

## General Description

The DA14580 integrated circuit has a fully integrated radio transceiver and baseband processor for *Bluetooth® low energy*. It can be used as a standalone application processor or as a data pump in hosted systems.

The DA14580 supports a flexible memory architecture for storing Bluetooth profiles and custom application code, which can be updated over the air (OTA). The qualified *Bluetooth low energy* protocol stack is stored in a dedicated ROM. All software runs on the ARM® Cortex®-M0 processor via a simple scheduler.

The *Bluetooth low energy* firmware includes the L2CAP service layer protocols, Security Manager (SM), Attribute Protocol (ATT), the Generic Attribute Profile (GATT) and the Generic Access Profile (GAP). All profiles published by the Bluetooth SIG as well as custom profiles are supported.

The transceiver interfaces directly to the antenna and is fully compliant with the *Bluetooth 4.2* standard.

The DA14580 has dedicated hardware for the Link Layer implementation of *Bluetooth low energy* and interface controllers for enhanced connectivity capabilities.

## Features

- Complies with *Bluetooth V4.2*, ETSI EN 300 328 and EN 300 440 Class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan)
- Processing power
  - 16 MHz 32 bit ARM Cortex-M0 with SWD interface
  - Dedicated Link Layer Processor
  - AES-128 bit encryption Processor
- Memories
  - 32 kB One-Time-Programmable (OTP) memory
  - 42 kB System SRAM
  - 84 kB ROM
  - 8 kB Retention SRAM
- Power management
  - Integrated Buck/Boost DC-DC converter
  - P0, P1, P2 and P3 ports with 3.3 V tolerance
  - Easy decoupling of only 4 supply pins
  - Supports coin (typ. 3.0 V) and alkaline (typ. 1.5 V) battery cells
  - 10-bit ADC for battery voltage measurement
- Digital controlled oscillators
  - 16 MHz crystal ( $\pm 20$  ppm max) and RC oscillator
  - 32 kHz crystal ( $\pm 50$  ppm,  $\pm 500$  ppm max) and RCX oscillator
- General purpose, Capture and Sleep timers
- Digital interfaces
  - General purpose I/Os: 14 (WLCSP34 package), 24 (QFN40 package), 32 (QFN48 package)
  - 2 UARTs with hardware flow control up to 1 MBd
  - SPI+™ interface
  - I2C bus at 100 kHz, 400 kHz
  - 3-axes capable Quadrature Decoder
- Analog interfaces
  - 4-channel 10-bit ADC
- Radio transceiver
  - Fully integrated 2.4 GHz CMOS transceiver
  - Single wire antenna: no RF matching or RX/TX switching required
  - Supply current at VBAT3V:
    - TX: 3.4 mA, RX: 3.7 mA (with ideal DC-DC)
  - 0 dBm transmit output power
  - -20 dBm output power in “Near Field Mode”
  - -93 dBm receiver sensitivity
- Packages:
  - WLCSP 34 pins, 2.436 mm x 2.436 mm
  - QFN 40 pins, 5 mm x 5 mm
  - QFN 48 pins, 6 mm x 6 mm
  - KGD (wafer, dice)

## System Diagram



## Contents

|  |            |
|--|------------|
| <b>General Description</b> .....           | <b>1</b>   |
| <b>Features</b> .....                      | <b>1</b>   |
| <b>System Diagram</b> .....                | <b>1</b>   |
| <b>Contents</b> .....                      | <b>2</b>   |
| <b>1 Block Diagram</b> .....               | <b>3</b>   |
| <b>2 Pinout</b> .....                      | <b>4</b>   |
| <b>3 Ordering Information</b> .....        | <b>8</b>   |
| <b>4 System Overview</b> .....             | <b>9</b>   |
| 4.1 ARM CORTEX-M0 CPU .....                | 9          |
| 4.2 BLUETOOTH SMART .....                  | 9          |
| 4.2.1 BLE Core .....                       | 9          |
| 4.2.2 Radio Transceiver .....              | 10         |
| 4.2.3 SmartSnippets™ .....                 | 10         |
| 4.3 MEMORIES .....                         | 11         |
| 4.4 FUNCTIONAL MODES .....                 | 11         |
| 4.5 POWER MODES .....                      | 12         |
| 4.6 INTERFACES .....                       | 12         |
| 4.6.1 UARTs .....                          | 12         |
| 4.6.2 SPI+ .....                           | 12         |
| 4.6.3 I2C Interface .....                  | 12         |
| 4.6.4 General Purpose ADC .....            | 13         |
| 4.6.5 Quadrature Decoder .....             | 13         |
| 4.6.6 Keyboard Controller .....            | 13         |
| 4.6.7 Input/Output Ports .....             | 13         |
| 4.7 TIMERS .....                           | 13         |
| 4.7.1 General Purpose Timers .....         | 13         |
| 4.7.2 Wake-Up timer .....                  | 14         |
| 4.7.3 Watchdog Timer .....                 | 14         |
| 4.8 CLOCK/RESET .....                      | 14         |
| 4.8.1 Clocks .....                         | 14         |
| 4.8.2 Reset .....                          | 15         |
| 4.9 POWER MANAGEMENT .....                 | 15         |
| <b>5 Registers</b> .....                   | <b>18</b>  |
| <b>6 Specifications</b> .....              | <b>139</b> |
| <b>7 Package Information</b> .....         | <b>151</b> |
| 7.1 MOISTURE SENSITIVITY LEVEL (MSL) ..... | 151        |
| 7.2 WLCSP HANDLING .....                   | 151        |
| 7.3 SOLDERING INFORMATION .....            | 151        |
| 7.4 PACKAGE OUTLINES .....                 | 152        |

1 Block Diagram

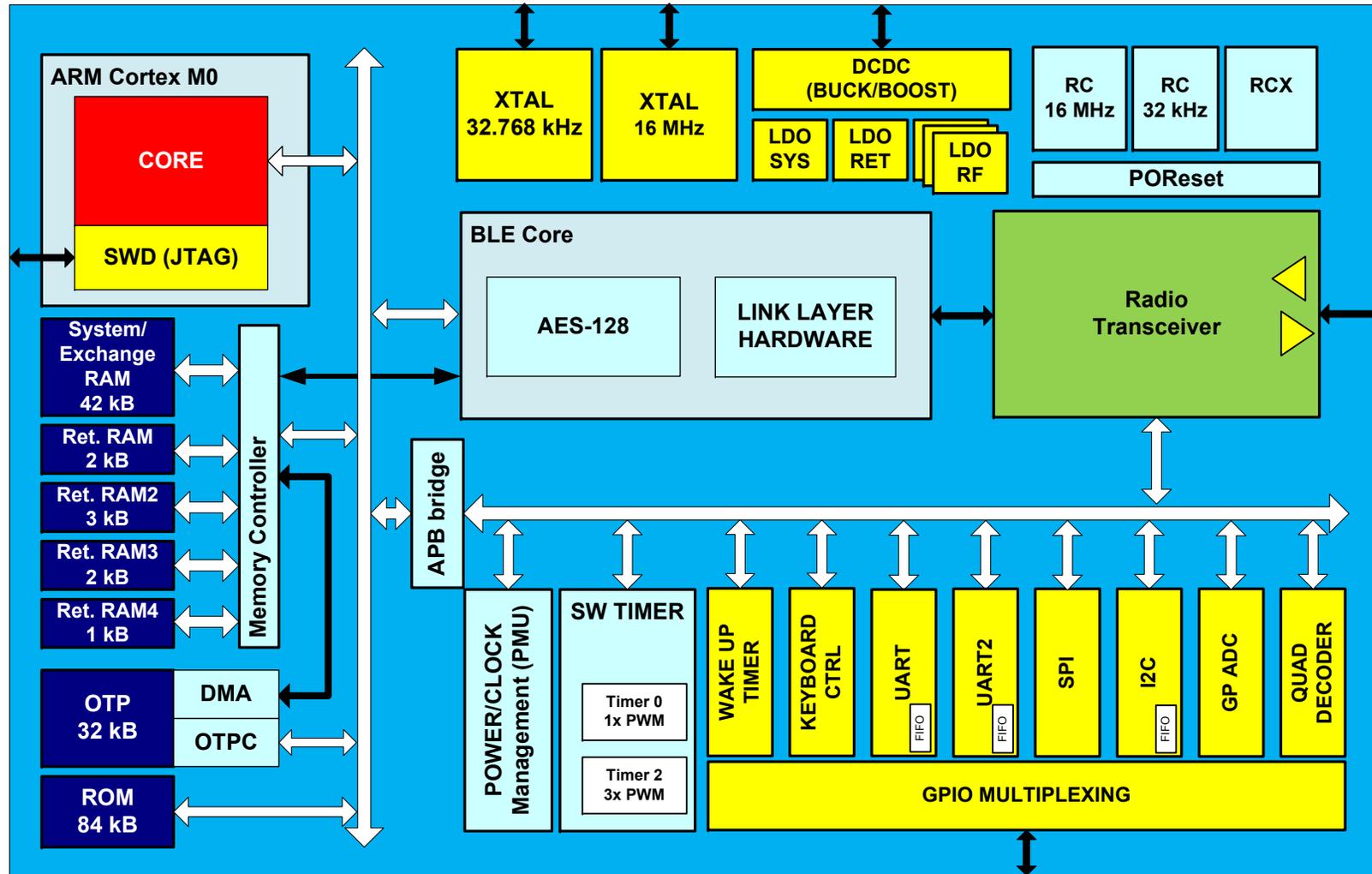


Figure 1: DA14580 Block Diagram

## 2 Pinout

The DA14580 comes in three packages:

- Wafer Level Chip Scale Package (WLCSP) with 34 balls

- Quad Flat Package No Leads (QFN) with 48 pins
- Quad Flat Package No Leads (QFN) with 40 pins

The actual pin/ball assignment is depicted in the following figures:

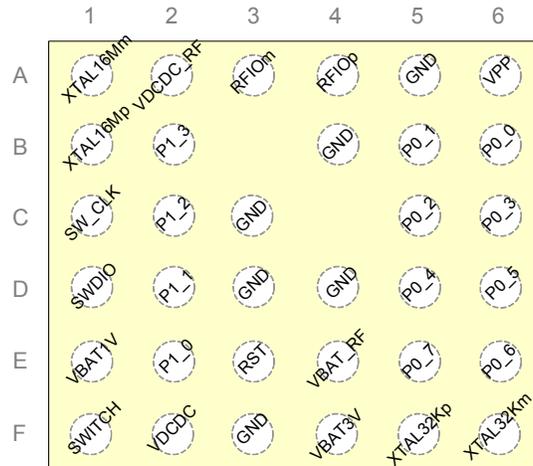


Figure 2: WLCSP34 ball assignment

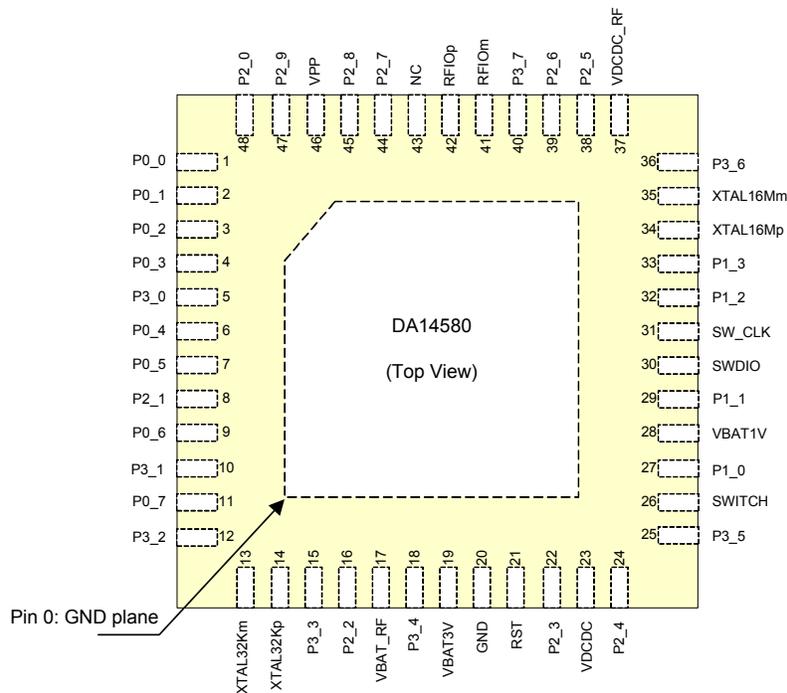


Figure 3: QFN48 Pin Assignment

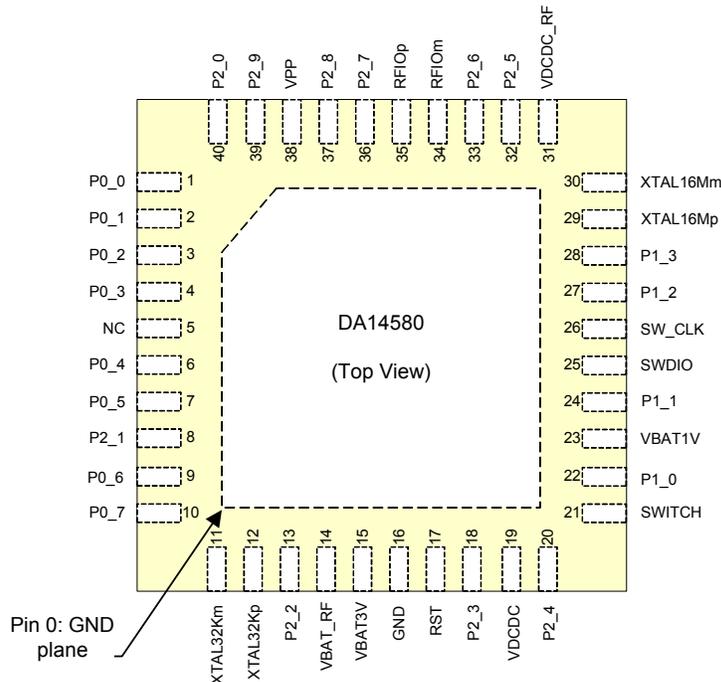


Figure 4: QFN40 Pin Assignment

Table 1: Pin Description

| Pin Name   | Type   | Drive (mA) | Reset State  | Description  |
|--|--|------------|--|--|
| <b>General Purpose I/Os</b>  |  |            |  |  |
| P0_0<br>P0_1<br>P0_2<br>P0_3<br>P0_4<br>P0_5<br>P0_6<br>P0_7                 | DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO               | 4.8        | I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD                 | INPUT/OUTPUT with selectable pull up/down resistor. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.  |
| P1_0<br>P1_1<br>P1_2<br>P1_3<br>P1_4/SWCLK<br>P1_5/SW_DIO                    | DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO                             | 4.8        | I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PU                                 | INPUT/OUTPUT with selectable pull up/down resistor. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.<br>This signal is the JTAG clock by default<br>This signal is the JTAG data I/O by default |
| P2_0<br>P2_1<br>P2_2<br>P2_3<br>P2_4<br>P2_5<br>P2_6<br>P2_7<br>P2_8<br>P2_9 | DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO<br>DIO | 4.8        | I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD<br>I-PD | INPUT/OUTPUT with selectable pull up/down resistor. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.<br>NOTE: This port is only available on the QFN40/QFN48 packages.                          |

Table 1: Pin Description

| Pin Name                  | Type     | Drive (mA) | Reset State | Description   |
|---------------------------|----------|------------|-------------|---|
| P3_0                      | DIO      | 4.8        | I-PD        | INPUT/OUTPUT with selectable pull up/down resistor. Pull-down enabled during and after reset. General purpose I/O port bit or alternate function nodes. Contain state retention mechanism during power down.<br>NOTE: This port is only available on the QFN48 package. |
| P3_1                      | DIO      |            | I-PD        |   |
| P3_2                      | DIO      |            | I-PD        |   |
| P3_3                      | DIO      |            | I-PD        |   |
| P3_4                      | DIO      |            | I-PD        |   |
| P3_5                      | DIO      |            | I-PD        |   |
| P3_6                      | DIO      |            | I-PD        |   |
| P3_7                      | DIO      |            | I-PD        |   |
| <b>Debug Interface</b>    |          |            |             |   |
| SWDIO/P1_5                | DIO      | 4.8        | I-PU        | INPUT/OUTPUT. JTAG Data input/output. Bidirectional data and control communication. Can also be used as a GPIO  |
| SW_CLK/P1_4               | DIO      | 4.8        | I-PD        | INPUT JTAG clock signal. Can also be used as a GPIO   |
| <b>Clocks</b>             |          |            |             |   |
| XTAL16Mp                  | AI       |            |             | INPUT. Crystal input for the 16 MHz XTAL  |
| XTAL16Mm                  | AO       |            |             | OUTPUT. Crystal output for the 16 MHz XTAL  |
| XTAL32kp                  | AI       |            |             | INPUT. Crystal input for the 32.768 kHz XTAL  |
| XTAL32km                  | AO       |            |             | OUTPUT. Crystal output for the 32.768 kHz XTAL  |
| <b>Quadrature Decoder</b> |          |            |             |   |
| QD_CHA_X                  | DI       |            |             | INPUT. Channel A for the X axis. Mapped on Px ports   |
| QD_CHB_X                  | DI       |            |             | INPUT. Channel B for the X axis. Mapped on Px ports   |
| QD_CHA_Y                  | DI       |            |             | INPUT. Channel A for the Y axis. Mapped on Px ports   |
| QD_CHB_Y                  | DI       |            |             | INPUT. Channel B for the Y axis. Mapped on Px ports   |
| QD_CHA_Z                  | DI       |            |             | INPUT. Channel A for the Z axis. Mapped on Px ports   |
| QD_CHB_Z                  | DI       |            |             | INPUT. Channel B for the Z axis. Mapped on Px ports   |
| <b>SPI Bus Interface</b>  |          |            |             |   |
| SPI_CLK                   | DO       |            |             | INPUT/OUTPUT. SPI Clock. Mapped on Px ports   |
| SPI_DI                    | DI       |            |             | INPUT. SPI Data input. Mapped on Px ports   |
| SPI_DO                    | DO       |            |             | OUTPUT. SPI Data output. Mapped on Px ports   |
| SPI_EN                    | DI       |            |             | INPUT. SPI Clock enable (active LOW). Mapped on Px ports  |
| <b>I2C Bus Interface</b>  |          |            |             |   |
| SDA                       | DIO/DIOD |            |             | INPUT/OUTPUT. I2C bus Data with open drain port. Mapped on Px ports   |
| SCL                       | DIO/DIOD |            |             | INPUT/OUTPUT. I2C bus Clock with open drain port. In open drain mode, SCL is monitored to support bit stretching by a slave. Mapped on Px ports.  |
| <b>UART Interface</b>     |          |            |             |   |
| UTX                       | DO       |            |             | OUTPUT. UART transmit data. Mapped on Px ports  |
| URX                       | DI       |            |             | INPUT. UART receive data. Mapped on Px ports  |
| URTS                      | DO       |            |             | OUTPUT. UART Request to Send. Mapped on Px ports  |
| UCTS                      | DI       |            |             | INPUT. UART Clear to Send. Mapped on Px ports   |
| UTX2                      | DO       |            |             | OUTPUT. UART 2 transmit data. Mapped on Px ports  |
| URX2                      | DI       |            |             | INPUT. UART 2 receive data. Mapped on Px ports  |
| URTS2                     | DO       |            |             | OUTPUT. UART 2 Request to Send. Mapped on Px ports  |
| UCTS2                     | DI       |            |             | INPUT. UART 2 Clear to Send. Mapped on Px ports   |

Table 1: Pin Description

| Pin Name                 | Type | Drive (mA) | Reset State | Description   |
|--------------------------|------|------------|-------------|---|
| <b>Analog Interface</b>  |      |            |             |   |
| ADC[0]                   | AI   |            |             | INPUT. Analog to Digital Converter input 0. Mapped on P0[0]   |
| ADC[1]                   | AI   |            |             | INPUT. Analog to Digital Converter input 1. Mapped on P0[1]   |
| ADC[2]                   | AI   |            |             | INPUT. Analog to Digital Converter input 2. Mapped on P0[2]   |
| ADC[3]                   | AI   |            |             | INPUT. Analog to Digital Converter input 3. Mapped on P0[3]   |
| <b>Radio Transceiver</b> |      |            |             |   |
| RFIOp                    | AIO  |            |             | RF input/output. Impedance 50 Ω.  |
| RFIOm                    | AIO  |            |             | RF ground   |
| <b>Miscellaneous</b>     |      |            |             |   |
| RST                      | DI   |            |             | INPUT. Reset signal (active high). Must be connected to GND if not used.  |
| VBAT_RF                  | AIO  |            |             | Connect to VBAT3V on the PCB  |
| VDCDC_RF                 | AIO  |            |             | Connect to VDCDC on the PCB   |
| VPP                      | AI   |            |             | INPUT. This pin is used while OTP programming and testing.<br>OTP programming: VPP = 6.7 V ± 0.1 V<br>OTP Normal operation: leave VPP floating  |
| <b>Power Supply</b>      |      |            |             |   |
| VBAT3V                   | AIO  |            |             | INPUT/OUTPUT. Battery connection. Used for a single coin battery (3 V). If an alkaline or a NiMH battery (1.5 V) is attached to pin VBAT1V, this is the second output of the DC-DC converter. |
| VBAT1V                   | AI   |            |             | INPUT. Battery connection. Used for an alkaline or a NiMH battery (1.5 V). If a single coin battery (3 V) is attached to pin VBAT3V, this pin must be connected to GND.                       |
| SWITCH                   | AIO  |            |             | INPUT/OUTPUT. Connection for the external DC-DC converter inductor.   |
| VDCDC                    | AO   |            |             | Output of the DC-DC converter   |
| GND                      | AIO  | -          | -           | Ground  |

### 3 Ordering Information

**Table 2: Ordering Information (Samples)**

| Part Number   | Package | Size (mm)     | Shipment Form | Pack Quantity |
|---------------|---------|---------------|---------------|---------------|
| DA14580-01UNA | WLCSP34 | 2.436 x 2.436 | Mini-reel     | 50/100/1000   |
| DA14580-01A31 | QFN48   | 6 x 6         | Tray          | 50            |
| DA14580-01AT1 | QFN40   | 5 x 5         | Tray          | 50            |

**Table 3: Ordering Information (Production)**

| Part Number   | Package | Size (mm)     | Shipment Form                             | Pack Quantity |
|---------------|---------|---------------|---|---------------|
| DA14580-01UNA | WLCSP34 | 2.436 x 2.436 | Mini-reel                                 | 5000          |
| DA14580-01A32 | QFN48   | 6 x 6         | Reel                                      | 4000          |
| DA14580-01AT2 | QFN40   | 5 x 5         | Reel                                      | 5000          |
| DA14580-01WO4 | KGD     | wafer         | Contact Dialog Semiconductor sales office |               |
| DA14580-01WC4 | KGD     | dice          | Contact Dialog Semiconductor sales office |               |

**Table 4: Ordering Information (Preprogrammed OTP)**

| Part Number     | Package | Shipment Form | Pack Quantity | Description                  |
|-----------------|---------|---------------|---------------|------------------------------|
| DA14580-01PxA31 | QFN48   | Tray          | 50            | Preprogrammed OTP, version x |
| DA14580-01PxAT1 | QFN40   | Tray          | 50            | Preprogrammed OTP, version x |
| DA14580-01PxUNA | WLCSP34 | Mini-reel     | 5000          | Preprogrammed OTP, version x |
| DA14580-01PxA32 | QFN48   | Reel          | 4000          | Preprogrammed OTP, version x |
| DA14580-01PxAT2 | QFN40   | Reel          | 4000          | Preprogrammed OTP, version x |

**Part Number Legend:**

DA14580-nn[ABC]XYZ

nn: chip revision number

A, AB or ABC: special version (optional)

XY: package code

Z: packing method



## 4 System Overview

The DA14580 contains the following internal blocks:

### 4.1 ARM CORTEX-M0 CPU

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.

#### 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 24 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.

### 4.2 BLUETOOTH SMART

#### 4.2.1 BLE Core

The BLE (Bluetooth low energy) core is a qualified Bluetooth baseband controller compatible with the Bluetooth low energy 4.2 specification and it is in charge of packet encoding/decoding and frame scheduling.

#### Features

- All device classes support (Broadcaster, Central, Observer, Peripheral)

- All packet types (Advertising / Data / Control)
- Encryption (AES / CCM)
- Bit stream processing (CRC, Whitening)
- FDMA/TDMA/events formatting and synchronization
- Frequency hopping calculation
- Operating clock 16 MHz or 8 MHz
- Low power modes supporting 32.0 kHz or 32.768 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

#### 4.2.2 Radio Transceiver

The Radio Transceiver implements the RF part of the Bluetooth low energy protocol. Together with the Bluetooth 4.2 PHY layer, this provides a 93 dB RF link budget for reliable wireless communication.

