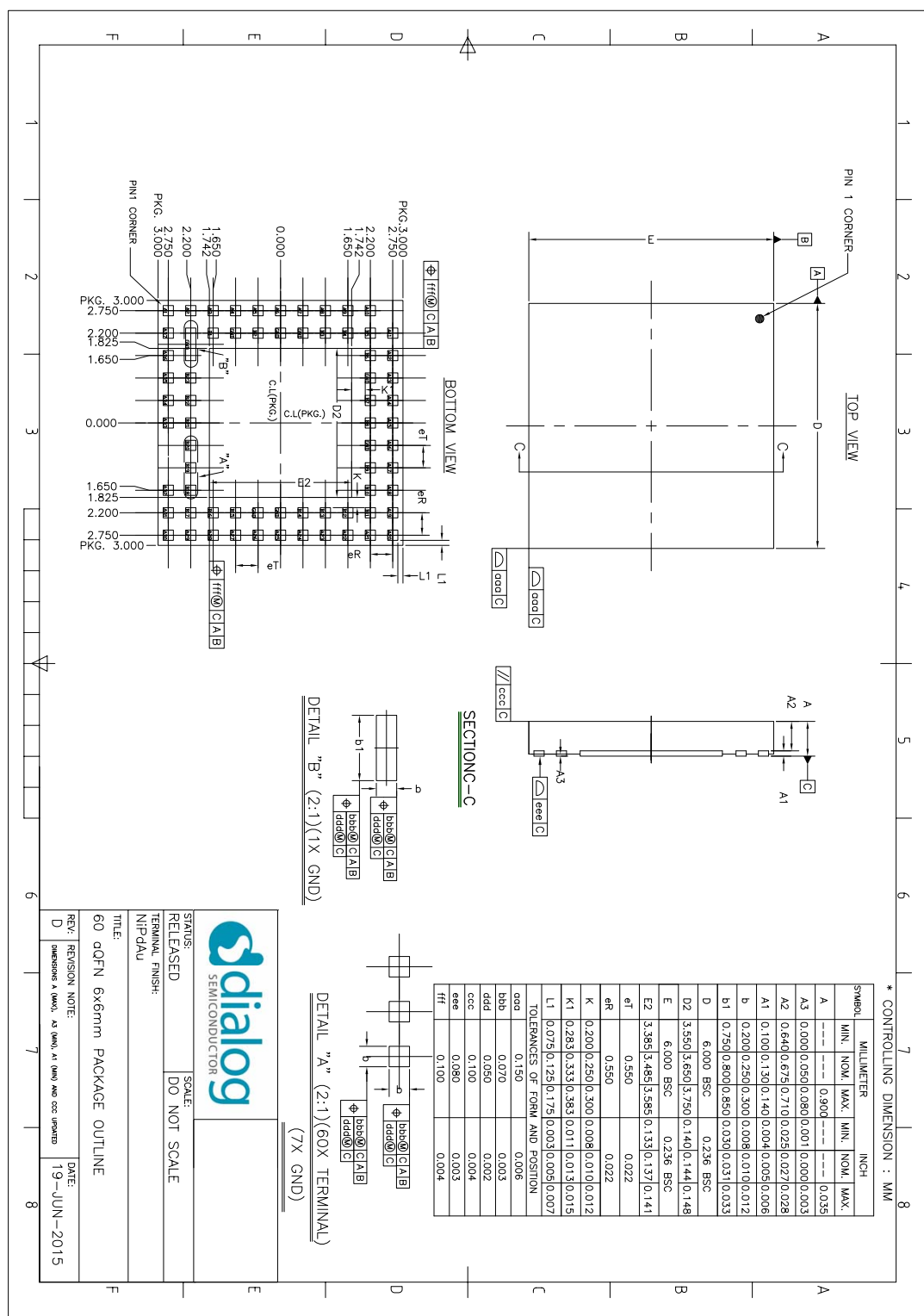
AQFN packages are qualified for MSL 3.

| MSL Level | Floor Life Time              |
|-----------|------------------------------|
| MSL 4     | 72 hours                     |
| MSL 3     | 168 hours                    |
| MSL 2A    | 4 weeks                      |
| MSL 2     | 1 year                       |
| MSL 1     | Unlimited at 30 °C / 85 % RH |

### 39.2 SOLDERING INFORMATION

Refer to the JEDEC standard J-STD-020 for relevant soldering information.