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 Bluetooth LE radio comprises the Receiver, Transmitter, Synthesizer, Rx/Tx combiner block, and Biasing LDOs.

#### Features

- Single ended RFIO interface, 50  $\Omega$  matched
- Alignment free operation
- -93 dBm receiver sensitivity
- 0 dBm transmit output power
- Ultra low power consumption
- Fast frequency tuning minimizes overhead

#### 4.2.3 SmartSnippets™

The DA14580 comes complete with Dialog's SmartSnippets™ Bluetooth Software platform which includes a qualified Bluetooth Smart single-mode stack on chip. Numerous Bluetooth Smart profiles for consumer wellness, sport, fitness, security and proximity applications are supplied as standard, while additional customer profiles can be developed and added as needed.

The SmartSnippets™ software development environment is based on Keil™'s uVision mature tools and contains example application code for both embedded and hosted modes.

### smart{snippets}

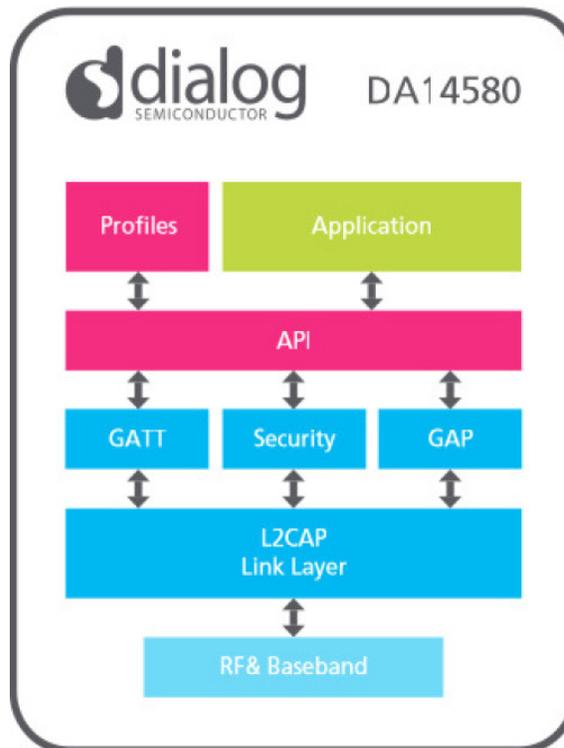


Figure 5: SmartSnippets Stack

Apart from the protocol stack, the Software platform supports a Hardware Abstraction Layer (HAL) which enables easy access to peripheral's features from a programmer's point of view, as presented in the following figure.

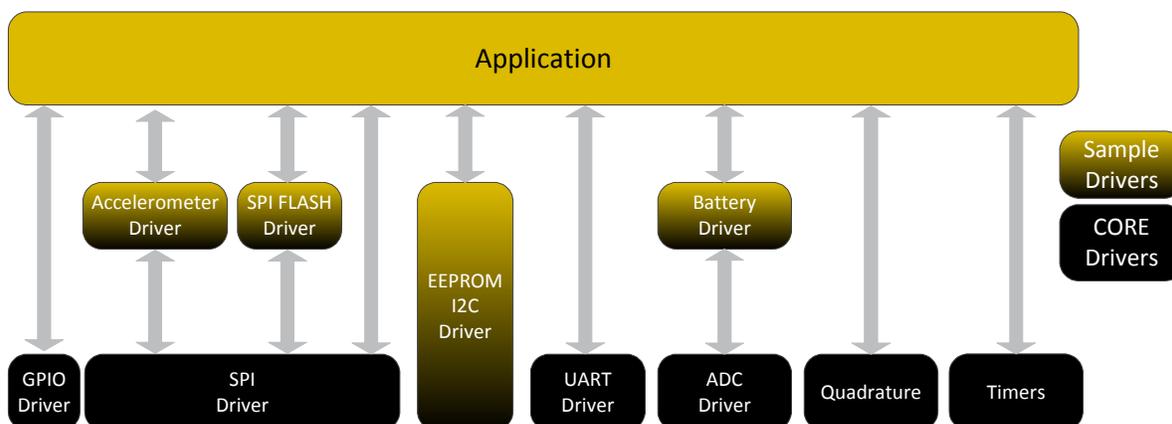


Figure 6: Hardware Abstraction Layer

Core drivers are provided for each interface of the DA14580 enabling optimized usage of the hardware's capabilities. These drivers provide an easy-to-use interface towards the hardware engines without having to interfere with the register programming directly.

On top of the core drivers, a number of sample drivers is also provided enabling communication with basic Bluetooth Smart application components: accelerometers, FLASH/EEPROM non-volatile memories, etc.

### 4.3 MEMORIES

The following memories are part of the DA14580's internal blocks:

**ROM.** This is a 84 kB ROM containing the Bluetooth low energy protocol stack as well as the boot code sequence.

**OTP.** This is a 32 kB One-Time Programmable memory array, used to store the application code as well as Bluetooth low energy profiles. It also contains the system configuration and calibration data.

**System SRAM.** This is a 42 kB system SRAM (System RAM) which is primarily used for mirroring the program code from the OTP when the system wakes/powers up. It also serves as Data RAM for intermediate variables and various data that the protocol requires. Optionally, it can be used as extra memory space for the BLE TX and RX data structures.

**Retention RAMs.** These are 4 special low leakage SRAM cells (2 kB + 2 kB + 3 kB + 1 kB) used to store various data of the Bluetooth low energy protocol as well as the system's global variables and processor stack when the system goes into Deep Sleep mode. Storage of this data ensures secure and quick configuration of the BLE Core after the system wakes up. Every cell can be powered on or off according to the application needs for retention area when in Deep

Sleep mode.

### 4.4 FUNCTIONAL MODES

The DA14580 is optimized for deeply 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 OTP. In general, the system has three functional modes of operation:

**A. Development Mode:** During this phase application code is developed using the ARM Cortex-M0 SW environment. The compiled code is then downloaded into the System RAM or any Retention RAMs by means of SWD (JTAG) or any serial interface (e.g. UART). Address 0x00 is remapped to the physical memory that contains the code and the CPU is configured to reset and execute code from the remapped device. This mode is enabling application development, debugging and on-the-fly testing.

**B. Normal Mode:** After the application is ready and verified, the code can be burned into the OTP. When the system boots/wakes up, the DMA of the OTP controller will automatically copy the program code from the OTP into the system RAM. Next, a SW reset or a jump to the System RAM occurs and code execution is started. Hence, in this mode, the system is autonomous, contains the required SW in OTP and is ready for integration into the final product.

**C. Calibration Mode:** Between Development and Normal mode, there is an intermediate stage where the chip needs to be calibrated with respect to two important features:

- Programming of the Bluetooth device address
- Programming of the trimming value for the external 16 MHz crystal.

This mode of operation applies to the final product and is performed by the customer. During this phase, certain fields in the OTP should be programmed.

#### 4.5 POWER MODES

There are four different power modes in the DA14580:

- *Active mode*: System is active and operates at full speed.
- *Sleep mode*: No power gating has been programmed, the ARM CPU is idle, waiting for an interrupt. PD\_SYS is on. PD\_PER and PED\_RAD depending on the programmed enabled value.
- *Extended Sleep mode*: All power domains are off except for the PD\_AON, the programmed PD\_RRx and the PD\_SR. Since the SysRAM retains its data, no OTP mirroring is required upon waking up the system.
- *Deep Sleep mode*: All power domains are off except for the PD\_AON and the programmed PD\_RRx. This mode dissipates the minimum leakage power. However, since the SysRAM has not retained its data, an OTP mirror action is required upon waking up the system.

#### 4.6 INTERFACES

##### 4.6.1 UARTs

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 no DMA support on the UART block since its contains internal FIFOs. Both UARTs support hardware flow control signals (RTS, CTS, DTR, DSR).

##### Features

- 16 bytes Transmit and receive FIFOs
- Hardware flow control support (CTS/RTS)
- 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: baud rate = (serial clock frequency)/(divisor).

##### 4.6.2 SPI+

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.

SPI is a trademark of Motorola, Inc.

##### Features

- Slave and Master mode
- 8 bit, 9 bit, 16 bit or 32 bit operation
- Clock speeds up to 16 MHz for the SPI controller. Programmable output frequencies of SPI source clock divided by 1, 2, 4, 8
- SPI clock line speed up to 8 MHz
- 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.

##### 4.6.3 I2C Interface

The I2C interface 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.

##### Features

- Two-wire I2C serial interface consists of a serial data line (SDA) and a serial clock (SCL)
- Two speeds are supported:
- Standard mode (0 to 100 kbit/s)
- Fast mode (<= 400 kbit/s)
- Clock synchronization
- 32 deep transmit/receive 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

#### 4.6.4 General Purpose ADC

The DA14580 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.

##### Features

- 10-bit dynamic ADC with 65 ns conversion time
- Maximum sampling rate 3.3 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
- Four single-ended or two differential external input channels
- Battery monitoring function
- Chopper function
- Offset and zero scale adjust
- Common-mode input level adjust

#### 4.6.5 Quadrature Decoder

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

The integrated quadrature decoder can automatically decode the signals for the X, Y and Z axes of a HID input device, reporting step count and direction: the channels are expected to provide a pulse train with 90 degrees phase difference; depending on whether the reference channel is leading or lagging, the direction can be determined.

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.

##### 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

#### 4.6.6 Keyboard Controller

The Keyboard controller can be used for debouncing the incoming GPIO signals when implementing a keyboard scanning engine. It generates an interrupt to the CPU (KEYBR\_IRQ).

In parallel, five extra interrupt lines can be triggered by a state change on 32 selectable GPIOs (GPIOx\_IRQ).

##### Features

- Monitors any of the 32 available GPIOs (12 in the WLCSP package, 22 in the QFN40 and 32 in the QFN48)
- Generates a keyboard interrupt on key press or key release
- Implements debouncing time from 0 up to 63 ms

Supports five separate interrupt generation lines from GPIO toggling

#### 4.6.7 Input/Output Ports

The DA14580 has software-configurable I/O pin assignment, organized into ports Port 0, Port1, Port2 and Port 3. Port 2 is only available at the QFN40 package while ports 2 and 3 are available at the QFN48 package.

##### Features

- Port 0: 8 pins, Port 1: 6 pins (including SW\_CLK and SWDIO), Port 2: 10 pins, Port 3: 8 pins
- Fully programmable pin assignment
- Selectable 25 k $\Omega$  pull-up, pull-down resistors per pin
- Pull-up voltage either VBAT3V (BUCK mode) or VBAT1V (BOOST mode) configurable per pin
- Fixed assignment for analog pin ADC[3:0]
- Pins retain their last state when system enters the Extended or Deep Sleep mode.

### 4.7 TIMERS

#### 4.7.1 General Purpose Timers

The Timer block contains 2 timer modules that are software controlled, programmable and can be used for various tasks.

##### Timer 0

- 16-bit general purpose timer
- Ability to generate 2 Pulse Width Modulated signals (PWM0 and PWM1, with common programming)
- Programmable output frequency:

$$f = \frac{(16, 8, 4, 2 \text{ MHz or } 32 \text{ kHz})}{(M + 1) + (N + 1)}$$

with N = 0 to (2<sup>16</sup>-1), M = 0 to (2<sup>16</sup>-1)

- Programmable duty cycle:

$$\delta = \frac{M+1}{(M+1)+(N+1)} \times 100 \%$$

- Separately programmable interrupt timer:

$$T = \frac{(16, 8, 4, 2 \text{ MHz or } 32 \text{ kHz})}{(ON+1)}$$

### Timer 2

- 14-bit general purpose timer
- Ability to generate 3 Pulse Width Modulated signals (PWM2, PWM3 and PWM4)
- Input clock frequency:

$$f_{IN} = \frac{\text{sys\_clk}}{N} \text{ with } N = 1, 2, 4 \text{ or } 8$$

and sys\_clk = 16 MHz or 32 kHz

- Programmable output frequency:

$$f_{OUT} = \left(\frac{f_{IN}}{2}\right) \text{ to } \left(\frac{f_{IN}}{2^{14}-1}\right)$$

- Three outputs with Programmable duty cycle from 0 % to 100 %
- Used for white LED intensity (on/off) control

### 4.7.2 Wake-Up timer

The Wake-up timer can be programmed to wake up the DA14580 from power down mode after a preprogrammed number of GPIO events.

#### Features

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

A minimum pulse duration of 2 sleep clock cycles must be applied to the GPIO to ensure a successful system wake-up.

### 4.7.3 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).

#### 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.

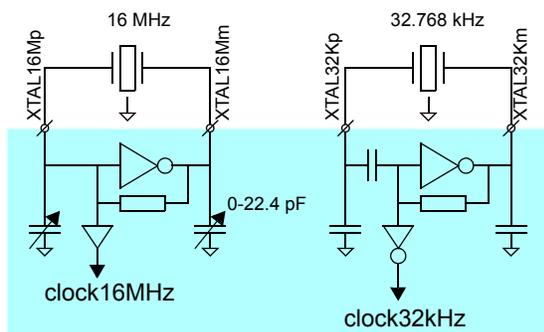
Maskable Watchdog freeze by user program. Note that if the system is not remapped, i.e. SysRAM is at address 0x20000000, then a watchdog fire will trigger the BootROM code to be executed again.

## 4.8 CLOCK/RESET

### 4.8.1 Clocks

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 7](#) below. No external components to the DA14580 are required other than the crystal itself. If the crystal has a case connection, it is advised to connect the case to ground.



**Figure 7: Crystal Oscillator Circuits**

There are 3 RC oscillators in the DA14580: one providing 16 MHz (RC16M), one providing 32 kHz (RC32K) and one providing a frequency in the range of 10.5 kHz (RCX).

#### 4.8.2 Reset

The DA14580 comprises an RST pad which is active high. It contains an RC filter for spikes suppression with 400 k $\Omega$  and 2.8 pF for the resistor and the capacitor respectively. It also contains a 25 k $\Omega$  pull-down resistor. This pad should be connected to ground if not needed by the application. The typical latency of the RST pad is in the range of 2  $\mu$ s.

### 4.9 POWER MANAGEMENT

The DA14580 has a complete power management function integrated with Buck or Boost DC-DC converter and separate LDOs for the different power domains of the system.

#### Features

- On-chip LDOs, without external capacitors
- Synchronous DC-DC converter which can be configured as either:
  - Boost (step-up) converter, starting from 0.9 V, when running from an Alkaline/NiMH cell.
  - Buck (step-down) converter for increased efficiency when running from a Lithium coin-cell or 2 Alkaline batteries down to 2.35 V.
- Battery voltage measurement ADC (multiplexed input from general purpose ADC)
- Use of small external components (2.2  $\mu$ H inductor and 1  $\mu$ F capacitor)

The Power Block contains a DC-DC converter which can be configured to operate as a Step-Up or a Step-Down converter. The converter provides power to four

LDO groups in the system:

1. LDO RET: This is the LDO providing power to the Retention domain (PD\_AON). It powers the Retention RAMs and the digital part which is always on.
2. LDO OTP: This is the LDO powering the OTP macro cell. This is the reason for using the step-up DC-DC converter when running from an Alkaline battery.
3. LDO SYS: This is the LDO providing the system with the actual VDD power required for the digital part to operate. Note that the Power Block implements seamless switching from the LDO SYS to the LDO RET when the system enters Deep Sleep mode. In the latter case, a low voltage is applied to the PD\_AON power domain to further reduce leakage.
4. LDO (various): This a group of LDOs used for the elaborate control of the powering up/down of the Radio, the GP ADC and the XTAL16M oscillator.

There are two ways of connecting external batteries to the Power Block of the DA14580. They depend on the specific battery cell used and its voltage range. Battery cells are distinguished into Lithium coin cells (2.35 V to 3.3 V) and Alkaline cells (1.0 V to 1.8 V). The connection diagrams are presented in [Figure 9](#) and [Figure 8](#) respectively:

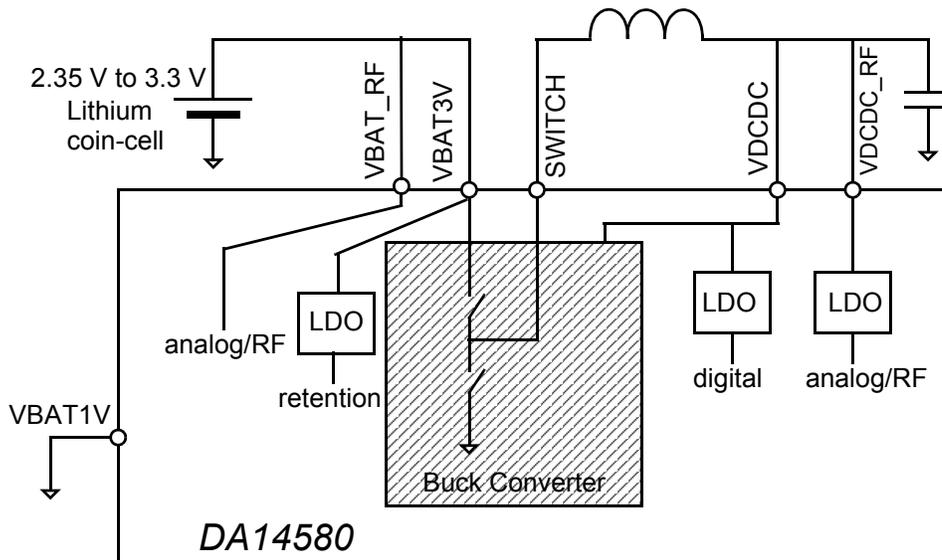


Figure 8: Supply Overview, Coin-Cell Application

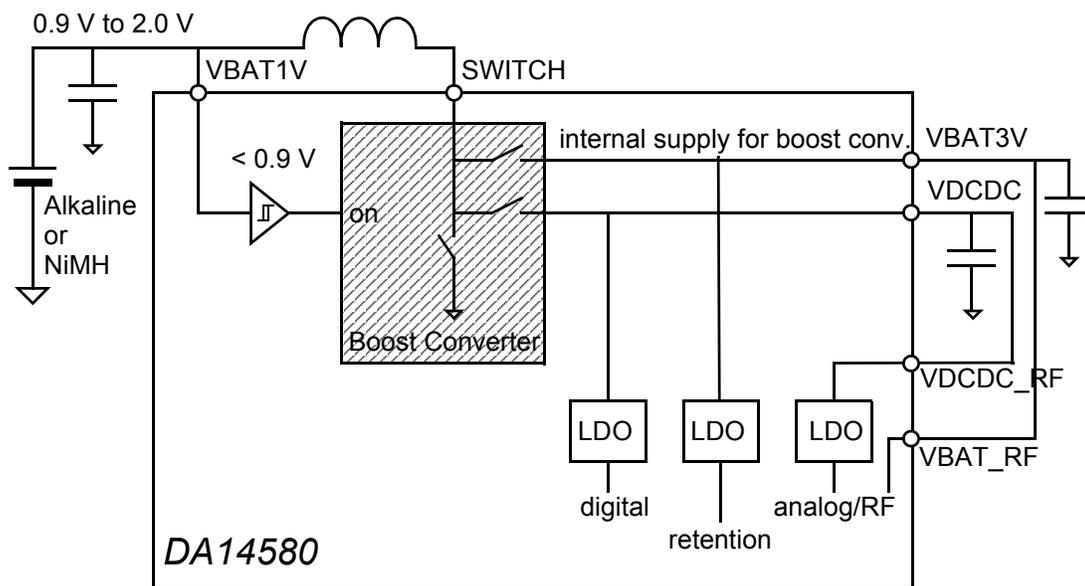
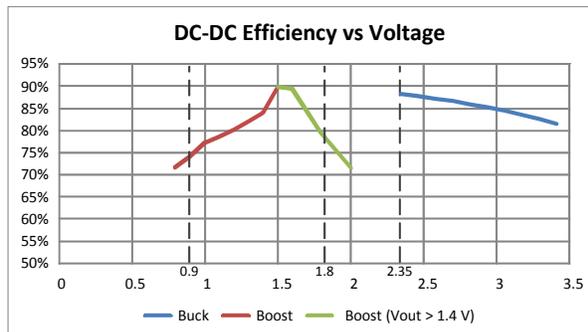


Figure 9: Supply Overview, Alkaline-Cell Application

The usage of Boost or Buck mode with respect to the provided voltage ranges is illustrated in the following figure which also illustrates the efficiency of the engine assuming a 10 mA constant load.





**Figure 10: DC-DC Efficiency in Buck/Boost Mode at Various Voltage Levels**

The X axis represents the supply voltage. BOOST mode should be used when voltage ranges from 0.9 V to 2.0 V to sustain a decent efficiency over 70 %. From that point on, the power dissipation becomes quite large.

BUCK mode can operate correctly with voltages in the range of 2.35 V to 3.3 V.

There are two voltage areas in [Figure 10](#) designated by dashed lines. The first one (0 V to 0.9 V) indicates that the DA14580 is not operational when the voltage is below 0.9 V. This is the absolute threshold for the DC-DC converter Boost mode.

The second area (1.8 V to 2.2 V) indicates that Deep Sleep mode is not allowed when the DC-DC converter is configured in BUCK mode and the voltage is within this range, because the OTP will not be readable any more. However, this part of the voltage range can be covered by the BOOST mode. Furthermore, when BUCK mode is mandatory, Extended Sleep mode can be activated instead of Deep Sleep mode, thus not using the OTP for the code mirroring but retain the code in SysRAM.

**Note:** The system should never be cold booted when the supply voltage is less than 2.5 V. A manual power up with a power supply less than 2.5 V in buck mode might create instability.

## 5 Registers

This section contains a detailed view of the DA14580 registers. It is organized as follows: An overview table is presented initially, which depicts all register names, addresses and descriptions. A detailed bit level description of each register follows.

The register file of the ARM Cortex-M0 can be found in the following documents, available on the ARM website:

**Devices Generic User Guide:**

DUI0497A\_cortex\_m0\_r0p0\_generic\_ug.pdf

**Technical Reference Manual:**

DDI0432C\_cortex\_m0\_r0p0\_trm.pdf

These documents contain the register descriptions for the Nested Vectored Interrupt Controller (NVIC), the System Control Block (SCB) and the System Timer (SysTick).