This document can be downloaded from <http://www.jedec.org>.



## Status definitions

| Version | Datasheet status | Product status | Definition                                                                                                                                                                                                                                                            |
|---------|------------------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.<n>   | Target           | Development    | This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.                                                                                                                                    |
| 2.<n>   | Preliminary      | Qualification  | This datasheet contains the specifications and preliminary characterisation data for products in pre-production. Specifications may be changed at any time without notice in order to improve the design.                                                             |
| 3.<n>   | Final            | Production     | This datasheet contains the final specifications for products in volume production. The specifications may be changed at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via Customer Product Notifications. |
| 4.<n>   | Obsolete         | Archived       | This datasheet contains the specifications for discontinued products. The information is provided for reference only.                                                                                                                                                 |

## Disclaimer

Information in this document is believed to be accurate and reliable. However, Dialog Semiconductor does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. Dialog Semiconductor furthermore takes no responsibility whatsoever for the content in this document if provided by any information source outside of Dialog Semiconductor.

Dialog Semiconductor reserves the right to change without notice the information published in this document, including without limitation the specification and design of the related semiconductor products, software and applications.

Applications, software, and semiconductor products described in this document are for illustrative purposes only. Dialog Semiconductor makes no representation or warranty that such applications, software and semiconductor products will be suitable for the specified use without further testing or modification. Unless otherwise agreed in writing, such testing or modification is the sole responsibility of the customer and Dialog Semiconductor excludes all liability in this respect.

Customer notes that nothing in this document may be construed as a license for customer to use the Dialog Semiconductor products, software and applications referred to in this document. Such license must be separately sought by customer with Dialog Semiconductor.

All use of Dialog Semiconductor products, software and applications referred to in this document are subject to Dialog Semiconductor's Standard Terms and Conditions of Sale, unless otherwise stated.

© Dialog Semiconductor. All rights reserved.

## RoHS compliance

Dialog Semiconductor complies to European Directive 2001/95/EC and from 2 January 2013 onwards to European Directive 2011/65/EU concerning Restriction of Hazardous Substances (RoHS/RoHS2). Dialog Semiconductor's statement on RoHS can be found on the customer portal <https://support.diasemi.com/>. RoHS certificates from our suppliers are available on request.

## Contacting Dialog Semiconductor

### United Kingdom (Headquarters)

Dialog Semiconductor (UK) LTD  
Phone: +44 1793 757700

### Germany

Dialog Semiconductor GmbH  
Phone: +49 7021 805-0

### The Netherlands

Dialog Semiconductor B.V.  
Phone: +31 73 640 8822

### Email:

[enquiry@diasemi.com](mailto:enquiry@diasemi.com)

### North America

Dialog Semiconductor Inc.  
Phone: +1 408 845 8500

### Japan

Dialog Semiconductor K. K.  
Phone: +81 3 5425 4567

### Taiwan

Dialog Semiconductor Taiwan  
Phone: +886 281 786 222

### Web site:

[www.dialog-semiconductor.com](http://www.dialog-semiconductor.com)

### Singapore

Dialog Semiconductor Singapore  
Phone: +65 64 8499 29

### Hong Kong

Dialog Semiconductor Hong Kong  
Phone: +852 3769 5200

### Korea

Dialog Semiconductor Korea  
Phone: +82 2 3469 8200

### China (Shenzhen)

Dialog Semiconductor China  
Phone: +86 755 2981 3669

### China (Shanghai)

Dialog Semiconductor China  
Phone: +86 21 5424 9058