Table 5: Register Map

| Address    | Port                | Description   |
|------------|---------------------|---|
| 0x40008000 | OTPC_MODE_REG       | Mode register   |
| 0x40008004 | OTPC_PCTRL_REG      | Bit-programming control register                          |
| 0x40008008 | OTPC_STAT_REG       | Status register   |
| 0x4000800C | OTPC_AHBADR_REG     | AHB master start address                                  |
| 0x40008010 | OTPC_CELADR_REG     | Macrocell start address                                   |
| 0x40008014 | OTPC_NWORDS_REG     | Number of words   |
| 0x40008018 | OTPC_FFPRT_REG      | Ports access to fifo logic                                |
| 0x4000801C | OTPC_FFRD_REG       | Latest read data from the OTPC_FFPRT_REG                  |
| 0x50000000 | CLK_AMBA_REG        | HCLK, PCLK, divider and clock gates                       |
| 0x50000002 | CLK_FREQ_TRIM_REG   | Xtal frequency trimming register                          |
| 0x50000004 | CLK_PER_REG         | Peripheral divider register                               |
| 0x50000008 | CLK_RADIO_REG       | Radio PLL control register                                |
| 0x5000000A | CLK_CTRL_REG        | Clock control register                                    |
| 0x50000010 | PMU_CTRL_REG        | Power Management Unit control register                    |
| 0x50000012 | SYS_CTRL_REG        | System Control register                                   |
| 0x50000014 | SYS_STAT_REG        | System status register                                    |
| 0x50000016 | TRIM_CTRL_REG       | Control trimming of the XTAL16M                           |
| 0x50000020 | CLK_32K_REG         | 32 kHz oscillator register                                |
| 0x50000022 | CLK_16M_REG         | 16 MHz RC-oscillator register                             |
| 0x50000024 | CLK_RCX20K_REG      | 20 kHz RXC-oscillator control register                    |
| 0x50000028 | BANDGAP_REG         | Bandgap trimming  |
| 0x5000002A | ANA_STATUS_REG      | Status bit of analog (power management) circuits          |
| 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_POL_P0_REG     | Select the sensitivity polarity for each P0 input         |
| 0x50000114 | WKUP_POL_P1_REG     | Select the sensitivity polarity for each P1 input         |
| 0x50000116 | WKUP_POL_P2_REG     | Select the sensitivity polarity for each P2 input         |
| 0x50000118 | WKUP_POL_P3_REG     | Select the sensitivity polarity for each P3 input         |
| 0x50000200 | QDEC_CTRL_REG       | Quad Decoder control register                             |
| 0x50000202 | QDEC_XCNT_REG       | Counter value of the X Axis                               |
| 0x50000204 | QDEC_YCNT_REG       | Counter value of the Y Axis                               |
| 0x50000206 | QDEC_CLOCKDIV_REG   | Clock divider register                                    |
| 0x50000208 | QDEC_CTRL2_REG      | Quad Decoder control register                             |

Table 5: Register Map

| Address    | Port                  | Description   |
|------------|-----------------------|---|
| 0x5000020A | QDEC_ZCNT_REG         | Z_counter   |
| 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/FIFO Control Register |
| 0x5000100C | UART_LCR_REG          | Line Control Register                                   |
| 0x50001010 | UART_MCR_REG          | Modem Control Register                                  |
| 0x50001014 | UART_LSR_REG          | Line Status Register                                    |
| 0x50001018 | UART_MSR_REG          | Modem Status Register                                   |
| 0x5000101C | UART_SCR_REG          | Scratchpad Register                                     |
| 0x50001020 | UART_LPDLL_REG        | Low Power Divisor Latch Low                             |
| 0x50001024 | UART_LPDH_REG         | Low Power Divisor Latch High                            |
| 0x50001030 | UART_SRBR_STHR0_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001034 | UART_SRBR_STHR1_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001038 | UART_SRBR_STHR2_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x5000103C | UART_SRBR_STHR3_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001040 | UART_SRBR_STHR4_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001044 | UART_SRBR_STHR5_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001048 | UART_SRBR_STHR6_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x5000104C | UART_SRBR_STHR7_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001050 | UART_SRBR_STHR8_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001054 | UART_SRBR_STHR9_REG   | Shadow Receive/Transmit Buffer Register                 |
| 0x50001058 | UART_SRBR_STHR10_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x5000105C | UART_SRBR_STHR11_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001060 | UART_SRBR_STHR12_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001064 | UART_SRBR_STHR13_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x50001068 | UART_SRBR_STHR14_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x5000106C | UART_SRBR_STHR15_REG  | Shadow Receive/Transmit Buffer Register                 |
| 0x5000107C | UART_USR_REG          | UART Status register.                                   |
| 0x50001080 | UART_TFL_REG          | Transmit FIFO Level                                     |
| 0x50001084 | UART_RFL_REG          | Receive FIFO Level.                                     |
| 0x50001088 | UART_SRR_REG          | Software Reset Register.                                |
| 0x5000108C | UART_SRTS_REG         | Shadow Request to Send                                  |
| 0x50001090 | UART_SBCR_REG         | Shadow Break Control Register                           |
| 0x50001094 | UART_SDMAM_REG        | Shadow DMA Mode   |
| 0x50001098 | UART_SFE_REG          | Shadow FIFO Enable                                      |
| 0x5000109C | UART_SRT_REG          | Shadow RCVR Trigger                                     |
| 0x500010A0 | UART_STET_REG         | Shadow TX Empty Trigger                                 |
| 0x500010A4 | UART_HTX_REG          | Halt TX   |
| 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 |

Table 5: Register Map

| Address    | Port                  | Description                             |
|------------|-----------------------|---|
| 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                     |
| 0x50001120 | UART2_LPDLL_REG       | Low Power Divisor Latch Low             |
| 0x50001124 | UART2_LPDLH_REG       | Low Power Divisor Latch High            |
| 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 |
| 0x5000117C | UART2_USR_REG         | UART Status register.                   |
| 0x50001180 | UART2_TFL_REG         | Transmit FIFO Level                     |
| 0x50001184 | UART2_RFL_REG         | Receive FIFO Level.                     |
| 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                                 |
| 0x500011F4 | UART2_CPR_REG         | Component Parameter Register            |
| 0x500011F8 | UART2_UCV_REG         | Component Version                       |
| 0x500011FC | UART2_CTR_REG         | Component Type Register                 |
| 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 | I2C_CON_REG           | I2C Control Register                    |
| 0x50001304 | I2C_TAR_REG           | I2C Target Address Register             |

Table 5: Register Map

| Address    | Port                     | Description   |
|------------|--------------------------|---|
| 0x50001308 | I2C_SAR_REG              | I2C Slave Address Register                          |
| 0x50001310 | I2C_DATA_CMD_REG         | I2C Rx/Tx Data Buffer and Command Register          |
| 0x50001314 | I2C_SS_SCL_HCNT_REG      | Standard Speed I2C Clock SCL High Count Register    |
| 0x50001318 | I2C_SS_SCL_LCNT_REG      | Standard Speed I2C Clock SCL Low Count Register     |
| 0x5000131C | I2C_FS_SCL_HCNT_REG      | Fast Speed I2C Clock SCL High Count Register        |
| 0x50001320 | I2C_FS_SCL_LCNT_REG      | Fast Speed I2C Clock SCL Low Count Register         |
| 0x5000132C | I2C_INTR_STAT_REG        | I2C Interrupt Status Register                       |
| 0x50001330 | I2C_INTR_MASK_REG        | I2C Interrupt Mask Register                         |
| 0x50001334 | I2C_RAW_INTR_STAT_REG    | I2C Raw Interrupt Status Register                   |
| 0x50001338 | I2C_RX_TL_REG            | I2C Receive FIFO Threshold Register                 |
| 0x5000133C | I2C_TX_TL_REG            | I2C Transmit FIFO Threshold Register                |
| 0x50001340 | I2C_CLR_INTR_REG         | Clear Combined and Individual Interrupt Register    |
| 0x50001344 | I2C_CLR_RX_UNDER_REG     | Clear RX_UNDER Interrupt Register                   |
| 0x50001348 | I2C_CLR_RX_OVER_REG      | Clear RX_OVER Interrupt Register                    |
| 0x5000134C | I2C_CLR_TX_OVER_REG      | Clear TX_OVER Interrupt Register                    |
| 0x50001350 | I2C_CLR_RD_REQ_REG       | Clear RD_REQ Interrupt Register                     |
| 0x50001354 | I2C_CLR_TX_ABRT_REG      | Clear TX_ABRT Interrupt Register                    |
| 0x50001358 | I2C_CLR_RX_DONE_REG      | Clear RX_DONE Interrupt Register                    |
| 0x5000135C | I2C_CLR_ACTIVITY_REG     | Clear ACTIVITY Interrupt Register                   |
| 0x50001360 | I2C_CLR_STOP_DET_REG     | Clear STOP_DET Interrupt Register                   |
| 0x50001364 | I2C_CLR_START_DET_REG    | Clear START_DET Interrupt Register                  |
| 0x50001368 | I2C_CLR_GEN_CALL_REG     | Clear GEN_CALL Interrupt Register                   |
| 0x5000136C | I2C_ENABLE_REG           | I2C Enable Register                                 |
| 0x50001370 | I2C_STATUS_REG           | I2C Status Register                                 |
| 0x50001374 | I2C_TXFLR_REG            | I2C Transmit FIFO Level Register                    |
| 0x50001378 | I2C_RXFLR_REG            | I2C Receive FIFO Level Register                     |
| 0x5000137C | I2C_SDA_HOLD_REG         | I2C SDA Hold Time Length Register                   |
| 0x50001380 | I2C_TX_ABRT_SOURCE_REG   | I2C Transmit Abort Source Register                  |
| 0x50001394 | I2C_SDA_SETUP_REG        | I2C SDA Setup Register                              |
| 0x50001398 | I2C_ACK_GENERAL_CALL_REG | I2C ACK General Call Register                       |
| 0x5000139C | I2C_ENABLE_STATUS_REG    | I2C Enable Status Register                          |
| 0x500013A0 | I2C_IC_FS_SPKLEN_REG     | I2C SS and FS spike suppression limit Size          |
| 0x50001400 | GPIO_IRQ0_IN_SEL_REG     | GPIO interrupt selection for GPIO_IRQ0              |
| 0x50001402 | GPIO_IRQ1_IN_SEL_REG     | GPIO interrupt selection for GPIO_IRQ1              |
| 0x50001404 | GPIO_IRQ2_IN_SEL_REG     | GPIO interrupt selection for GPIO_IRQ2              |
| 0x50001406 | GPIO_IRQ3_IN_SEL_REG     | GPIO interrupt selection for GPIO_IRQ3              |
| 0x50001408 | GPIO_IRQ4_IN_SEL_REG     | GPIO interrupt selection for GPIO_IRQ4              |
| 0x5000140C | GPIO_DEBOUNCE_REG        | debounce counter value for GPIO inputs              |
| 0x5000140E | GPIO_RESET_IRQ_REG       | GPIO interrupt reset register                       |
| 0x50001410 | GPIO_INT_LEVEL_CTRL_REG  | high or low level select for GPIO interrupts        |
| 0x50001412 | KBRD_IRQ_IN_SEL0_REG     | GPIO interrupt selection for KBRD_IRQ for P0        |
| 0x50001414 | KBRD_IRQ_IN_SEL1_REG     | GPIO interrupt selection for KBRD_IRQ for P1 and P2 |
| 0x50001416 | KBRD_IRQ_IN_SEL2_REG     | GPIO interrupt selection for KBRD_IRQ for P3        |
| 0x50001500 | GP_ADC_CTRL_REG          | General Purpose ADC Control Register                |

Table 5: Register Map

| Address    | Port                 | Description                                  |
|------------|----------------------|--|
| 0x50001502 | GP_ADC_CTRL2_REG     | General Purpose ADC Second Control Register  |
| 0x50001504 | GP_ADC_OFFP_REG      | General Purpose ADC Positive Offset Register |
| 0x50001506 | GP_ADC_OFFN_REG      | General Purpose ADC Negative Offset Register |
| 0x50001508 | GP_ADC_CLEAR_INT_REG | General Purpose ADC Clear Interrupt Register |
| 0x5000150A | GP_ADC_RESULT_REG    | General Purpose ADC Result Register          |
| 0x5000150C | GP_ADC_DELAY_REG     | General Purpose ADC Delay Register           |
| 0x5000150E | GP_ADC_DELAY2_REG    | General Purpose ADC Second Delay Register    |
| 0x50001600 | CLK_REF_SEL_REG      | Select clock for oscillator calibration      |
| 0x50001602 | CLK_REF_CNT_REG      | Count value for oscillator calibration       |
| 0x50001604 | CLK_REF_VAL_L_REG    | XTAL16M reference cycles, lower 16 bits      |
| 0x50001606 | CLK_REF_VAL_H_REG    | XTAL16M reference cycles, upper 16 bits      |
| 0x50003000 | P0_DATA_REG          | P0 Data input / output register              |
| 0x50003002 | P0_SET_DATA_REG      | P0 Set port pins register                    |
| 0x50003004 | P0_RESET_DATA_REG    | P0 Reset port pins register                  |
| 0x50003006 | P00_MODE_REG         | P00 Mode Register                            |
| 0x50003008 | P01_MODE_REG         | P01 Mode Register                            |
| 0x5000300A | P02_MODE_REG         | P02 Mode Register                            |
| 0x5000300C | P03_MODE_REG         | P03 Mode Register                            |
| 0x5000300E | P04_MODE_REG         | P04 Mode Register                            |
| 0x50003010 | P05_MODE_REG         | P05 Mode Register                            |
| 0x50003012 | P06_MODE_REG         | P06 Mode Register                            |
| 0x50003014 | P07_MODE_REG         | P07 Mode Register                            |
| 0x50003020 | P1_DATA_REG          | P1 Data input / output register              |
| 0x50003022 | P1_SET_DATA_REG      | P1 Set port pins register                    |
| 0x50003024 | P1_RESET_DATA_REG    | P1 Reset port pins register                  |
| 0x50003026 | P10_MODE_REG         | P10 Mode Register                            |
| 0x50003028 | P11_MODE_REG         | P11 Mode Register                            |
| 0x5000302A | P12_MODE_REG         | P12 Mode Register                            |
| 0x5000302C | P13_MODE_REG         | P13 Mode Register                            |
| 0x5000302E | P14_MODE_REG         | P14 Mode Register                            |
| 0x50003030 | P15_MODE_REG         | P15 Mode Register                            |
| 0x50003040 | P2_DATA_REG          | P2 Data input / output register              |
| 0x50003042 | P2_SET_DATA_REG      | P2 Set port pins register                    |
| 0x50003044 | P2_RESET_DATA_REG    | P2 Reset port pins register                  |
| 0x50003046 | P20_MODE_REG         | P20 Mode Register                            |
| 0x50003048 | P21_MODE_REG         | P21 Mode Register                            |
| 0x5000304A | P22_MODE_REG         | P22 Mode Register                            |
| 0x5000304C | P23_MODE_REG         | P23 Mode Register                            |
| 0x5000304E | P24_MODE_REG         | P24 Mode Register                            |
| 0x50003050 | P25_MODE_REG         | P25 Mode Register                            |
| 0x50003052 | P26_MODE_REG         | P26 Mode Register                            |
| 0x50003054 | P27_MODE_REG         | P27 Mode Register                            |
| 0x50003056 | P28_MODE_REG         | P28 Mode Register                            |
| 0x50003058 | P29_MODE_REG         | P29 Mode Register                            |

Table 5: Register Map

| Address    | Port                 | Description                                     |
|------------|----------------------|---|
| 0x50003070 | P01_PADPWR_CTRL_REG  | Ports 0 and 1 Output Power Control Register     |
| 0x50003072 | P2_PADPWR_CTRL_REG   | Port 2 Output Power Control Register            |
| 0x50003074 | P3_PADPWR_CTRL_REG   | Port 3 Output Power Control Register            |
| 0x50003080 | P3_DATA_REG          | P3 Data input / output register                 |
| 0x50003082 | P3_SET_DATA_REG      | P3 Set port pins register                       |
| 0x50003084 | P3_RESET_DATA_REG    | P3 Reset port pins register                     |
| 0x50003086 | P30_MODE_REG         | P30 Mode Register                               |
| 0x50003088 | P31_MODE_REG         | P31 Mode Register                               |
| 0x5000308A | P32_MODE_REG         | P32 Mode Register                               |
| 0x5000308C | P33_MODE_REG         | P33 Mode Register                               |
| 0x5000308E | P34_MODE_REG         | P34 Mode Register                               |
| 0x50003090 | P35_MODE_REG         | P35 Mode Register                               |
| 0x50003092 | P36_MODE_REG         | P36 Mode Register                               |
| 0x50003094 | P37_MODE_REG         | P37 Mode Register                               |
| 0x50003100 | WATCHDOG_REG         | Watchdog timer register.                        |
| 0x50003102 | WATCHDOG_CTRL_REG    | Watchdog control register.                      |
| 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.                         |
| 0x50003300 | SET_FREEZE_REG       | Controls freezing of various timers/counters.   |
| 0x50003302 | RESET_FREEZE_REG     | Controls unfreezing of various timers/counters. |
| 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.        |
| 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_DUTY_CYCLE      | Duty Cycle for PWM2                             |
| 0x5000340A | PWM3_DUTY_CYCLE      | Duty Cycle for PWM3                             |
| 0x5000340C | PWM4_DUTY_CYCLE      | Duty Cycle for PWM4                             |
| 0x5000340E | TRIPLE_PWM_FREQUENCY | Frequency for PWM 2,3 and 4                     |
| 0x50003410 | TRIPLE_PWM_CTRL_REG  | PWM 2 3 4 Control                               |

Table 6: OTPC\_MODE\_REG (0x40008000)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 31:30 | -    | -      | Reserved    | 0x0   |



Table 6: OTPC\_MODE\_REG (0x40008000)

| Bit   | Mode | Symbol                 | Description  | Reset |
|-------|------|------------------------|--|-------|
| 29:28 | R/W  | OTPC_MODE_PRG_PORT_MUX | Selects the source that is connected to the prg_port port of the controller.<br>00 - {16'd0, BANDGAP_REG[15:0]}<br>01 - {RF_RSSI_COMP_CTRL_REG[15:0], 8'd0, RFIO_CTRL1_REG[7:0]}<br>10 - {3'd0, RF_LNA_CTRL3_REG[4:0], RF_LNA_CTRL2_REG[11:0], RF_LNA_CTRL1_REG[11:0]}<br>11 - {28'd0, RF_VCO_CTRL_REG[3:0]}<br><br>See OTPC_MODE_PRG_PORT_SEL about the use of the prg_port   | 0x0   |
| 27:9  | -    | -                      | Reserved   | 0x0   |
| 8     | R/W  | OTPC_MODE_PRG_FAST     | Defines the timing that will be used for all the programming activities (APROG, MPROG and TWR)<br>0 - Selects the normal timing<br>1 - Selects the fast timing   | 0     |
| 7     | R/W  | OTPC_MODE_PRG_PORT_SEL | Selects an alternative data source for the programming of the OTP macrocells, when the controller is configured in APROG mode.<br>0 - The fifo will be used as the data source. The fifo will be filled with a way defined by the register OTPC_MODE_USE_DMA. The number of words that will be programmed is defined by OTPC_NWORDS.<br>1 - Only one word will programmed. The value of the word is contained in the prg_port port of the controller. The values of the registers OTPC_MODE_USE_DMA, OTPC_NWORDS and the contents of the FIFO will not be used.  | 0x0   |
| 6     | R/W  | OTPC_MODE_TWO_CC_ACC   | Defines the duration of each read from the OTP macrocells.<br>0 - Reads 16 bits of data every one clock cycle.<br>1 - Reads 16 bits of data every two clock cycles.  | 0x0   |
| 5     | R/W  | OTPC_MODE_FIFO_FLUSH   | Writing 1, removes any content from the FIFO. This bit returns automatically to 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 - DMA is not used. The data should be transferred from/to controller through OTPC_FFPRT_REG<br>1 - DMA is used. Data transfers from/to controller are performed automatically. The AHB base address should be configured in OTPC_AHBADR_REG before the selection of the mode.<br><br>If programming of the OTPC_MODE_REG is performed through the serial interface, the OTPC_MODE_USE_DMA will be set to 0 automatically.<br>If the controller is in APROG mode and the OTPC_MODE_PRG_PORT_SEL is enabled, the dma will stay inactive. | 0x0   |
| 3     | -    | -                      | Reserved   | 0x0   |

Table 6: OTPC\_MODE\_REG (0x40008000)

| Bit | Mode | Symbol         | Description   | Reset |
|-----|------|----------------|---|-------|
| 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 - Test mode. Reserved<br>110 - Test mode. Reserved<br>111 - Test mode. Reserved<br>To manually move between modes, always return to STBY mode first. | 0x0   |

Table 7: OTPC\_PCTRL\_REG (0x40008004)

| Bit   | Mode | Symbol           | Description  | Reset |
|-------|------|------------------|--|-------|
| 31:28 | -    | -                | Reserved   | 0x0   |
| 27    | R/W  | OTPC_PCTRL_ENU   | Enables the programming in the upper bank of the OTP.<br>0 - Programming sequence is not applied in the upper bank.<br>1 - Programming sequence is applied in the upper bank.  | 0x0   |
| 26    | R/W  | OTPC_PCTRL_BITU  | Defines the value of the selected bit in the upper bank, after the programming sequence.   | 0x0   |
| 25    | R/W  | OTPC_PCTRL_ENL   | Enables the programming in the lower bank.<br>0 - The programming sequence is not applied in the lower bank.<br>1 - The programming sequence is applied in the lower bank.   | 0x0   |
| 24    | R/W  | OTPC_PCTRL_BITL  | Defines the value of the selected bit in the lower bank, after the programming sequence.   | 0x0   |
| 23    | R/W  | OTPC_PCTRL_BSELU | Selects between the U1 and U0 byte for the programming sequence in the upper bank.<br>0 - Program the U0 byte<br>1 - Program the U1 byte   | 0x0   |
| 22:20 | R/W  | OTPC_PCTRL_BADRU | Selects the bit inside the Ux (x=0,1) byte, which will be programmed in the upper bank.  | 0x0   |
| 19    | R/W  | OTPC_PCTRL_BSELL | Selects between the L1 and L0 byte for the programming sequence in the lower bank.<br>0 - Program the L0 byte<br>1 - Program the L1 byte   | 0x0   |
| 18:16 | R/W  | OTPC_PCTRL_BADRL | Selects the bit inside the Lx (x=0,1) byte, which will be programmed in the lower bank.  | 0x0   |
| 15:13 | -    | -                | Reserved   | 0x0   |
| 12:0  | R/W  | OTPC_PCTRL_WADDR | Defines the address of a 32 bits word {U1,L1,U0,L0} in the macrocells, where one or two bits will be programmed. There are two macrocell banks, with 8 bits each. Each bank contribute with two memory positions for each 32 bits word. The Ux, Lx represent the bytes of the upper and lower bank respectively. | 0x0   |

Table 8: OTPC\_STAT\_REG (0x40008008)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 31:29 | -    | -      | Reserved    | 0x0   |

Table 8: OTPC\_STAT\_REG (0x40008008)

| Bit   | Mode | Symbol           | Description  | Reset |
|-------|------|------------------|--|-------|
| 28:16 | R    | OTPC_STAT_NWORS  | Contains the current value of the words to be processed.   | 0     |
| 15    | R    | OTPC_STAT_TERR_U | Indicates the upper bank as the source of a test error. This value is valid when OTPC_STAT_TERROR is valid.<br>0 - There is no test error in the upper bank<br>1 - A test error has occurred in the upper bank   | 0x0   |
| 14    | R    | OTPC_STAT_TERR_L | Indicates the lower bank as the source of a test error. The value is valid when OTPC_STAT_TERROR is valid.<br>0 - There is no test error in the lower bank<br>1 - A test error has occurred in the lower bank  | 0x0   |
| 13    | R    | OTPC_STAT_PERR_U | Indicates the upper bank as the source of a programming error. The value is valid when OTPC_STAT_PERROR is valid.<br>0 - There is no programming error in the upper bank<br>1 - A programming error has occurred in the upper bank   | 0x0   |
| 12    | R    | OTPC_STAT_PERR_L | Indicates the lower bank as the source of a programming error. The value is valid when OTPC_STAT_PERROR is valid.<br>0 - There is no programming error in the lower bank<br>1 - A programming error has occurred in the lower bank   | 0x0   |
| 11:8  | R    | OTPC_STAT_FWORS  | Indicates the number of words which contained in the fifo of the controller.   | 0x0   |
| 7:5   | -    | -                | Reserved   | 0x0   |
| 4     | R    | OTPC_STAT_ARDY   | Monitors the progress of read or programming operations while in the AREAD or APROG modes.<br>0 - The controller is busy while reading or programming (AREAD or APROG modes).<br>1 - The controller is not busy in AREAD or APROG mode.  | 0x1   |
| 3     | 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   |
| 2     | 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. A test mode is in progress.<br>1 - There is no active test mode.  | 0x1   |
| 1     | R    | OTPC_STAT_PERROR | Indicates that an error has occurred during the bit-programming process.<br>0 - No error during the bit-programming process.<br>1 - The process of bit-programming failed.<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 is normal to change periodically. Upon finishing the operation in the APROG mode (OTPC_STAT_ARDY= 1), this field indicates if the programming has failed or ended successfully. | 0x0   |
| 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 9: OTPC\_AHBADR\_REG (0x4000800C)

| Bit  | Mode | Symbol      | Description   | Reset |
|------|------|-------------|---|-------|
| 31:2 | R/W  | OTPC_AHBADR | The AHB address used by the AHB master interface of the controller (bits [31:2]). | 0x0   |
| 1:0  | -    | -           | Reserved  | 0x0   |

Table 10: OTPC\_CELADR\_REG (0x40008010)

| Bit   | Mode | Symbol      | Description  | Reset |
|-------|------|-------------|--|-------|
| 31:13 | -    | -           | Reserved   | 0x0   |
| 12:0  | R/W  | OTPC_CELADR | Defines a word address inside the macrocell. Used in modes AREAD and APROG and is automatically updated. | 0x0   |

Table 11: OTPC\_NWORDS\_REG (0x40008014)

| Bit   | Mode | Symbol      | Description  | Reset |
|-------|------|-------------|--|-------|
| 31:13 | -    | -           | Reserved   | 0x0   |
| 12:0  | R/W  | OTPC_NWORDS | The number of words (minus one) for reading/programming during the AREAD/APROG mode.<br>If in APROG mode, and the OTPC_MODE_PRG_PORT_SEL is enabled (=1), this register will not be used and will stay unchanged.<br>During mirroring, this register reflects the current amount of data that will be copied. It keeps its value until be written by the software with a new value. The number of the words that remaining to be processed by the controller is contained in the field OTPC_STAT_NWORDS. | 0x0   |

Table 12: OTPC\_FFPRT\_REG (0x40008018)

| Bit  | Mode | Symbol     | Description   | Reset |
|------|------|------------|---|-------|
| 31:0 | R/W  | OTPC_FFPRT | Provides access to the fifo through an access port. Write 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.<br>Check OTPC_STAT_FWORDS register for data/space availability, before accessing the fifo. | 0x0   |

Table 13: OTPC\_FFRD\_REG (0x4000801C)

| Bit  | Mode | Symbol    | Description   | Reset |
|------|------|-----------|---|-------|
| 31:0 | R    | OTPC_FFRD | Contains the value read from the fifo, after a read of the OTPC_FFPRT_REG register. | 0x0   |

Table 14: CLK\_AMBA\_REG (0x50000000)

| Bit  | Mode | Symbol     | Description                     | Reset |
|------|------|------------|---------------------------------|-------|
| 15:8 | -    | -          | Reserved                        | 0x0   |
| 7    | R/W  | OTP_ENABLE | Clock enable for OTP controller | 0x0   |
| 6    | -    | -          | Reserved                        | 0x0   |

Table 14: CLK\_AMBA\_REG (0x50000000)

| Bit | Mode | Symbol   | Description   | Reset |
|-----|------|----------|---|-------|
| 5:4 | R/W  | PCLK_DIV | APB interface clock (PCLK). Divider is cascaded with HCLK_DIV. PCLK is HCLK divided by:<br>0x0: divide by 1<br>0x1: divide by 2<br>0x2: divide by 4<br>0x3: divide by 8 | 0x2   |
| 3:2 | -    | -        | Reserved  | 0x0   |
| 1:0 | R/W  | HCLK_DIV | AHB interface and microprocessor clock (HCLK). HCLK is source clock divided by:<br>0x0: divide by 1<br>0x1: divide by 2<br>0x2: divide by 4<br>0x3: divide by 8         | 0x2   |

Table 15: CLK\_FREQ\_TRIM\_REG (0x50000002)

| Bit   | Mode | Symbol     | Description   | Reset |
|-------|------|------------|---|-------|
| 15:11 | -    | -          | Reserved  | 0x0   |
| 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 200 us 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 16: CLK\_PER\_REG (0x50000004)

| Bit   | Mode | Symbol          | Description  | Reset |
|-------|------|-----------------|--|-------|
| 15    | R/W  | QUAD_ENABLE     | Enable the Quadrature clock  | 0x0   |
| 14:12 | -    | -               | Reserved   | 0x0   |
| 11    | R/W  | SPI_ENABLE      | Enable SPI clock   | 0x0   |
| 10    | -    | -               | Reserved   | 0x0   |
| 9:8   | R/W  | SPI_DIV         | Division factor for SPI<br>0x0: divide by 1<br>0x1: divide by 2<br>0x2: divide by 4<br>0x3: divide by 8    | 0x0   |
| 7     | R/W  | UART1_ENABLE    | Enable UART1 clock   | 0x0   |
| 6     | R/W  | UART2_ENABLE    | Enable UART2 clock   | 0x0   |
| 5     | R/W  | I2C_ENABLE      | Enable I2C clock   | 0x0   |
| 4     | R/W  | WAKEUPCT_ENABLE | Enable Wakeup Capture Timer clock  | 0x0   |
| 3     | R/W  | TMR_ENABLE      | Enable TIMER0 and TIMER2 clock   | 0x0   |
| 2     | -    | -               | Reserved   | 0x0   |
| 1:0   | R/W  | TMR_DIV         | Division factor for TIMER0<br>0x0: divide by 1<br>0x1: divide by 2<br>0x2: divide by 4<br>0x3: divide by 8 | 0x0   |

Table 17: CLK\_RADIO\_REG (0x50000008)

| Bit  | Mode | Symbol       | Description  | Reset |
|------|------|--------------|--|-------|
| 15:8 | -    | -            | Reserved   | 0x0   |
| 7    | R/W  | BLE_ENABLE   | Enable the BLE core clocks   | 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<br>0x0: divide by 1<br>0x1: divide by 2<br>0x2: divide by 4<br>0x3: divide by 8<br>The programmed frequency should not be lower than 8 MHz 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   |
| 2    | -    | -            | Reserved   | 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>The programmed frequency must be exactly 8 MHz.   | 0x0   |

Table 18: CLK\_CTRL\_REG (0x5000000A)

| Bit  | Mode | Symbol                    | Description  | Reset |
|------|------|---------------------------|--|-------|
| 15:8 | -    | -                         | Reserved   | 0x0   |
| 7    | R    | RUNNING_AT_XTAL16M        | Indicates that the XTAL16M clock is used as clock, and may not be switched off   | 0x1   |
| 6    | R    | RUNNING_AT_RC16M          | Indicates that the RC16M clock is used as clock  | 0x0   |
| 5    | R    | RUNNING_AT_32K            | Indicates that either the RC32k or XTAL32k is being used as clock  | 0x0   |
| 4    | -    | -                         | Reserved   | 0x0   |
| 3    | R/W  | XTAL16M_SPIKE_FLT_DISABLE | Disable spikefilter in digital clock   | 0x0   |
| 2    | R/W  | XTAL16M_DISABLE           | Setting this bit instantaneously disables the 16 MHz crystal oscillator. Also, after sleep/wakeup cycle, the oscillator will not be enabled. This bit may not be set to '1' when "RUNNING_AT_XTAL16M is '1' to prevent deadlock. After resetting this bit, wait for XTAL16_SETTLED or 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_SETTLED and XTAL16_TRIM_READY bits!!)<br>0x1: RC16M<br>0x2/0x3: either RC32k or XTAL32k is used  | 0x0   |

Table 19: PMU\_CTRL\_REG (0x50000010)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:12 | -    | -      | Reserved    | 0x0   |

Table 19: PMU\_CTRL\_REG (0x50000010)

| Bit  | Mode | Symbol          | Description  | Reset |
|------|------|-----------------|--|-------|
| 11:8 | R/W  | RETENTION_MODE  | Select the retainability of the 4 retention RAM macros.<br>'1' is retainable, '0' is power gated.<br>(3) is RETRAM4<br>(2) is RETRAM3<br>(1) is RETRAM2<br>(0) is RETRAM1  | 0x0   |
| 7    | R/W  | FORCE_BOOST     | Force the DCDC into boost mode at next wakeup.<br>Setting this bit reduces the deepsleep current.<br>FORCE_BOOST has highest priority.<br>When either FORCE_BOOST or FORCE_BUCK have been written, these bits cannot be changed. | 0x0   |
| 6    | R/W  | FORCE_BUCK      | Force the DCDC into buck mode at next wakeup.<br>Setting this bit reduces the deepsleep current.<br>FORCE_BOOST has highest priority.<br>When either FORCE_BOOST or FORCE_BUCK have been written, these bits cannot be changed.  | 0x0   |
| 5:4  | R/W  | OTP_COPY_DIV    | Sets the HCLK division during OTP mirroring  | 0x0   |
| 2    | R/W  | RADIO_SLEEP     | Put the digital part of the radio in powerdown   | 0x1   |
| 1    | R/W  | PERIPH_SLEEP    | Put all peripherals (I2C, UART, SPI, ADC) in powerdown   | 0x1   |
| 0    | R/W  | RESET_ON_WAKEUP | Perform a Hardware Reset after waking up. Booter will be started.  | 0x0   |

Table 20: SYS\_CTRL\_REG (0x50000012)

| Bit | Mode | Symbol          | Description  | Reset |
|-----|------|-----------------|--|-------|
| 15  | W    | SW_RESET        | Writing a '1' to this bit will reset the device, except for:<br>SYS_CTRL_REG<br>CLK_FREQ_TRIM_REG<br>...   | 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   |
| 8   | -    | -               | Reserved   | 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   | R/W  | OTP_COPY        | Enables OTP to SysRAM copy action after waking up<br>PD_SYS  | 0x0   |
| 3   | R/W  | CLK32_SOURCE    | Sets the clock source of the 32 kHz clock<br>0 = RC-oscillator<br>1 = 32 kHz crystal oscillator  | 0x0   |

Table 20: SYS\_CTRL\_REG (0x50000012)

| Bit | Mode | Symbol     | Description   | Reset |
|-----|------|------------|---|-------|
| 2   | R/W  | RET_SYSRAM | Sets the development phase mode.<br>The PD_SYS is not actually power gated (SysRAM is retained).<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   |
| 1:0 | R/W  | REMAP_ADR0 | Controls which memory is located at address 0x0000 for execution.<br>0x0: ROM<br>0x1: OTP<br>0x2: SysRAM<br>0x3: RetRAM   | 0x0   |

Table 21: SYS\_STAT\_REG (0x50000014)

| Bit  | Mode | Symbol            | Description  | Reset |
|------|------|-------------------|--|-------|
| 15:8 | -    | -                 | Reserved   | 0x0   |
| 7    | R    | XTAL16_SETTLED    | Indicates that XTAL16 has had > 2 ms of settle time  | 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_UP         | Indicates that PD_DBG is functional  | 0x0   |
| 4    | R    | DBG_IS_DOWN       | Indicates that PD_DBG is in power down   | 0x1   |
| 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 22: TRIM\_CTRL\_REG (0x50000016)

| Bit | Mode | Symbol      | Description  | Reset |
|-----|------|-------------|--|-------|
| 7:4 | R/W  | TRIM_TIME   | Defines the delay between XTAL16M enable and applying the CLK_FREQ_TRIM_REG in steps of 250 us.<br>0x0: apply directly<br>0x1: wait between 0 and 250 us<br>0x2: wait between 250 us and 500 us<br>etc.<br><b>(Note 1)</b> | 0xA   |
| 3:0 | R/W  | SETTLE_TIME | Defines the delay between applying CLK_FREQ_TRIM_REG and XTAL16_SETTLED in steps of 250 us.<br>0x0: XTAL16_SETTLED is set directly<br>0x1: wait between 0 and 250 us<br>0x2: wait between 250 us and 500 us<br>etc.        | 0x2   |

**Note 1:** The period duration of 250 us is derived by dividing the RC16M clock signal by 4000. Consequently, the period duration may vary over temperature.

Table 23: CLK\_32K\_REG (0x50000020)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:13 | -    | -      | Reserved    | 0x0   |



Table 23: 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>Set this bit to '1' for an external clock applied at XTAL32Kp. Keep this bit '0' with a crystal between XTAL32Kp and XTAL32Km. | 0x0   |
| 11:8 | R/W  | RC32K_TRIM             | Controls the frequency of the RC32K oscillator.<br>0x0: lowest frequency<br>0x7: default<br>0xF: highest frequency  | 0x7   |
| 7    | R/W  | RC32K_ENABLE           | Enables the 32 kHz RC oscillator  | 0x1   |
| 6:3  | R/W  | XTAL32K_CUR            | Bias current for the 32kHz XTAL oscillator.<br>0x0: minimum<br>0x3: default<br>0xF: maximum<br>For each application there is an optimal setting for which the startup behavior is optimal.                        | 0x3   |
| 2:1  | R/W  | XTAL32K_RBIAS          | Setting for the bias resistor of the 32 kHz XTAL oscillator.<br>0x0: maximum<br>0x3: minimum<br>Preferred setting will be provided by Dialog.   | 0x2   |
| 0    | R/W  | XTAL32K_ENABLE         | Enables the 32 kHz XTAL oscillator  | 0x0   |

Table 24: CLK\_16M\_REG (0x50000022)

| Bit   | Mode | Symbol                         | Description  | Reset |
|-------|------|--------------------------------|--|-------|
| 15:10 | -    | -                              | Reserved   | 0x0   |
| 9     | R/W  | XTAL16_NOISE_FILTER_ENABLE     | Enables noise filter in 16 MHz crystal oscillator  | 0x0   |
| 8     | R/W  | XTAL16_BIAS_SAMPLE_HOLD_ENABLE | Enables Ibias sample/hold function in 16 MHz crystal oscillator. This bit should be set when the system wake up and reset before entering deep or extended sleep mode. | 0x0   |
| 7:5   | R/W  | XTAL16_CUR_SET                 | Bias current for the 16 MHz XTAL oscillator.<br>0x0: minimum<br>0x7: maximum   | 0x5   |
| 4:1   | R/W  | RC16M_TRIM                     | Controls the frequency of the RC16M oscillator.<br>0x0: lowest frequency<br>0xF: highest frequency   | 0x0   |
| 0     | R/W  | RC16M_ENABLE                   | Enables the 16 MHz RC oscillator   | 0x0   |

Table 25: CLK\_RCX20K\_REG (0x50000024)

| Bit | Mode | Symbol        | Description  | Reset |
|-----|------|---------------|--|-------|
| 12  | R/W  | RCX20K_SELECT | Selects RCX oscillator.<br>0 : RC32K oscillator<br>1: RCX oscillator | 0     |
| 11  | R/W  | RCX20K_ENABLE | Enable the RCX 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     |

Table 25: CLK\_RCX20K\_REG (0x50000024)

| Bit | Mode | Symbol      | Description  | Reset |
|-----|------|-------------|--|-------|
| 3:0 | R/W  | RCX20K_TRIM | Controls the frequency of the RCX oscillator.<br>0x0: lowest frequency<br>0x7: default<br>0xF: highest frequency | 8     |

Table 26: BANDGAP\_REG (0x50000028)

| Bit   | Mode | Symbol       | Description   | Reset |
|-------|------|--------------|---|-------|
| 15    | -    | -            | Reserved  | 0x0   |
| 14    | R/W  | BGR_LOWPOWER | Test-mode, do not use.<br>It disables the bandgap core (voltages will continue for some time, but will slowly drift away) | 0x0   |
| 13:10 | R/W  | LDO_RET_TRIM | (Note 2)  | 0x0   |
| 9:5   | R/W  | BGR_ITRIM    | Current trimming for bias   | 0x0   |
| 4:0   | R/W  | BGR_TRIM     | Trim register for bandgap   | 0x0   |

**Note 2:** 0xF is the lowest voltage, but is too low for reliable startup at high temperature in combination with extended sleep. 0xA is 100 mV higher and considered to be the lowest value which is safe to use. 0x0 or 0x1 is again 100 mV higher and 0x0 is the reset value. 0x4 is the maximum voltage.

Table 27: ANA\_STATUS\_REG (0x5000002A)

| Bit   | Mode | Symbol           | Description  | Reset |
|-------|------|------------------|--|-------|
| 15:10 | -    | -                | Reserved   | 0x0   |
| 9     | R    | BOOST_SELECTED   | Indicates that DCDC is in boost mode   | 0x0   |
| 8     | -    | -                | Reserved   | 0x0   |
| 7     | R    | BANDGAP_OK       | Indicates that BANDGAP is OK   | 0x1   |
| 6     | R    | BOOST_VBAT_OK    | Indicates that VBAT is above threshold while in BOOST converter mode.                      | 0x0   |
| 5     | R    | LDO_ANA_OK       | Indicates that LDO_ANA is in regulation. This LDO is used for the general-purpose ADC only | 0x0   |
| 4     | R    | LDO_VDD_OK       | Indicates that LDO_VDD is in regulation  | 0x1   |
| 3     | R    | LDO_OTP_OK       | Indicates that LDO_OTP is in regulation  | 0x0   |
| 2     | R    | VDCDC_OK         | Indicates that VDCDC is above threshold.   | 0x0   |
| 1     | R    | VBAT1V_OK        | Indicates that VBAT1V is above threshold.  | 0x0   |
| 0     | R    | VBAT1V_AVAILABLE | Indicates that VBAT1V is available.  | 0x0   |

Table 28: WKUP\_CTRL\_REG (0x50000100)

| Bit   | Mode | Symbol          | Description  | Reset |
|-------|------|-----------------|--|-------|
| 15:14 | -    | -               | 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>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 (N*1 ms with N = 1 to 63).<br>0x0: no debouncing<br>0x1 to 0x3F: 1 ms to 63 ms debounce time  | 0x0   |

Table 29: 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 30: 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. reading always returns 0. | 0x0   |

Table 31: 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 32: 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 33: 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 34: WKUP\_SELECT\_P1\_REG (0x5000010C)

| Bit | Mode | Symbol         | Description  | Reset |
|-----|------|----------------|--|-------|
| 5:0 | 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 35: WKUP\_SELECT\_P2\_REG (0x5000010E)

| Bit | Mode | Symbol         | Description  | Reset |
|-----|------|----------------|--|-------|
| 9:0 | 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 36: 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 37: WKUP\_POL\_P0\_REG (0x50000112)

| 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 38: WKUP\_POL\_P1\_REG (0x50000114)

| Bit | Mode | Symbol      | Description   | Reset |
|-----|------|-------------|---|-------|
| 5:0 | 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 39: WKUP\_POL\_P2\_REG (0x50000116)

| Bit | Mode | Symbol      | Description   | Reset |
|-----|------|-------------|---|-------|
| 9:0 | 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 40: WKUP\_POL\_P3\_REG (0x50000118)

| 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 41: QDEC\_CTRL\_REG (0x50000200)

| Bit   | Mode | Symbol        | Description  | Reset |
|-------|------|---------------|--|-------|
| 15:10 | -    | -             | Reserved   | 0x0   |
| 9:3   | R/W  | QD_IRQ_THRES  | The number of events on either counter (X or Y) 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 42: QDEC\_XCNT\_REG (0x50000202)

| Bit  | Mode | Symbol    | Description  | Reset |
|------|------|-----------|--|-------|
| 15:0 | R    | X_COUNTER | Contains a signed value of the events. Zero when channel is disabled | 0x0   |

Table 43: QDEC\_YCNT\_REG (0x50000204)

| Bit  | Mode | Symbol    | Description  | Reset |
|------|------|-----------|--|-------|
| 15:0 | R    | Y_COUNTER | Contains a signed value of the events. Zero when channel is disabled | 0x0   |

Table 44: QDEC\_CLOCKDIV\_REG (0x50000206)

| 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   |

Table 45: QDEC\_CTRL2\_REG (0x50000208)

| Bit   | Mode | Symbol       | Description   | Reset |
|-------|------|--------------|---|-------|
| 15:12 | -    | -            | Reserved  | 0     |
| 11:8  | R/W  | CHZ_PORT_SEL | Defines which GPIOs are mapped on Channel Z<br>0: none<br>1: P0[0] -> CHZ_A, P0[1] -> CHZ_B<br>2: P0[2] -> CHZ_A, P0[3] -> CHZ_B<br>3: P0[4] -> CHZ_A, P0[5] -> CHZ_B<br>4: P0[6] -> CHZ_A, P0[7] -> CHZ_B<br>5: P1[0] -> CHZ_A, P1[1] -> CHZ_B<br>6: P1[2] -> CHZ_A, P1[3] -> CHZ_B<br>7: P2[3] -> CHZ_A, P2[4] -> CHZ_B<br>8: P2[5] -> CHZ_A, P2[6] -> CHZ_B<br>9: P2[7] -> CHZ_A, P2[8] -> CHZ_B<br>10: P2[9] -> CHZ_A, P2[0] -> CHZ_B<br>11..15: None | 0     |
| 7:4   | R/W  | CHY_PORT_SEL | Defines which GPIOs are mapped on Channel Y<br>0: none<br>1: P0[0] -> CHY_A, P0[1] -> CHY_B<br>2: P0[2] -> CHY_A, P0[3] -> CHY_B<br>3: P0[4] -> CHY_A, P0[5] -> CHY_B<br>4: P0[6] -> CHY_A, P0[7] -> CHY_B<br>5: P1[0] -> CHY_A, P1[1] -> CHY_B<br>6: P1[2] -> CHY_A, P1[3] -> CHY_B<br>7: P2[3] -> CHY_A, P2[4] -> CHY_B<br>8: P2[5] -> CHY_A, P2[6] -> CHY_B<br>9: P2[7] -> CHY_A, P2[8] -> CHY_B<br>10: P2[9] -> CHY_A, P2[0] -> CHY_B<br>11..15: None | 0     |
| 3:0   | R/W  | CHX_PORT_SEL | Defines which GPIOs are mapped on Channel X<br>0: none<br>1: P0[0] -> CHX_A, P0[1] -> CHX_B<br>2: P0[2] -> CHX_A, P0[3] -> CHX_B<br>3: P0[4] -> CHX_A, P0[5] -> CHX_B<br>4: P0[6] -> CHX_A, P0[7] -> CHX_B<br>5: P1[0] -> CHX_A, P1[1] -> CHX_B<br>6: P1[2] -> CHX_A, P1[3] -> CHX_B<br>7: P2[3] -> CHX_A, P2[4] -> CHX_B<br>8: P2[5] -> CHX_A, P2[6] -> CHX_B<br>9: P2[7] -> CHX_A, P2[8] -> CHX_B<br>10: P2[9] -> CHX_A, P2[0] -> CHX_B<br>11..15: None | 0     |

Table 46: QDEC\_ZCNT\_REG (0x5000020A)

| Bit  | Mode | Symbol    | Description  | Reset |
|------|------|-----------|--|-------|
| 15:0 | R    | Z_COUNTER | Contains a signed value of the events. Zero when channel is disabled | 0     |

Table 47: UART\_RBR\_THR\_DLL\_REG (0x50001000)

| Bit  | Mode | Symbol      | Description   | Reset |
|------|------|-------------|---|-------|
| 15:8 | -    | -           | Reserved  | 0x0   |
| 7:0  | R/W  | RBR_THR_DLL | <p>Receive Buffer Register: 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. Transmit Holding Register: 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. Divisor Latch (Low): 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: <math>\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> | 0x0   |

Table 48: 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   |

Table 48: UART\_IER\_DLH\_REG (0x50001004)

| Bit | Mode | Symbol     | Description   | Reset |
|-----|------|------------|---|-------|
| 3   | R/W  | EDSSI_DLH3 | Interrupt Enable Register: EDSSI, Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[3] of the 8 bit DLH register  | 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 49: UART\_IIR\_FCR\_REG (0x50001008)

| 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. 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: 0000 = modem status. 0001 = no interrupt pending. 0010 = THR empty. 0100 = received data available. 0110 = receiver line status. 0111 = busy detect. 1100 = character timeout. 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 50: UART\_LCR\_REG (0x5000100C)

| Bit  | Mode | Symbol    | Description  | Reset |
|------|------|-----------|--|-------|
| 15:8 | -    | -         | Reserved   | 0x0   |
| 7    | R/W  | UART_DLAB | <p>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.</p> | 0x0   |



Table 50: UART\_LCR\_REG (0x5000100C)

| 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 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.<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.<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 51: 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" on page 53.<br>0 = IrDA SIR Mode disabled<br>1 = IrDA SIR Mode enabled  | 0x0   |
| 5    | R/W  | UART_AFCE | Auto Flow Control Enable.<br>When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, hardware Auto Flow Control is enabled via CTS and RTS.<br>0 = Auto Flow Control Mode disabled<br>1 = Auto Flow Control Mode enabled | 0x0   |

Table 51: UART\_MCR\_REG (0x50001010)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 4   | R/W  | UART_LB   | <p>LoopBack Bit.</p> <p>This is used to put the UART into a diagnostic mode for test purposes.</p> <p>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.</p> <p>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.</p>   | 0x0   |
| 3   | R/W  | UART_OUT2 | <p>OUT2.</p> <p>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:</p> <p>0 = out2_n de-asserted (logic 1)</p> <p>1 = out2_n asserted (logic 0)</p> <p>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.</p>   | 0x0   |
| 2   | R/W  | UART_OUT1 | <p>OUT1.</p> <p>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:</p> <p>0 = out1_n de-asserted (logic 1)</p> <p>1 = out1_n asserted (logic 0)</p> <p>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.</p>   | 0x0   |
| 1   | R/W  | UART_RTS  | <p>Request to Send.</p> <p>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.</p> <p>When Auto Flow Control is disabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. When Auto Flow Control is enabled (MCR[5] set to one) and FIFOs are enabled (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.</p> <p>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.</p> | 0x0   |
| 0   | -    | -         | Reserved  | 0x0   |

Table 52: UART\_LSR\_REG (0x50001014)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 52: UART\_LSR\_REG (0x50001014)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 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_B1   | 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 52: 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>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 53: UART\_MSR\_REG (0x50001018)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 53: UART\_MSR\_REG (0x50001018)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 7   | R    | UART_DCD  | <p>Data Carrier Detect.</p> <p>This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set.</p> <p>0 = dcd_n input is de-asserted (logic 1)<br/>1 = dcd_n input is asserted (logic 0)</p> <p>In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2).</p>  | 0x0   |
| 6   | R    | UART_R1   | <p>Ring Indicator.</p> <p>This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set.</p> <p>0 = ri_n input is de-asserted (logic 1)<br/>1 = ri_n input is asserted (logic 0)</p> <p>In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1).</p>   | 0x0   |
| 5   | -    | -         | Reserved  | 0x0   |
| 4   | R    | UART_CTS  | <p>Clear to Send.</p> <p>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.</p> <p>0 = cts_n input is de-asserted (logic 1)<br/>1 = cts_n input is asserted (logic 0)</p> <p>In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).</p>  | 0x0   |
| 3   | R    | UART_DDCD | <p>Delta Data Carrier Detect.</p> <p>This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.</p> <p>0 = no change on dcd_n since last read of MSR<br/>1 = change on dcd_n since last read of MSR</p> <p>Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.</p> | 0x0   |
| 2   | R    | UART_TERI | <p>Trailing Edge of Ring Indicator.</p> <p>This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.</p> <p>0 = no change on ri_n since last read of MSR<br/>1 = change on ri_n since last read of MSR</p> <p>Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low.</p>   | 0x0   |
| 1   | -    | -         | Reserved  | 0x0   |

Table 53: UART\_MSR\_REG (0x50001018)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 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 54: 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 55: UART\_LPDLL\_REG (0x50001020)

| Bit  | Mode | Symbol     | Description   | Reset |
|------|------|------------|---|-------|
| 15:8 | -    | -          | Reserved  | 0x0   |
| 7:0  | R/W  | UART_LPDLL | This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. This register may be accessed only when the DLAB bit (LCR[7]) is set.<br>The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows:<br>Low power baud rate = (serial clock frequency)/(16* divisor)<br>Therefore, a divisor must be selected to give a baud rate of 115.2K.<br>NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLL is set, at least eight clock cycles of the slowest UART Ctrl clock should be allowed to pass before transmitting or receiving data. | 0x0   |

Table 56: UART\_LPDLH\_REG (0x50001024)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 56: UART\_LPDH\_REG (0x50001024)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 7:0 | R/W  | UART_LPDH | <p>This register makes up the upper 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. This register may be accessed only when the DLAB bit (LCR[7]) is set.</p> <p>The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows:</p> $\text{Low power baud rate} = (\text{serial clock frequency}) / (16 * \text{divisor})$ <p>Therefore, a divisor must be selected to give a baud rate of 115.2K.</p> <p>NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLH is set, at least eight clock cycles of the slowest UART Ctrl clock should be allowed to pass before transmitting or receiving data.</p> | 0x0   |

Table 57: UART\_SRBR\_STHR0\_REG (0x50001030)

| Bit  | Mode | Symbol     | Description   | Reset |
|------|------|------------|---|-------|
| 15:8 | -    | -          | Reserved  | 0x0   |
| 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 58: UART\_SRBR\_STHR1\_REG (0x50001034)

| Bit  | Mode | Symbol     | Description   | Reset |
|------|------|------------|---|-------|
| 15:8 | -    | -          | Reserved  | 0x0   |
| 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 59: UART\_SRBR\_STHR2\_REG (0x50001038)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |



Table 59: UART\_SRBR\_STHR2\_REG (0x50001038)

| 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 60: UART\_SRBR\_STHR3\_REG (0x5000103C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 60: UART\_SRBR\_STHR3\_REG (0x5000103C)

| 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 61: UART\_SRBR\_STHR4\_REG (0x50001040)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 61: UART\_SRBR\_STHR4\_REG (0x50001040)

| 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 62: UART\_SRBR\_STHR5\_REG (0x50001044)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 62: UART\_SRBR\_STHR5\_REG (0x50001044)

| 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 63: UART\_SRBR\_STHR6\_REG (0x50001048)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 63: UART\_SRBR\_STHR6\_REG (0x50001048)

| 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 64: UART\_SRBR\_STHR7\_REG (0x5000104C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 64: UART\_SRBR\_STHR7\_REG (0x5000104C)

| 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 65: UART\_SRBR\_STHR8\_REG (0x50001050)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 65: UART\_SRBR\_STHR8\_REG (0x50001050)

| 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 66: UART\_SRBR\_STHR9\_REG (0x50001054)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 66: UART\_SRBR\_STHR9\_REG (0x50001054)

| 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 67: UART\_SRBR\_STHR10\_REG (0x50001058)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |



Table 67: UART\_SRBR\_STHR10\_REG (0x50001058)

| 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 68: UART\_SRBR\_STHR11\_REG (0x5000105C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 68: UART\_SRBR\_STHR11\_REG (0x5000105C)

| 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 69: UART\_SRBR\_STHR12\_REG (0x50001060)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 69: UART\_SRBR\_STHR12\_REG (0x50001060)

| 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 70: UART\_SRBR\_STHR13\_REG (0x50001064)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 70: UART\_SRBR\_STHR13\_REG (0x50001064)

| 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 71: UART\_SRBR\_STHR14\_REG (0x50001068)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 71: UART\_SRBR\_STHR14\_REG (0x50001068)

| 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 72: UART\_SRBR\_STHR15\_REG (0x5000106C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 72: UART\_SRBR\_STHR15\_REG (0x5000106C)

| 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 73: UART\_USR\_REG (0x5000107C)

| Bit  | Mode | Symbol    | Description   | Reset |
|------|------|-----------|---|-------|
| 15:5 | -    | -         | Reserved  | 0x0   |
| 4    | R    | UART_RFF  | <p>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.</p>                | 0x0   |
| 3    | R    | UART_RFNE | <p>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.</p>  | 0x0   |
| 2    | R    | UART_TFE  | <p>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.</p> | 0x1   |

Table 73: UART\_USR\_REG (0x5000107C)

| Bit | Mode | Symbol    | Description  | Reset |
|-----|------|-----------|--|-------|
| 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   | -    | -         | Reserved   | 0x0   |

Table 74: UART\_TFL\_REG (0x50001080)

| Bit  | Mode | Symbol                   | Description   | Reset |
|------|------|--------------------------|---|-------|
| 15:0 | R    | UART_TRANSMIT_FIFO_LEVEL | Transmit FIFO Level.<br>This indicates the number of data entries in the transmit FIFO. | 0x0   |

Table 75: UART\_RFL\_REG (0x50001084)

| Bit  | Mode | Symbol                  | Description   | Reset |
|------|------|-------------------------|---|-------|
| 15:0 | R    | UART_RECEIVE_FIFO_LEVEL | Receive FIFO Level.<br>This indicates the number of data entries in the receive FIFO. | 0x0   |

Table 76: UART\_SRR\_REG (0x50001088)

| 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.<br>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 77: UART\_SRTS\_REG (0x5000108C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 77: UART\_SRTS\_REG (0x5000108C)

| Bit | Mode | Symbol                      | Description  | Reset |
|-----|------|-----------------------------|--|-------|
| 0   | R/W  | UART_SHADOW_REQUEST_TO_SEND | <p>Shadow Request to Send.</p> <p>This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to perform 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.</p> <p>When Auto Flow Control is disabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high.</p> <p>When Auto Flow Control is enabled (MCR[5] = 1) and FIFOs are enabled (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).</p> <p>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.</p> | 0x0   |

Table 78: UART\_SBCR\_REG (0x50001090)

| Bit  | Mode | Symbol                    | Description   | Reset |
|------|------|---------------------------|---|-------|
| 15:1 | -    | -                         | Reserved  | 0x0   |
| 0    | R/W  | UART_SHADOW_BREAK_CONTROL | <p>Shadow Break Control Bit.</p> <p>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.</p> <p>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.</p> <p>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.</p> | 0x0   |

Table 79: UART\_SDMAM\_REG (0x50001094)

| Bit  | Mode | Symbol               | Description  | Reset |
|------|------|----------------------|--|-------|
| 15:1 | -    | -                    | Reserved   | 0x0   |
| 0    | R/W  | UART_SHADOW_DMA_MODE | <p>Shadow DMA Mode.</p> <p>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.</p> <p>0 = mode 0<br/>1 = mode 1</p> | 0x0   |

Table 80: UART\_SFE\_REG (0x50001098)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |



Table 80: UART\_SFE\_REG (0x50001098)

| Bit | Mode | Symbol                  | Description   | Reset |
|-----|------|-------------------------|---|-------|
| 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 81: UART\_SRT\_REG (0x5000109C)

| Bit  | Mode | Symbol                   | Description  | Reset |
|------|------|--------------------------|--|-------|
| 15:2 | -    | -                        | Reserved   | 0x0   |
| 1:0  | R/W  | UART_SHADOW_RCVR_TRIGGER | Shadow RCVR Trigger.<br>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.<br>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:<br>00 = 1 character in the FIFO<br>01 = FIFO $\hat{A}$ ¼ full<br>10 = FIFO $\hat{A}$ ½ full<br>11 = FIFO 2 less than full | 0x0   |

Table 82: UART\_STET\_REG (0x500010A0)

| Bit  | Mode | Symbol                       | Description  | Reset |
|------|------|------------------------------|--|-------|
| 15:2 | -    | -                            | Reserved   | 0x0   |
| 1:0  | R/W  | UART_SHADOW_TX_EMPTY_TRIGGER | Shadow TX Empty Trigger.<br>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.<br>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:<br>00 = FIFO empty<br>01 = 2 characters in the FIFO<br>10 = FIFO $\hat{A}$ ¼ full<br>11 = FIFO $\hat{A}$ ½ full | 0x0   |

Table 83: UART\_HTX\_REG (0x500010A4)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

**Table 83: UART\_HTX\_REG (0x500010A4)**

| Bit | Mode | Symbol       | Description   | Reset |
|-----|------|--------------|---|-------|
| 0   | R/W  | UART_HALT_TX | 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.<br>0 = Halt TX disabled<br>1 = Halt TX enabled<br>Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation. | 0x0   |

**Table 84: UART\_CPR\_REG (0x500010F4)**

| Bit  | Mode | Symbol | Description                  | Reset |
|------|------|--------|------------------------------|-------|
| 15:0 | R    | CPR    | Component Parameter Register | 0x0   |

**Table 85: UART\_UCV\_REG (0x500010F8)**

| Bit  | Mode | Symbol | Description       | Reset          |
|------|------|--------|-------------------|----------------|
| 15:0 | R    | UCV    | Component Version | 0x33303<br>82A |

**Table 86: UART\_CTR\_REG (0x500010FC)**

| Bit  | Mode | Symbol | Description             | Reset          |
|------|------|--------|-------------------------|----------------|
| 15:0 | R    | CTR    | Component Type Register | 0x44570<br>110 |

**Table 87: UART2\_RBR\_THR\_DLL\_REG (0x50001100)**

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 87: UART2\_RBR\_THR\_DLL\_REG (0x50001100)

| Bit | Mode | Symbol      | Description   | Reset |
|-----|------|-------------|---|-------|
| 7:0 | R/W  | RBR_THR_DLL | <p>Receive Buffer Register: 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. Transmit Holding Register: 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. Divisor Latch (Low): 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: <math>\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> | 0x0   |

Table 88: 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    | R/W  | EDSSI_DLH3 | Interrupt Enable Register: EDSSI, Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[3] of the 8 bit DLH register | 0x0   |

Table 88: UART2\_IER\_DLH\_REG (0x50001104)

| 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 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 89: 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. 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: 0000 = modem status. 0001 = no interrupt pending. 0010 = THR empty. 0100 = received data available. 0110 = receiver line status. 0111 = busy detect. 1100 = character timeout. 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 90: UART2\_LCR\_REG (0x5000110C)

| Bit  | Mode | Symbol    | Description  | Reset |
|------|------|-----------|--|-------|
| 15:8 | -    | -         | Reserved   | 0x0   |
| 7    | R/W  | UART_DLAB | <p>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.</p> | 0x0   |

Table 90: 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 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.<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.<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 91: 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" on page 53.<br>0 = IrDA SIR Mode disabled<br>1 = IrDA SIR Mode enabled  | 0x0   |
| 5    | R/W  | UART_AFCE | Auto Flow Control Enable.<br>When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, hardware Auto Flow Control is enabled via CTS and RTS.<br>0 = Auto Flow Control Mode disabled<br>1 = Auto Flow Control Mode enabled | 0x0   |

Table 91: UART2\_MCR\_REG (0x50001110)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 4   | R/W  | UART_LB   | <p>LoopBack Bit.</p> <p>This is used to put the UART into a diagnostic mode for test purposes.</p> <p>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.</p> <p>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.</p>   | 0x0   |
| 3   | R/W  | UART_OUT2 | <p>OUT2.</p> <p>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:</p> <p>0 = out2_n de-asserted (logic 1)</p> <p>1 = out2_n asserted (logic 0)</p> <p>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.</p>   | 0x0   |
| 2   | R/W  | UART_OUT1 | <p>OUT1.</p> <p>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:</p> <p>0 = out1_n de-asserted (logic 1)</p> <p>1 = out1_n asserted (logic 0)</p> <p>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.</p>   | 0x0   |
| 1   | R/W  | UART_RTS  | <p>Request to Send.</p> <p>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.</p> <p>When Auto Flow Control is disabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. When Auto Flow Control is enabled (MCR[5] set to one) and FIFOs are enabled (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.</p> <p>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.</p> | 0x0   |
| 0   | -    | -         | Reserved  | 0x0   |

Table 92: UART2\_LSR\_REG (0x50001114)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 92: UART2\_LSR\_REG (0x50001114)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 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_B1   | 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 92: 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 93: UART2\_MSR\_REG (0x50001118)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 93: UART2\_MSR\_REG (0x50001118)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 7   | R    | UART_DCD  | <p>Data Carrier Detect.</p> <p>This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set.</p> <p>0 = dcd_n input is de-asserted (logic 1)<br/>1 = dcd_n input is asserted (logic 0)</p> <p>In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2).</p>  | 0x0   |
| 6   | R    | UART_R1   | <p>Ring Indicator.</p> <p>This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set.</p> <p>0 = ri_n input is de-asserted (logic 1)<br/>1 = ri_n input is asserted (logic 0)</p> <p>In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1).</p>   | 0x0   |
| 5   | -    | -         | Reserved  | 0x0   |
| 4   | R    | UART_CTS  | <p>Clear to Send.</p> <p>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.</p> <p>0 = cts_n input is de-asserted (logic 1)<br/>1 = cts_n input is asserted (logic 0)</p> <p>In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).</p>  | 0x0   |
| 3   | R    | UART_DDCD | <p>Delta Data Carrier Detect.</p> <p>This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.</p> <p>0 = no change on dcd_n since last read of MSR<br/>1 = change on dcd_n since last read of MSR</p> <p>Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.</p> | 0x0   |
| 2   | R    | UART_TERI | <p>Trailing Edge of Ring Indicator.</p> <p>This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.</p> <p>0 = no change on ri_n since last read of MSR<br/>1 = change on ri_n since last read of MSR</p> <p>Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low.</p>   | 0x0   |
| 1   | -    | -         | Reserved  | 0x0   |

Table 93: UART2\_MSR\_REG (0x50001118)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 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 94: 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 95: UART2\_LPDLL\_REG (0x50001120)

| Bit  | Mode | Symbol     | Description   | Reset |
|------|------|------------|---|-------|
| 15:8 | -    | -          | Reserved  | 0x0   |
| 7:0  | R/W  | UART_LPDLL | This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. This register may be accessed only when the DLAB bit (LCR[7]) is set.<br>The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows:<br>Low power baud rate = (serial clock frequency)/(16* divisor)<br>Therefore, a divisor must be selected to give a baud rate of 115.2K.<br>NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLL is set, at least eight clock cycles of the slowest UART Ctrl clock should be allowed to pass before transmitting or receiving data. | 0x0   |

Table 96: UART2\_LPDH\_REG (0x50001124)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 96: UART2\_LPDH\_REG (0x50001124)

| Bit | Mode | Symbol    | Description   | Reset |
|-----|------|-----------|---|-------|
| 7:0 | R/W  | UART_LPDH | <p>This register makes up the upper 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. This register may be accessed only when the DLAB bit (LCR[7]) is set.</p> <p>The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows:</p> $\text{Low power baud rate} = (\text{serial clock frequency}) / (16 * \text{divisor})$ <p>Therefore, a divisor must be selected to give a baud rate of 115.2K.</p> <p>NOTE: When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLH is set, at least eight clock cycles of the slowest UART Ctrl clock should be allowed to pass before transmitting or receiving data.</p> | 0x0   |

Table 97: UART2\_SRBR\_STHR0\_REG (0x50001130)

| Bit  | Mode | Symbol     | Description   | Reset |
|------|------|------------|---|-------|
| 15:8 | -    | -          | Reserved  | 0x0   |
| 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 98: UART2\_SRBR\_STHR1\_REG (0x50001134)

| Bit  | Mode | Symbol     | Description   | Reset |
|------|------|------------|---|-------|
| 15:8 | -    | -          | Reserved  | 0x0   |
| 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 99: UART2\_SRBR\_STHR2\_REG (0x50001138)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 99: 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 100: UART2\_SRBR\_STHR3\_REG (0x5000113C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 100: 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 101: UART2\_SRBR\_STHR4\_REG (0x50001140)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 101: 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 (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 102: UART2\_SRBR\_STHR5\_REG (0x50001144)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |



Table 102: 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 103: UART2\_SRBR\_STHR6\_REG (0x50001148)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 103: 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 104: UART2\_SRBR\_STHR7\_REG (0x5000114C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 104: 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 105: UART2\_SRBR\_STHR8\_REG (0x50001150)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 105: 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 106: UART2\_SRBR\_STHR9\_REG (0x50001154)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 106: 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 107: UART2\_SRBR\_STHR10\_REG (0x50001158)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 107: 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 108: UART2\_SRBR\_STHR11\_REG (0x5000115C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 108: 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 (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 109: UART2\_SRBR\_STHR12\_REG (0x50001160)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 109: 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 110: UART2\_SRBR\_STHR13\_REG (0x50001164)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |



Table 110: 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 111: UART2\_SRBR\_STHR14\_REG (0x50001168)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 111: 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 112: UART2\_SRBR\_STHR15\_REG (0x5000116C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:8 | -    | -      | Reserved    | 0x0   |

Table 112: 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 113: UART2\_USR\_REG (0x5000117C)

| Bit  | Mode | Symbol    | Description   | Reset |
|------|------|-----------|---|-------|
| 15:5 | -    | -         | Reserved  | 0x0   |
| 4    | R    | UART_RFF  | <p>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.</p>                | 0x0   |
| 3    | R    | UART_RFNE | <p>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.</p>  | 0x0   |
| 2    | R    | UART_TFE  | <p>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.</p> | 0x1   |

Table 113: UART2\_USR\_REG (0x5000117C)

| Bit | Mode | Symbol    | Description  | Reset |
|-----|------|-----------|--|-------|
| 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   | -    | -         | Reserved   | 0x0   |

Table 114: UART2\_TFL\_REG (0x50001180)

| Bit  | Mode | Symbol                   | Description   | Reset |
|------|------|--------------------------|---|-------|
| 15:0 | R    | UART_TRANSMIT_FIFO_LEVEL | Transmit FIFO Level.<br>This indicates the number of data entries in the transmit FIFO. | 0x0   |

Table 115: UART2\_RFL\_REG (0x50001184)

| Bit  | Mode | Symbol                  | Description   | Reset |
|------|------|-------------------------|---|-------|
| 15:0 | R    | UART_RECEIVE_FIFO_LEVEL | Receive FIFO Level.<br>This indicates the number of data entries in the receive FIFO. | 0x0   |

Table 116: 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.<br>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 117: UART2\_SRTS\_REG (0x5000118C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 117: UART2\_SRTS\_REG (0x5000118C)

| Bit | Mode | Symbol                      | Description  | Reset |
|-----|------|-----------------------------|--|-------|
| 0   | R/W  | UART_SHADOW_REQUEST_TO_SEND | <p>Shadow Request to Send.</p> <p>This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to perform 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.</p> <p>When Auto Flow Control is disabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high.</p> <p>When Auto Flow Control is enabled (MCR[5] = 1) and FIFOs are enabled (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).</p> <p>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.</p> | 0x0   |

Table 118: UART2\_SBCR\_REG (0x50001190)

| Bit  | Mode | Symbol                    | Description   | Reset |
|------|------|---------------------------|---|-------|
| 15:1 | -    | -                         | Reserved  | 0x0   |
| 0    | R/W  | UART_SHADOW_BREAK_CONTROL | <p>Shadow Break Control Bit.</p> <p>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.</p> <p>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.</p> <p>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.</p> | 0x0   |

Table 119: UART2\_SDMAM\_REG (0x50001194)

| Bit  | Mode | Symbol               | Description  | Reset |
|------|------|----------------------|--|-------|
| 15:1 | -    | -                    | Reserved   | 0x0   |
| 0    | R/W  | UART_SHADOW_DMA_MODE | <p>Shadow DMA Mode.</p> <p>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.</p> <p>0 = mode 0<br/>1 = mode 1</p> | 0x0   |

Table 120: UART2\_SFE\_REG (0x50001198)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 120: UART2\_SFE\_REG (0x50001198)

| Bit | Mode | Symbol                  | Description   | Reset |
|-----|------|-------------------------|---|-------|
| 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 121: UART2\_SRT\_REG (0x5000119C)

| Bit  | Mode | Symbol                   | Description  | Reset |
|------|------|--------------------------|--|-------|
| 15:2 | -    | -                        | Reserved   | 0x0   |
| 1:0  | R/W  | UART_SHADOW_RCVR_TRIGGER | Shadow RCVR Trigger.<br>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.<br>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:<br>00 = 1 character in the FIFO<br>01 = FIFO $\hat{A}$ ¼ full<br>10 = FIFO $\hat{A}$ ½ full<br>11 = FIFO 2 less than full | 0x0   |

Table 122: UART2\_STET\_REG (0x500011A0)

| Bit  | Mode | Symbol                       | Description  | Reset |
|------|------|------------------------------|--|-------|
| 15:2 | -    | -                            | Reserved   | 0x0   |
| 1:0  | R/W  | UART_SHADOW_TX_EMPTY_TRIGGER | Shadow TX Empty Trigger.<br>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.<br>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:<br>00 = FIFO empty<br>01 = 2 characters in the FIFO<br>10 = FIFO $\hat{A}$ ¼ full<br>11 = FIFO $\hat{A}$ ½ full | 0x0   |

Table 123: UART2\_HTX\_REG (0x500011A4)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 123: UART2\_HTX\_REG (0x500011A4)

| Bit | Mode | Symbol       | Description   | Reset |
|-----|------|--------------|---|-------|
| 0   | R/W  | UART_HALT_TX | 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.<br>0 = Halt TX disabled<br>1 = Halt TX enabled<br>Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation. | 0x0   |

Table 124: UART2\_CPR\_REG (0x500011F4)

| Bit  | Mode | Symbol | Description                  | Reset |
|------|------|--------|------------------------------|-------|
| 15:0 | R    | CPR    | Component Parameter Register | 0x0   |

Table 125: UART2\_UCV\_REG (0x500011F8)

| Bit  | Mode | Symbol | Description       | Reset          |
|------|------|--------|-------------------|----------------|
| 15:0 | R    | UCV    | Component Version | 0x33303<br>82A |

Table 126: UART2\_CTR\_REG (0x500011FC)

| Bit  | Mode | Symbol | Description             | Reset          |
|------|------|--------|-------------------------|----------------|
| 15:0 | R    | CTR    | Component Type Register | 0x44570<br>110 |

Table 127: 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 the Interrupt Controller<br>1 = Enable SPI_INT_BIT to the Interrupt Controller   | 0x0   |
| 13  | R    | SPI_INT_BIT  | 0 = RX Register or FIFO is empty.<br>1 = SPI interrupt. Data has been transmitted and received-<br>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   |

Table 127: SPI\_CTRL\_REG (0x50001200)

| Bit | Mode | Symbol  | Description  | Reset |
|-----|------|---------|--|-------|
| 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 frequency in master mode:00 = (XTAL) / (CLK_PER_REG *8)<br>01 = (XTAL) / (CLK_PER_REG *4)<br>10 = (XTAL) / (CLK_PER_REG *2)<br>11 = (XTAL) / (CLK_PER_REG *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   |
| 0   | R/W  | SPI_ON  | 0 = SPI Module switched off (power saving). Everything is reset except SPI_CTRL_REG0 and SPI_CTRL_REG1. 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 128: 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 129: 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 130: 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 131: 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   |



Table 131: SPI\_CTRL\_REG1 (0x50001208)

| Bit | Mode | Symbol        | Description   | Reset |
|-----|------|---------------|---|-------|
| 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   |
| 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 132: I2C\_CON\_REG (0x50001300)

| 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 133: I2C\_TAR\_REG (0x50001304)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:12 | -    | -      | Reserved    | 0x0   |

Table 133: I2C\_TAR\_REG (0x50001304)

| Bit | Mode | Symbol      | Description  | Reset |
|-----|------|-------------|--|-------|
| 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 | 0x55  |

Table 134: I2C\_SAR\_REG (0x50001308)

| 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 135: I2C\_DATA\_CMD\_REG (0x50001310)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:9 | -    | -      | Reserved    | 0x0   |

Table 135: I2C\_DATA\_CMD\_REG (0x50001310)

| 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 136: I2C\_SS\_SCL\_HCNT\_REG (0x50001314)

| 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 137: I2C\_SS\_SCL\_LCNT\_REG (0x50001318)

| 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 138: I2C\_FS\_SCL\_HCNT\_REG (0x5000131C)

| 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 139: I2C\_FS\_SCL\_LCNT\_REG (0x50001320)

| 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 140: I2C\_INTR\_STAT\_REG (0x5000132C)

| 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 140: I2C\_INTR\_STAT\_REG (0x5000132C)

| 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_ABORT  | 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_ABORT_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_ABORT 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 140: I2C\_INTR\_STAT\_REG (0x5000132C)

| 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 141: I2C\_INTR\_MASK\_REG (0x50001330)

| 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 142: I2C\_RAW\_INTR\_STAT\_REG (0x50001334)

| 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 142: I2C\_RAW\_INTR\_STAT\_REG (0x50001334)

| 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 143: I2C\_RX\_TL\_REG (0x50001338)

| 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 144: I2C\_TX\_TL\_REG (0x5000133C)

| 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 145: I2C\_CLR\_INTR\_REG (0x50001340)

| 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 146: I2C\_CLR\_RX\_UNDER\_REG (0x50001344)

| 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 147: I2C\_CLR\_RX\_OVER\_REG (0x50001348)

| 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 148: I2C\_CLR\_TX\_OVER\_REG (0x5000134C)

| 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 149: I2C\_CLR\_RD\_REQ\_REG (0x50001350)

| 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 150: I2C\_CLR\_TX\_ABRT\_REG (0x50001354)

| 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 151: I2C\_CLR\_RX\_DONE\_REG (0x50001358)

| 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 152: I2C\_CLR\_ACTIVITY\_REG (0x5000135C)

| 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 153: I2C\_CLR\_STOP\_DET\_REG (0x50001360)

| 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 154: I2C\_CLR\_START\_DET\_REG (0x50001364)

| 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 155: I2C\_CLR\_GEN\_CALL\_REG (0x50001368)

| 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 156: I2C\_ENABLE\_REG (0x5000136C)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:1 | -    | -      | Reserved    | 0x0   |

Table 156: I2C\_ENABLE\_REG (0x5000136C)

| 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 157: I2C\_STATUS\_REG (0x50001370)

| 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 157: I2C\_STATUS\_REG (0x50001370)

| Bit | Mode | Symbol       | Description          | Reset |
|-----|------|--------------|----------------------|-------|
| 0   | R    | I2C_ACTIVITY | I2C Activity Status. | 0x0   |

Table 158: I2C\_TXFLR\_REG (0x50001374)

| 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 159: I2C\_RXFLR\_REG (0x50001378)

| 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 160: I2C\_SDA\_HOLD\_REG (0x5000137C)

| Bit  | Mode | Symbol      | Description   | Reset |
|------|------|-------------|---------------|-------|
| 15:0 | R/W  | IC_SDA_HOLD | SDA Hold time | 0x1   |

Table 161: I2C\_TX\_ABRT\_SOURCE\_REG (0x50001380)

| 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_NORSTR   | 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 161: I2C\_TX\_ABRT\_SOURCE\_REG (0x50001380)

| 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_ACK DET  | 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_REA D    | 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_NOA CK   | 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_NO ACK  | 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_N OACK | 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_N OACK | 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_N OACK | 1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.   | 0x0   |

Table 162: I2C\_SDA\_SETUP\_REG (0x50001394)

| 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 163: I2C\_ACK\_GENERAL\_CALL\_REG (0x50001398)

| 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 164: I2C\_ENABLE\_STATUS\_REG (0x5000139C)

| Bit  | Mode | Symbol                  | Description  | Reset |
|------|------|-------------------------|--|-------|
| 15:3 | -    | -                       | Reserved   | 0x0   |
| 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   |

Table 164: I2C\_ENABLE\_STATUS\_REG (0x5000139C)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 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.<br>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 165: I2C\_IC\_FS\_SPKLEN\_REG (0x500013A0)

| Bit  | Mode | Symbol       | Description   | Reset |
|------|------|--------------|---|-------|
| 15:8 | -    | -            | Reserved  | 0x0   |
| 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 166: GPIO\_IRQ0\_IN\_SEL\_REG (0x50001400)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:6 | -    | -      | Reserved    | 0x0   |

Table 166: GPIO\_IRQ0\_IN\_SEL\_REG (0x50001400)

| Bit | Mode | Symbol        | Description  | Reset |
|-----|------|---------------|--|-------|
| 5:0 | R/W  | KBRD_IRQ0_SEL | input selection that can generate a GPIO interrupt<br>0: no input selected<br>1: P0[0] is selected<br>2: P0[1] is selected<br>3: P0[2] is selected<br>4: P0[3] is selected<br>5: P0[4] is selected<br>6: P0[5] is selected<br>7: P0[6] is selected<br>8: P0[7] is selected<br>9: P1[0] is selected<br>10: P1[1] is selected<br>11: P1[2] is selected<br>12: P1[3] is selected<br>13: P1[4] is selected<br>14: P1[5] is selected<br>15: P2[0] is selected<br>16: P2[1] is selected<br>17: P2[2] is selected<br>18: P2[3] is selected<br>19: P2[4] is selected<br>20: P2[5] is selected<br>21: P2[6] is selected<br>22: P2[7] is selected<br>23: P2[8] is selected<br>24: P2[9] is selected<br>25: P3[0] is selected<br>26: P3[1] is selected<br>27: P3[2] is selected<br>28: P3[3] is selected<br>29: P3[4] is selected<br>30: P3[5] is selected<br>31: P3[6] is selected<br>32: P3[7] is selected<br>all others: no input selected | 0x0   |

Table 167: GPIO\_IRQ1\_IN\_SEL\_REG (0x50001402)

| Bit  | Mode | Symbol        | Description       | Reset |
|------|------|---------------|-------------------|-------|
| 15:5 | -    | -             | Reserved          | 0x0   |
| 5:0  | R/W  | KBRD_IRQ1_SEL | see KBRD_IRQ0_SEL | 0x0   |

Table 168: GPIO\_IRQ2\_IN\_SEL\_REG (0x50001404)

| Bit  | Mode | Symbol        | Description       | Reset |
|------|------|---------------|-------------------|-------|
| 15:5 | -    | -             | Reserved          | 0x0   |
| 5:0  | R/W  | KBRD_IRQ2_SEL | see KBRD_IRQ0_SEL | 0x0   |

Table 169: GPIO\_IRQ3\_IN\_SEL\_REG (0x50001406)

| Bit  | Mode | Symbol        | Description       | Reset |
|------|------|---------------|-------------------|-------|
| 15:5 | -    | -             | Reserved          | 0x0   |
| 5:0  | R/W  | KBRD_IRQ3_SEL | see KBRD_IRQ0_SEL | 0x0   |



Table 170: GPIO\_IRQ4\_IN\_SEL\_REG (0x50001408)

| Bit  | Mode | Symbol        | Description       | Reset |
|------|------|---------------|-------------------|-------|
| 15:5 | -    | -             | Reserved          | 0x0   |
| 5:0  | R/W  | KBRD_IRQ4_SEL | see KBRD_IRQ0_SEL | 0x0   |

Table 171: GPIO\_DEBOUNCE\_REG (0x5000140C)

| Bit   | Mode | Symbol          | Description   | Reset |
|-------|------|-----------------|---|-------|
| 15:14 | -    | -               | Reserved  | 0x0   |
| 13    | R/W  | DEB_ENABLE_KBRD | enables the debounce counter for the KBRD interface   | 0x0   |
| 12    | R/W  | DEB_ENABLE4     | enables the debounce counter for GPIO IRQ4  | 0x0   |
| 11    | R/W  | DEB_ENABLE3     | enables the debounce counter for GPIO IRQ3  | 0x0   |
| 10    | R/W  | DEB_ENABLE2     | enables the debounce counter for GPIO IRQ2  | 0x0   |
| 9     | R/W  | DEB_ENABLE1     | enables the debounce counter for GPIO IRQ1  | 0x0   |
| 8     | R/W  | DEB_ENABLE0     | enables the debounce counter for GPIO IRQ0  | 0x0   |
| 7:6   | -    | -               | Reserved  | 0x0   |
| 5:0   | R/W  | DEB_VALUE       | Keyboard debounce time if enabled. Generate KEYB_INT after specified time.<br>Debounce time: $N * 1 \text{ ms}$ . $N = 0..63$ | 0x0   |

Table 172: GPIO\_RESET\_IRQ\_REG (0x5000140E)

| Bit  | Mode | Symbol          | Description  | Reset |
|------|------|-----------------|--|-------|
| 15:6 | -    | -               | Reserved   | 0x0   |
| 5    | R0/W | RESET_KBRD_IRQ  | writing a 1 to this bit will reset the KBRD IRQ. Reading returns 0.  | 0x0   |
| 4    | R0/W | RESET_GPIO4_IRQ | writing a 1 to this bit will reset the GPIO4 IRQ. Reading returns 0. | 0x0   |
| 3    | R0/W | RESET_GPIO3_IRQ | writing a 1 to this bit will reset the GPIO3 IRQ. Reading returns 0. | 0x0   |
| 2    | R0/W | RESET_GPIO2_IRQ | writing a 1 to this bit will reset the GPIO2 IRQ. Reading returns 0. | 0x0   |
| 1    | R0/W | RESET_GPIO1_IRQ | writing a 1 to this bit will reset the GPIO1 IRQ. Reading returns 0. | 0x0   |
| 0    | R0/W | RESET_GPIO0_IRQ | writing a 1 to this bit will reset the GPIO0 IRQ. Reading returns 0. | 0x0   |

Table 173: GPIO\_INT\_LEVEL\_CTRL\_REG (0x50001410)

| Bit   | Mode | Symbol       | Description   | Reset |
|-------|------|--------------|---|-------|
| 15:14 | -    | -            | Reserved  | 0x0   |
| 12    | R/W  | EDGE_LEVELN4 | see EDGE_LEVELN0, but for GPIO IRQ4   | 0x0   |
| 11    | R/W  | EDGE_LEVELN3 | see EDGE_LEVELN0, but for GPIO IRQ3   | 0x0   |
| 10    | R/W  | EDGE_LEVELN2 | see EDGE_LEVELN0, but for GPIO IRQ2   | 0x0   |
| 9     | R/W  | EDGE_LEVELN1 | see EDGE_LEVELN0, but for GPIO IRQ1   | 0x0   |
| 8     | R/W  | EDGE_LEVELN0 | 0: do not wait for key release after interrupt was reset for GPIO IRQ0, so a new interrupt can be initiated immediately<br>1: wait for key release after interrupt was reset for IRQ0 | 0x0   |
| 7:6   | -    | -            | Reserved  | 0x0   |

Table 173: GPIO\_INT\_LEVEL\_CTRL\_REG (0x50001410)

| Bit | Mode | Symbol       | Description   | Reset |
|-----|------|--------------|---|-------|
| 4   | R/W  | INPUT_LEVEL4 | see INPUT_LEVEL0, but for GPIO IRQ4   | 0x0   |
| 3   | R/W  | INPUT_LEVEL3 | see INPUT_LEVEL0, but for GPIO IRQ3   | 0x0   |
| 2   | R/W  | INPUT_LEVEL2 | see INPUT_LEVEL0, but for GPIO IRQ2   | 0x0   |
| 1   | R/W  | INPUT_LEVEL1 | see INPUT_LEVEL0, but for GPIO IRQ1   | 0x0   |
| 0   | R/W  | INPUT_LEVEL0 | 0 = selected input will generate GPIO IRQ0 if that input is high.<br>1 = selected input will generate GPIO IRQ0 if that input is low. | 0x0   |

Table 174: KBRD\_IRQ\_IN\_SEL0\_REG (0x50001412)

| Bit  | Mode | Symbol      | Description  | Reset |
|------|------|-------------|--|-------|
| 15   | R/W  | KBRD_REL    | 0 = No interrupt on key release<br>1 = Interrupt also on key release (also debouncing if enabled)  | 0x0   |
| 14   | R/W  | KBRD_LEVEL  | 0 = enabled input will generate KBRD IRQ if that input is high.<br>1 = enabled input will generate KBRD IRQ if that input is low.  | 0x0   |
| 13:8 | R/W  | KEY_REPEAT  | While key is pressed, automatically generate repeating KEYB_INT after specified time unequal to 0.<br>Repeat time: $N \cdot 1$ ms. $N = 1..63$ , $N=0$ disables the timer. | 0x0   |
| 7    | R/W  | KBRD_P07_EN | enable P0[7] for the keyboard interrupt  | 0x0   |
| 6    | R/W  | KBRD_P06_EN | enable P0[6] for the keyboard interrupt  | 0x0   |
| 5    | R/W  | KBRD_P05_EN | enable P0[5] for the keyboard interrupt  | 0x0   |
| 4    | R/W  | KBRD_P04_EN | enable P0[4] for the keyboard interrupt  | 0x0   |
| 3    | R/W  | KBRD_P03_EN | enable P0[3] for the keyboard interrupt  | 0x0   |
| 2    | R/W  | KBRD_P02_EN | enable P0[2] for the keyboard interrupt  | 0x0   |
| 1    | R/W  | KBRD_P01_EN | enable P0[1] for the keyboard interrupt  | 0x0   |
| 0    | R/W  | KBRD_P00_EN | enable P0[0] for the keyboard interrupt  | 0x0   |

Table 175: KBRD\_IRQ\_IN\_SEL1\_REG (0x50001414)

| Bit | Mode | Symbol      | Description                             | Reset |
|-----|------|-------------|---|-------|
| 15  | R/W  | KBRD_P15_EN | enable P1[5] for the keyboard interrupt | 0x0   |
| 14  | R/W  | KBRD_P14_EN | enable P1[4] for the keyboard interrupt | 0x0   |
| 13  | R/W  | KBRD_P13_EN | enable P1[3] for the keyboard interrupt | 0x0   |
| 12  | R/W  | KBRD_P12_EN | enable P1[2] for the keyboard interrupt | 0x0   |
| 11  | R/W  | KBRD_P11_EN | enable P1[1] for the keyboard interrupt | 0x0   |
| 10  | R/W  | KBRD_P10_EN | enable P1[0] for the keyboard interrupt | 0x0   |
| 9   | R/W  | KBRD_P29_EN | enable P2[9] for the keyboard interrupt | 0x0   |
| 8   | R/W  | KBRD_P28_EN | enable P2[8] for the keyboard interrupt | 0x0   |
| 7   | R/W  | KBRD_P27_EN | enable P2[7] for the keyboard interrupt | 0x0   |
| 6   | R/W  | KBRD_P26_EN | enable P2[6] for the keyboard interrupt | 0x0   |
| 5   | R/W  | KBRD_P25_EN | enable P2[5] for the keyboard interrupt | 0x0   |
| 4   | R/W  | KBRD_P24_EN | enable P2[4] for the keyboard interrupt | 0x0   |
| 3   | R/W  | KBRD_P23_EN | enable P2[3] for the keyboard interrupt | 0x0   |
| 2   | R/W  | KBRD_P22_EN | enable P2[2] for the keyboard interrupt | 0x0   |

Table 175: KBRD\_IRQ\_IN\_SEL1\_REG (0x50001414)

| Bit | Mode | Symbol      | Description                             | Reset |
|-----|------|-------------|---|-------|
| 1   | R/W  | KBRD_P21_EN | enable P2[1] for the keyboard interrupt | 0x0   |
| 0   | R/W  | KBRD_P20_EN | enable P2[0] for the keyboard interrupt | 0x0   |

Table 176: KBRD\_IRQ\_IN\_SEL2\_REG (0x50001416)

| Bit | Mode | Symbol      | Description                             | Reset |
|-----|------|-------------|---|-------|
| 7   | R/W  | KBRD_P37_EN | enable P3[7] for the keyboard interrupt | 0x0   |
| 6   | R/W  | KBRD_P36_EN | enable P3[6] for the keyboard interrupt | 0x0   |
| 5   | R/W  | KBRD_P35_EN | enable P3[5] for the keyboard interrupt | 0x0   |
| 4   | R/W  | KBRD_P34_EN | enable P3[4] for the keyboard interrupt | 0x0   |
| 3   | R/W  | KBRD_P33_EN | enable P3[3] for the keyboard interrupt | 0x0   |
| 2   | R/W  | KBRD_P32_EN | enable P3[2] for the keyboard interrupt | 0x0   |
| 1   | R/W  | KBRD_P31_EN | enable P3[1] for the keyboard interrupt | 0x0   |
| 0   | R/W  | KBRD_P30_EN | enable P3[0] for the keyboard interrupt | 0x0   |

Table 177: GP\_ADC\_CTRL\_REG (0x50001500)

| Bit | Mode | Symbol          | Description  | Reset |
|-----|------|-----------------|--|-------|
| 15  | R/W  | GP_ADC_LDO_ZERO | Forces LDO-output to 0V.   | 0x0   |
| 14  | R/W  | GP_ADC_LDO_EN   | Turns on LDO.  | 0x0   |
| 13  | R/W  | GP_ADC_CHOP     | Takes two samples with opposite GP_ADC_SIGN to cancel the internal offset voltage of the ADC; Highly recommended for DC-measurements.  | 0x0   |
| 12  | R/W  | GP_ADC_MUTE     | 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   |
| 11  | R/W  | GP_ADC_SE       | 0 = Differential mode<br>1 = Single ended mode   | 0x0   |
| 10  | 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   |
| 9:6 | R/W  | GP_ADC_SEL      | ADC input selection which must be set before the GP_ADC_START bit is enabled.<br>If GP_ADC_SE = 1 (single ended mode):<br>0000 = P0[0]<br>0001 = P0[1]<br>0010 = P0[2]<br>0011 = P0[3]<br>0100 = AVS<br>0101 = VDD_REF<br>0110 = VDD_RTT<br>0111 = VBAT3V<br>1000 = VDCDC<br>1001 = VBAT1V<br>All other combinations are reserved.<br>If GP_ADC_SE = 0 (differential mode):<br>0000 = P0[0] vs P0[1]<br>All other combinations are P0[2] vs P0[3]. | 0x0   |
| 5   | R/W  | GP_ADC_MINT     | 0 = Disable (mask) GP_ADC_INT.<br>1 = Enable GP_ADC_INT to ICU.  | 0x0   |

Table 177: GP\_ADC\_CTRL\_REG (0x50001500)

| 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.<br>1 = Digital clock used.  | 0x0   |
| 2   | -    | GP_ADC_TEST    | Reserved, keep 0.   | 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. | 0x0   |
| 0   | R/W  | GP_ADC_EN      | 0 = ADC is disabled and in reset.<br>1 = ADC is enabled and sampling of input is started.   | 0x0   |

Table 178: GP\_ADC\_CTRL2\_REG (0x50001502)

| Bit  | Mode | Symbol          | Description   | Reset |
|------|------|-----------------|---|-------|
| 15:4 | -    | -               | Reserved  | 0x0   |
| 3    | R/W  | GP_ADC_I20U     | Adds 20uA constant load current at the ADC LDO to minimize ripple on the reference voltage of the ADC.  | 0x0   |
| 2    | R/W  | GP_ADC_IDYN     | Enables dynamic load current at the ADC LDO to minimize ripple on the reference voltage of the ADC.   | 0x0   |
| 1    | 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.                                    | 0x0   |
| 0    | R/W  | GP_ADC_DELAY_EN | Enables delay function for several signals. This is not auto-cleared. Toggle this bit before every sampling to enable successive conversions. | 0x0   |

Table 179: GP\_ADC\_OFFP\_REG (0x50001504)

| 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 180: GP\_ADC\_OFFN\_REG (0x50001506)

| 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 181: GP\_ADC\_CLEAR\_INT\_REG (0x50001508)

| 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 182: GP\_ADC\_RESULT\_REG (0x5000150A)

| Bit   | Mode | Symbol     | Description   | Reset |
|-------|------|------------|---|-------|
| 15:10 | -    | -          | Reserved  | 0x0   |
| 9:0   | R    | GP_ADC_VAL | Returns the 10 bits linear value of the last AD conversion. | 0x0   |

Table 183: GP\_ADC\_DELAY\_REG (0x5000150C)

| Bit  | Mode | Symbol     | Description  | Reset |
|------|------|------------|--|-------|
| 15:8 | -    | -          | Reserved   | 0x0   |
| 7:0  | R/W  | DEL_LDO_EN | Defines the delay before the LDO enable (GP_ADC_LDO_EN). Reset value is 0 $\mu$ s since the LDO enable should be the first thing to be programmed in the sequence of bringing the GP ADC up. | 0x0   |

Table 184: GP\_ADC\_DELAY2\_REG (0x5000150E)

| Bit  | Mode | Symbol        | Description  | Reset |
|------|------|---------------|--|-------|
| 15:8 | R/W  | DEL_ADC_START | Defines the delay for the GP_ADC_START bit. Reset value is 17 $\mu$ s which is the recommended value to wait before starting the GP ADC. This is the third and last step of bringing up the GP ADC | 0x88  |
| 7:0  | R/W  | DEL_ADC_EN    | Defines the delay for the GP_ADC_EN bit. Reset value is 16 $\mu$ s which is the recommended value to wait after enabling the LDO. This is the second step in bringing up the GP ADC.               | 0x80  |

Table 185: CLK\_REF\_SEL\_REG (0x50001600)

| 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 : RC32KHz oscillator<br><br>0x1 : RC16MHz oscillator<br><br>0x2 : XTAL32KHz oscillator<br><br>0x3 : RCX32KHz oscillator | 0x0   |

Table 186: CLK\_REF\_CNT\_REG (0x50001602)

| Bit  | Mode | Symbol      | Description  | Reset |
|------|------|-------------|--|-------|
| 15:0 | R/W  | REF_CNT_VAL | Indicates the calibration time, with a decrement counter to 1. | 0x0   |

Table 187: CLK\_REF\_VAL\_L\_REG (0x50001604)

| Bit  | Mode | Symbol       | Description  | Reset |
|------|------|--------------|--|-------|
| 15:0 | R    | XTAL_CNT_VAL | Returns the lower 16 bits of XTAL16 clock cycles during the calibration time, defined with REF_CNT_VAL | 0x0   |

Table 188: CLK\_REF\_VAL\_H\_REG (0x50001606)

| Bit  | Mode | Symbol       | Description   | Reset |
|------|------|--------------|---|-------|
| 15:0 | R    | XTAL_CNT_VAL | Returns the upper 16 bits of XTAL 16 clock cycles during the calibration time, defined with REF_CNT_VAL | 0x0   |

Table 189: 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 190: P0\_SET\_DATA\_REG (0x50003002)

| 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 191: P0\_RESET\_DATA\_REG (0x50003004)

| 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 192: P00\_MODE\_REG (0x50003006)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 192: P00\_MODE\_REG (0x50003006)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSN<br>21 = UART2_CTSN<br>22 = UART2_RTSN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 193: P01\_MODE\_REG (0x50003008)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 193: P01\_MODE\_REG (0x50003008)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSN<br>21 = UART2_CTSN<br>22 = UART2_RTSN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 194: P02\_MODE\_REG (0x5000300A)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |



Table 194: P02\_MODE\_REG (0x5000300A)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSN<br>21 = UART2_CTSN<br>22 = UART2_RTSN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 195: P03\_MODE\_REG (0x5000300C)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 195: P03\_MODE\_REG (0x5000300C)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSN<br>21 = UART2_CTSN<br>22 = UART2_RTSN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 196: P04\_MODE\_REG (0x5000300E)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 196: P04\_MODE\_REG (0x5000300E)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSN<br>21 = UART2_CTSN<br>22 = UART2_RTSN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 197: P05\_MODE\_REG (0x50003010)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 197: P05\_MODE\_REG (0x50003010)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSN<br>21 = UART2_CTSN<br>22 = UART2_RTSN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 198: P06\_MODE\_REG (0x50003012)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 198: P06\_MODE\_REG (0x50003012)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSN<br>21 = UART2_CTSN<br>22 = UART2_RTSN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 199: P07\_MODE\_REG (0x50003014)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 199: P07\_MODE\_REG (0x50003014)

| Bit | Mode | Symbol | Description  | Reset |
|-----|------|--------|--|-------|
| 4:0 | R/W  | PID    | Function of port<br>0 = Port function, PUPD as set above<br>1 = UART1_RX<br>2 = UART1_TX<br>3 = UART2_RX<br>4 = UART2_TX<br>5 = SPI_DI<br>6 = SPI_DO<br>7 = SPI_CLK<br>8 = SPI_EN<br>9 = I2C_SCL<br>10 = I2C_SDA<br>11 = UART1_IRDA_RX<br>12 = UART1_IRDA_TX<br>13 = UART2_IRDA_RX<br>14 = UART2_IRDA_TX<br>15 = ADC (only for P0[3:0])<br>16 = PWM0<br>17 = PWM1<br>18 = BLE_DIAG (only for P0[7:0])<br>19 = UART1_CTSN<br>20 = UART1_RTSEN<br>21 = UART2_CTSN<br>22 = UART2_RTSEN<br>23 = PWM2<br>24 = PWM3<br>25 = PWM4<br>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. | 0x0   |

Table 200: P1\_DATA\_REG (0x50003020)

| 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 201: P1\_SET\_DATA\_REG (0x50003022)

| 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 202: P1\_RESET\_DATA\_REG (0x50003024)

| Bit  | Mode | Symbol   | Description   | Reset |
|------|------|----------|---|-------|
| 15:8 | -    | -        | Reserved  | 0x0   |
| 7:0  | R/W  | P1_RESET | Writing a 1 to P1[y] sets P1[y] to 0. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 203: P10\_MODE\_REG (0x50003026)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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 analog mode, these bits are don't care<br>P14_MODE_REG and P15_MODE_REG reset value is 1 (i.e. pulled up) | 0x2   |
| 7:5   | -    | -      | Reserved  | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0x0   |

Table 204: P11\_MODE\_REG (0x50003028)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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 analog mode, these bits are don't care<br>P14_MODE_REG and P15_MODE_REG reset value is 1 (i.e. pulled up) | 0x2   |
| 7:5   | -    | -      | Reserved  | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0x0   |

Table 205: P12\_MODE\_REG (0x5000302A)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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 analog mode, these bits are don't care<br>P14_MODE_REG and P15_MODE_REG reset value is 1 (i.e. pulled up) | 0x2   |
| 7:5   | -    | -      | Reserved  | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0x0   |

Table 206: P13\_MODE\_REG (0x5000302C)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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 analog mode, these bits are don't care<br>P14_MODE_REG and P15_MODE_REG reset value is 1 (i.e. pulled up) | 0x2   |
| 7:5   | -    | -      | Reserved  | 0x0   |

Table 206: P13\_MODE\_REG (0x5000302C)

| Bit | Mode | Symbol | Description           | Reset |
|-----|------|--------|-----------------------|-------|
| 4:0 | R/W  | PID    | See P0x_MODE_REG[PID] | 0x0   |

Table 207: P14\_MODE\_REG (0x5000302E)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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 analog mode, these bits are don't care<br>P14_MODE_REG and P15_MODE_REG reset value is 1 (i.e. pulled up) | 0x2   |
| 7:5   | -    | -      | Reserved  | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0x0   |

Table 208: P15\_MODE\_REG (0x50003030)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 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 analog mode, these bits are don't care<br>P14_MODE_REG and P15_MODE_REG reset value is 1 (i.e. pulled up) | 0x1   |
| 7:5   | -    | -      | Reserved  | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0x0   |

Table 209: P2\_DATA\_REG (0x50003040)

| Bit   | Mode | Symbol  | Description   | Reset |
|-------|------|---------|---|-------|
| 15:10 | -    | -       | Reserved  | 0x0   |
| 9:0   | R/W  | P2_DATA | Set P2 output register when written; Returns the value of P2 port when read | 0x0   |

Table 210: P2\_SET\_DATA\_REG (0x50003042)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 0x0   |
| 9:0   | R/W  | P2_SET | Writing a 1 to P2[y] sets P2[y] to 1. Writing 0 is discarded; Reading returns 0 | 0x0   |

Table 211: P2\_RESET\_DATA\_REG (0x50003044)

| Bit   | Mode | Symbol   | Description   | Reset |
|-------|------|----------|---|-------|
| 15:10 | -    | -        | Reserved  | 0x0   |
| 9:0   | R/W  | P2_RESET | Writing a 1 to P2[y] sets P2[y] to 0. Writing 0 is discarded; Reading returns 0 | 0x0   |



Table 212: P20\_MODE\_REG (0x50003046)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 213: P21\_MODE\_REG (0x50003048)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 214: P22\_MODE\_REG (0x5000304A)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 215: P23\_MODE\_REG (0x5000304C)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 216: P24\_MODE\_REG (0x5000304E)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:10 | -    | -      | Reserved    | 0x0   |

Table 216: P24\_MODE\_REG (0x5000304E)

| 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 analog mode, these bits are don't care | 0x2   |
| 7:5 | -    | -      | Reserved   | 0x0   |
| 4:0 | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 217: P25\_MODE\_REG (0x50003050)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 218: P26\_MODE\_REG (0x50003052)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 219: P27\_MODE\_REG (0x50003054)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 220: P28\_MODE\_REG (0x50003056)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:10 | -    | -      | Reserved    | 0x0   |

Table 220: P28\_MODE\_REG (0x50003056)

| 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 analog mode, these bits are don't care | 0x2   |
| 7:5 | -    | -      | Reserved   | 0x0   |
| 4:0 | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 221: P29\_MODE\_REG (0x50003058)

| Bit   | Mode | Symbol | Description  | Reset |
|-------|------|--------|--|-------|
| 15:10 | -    | -      | Reserved   | 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 analog mode, these bits are don't care | 0x2   |
| 7:5   | -    | -      | Reserved   | 0x0   |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]  | 0x0   |

Table 222: P01\_PADPWR\_CTRL\_REG (0x50003070)

| Bit   | Mode | Symbol      | Description  | Reset |
|-------|------|-------------|--|-------|
| 15:12 | -    | -           | Reserved   | 0x0   |
| 13:8  | R/W  | P1_OUT_CTRL | 1 = P1_x port output is powered by the 1 V rail<br>0 = P1_x port output is powered by the 3 V rail<br>bit 8 controls the power of P1[0],<br>bit 13 controls the power of P1[5]<br>(Note 3) | 0x0   |
| 7:0   | R/W  | P0_OUT_CTRL | 1 = P0_x port output is powered by the 1 V rail<br>0 = P0_x port output is powered by the 3 V rail<br>bit 0 controls the power of P0[0],<br>bit 7 controls the power of P0[7]<br>(Note 4)  | 0x0   |

**Note 3:** In Buck mode the output must be powered by the 3 V rail. In Boost mode the outputs can be powered by the 1 V rail or by the 3 V rail. In Boost mode the 3 V rail can only supply a limited current, e.g. for switching a high-impedance input of an external device. See table 'Digital input/output characteristics'.

**Note 4:** In Buck mode the output must be powered by the 3 V rail. In Boost mode the outputs can be powered by the 1 V rail or by the 3 V rail. In Boost mode the 3 V rail can only supply a limited current, e.g. for switching a high-impedance input of an external device. See table 'Digital input/output characteristics'.

Table 223: P2\_PADPWR\_CTRL\_REG (0x50003072)

| Bit   | Mode | Symbol      | Description  | Reset |
|-------|------|-------------|--|-------|
| 15:10 | -    | -           | Reserved   | 0x0   |
| 9:0   | R/W  | P2_OUT_CTRL | 1 = P2_x port output is powered by the 1 V rail<br>0 = P2_x port output is powered by the 3 V rail<br>bit 0 controls the power of P2[0],<br>bit 9 controls the power of P2[9],<br>(Note 5) | 0x0   |

**Note 5:** In Buck mode the output must be powered by the 3 V rail. In Boost mode the outputs can be powered by the 1 V rail or by the 3 V rail. In Boost mode the 3 V rail can only supply a limited current, e.g. for switching a high-impedance input of an external device. See table 'Digital input/output characteristics'.

input/output characteristics'.

**Table 224: P3\_PADPWR\_CTRL\_REG (0x50003074)**

| Bit  | Mode | Symbol      | Description  | Reset |
|------|------|-------------|--|-------|
| 15:8 | -    | -           | Reserved   | 0     |
| 7:0  | R/W  | P3_OUT_CTRL | 1 = P3_x port output is powered by the 1 V rail<br>0 = P3_x port output is powered by the 3 V rail<br>bit 0 controls the power of P3[0],<br>bit 7 controls the power of P3[7],<br>(Note 6) | 0     |

**Note 6:** In Buck mode the output must be powered by the 3 V rail. In Boost mode the outputs can be powered by the 1 V rail or by the 3 V rail. In Boost mode the 3 V rail can only supply a limited current, e.g. for switching a high-impedance input of an external device. See table 'Digital input/output characteristics'.

**Table 225: P3\_DATA\_REG (0x50003080)**

| Bit  | Mode | Symbol  | Description   | Reset |
|------|------|---------|---|-------|
| 15:8 | -    | -       | Reserved  | 0     |
| 7:0  | R/W  | P3_DATA | Set P3 output register when written; Returns the value of P3 port when read | 0     |

**Table 226: P3\_SET\_DATA\_REG (0x50003082)**

| Bit  | Mode | Symbol | Description  | Reset |
|------|------|--------|--|-------|
| 15:8 | -    | -      | Reserved   | 0     |
| 7:0  | R0/W | P3_SET | Writing a 1 to P3[y] sets P3[y] to 1. Writing 0 is discarded;<br>Reading returns 0 | 0     |

**Table 227: P3\_RESET\_DATA\_REG (0x50003084)**

| Bit  | Mode | Symbol   | Description  | Reset |
|------|------|----------|--|-------|
| 15:8 | -    | -        | Reserved   | 0     |
| 7:0  | R0/W | P3_RESET | Writing a 1 to P0[y] sets P0[y] to 0. Writing 0 is discarded;<br>Reading returns 0 | 0     |

**Table 228: P30\_MODE\_REG (0x50003086)**

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 0     |
| 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 | 2     |
| 7:5   | -    | -      | Reserved  | 0     |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

**Table 229: P31\_MODE\_REG (0x50003088)**

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:10 | -    | -      | Reserved    | 0     |

Table 229: P31\_MODE\_REG (0x50003088)

| 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 | 2     |
| 7:5 | -    | -      | Reserved  | 0     |
| 4:0 | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

Table 230: P32\_MODE\_REG (0x5000308A)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 0     |
| 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 | 2     |
| 7:5   | -    | -      | Reserved  | 0     |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

Table 231: P33\_MODE\_REG (0x5000308C)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 0     |
| 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 | 2     |
| 7:5   | -    | -      | Reserved  | 0     |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

Table 232: P34\_MODE\_REG (0x5000308E)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 0     |
| 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 | 2     |
| 7:5   | -    | -      | Reserved  | 0     |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

Table 233: P35\_MODE\_REG (0x50003090)

| Bit   | Mode | Symbol | Description | Reset |
|-------|------|--------|-------------|-------|
| 15:10 | -    | -      | Reserved    | 0     |

Table 233: P35\_MODE\_REG (0x50003090)

| 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 | 2     |
| 7:5 | -    | -      | Reserved  | 0     |
| 4:0 | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

Table 234: P36\_MODE\_REG (0x50003092)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 0     |
| 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 | 2     |
| 7:5   | -    | -      | Reserved  | 0     |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

Table 235: P37\_MODE\_REG (0x50003094)

| Bit   | Mode | Symbol | Description   | Reset |
|-------|------|--------|---|-------|
| 15:10 | -    | -      | Reserved  | 0     |
| 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 | 2     |
| 7:5   | -    | -      | Reserved  | 0     |
| 4:0   | R/W  | PID    | See P0x_MODE_REG[PID]   | 0     |

Table 236: WATCHDOG\_REG (0x50003100)

| Bit  | Mode | Symbol       | Description   | Reset |
|------|------|--------------|---|-------|
| 15:9 | R0/W | WDOG_WEN     | 0000.000 = Write enable for Watchdog timer<br>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     | <u>Write</u> : Watchdog timer reload value. Note that all bits 15-9 must be 0 to reload this register.<br><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:<br>If WATCHDOG_CTRL_REG[NMI_RST] = 0 then<br>If WDOG_VAL = 0 -> NMI (Non Maskable Interrupt)<br>if WDOG_VAL = 1F0 <sub>16</sub> -> WDOG reset -> reload FF <sub>16</sub><br>If WATCHDOG_CTRL_REG[NMI_RST] = 1 then<br>if WDOG_VAL <= 0 -> WDOG reset -> reload FF <sub>16</sub> | 0xFF  |

Table 237: 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;math&gt;\leq -16&lt;/math&gt;. 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.<br/>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   |

Table 238: CHIP\_ID1\_REG (0x50003200)

| Bit | Mode | Symbol   | Description                                    | Reset |
|-----|------|----------|--|-------|
| 7:0 | R    | CHIP_ID1 | First character of device type "580" in ASCII. | 0x35  |

Table 239: CHIP\_ID2\_REG (0x50003201)

| Bit | Mode | Symbol   | Description                                     | Reset |
|-----|------|----------|---|-------|
| 7:0 | R    | CHIP_ID2 | Second character of device type "580" in ASCII. | 0x38  |

Table 240: CHIP\_ID3\_REG (0x50003202)

| Bit | Mode | Symbol   | Description                                    | Reset |
|-----|------|----------|--|-------|
| 7:0 | R    | CHIP_ID3 | Third character of device type "580" in ASCII. | 0x30  |

Table 241: CHIP\_SWC\_REG (0x50003203)

| Bit | Mode | Symbol   | Description  | Reset |
|-----|------|----------|--|-------|
| 7:4 | -    | -        | Reserved   | 0x0   |
| 3:0 | R    | CHIP_SWC | <p><b>SoftWare Compatibility code.</b><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.</p> | 0x0   |

Table 242: 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  |

Table 243: SET\_FREEZE\_REG (0x50003300)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:4 | -    | -      | Reserved    | 0x0   |

Table 243: SET\_FREEZE\_REG (0x50003300)

| Bit | Mode | Symbol      | Description  | Reset |
|-----|------|-------------|--|-------|
| 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_SWTIM   | 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 244: RESET\_FREEZE\_REG (0x50003302)

| Bit  | Mode | Symbol      | Description   | Reset |
|------|------|-------------|---|-------|
| 15:4 | -    | -           | Reserved  | 0x0   |
| 3    | R/W  | FRZ_WDOG    | If '1', the watchdog timer continues, '0' is discarded.       | 0x0   |
| 2    | R/W  | FRZ_BLETIM  | If '1', the the BLE master clock continues, '0' is discarded. | 0x0   |
| 1    | R/W  | FRZ_SWTIM   | 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 245: DEBUG\_REG (0x50003304)

| Bit  | Mode | Symbol           | Description   | Reset |
|------|------|------------------|---|-------|
| 15:1 | -    | -                | Reserved  | 0x0   |
| 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. | 0x1   |

Table 246: 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 247: GP\_CONTROL\_REG (0x50003308)

| Bit  | Mode | Symbol | Description | Reset |
|------|------|--------|-------------|-------|
| 15:6 | -    | -      | Reserved    | 0     |



Table 247: GP\_CONTROL\_REG (0x50003308)

| Bit | Mode | Symbol         | Description  | Reset |
|-----|------|----------------|--|-------|
| 5:1 | R/W  | EM_MAP         | Select the mapping of the Exchange memory pages.<br>0: EM size 0 kB, SysRAM size 42 kB<br>1: EM size 2 kB, SysRAM size 48 kB<br>2: EM size 3 kB, SysRAM size 47 kB<br>3: EM size 4 kB, SysRAM size 46 kB<br>4: EM size 5 kB, SysRAM size 45 kB<br>5: EM size 6 kB, SysRAM size 44 kB<br>6: EM size 7 kB, SysRAM size 43 kB<br>7: EM size 8 kB, SysRAM size 42 kB<br>8: Reserved<br>9: EM size 4 kB, SysRAM size 40 kB<br>10: EM size 5 kB, SysRAM size 40 kB<br>11: EM size 6 kB, SysRAM size 40 kB<br>12: EM size 7 kB, SysRAM size 40 kB<br>13: EM size 8 kB, SysRAM size 40 kB<br>14: EM size 9 kB, SysRAM size 40 kB<br>15: EM size 10 kB, SysRAM size 40 kB<br>16: Reserved<br>17: EM size 6 kB, SysRAM size 38 kB<br>18: EM size 7 kB, SysRAM size 38 kB<br>19: EM size 8 kB, SysRAM size 38 kB<br>20: EM size 9 kB, SysRAM size 38 kB<br>21: EM size 10 kB, SysRAM size 38 kB<br>22: EM size 11 kB, SysRAM size 38 kB<br>23: EM size 12 kB, SysRAM size 38 kB<br>other: Reserved. | 0x1   |
| 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 248: 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. So it will be in the range of 1 to 8 MHz. | 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 16, 8, 4 or 2 MHz (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 249: 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 250: 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<br>T0_CNTER is returned | 0x0   |

Table 251: 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 252: PWM2\_DUTY\_CYCLE (0x50003408)

| Bit  | Mode | Symbol     | Description        | Reset |
|------|------|------------|--------------------|-------|
| 13:0 | R/W  | DUTY_CYCLE | duty cycle for PWM | 0x0   |

Table 253: PWM3\_DUTY\_CYCLE (0x5000340A)

| Bit  | Mode | Symbol     | Description        | Reset |
|------|------|------------|--------------------|-------|
| 13:0 | R/W  | DUTY_CYCLE | duty cycle for PWM | 0x0   |

Table 254: PWM4\_DUTY\_CYCLE (0x5000340C)

| Bit  | Mode | Symbol     | Description        | Reset |
|------|------|------------|--------------------|-------|
| 13:0 | R/W  | DUTY_CYCLE | duty cycle for PWM | 0x0   |

Table 255: TRIPLE\_PWM\_FREQUENCY (0x5000340E)

| Bit  | Mode | Symbol | Description        | Reset |
|------|------|--------|--------------------|-------|
| 13:0 | R/W  | FREQ   | Freq for PWM 2 3 4 | 0x0   |

Table 256: TRIPLE\_PWM\_CTRL\_REG (0x50003410)

| 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<br>BLE | '1' = PWM 2 3 4 is enabled   | 0x0   |



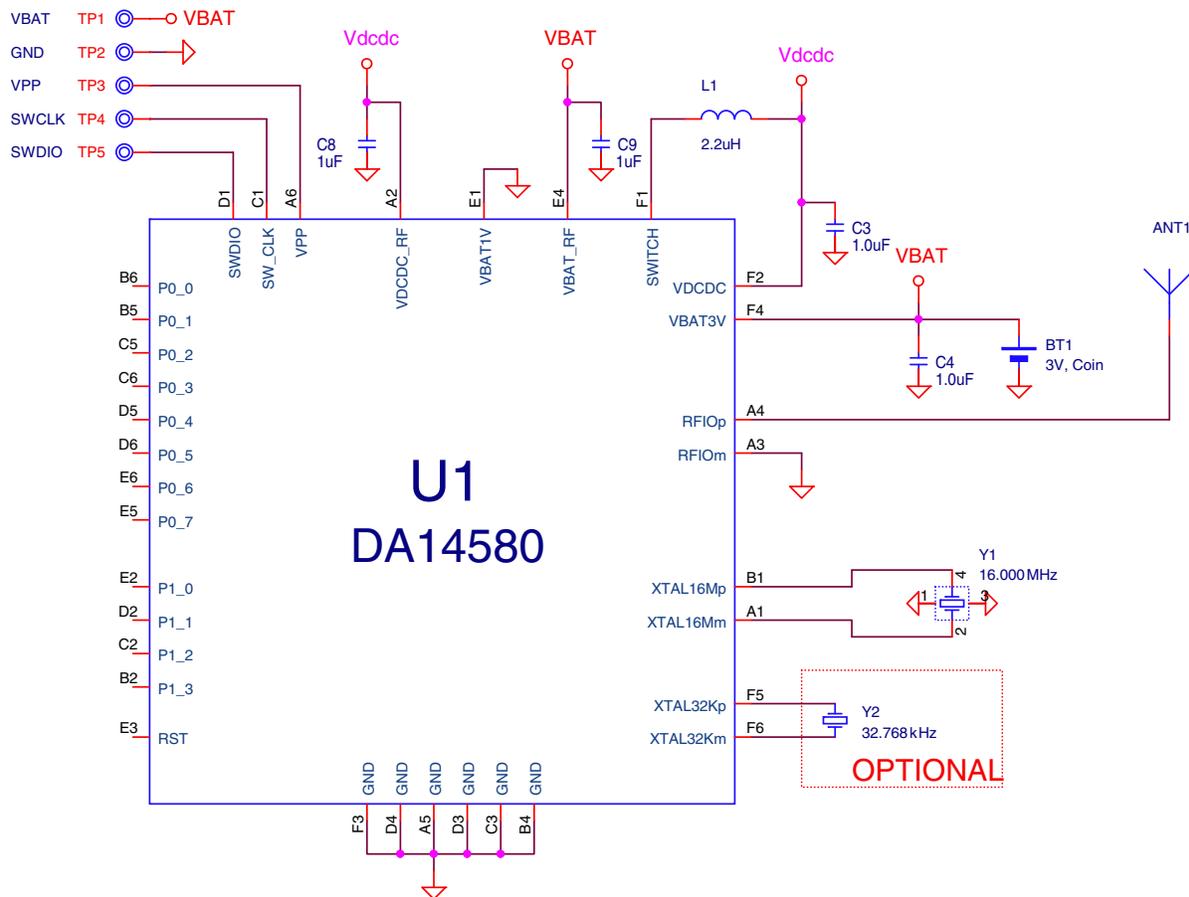


Figure 12: Lithium Coin Cell Powered System Diagram (Buck Mode)

Table 257: Absolute Maximum Ratings

| Parameter                         | Description  | Conditions  | Min  | Typ | Max                                | Unit |
|-----------------------------------|--|---|------|-----|------------------------------------|------|
| V <sub>PIN(LIM)</sub> (default)   | limiting voltage on a pin                              | Voltage between pin and GND<br>(Note 7)   | -0.1 |     | min{3.6, V <sub>BAT_RF</sub> +0.2} | V    |
| T <sub>STG</sub>                  | storage temperature                                    |   | -50  |     | 150                                | °C   |
| t <sub>R(SUP)</sub>               | supply rise time                                       | Power supply rise time  |      |     | 100                                | ms   |
| V <sub>BAT(LIM)</sub> (VBAT1V)    | limiting battery supply voltage                        | Supply voltage on VBAT1V in a boost converter application (VBAT3V is second output of boost-converter in this case)<br>(Note 7) | -0.1 |     | 3.6                                | V    |
| V <sub>BAT(LIM)</sub> (VBAT3V)    | limiting battery supply voltage                        | Supply voltage on VBAT3V and VBAT_RF in a buck-converter application, pin VBAT1V is connected to ground<br>(Note 7)             | -0.1 |     | 3.6                                | V    |
| V <sub>PIN(LIM)</sub> (1V2)       | limiting voltage on a pin                              | XTAL32Km, XTAL16Mp, XTAL16Mm<br>(Note 7)  | -0.2 |     | min(1.2, V <sub>BAT_RF</sub> +0.2) | V    |
| V <sub>PIN(LIM)</sub> (VDC DC_RF) | limiting voltage on the VDCDC_RF pin                   | Supply voltage on VDCDC_RF<br>(Note 7)  | -0.2 |     | min(3.3, V <sub>BAT_RF</sub> +0.2) | V    |
| V <sub>PIN(LIM)</sub> (XTAL32Kp)  | limiting voltage on a pin                              | XTAL32Kp  | -0.2 |     | min(1.5, V <sub>BAT_RF</sub> +0.2) | V    |
| V <sub>ESD(HBM)</sub> (WLCSP34)   | electrostatic discharge voltage (Human Body Model)     |   |      |     | 2000                               | V    |
| V <sub>ESD(HBM)</sub> (QFN40)     | electrostatic discharge voltage (Human Body Model)     |   |      |     | 4000                               | V    |
| V <sub>ESD(HBM)</sub> (QFN48)     | electrostatic discharge voltage (Human Body Model)     |   |      |     | 4000                               | V    |
| V <sub>ESD(MM)</sub> (WLCSP34)    | electrostatic discharge voltage (Machine Model)        |   |      |     | 200                                | V    |
| V <sub>ESD(MM)</sub> (QFN40)      | electrostatic discharge voltage (Machine Model)        |   |      |     | 200                                | V    |
| V <sub>ESD(MM)</sub> (QFN48)      | electrostatic discharge voltage (Machine Model)        |   |      |     | 200                                | V    |
| V <sub>ESD(CDM)</sub> (WLCSP34)   | electrostatic discharge voltage (Charged Device Model) |   |      |     | 500                                | V    |
| V <sub>ESD(CDM)</sub> (QFN40)     | electrostatic discharge voltage (Charged Device Model) |   |      |     | 1000                               | V    |
| V <sub>ESD(CDM)</sub> (QFN48)     | electrostatic discharge voltage (Charged Device Model) |   |      |     | 1000                               | V    |

**Note 7:** The device should not be exposed for prolonged periods of time to voltages between the Recommended Operating Conditions and the Absolute Maximum Ratings range.

**Table 258: Recommended Operating Conditions**

| Parameter                   | Description            | Conditions  | Min              | Typ | Max                                  | Unit |
|-----------------------------|------------------------|---|------------------|-----|--------------------------------------|------|
| V <sub>PP</sub>             | programming voltage    | Supply voltage on pin VPP during OTP programming; T <sub>J</sub> ≤ 50 °C  | 6.6              | 6.7 | 6.8                                  | V    |
| V <sub>BAT</sub> (VBAT1V)   | battery supply voltage | Supply voltage on VBAT1V in a boost converter application (VBAT3V is second output of boost-converter in this case) | 0.9              |     | 3.3                                  | V    |
| V <sub>BAT</sub> (VBAT3V)   | battery supply voltage | Supply voltage on VBAT3V and VBAT_RF in a buck-converter application, pin VBAT1V is connected to ground             | 2.35<br>(Note 8) |     | 3.3                                  | V    |
| V <sub>PIN</sub> (default)  | voltage on a pin       | Voltage between pin and GND   | 0                |     | min(3.3, V <sub>BAT_RF+</sub> - 0.2) | V    |
| V <sub>PIN</sub> (1V2)      | voltage on a pin       | XTAL32Km, XTAL16Mp, XTAL16Mm  | 0                |     | 1.2                                  | V    |
| V <sub>PIN</sub> (VDCDC_RF) | voltage on a pin       | Supply voltage on VDCDC_RF  | 0                |     | 3.3                                  | V    |
| T <sub>A</sub>              | ambient temperature    |   | -40              |     | 85                                   | °C   |

**Note 8:** Cold boot should not be performed if voltage is less than 2.5 V because of possible corruption during OTP data mirroring. Trim values programmed in the OTP as well as the application image, should be copied into RAM while VBAT3V ≥ 2.5 V.

**Table 259: DC Characteristics**

| Parameter                             | Description            | Conditions  | Min | Typ  | Max | Unit |
|---------------------------------------|------------------------|---|-----|------|-----|------|
| I <sub>BAT</sub> (DP_SLP)_BOOST_1kB   | battery supply current | Boost configuration in deep-sleep with 1 kB retention RAM active, running from RC32K oscillator at lowest frequency |     | 0.48 |     | μA   |
| I <sub>BAT</sub> (DP_SLP)_BOOST_2kB   | battery supply current | Boost configuration in deep-sleep with 2 kB retention RAM active, running from XTAL32K oscillator                   |     | 0.55 |     | μA   |
| I <sub>BAT</sub> (DP_SLP)_BOOST_8kB   | battery supply current | Typical boost-application in deep-sleep with 8 kB retention RAM active, running from XTAL32K oscillator             |     | 0.7  | 2   | μA   |
| I <sub>BAT</sub> (EXT_SLP)_BOOST_43KB | battery supply current | Typical boost-application in extended-sleep mode with 42 kB (SysRAM) and 1 kB (RetRAM) retained                     |     | 1.37 |     | μA   |

Table 259: DC Characteristics

| Parameter                        | Description            | Conditions   | Min | Typ  | Max | Unit    |
|----------------------------------|------------------------|--|-----|------|-----|---------|
| $I_{BAT}(EXT\_SLP)\_BOOST\_50kB$ | battery supply current | Typical boost-application in extended-sleep mode with 42 kB (SysRAM) and 8 kB (RetRAM) retained                    |     | 1.5  | 3   | $\mu A$ |
| $I_{BAT}(DP\_SLP)\_BUCK\_1kB$    | battery supply current | Buck configuration in deep-sleep with 1 kB retention RAM active, running from RC32K oscillator at lowest frequency |     | 0.4  |     | $\mu A$ |
| $I_{BAT}(DP\_SLP)\_BUCK\_2kB$    | battery supply current | Buck configuration in deep-sleep with 2 kB retention RAM active, running from XTAL32K oscillator                   |     | 0.45 |     | $\mu A$ |
| $I_{BAT}(DP\_SLP)\_BUCK\_8kB$    | battery supply current | Typical buck-application in deep-sleep with 8 kB retention RAM active, running from XTAL32K oscillator             |     | 0.6  | 2   | $\mu A$ |
| $I_{BAT}(EXT\_SLP)\_BUCK\_43kB$  | battery supply current | Typical buck-application in extended-sleep mode with 42 kB (SysRAM) and 1 kB (RetRAM) retained                     |     | 1.2  |     | $\mu A$ |
| $I_{BAT}(EXT\_SLP)\_BUCK\_50kB$  | battery supply current | Typical buck-application in extended-sleep mode with 42 kB (SysRAM) and 8 kB (RetRAM) retained                     |     | 1.4  | 3   | $\mu A$ |
| $I_{BAT}(ACT\_RX)\_BOOST$        | battery supply current | Typical application with boost converter and receiver active   |     | 13.4 | 16  | mA      |
| $I_{BAT}(ACT\_TX)\_BOOST$        | battery supply current | Typical application with boost converter and transmitter active  |     | 12.4 | 15  | mA      |
| $I_{BAT}(ACT\_RX)\_BUCK$         | battery supply current | Typical application with buck converter and receiver active  |     | 5.1  | 6   | mA      |
| $I_{BAT}(ACT\_TX)\_BUCK$         | battery supply current | Typical application with buck converter and transmitter active   |     | 4.8  | 6   | mA      |

Table 260: Timing Characteristics

| Parameter        | Description  | Conditions   | Min | Typ             | Max | Unit |
|------------------|--------------|--|-----|-----------------|-----|------|
| $t_{STA}(BOOST)$ | startup time | Boost-mode; time from deep-sleep to software start.<br>Typical application, running from retention RAM on 16 MHz RC oscillator |     | 1.2<br>(Note 9) |     | ms   |

Table 260: Timing Characteristics

| Parameter       | Description  | Conditions  | Min | Typ           | Max | Unit |
|-----------------|--------------|---|-----|---------------|-----|------|
| $t_{STA}(BUCK)$ | startup time | Buck-mode; time from deep-sleep to software start.<br>Typical application, running from retention RAM on 16 MHz RC oscillator |     | 1<br>(Note 9) |     | ms   |

Note 9: Worst-case value under Normal Operating Conditions.

Table 261: 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   | 10  |     | 12   | pF       |
| $C_0(16M)$                   | shunt capacitance            |   |     |     | 5    | pF       |
| $\Delta f_{XTAL}(16M)$       | crystal frequency tolerance  | After optional trimming; including aging and temperature drift<br>(Note 10)                                 | -20 |     | 20   | ppm      |
| $\Delta f_{XTAL}(16M)_{UNT}$ | crystal frequency tolerance  | Untrimmed; including aging and temperature drift<br>(Note 11)   | -40 |     | 40   | ppm      |
| $P_{DRV(MAX)}(16M)$          | maximum drive power          | (Note 12)   | 100 |     |      | $\mu W$  |
| $V_{CLK(EXT)}(16M)$          | external clock voltage       | Only in case of an external reference clock on XTAL16Mp (XTAL16Mm floating or connected to mid-level 0.6 V) | 1   | 1.2 |      | V        |
| $\phi_N(EXTERNAL)16M$        | phase noise                  | $f_C = 50$ kHz<br>in case of an external reference clock  |     |     | -130 | dBc/Hz   |

Note 10: Using the internal varicaps a wide range of crystals can be trimmed to the required tolerance.

Note 11: Maximum allowed frequency tolerance for compensation by the internal varicap trimming mechanism.

Note 12: Select a crystal which can handle a drive-level equal or more than this specification

Table 262: 16 MHz Crystal Oscillator: Timing Characteristics

| Parameter            | Description                     | Conditions | Min | Typ | Max | Unit |
|----------------------|---------------------------------|------------|-----|-----|-----|------|
| $t_{STA(XTAL)}(16M)$ | crystal oscillator startup time |            | 0.5 | 2   | 3   | ms   |



Table 263: 32 kHz Crystal Oscillator: Recommended Operating Conditions

| Parameter                  | Description                                   | Conditions  | Min  | Typ    | Max | Unit |
|----------------------------|---|---|------|--------|-----|------|
| V <sub>CLK(EXT)(32K)</sub> | external clock voltage                        | peak-peak voltage of external clock at XTAL32Kp, pin XTAL32Km floating. note: XTAL32Kp is internally AC coupled | 0.1  | 0.2    | 1.5 | V    |
| f <sub>XTAL(32k)</sub>     | crystal oscillator frequency                  | frequency range for an external clock (for a crystal, use either 32.000 kHz or 32.768 kHz)                      | 10   | 32.768 | 100 | kHz  |
| ESR(32k)                   | equivalent series resistance                  |   |      |        | 100 | kΩ   |
| C <sub>L(32k)</sub>        | load capacitance                              | no external capacitors are required for a 6 pF or 7 pF crystal  | 6    | 7      | 9   | pF   |
| C <sub>0(32k)</sub>        | shunt capacitance                             |   |      | 1      | 2   | pF   |
| Δf <sub>XTAL(32k)</sub>    | crystal frequency tolerance (including aging) | Timing accuracy is dominated by crystal accuracy. A much smaller value is preferred                             | -250 |        | 250 | ppm  |
| P <sub>DRV(MAX)(32k)</sub> | maximum drive power                           | <b>(Note 13)</b>  | 0.1  |        |     | μW   |

**Note 13:** Select a crystal that can handle a drive-level of at least this specification.

Table 264: 32 kHz Crystal Oscillator: Timing Characteristics

| Parameter                   | Description                     | Conditions  | Min | Typ | Max | Unit |
|-----------------------------|---------------------------------|---|-----|-----|-----|------|
| t <sub>STA(XTAL)(32k)</sub> | crystal oscillator startup time | Typical application, time until 1000 clocks are detected <b>(Note 14)</b> |     | 0.4 |     | s    |

**Note 14:** This parameter is very much dependent on crystal parameters

Table 265: DC-DC Converter: Recommended Operating Conditions

| Parameter                | Description                | Conditions   | Min | Typ | Max | Unit |
|--------------------------|----------------------------|--|-----|-----|-----|------|
| L                        | effective inductance       |  | 1.5 | 2.2 | 3   | μH   |
| C <sub>OUT(VDCDC)</sub>  | effective load capacitance | VDCDC and VDDCRF combined <b>(Note 15)</b>   | 0.5 | 1   | 10  | μF   |
| C <sub>OUT(VBAT3V)</sub> | effective load capacitance | VBATRF and VBAT3V combined are the second output of the boost-converter <b>(Note 15)</b> | 0.5 | 1   | 10  | μF   |

**Note 15:** A low value will result in lowest power consumption, keep this value at 1 uF or 2 uF.

Table 266: DC-DC Converter: DC Characteristics

| Parameter            | Description    | Conditions       | Min | Typ  | Max | Unit |
|----------------------|----------------|------------------|-----|------|-----|------|
| V <sub>O(BUCK)</sub> | output voltage | default settings |     | 1.41 |     | V    |

Table 266: DC-DC Converter: DC Characteristics

| Parameter                        | Description                   | Conditions  | Min  | Typ   | Max | Unit |
|----------------------------------|-------------------------------|---|------|-------|-----|------|
| V <sub>O</sub> (BOOST)           | output voltage                | default settings, VDCDC   |      | 1.41  |     | V    |
| $\eta_{CONV\_MAX}(BUCK)$         | maximum conversion efficiency |   |      | 86    |     | %    |
| $\eta_{CONV\_MAX}(BOOST)$        | maximum conversion efficiency |   |      | 80    |     | %    |
| $\Delta V_O / \Delta V_I(BUCK)$  | line regulation               | $2.35\text{ V} \leq V_{BAT3V} \leq 3.3\text{ V}$                            | -2   | 0.7   | 2   | %/V  |
| $\Delta V_O / \Delta V_I(BOOST)$ | line regulation               | $0.9\text{ V} \leq V_{BAT1V} \leq 1.2\text{ V}$<br>(Note 16)                | -2   | 1     | 4   | %/V  |
| $\Delta V_O / \Delta I_L(BUCK)$  | load regulation               | $V_{BAT3V} = 2.5\text{ V}$  | -0.2 | -0.02 | 0.2 | %/mA |
| $\Delta V_O / \Delta I_L(BOOST)$ | load regulation               | $V_{BAT1V} = 1.2\text{ V}$  | -0.2 | -0.07 | 0.2 | %/mA |
| V <sub>RPL</sub> (BUCK)          | ripple voltage                | buck mode; RMS ripple voltage   |      | 5     |     | mV   |
| V <sub>RPL</sub> (BOOST)         | ripple voltage                | $V_{BAT1V} \leq 1.2\text{ V}$ , boost mode; RMS ripple voltage<br>(Note 16) |      | 8     |     | mV   |

Note 16: When V<sub>BAT1V</sub> > VDCDC\_nominal, VDCDC will follow V<sub>BAT1V</sub>.

Table 267: Digital Input/Output: 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>IH</sub> (RST)                 | HIGH level input voltage  | RST pin  | 0.84 |     |      | V             |
| V <sub>IL</sub> (RST)                 | LOW level input voltage   | RST pin  |      |     | 0.36 | V             |
| V <sub>OH</sub> (V <sub>BAT1V</sub> ) | HIGH level output voltage | $I_{out} = -250\ \mu\text{A}$ , $V_{BAT3V} = 2.35\text{ V}$ , $V_{BAT1V} = 0.9\text{ V}$         | 0.72 |     |      | V             |
| V <sub>OH</sub> (V <sub>BAT3V</sub> ) | HIGH level output voltage | $I_{out} = -4.8\text{ mA}$ , $V_{BAT3V} = 2.35\text{ V}$ , $V_{BAT1V} = 0\text{ V}$<br>(Note 17) | 1.88 |     |      | V             |
| V <sub>OL</sub> (V <sub>BAT1V</sub> ) | LOW level output voltage  | $I_{out} = 250\ \mu\text{A}$ , $V_{BAT3V} = 2.35\text{ V}$ , $V_{BAT1V} = 0.9\text{ V}$          |      |     | 0.18 | V             |
| V <sub>OL</sub> (V <sub>BAT3V</sub> ) | LOW level output voltage  | $I_{out} = 4.8\text{ mA}$ , $V_{BAT3V} = 2.35\text{ V}$ , $V_{BAT1V} = 0\text{ V}$<br>(Note 18)  |      |     | 0.47 | V             |
| I <sub>IH</sub>                       | HIGH level input current  | $V_{in} = V_{BAT3V} = 2.5\text{ V}$  | -1   |     | 1    | $\mu\text{A}$ |
| I <sub>IL</sub>                       | LOW level input current   | $V_{in} = V_{SS} = 0\text{ V}$   | -1   |     | 1    | $\mu\text{A}$ |
| I <sub>IH</sub> (PD)                  | HIGH level input current  | $V_{in} = V_{BAT3V} = 2.5\text{ V}$  | 50   |     | 150  | $\mu\text{A}$ |
| I <sub>IL</sub> (PU)                  | LOW level input current   | $V_{in} = V_{SS} = 0\text{ V}$   | -150 |     | -50  | $\mu\text{A}$ |
| I <sub>IH</sub> (RST)                 | HIGH level input current  | RST pin, $V(RST) = 1.2\text{ V}$   | 25   |     | 75   | $\mu\text{A}$ |

Note 17: In Boost mode the output source current is limited to  $I_{out} = -250\ \mu\text{A}$ .

Note 18: In Boost mode the output sink current is limited to  $I_{out} = 250\ \mu\text{A}$ .

Table 268: General Purpose ADC: Recommended Operating Conditions

| Parameter              | Description                 | Conditions | Min | Typ | Max | Unit |
|------------------------|-----------------------------|------------|-----|-----|-----|------|
| N <sub>BIT</sub> (ADC) | number of bits (resolution) |            |     | 10  |     | bit  |

Table 269: General Purpose ADC: DC Characteristics

| Parameter           | Description                       | Conditions                             | Min  | Typ   | Max  | Unit |
|---------------------|-----------------------------------|--|------|-------|------|------|
| V <sub>I(ZS)</sub>  | zero-scale input voltage          | single-ended, calibrated at zero input | -2.5 | 0     | 2.5  | mV   |
| V <sub>I(FS)</sub>  | full-scale input voltage          | single-ended, calibrated at zero input | 1150 | 1180  | 1250 | mV   |
| V <sub>I(FSN)</sub> | negative full-scale input voltage | differential, calibrated at zero input |      | -1180 |      | mV   |
| V <sub>I(FSP)</sub> | positive full-scale input voltage | differential, calibrated at zero input |      | 1180  |      | mV   |
| INL                 | integral non-linearity            |  | -2   |       | 2    | LSB  |
| DNL                 | differential non-linearity        |  | -2   |       | 2    | LSB  |

Table 270: General Purpose ADC: Timing Characteristics

| Parameter               | Description     | Conditions   | Min | Typ  | Max | Unit |
|-------------------------|-----------------|--|-----|------|-----|------|
| t <sub>CONV</sub> (ADC) | conversion time | Excluding initial settling time of the LDO and the 3x-attenuation (if used):<br><br>LDO settling time is 20 μs (max), 3x-attenuation settling time = 1 μs (max)<br>Using internal ADC-clock (~200 MHz) |     | 0.25 | 0.4 | μs   |

Table 271: Radio: DC Characteristics

| Parameter               | Description            | Conditions  | Min | Typ | Max | Unit |
|-------------------------|------------------------|---|-----|-----|-----|------|
| I <sub>BAT</sub> (RF)RX | battery supply current | receive mode; radio receiver and synthesizer active; DCDC converter assumed ideal; T <sub>A</sub> = 25 °C<br><b>(Note 19)</b>     |     | 3.7 | 4.3 | mA   |
| I <sub>BAT</sub> (RF)TX | battery supply current | transmit mode; radio transmitter and synthesizer active; DCDC converter assumed ideal; T <sub>A</sub> = 25 °C<br><b>(Note 19)</b> |     | 3.4 | 4   | mA   |

**Note 19:** The DCDC-converter efficiency is assumed to be 100 % to enable benchmarking of the radio currents at battery supply domain (VBAT3V = 3 V).

Table 272: Radio: AC Characteristics

| Parameter                | Description                                       | Conditions   | Min | Typ   | Max | Unit |
|--------------------------|---|--|-----|-------|-----|------|
| P <sub>SENS(CLEAN)</sub> | sensitivity level                                 | DC-DC converter enabled; Dirty Transmitter disabled; T <sub>A</sub> = 25 °C; PER = 30.8 %<br><b>(Note 20)</b>  |     | -93   |     | dBm  |
| P <sub>SENS</sub>        | sensitivity level                                 | Normal Operating Conditions; DC-DC converter enabled; T <sub>A</sub> = 25 °C; PER = 30.8 %<br><b>(Note 20)</b>   |     | -92.5 |     | dBm  |
| P <sub>I(max)</sub>      | input power level                                 | DC-DC converter disabled; T <sub>A</sub> = 25 °C; PER ≤ 30.8 %<br><b>(Note 20)</b>   | 10  |       |     | dBm  |
| P <sub>INT(IMD)</sub>    | intermodulation distortion interferer power level | worst case interferer level @ f <sub>1</sub> , f <sub>2</sub> with 2f <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 %; T <sub>A</sub> = 25 °C<br><b>(Note 22)</b> | -35 | -31   |     | dBm  |
| CIR(0)                   | carrier to interferer ratio                       | n = 0; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz; T <sub>A</sub> = 25 °C<br><b>(Note 23)</b>  |     | 7     | 21  | dB   |
| CIR(1)                   | carrier to interferer ratio                       | n = ±1; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz; T <sub>A</sub> = 25 °C<br><b>(Note 23)</b>   |     | -3    | 15  | dB   |
| CIR(P2)                  | carrier to interferer ratio                       | n = +2 (image frequency); interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz; T <sub>A</sub> = 25 °C<br><b>(Note 23)</b>   |     | -20   | -9  | dB   |
| CIR(M2)                  | carrier to interferer ratio                       | n = -2; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz; T <sub>A</sub> = 25 °C<br><b>(Note 23)</b>   |     | -30   | -17 | dB   |
| CIR(P3)                  | carrier to interferer ratio                       | n = +3 (image frequency + 1 MHz); interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz; T <sub>A</sub> = 25 °C<br><b>(Note 23)</b>   |     | -30   | -15 | dB   |
| CIR(M3)                  | carrier to interferer ratio                       | n = -3; interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz; T <sub>A</sub> = 25 °C<br><b>(Note 23)</b>   |     | -35   | -27 | dB   |
| CIR(4)                   | carrier to interferer ratio                       | n  >= 4 (any other BLE channel); interferer @ f <sub>1</sub> = f <sub>0</sub> + n*1 MHz; T <sub>A</sub> = 25 °C<br><b>(Note 23)</b>  |     | -37   | -27 | dB   |

Table 272: Radio: AC Characteristics

| Parameter                       | Description                  | Conditions   | Min  | Typ   | Max   | Unit       |
|---------------------------------|------------------------------|--|------|-------|-------|------------|
| P <sub>BL(I)</sub>              | blocker power level          | 30 MHz ≤ f <sub>BL</sub> ≤ 2000 MHz; P <sub>WANTED</sub> = -67 dBm; T <sub>A</sub> = 25 °C<br><a href="#">(Note 24)</a>  | -5   |       |       | dBm        |
| P <sub>BL(II)</sub>             | blocker power level          | 2003 MHz ≤ f <sub>BL</sub> ≤ 2399 MHz; P <sub>WANTED</sub> = -67 dBm; T <sub>A</sub> = 25 °C<br><a href="#">(Note 24)</a>  | -15  |       |       | dBm        |
| P <sub>BL(III)</sub>            | blocker power level          | 2484 MHz ≤ f <sub>BL</sub> ≤ 2997 MHz; P <sub>WANTED</sub> = -67 dBm; T <sub>A</sub> = 25 °C<br><a href="#">(Note 24)</a>  | -15  |       |       | dBm        |
| P <sub>BL(IV)</sub>             | blocker power level          | 3000 MHz ≤ f <sub>BL</sub> ≤ 12.75 GHz; P <sub>WANTED</sub> = -67 dBm; T <sub>A</sub> = 25 °C<br><a href="#">(Note 24)</a>   | -5   |       |       | dBm        |
| P <sub>RSSI(min)</sub>          | RSSI power level             | absolute power level for RXRSSI[7:0] = 0; T <sub>A</sub> = 25 °C<br><a href="#">(Note 25)</a>  | -115 | -112  | -109  | dBm        |
| P <sub>RSSI(max)</sub>          | RSSI power level             | upper limit of monotonous range; T <sub>A</sub> = 25 °C  | -26  | -19   |       | dBm        |
| L <sub>ACC(RSSI)BO</sub><br>OST | level accuracy               | tolerance of 5 % to 95 % confidence interval of P <sub>RF</sub> : when RXRSSI[7:0] = X, 50 < X < 175; burst mode 1500 packets; T <sub>A</sub> = 25 °C; DC-DC converter in BOOST mode |      | 0     | 3     | dB         |
| L <sub>ACC(RSSI)BU</sub><br>CK  | level accuracy               | tolerance of 5 % to 95 % confidence interval of P <sub>RF</sub> : when RXRSSI[7:0] = X, 50 < X < 175; burst mode 1500 packets; T <sub>A</sub> = 25 °C; DC-DC converter in BUCK mode  |      | 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; T <sub>A</sub> = 25 °C  | 0.46 | 0.474 | 0.485 | dB/<br>LSB |
| ACP(2M)                         | adjacent channel power level | f <sub>OFFSET</sub> = 2 MHz; T <sub>A</sub> = 25 °C<br><a href="#">(Note 26)</a>   |      | -53   | -50   | dBm        |
| ACP(2M)(EOC)                    | adjacent channel power level | f <sub>OFFSET</sub> = 2 MHz; -40 °C ≤ T <sub>A</sub> ≤ +85 °C<br><a href="#">(Note 26)</a>   |      | -53   | -47   | dBm        |
| ACP(3M)                         | adjacent channel power level | f <sub>OFFSET</sub> ≥ 3 MHz; T <sub>A</sub> = 25 °C<br><a href="#">(Note 26)</a>   |      | -57   | -55   | dBm        |

Table 272: Radio: AC Characteristics

| Parameter            | Description                             | Conditions  | Min | Typ | Max | Unit |
|----------------------|---|---|-----|-----|-----|------|
| ACP(3M)(EOC)         | adjacent channel power level            | $f_{\text{OFFSET}} \geq 3 \text{ MHz}$ ; $-40 \text{ }^\circ\text{C} \leq T_A \leq +85 \text{ }^\circ\text{C}$<br>(Note 26) |     | -57 | -47 | dBm  |
| P <sub>O</sub>       | output power level                      | V <sub>DD</sub> = 3 V; maximum gain; T <sub>A</sub> = 25 °C   | -2  | -1  | 0   | dBm  |
| P <sub>O</sub> (HD2) | output power level (second harmonic)    | V <sub>DD</sub> = 3 V; maximum gain; T <sub>A</sub> = 25 °C   |     | -54 | -40 | dBm  |
| P <sub>O</sub> (HD3) | output power level (third harmonic)     | V <sub>DD</sub> = 3 V; maximum gain; T <sub>A</sub> = 25 °C   |     | -56 | -40 | dBm  |
| P <sub>O</sub> (HD4) | output power level (fourth harmonic)    | V <sub>DD</sub> = 3 V; maximum gain; T <sub>A</sub> = 25 °C   |     | -70 | -40 | dBm  |
| P <sub>O</sub> (HD5) | output power level (fifth harmonic)     | V <sub>DD</sub> = 3 V; maximum gain; T <sub>A</sub> = 25 °C   |     | -70 | -40 | dBm  |
| P <sub>O</sub> (NFM) | output power level in 'Near Field Mode' | V <sub>DD</sub> = 3 V; maximum gain; T <sub>A</sub> = 25 °C<br>(Note 27)  | -25 | -20 | -15 | dBm  |

**Note 20:** Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.4.1.

**Note 21:** Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.4.2.

**Note 22:** Measured according to Bluetooth® Core Technical Specification document, version 4.2, 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 23:** Measured according to Bluetooth® Core Technical Specification document, version 4.2, volume 6, section 4.2.

**Note 24:** Measured according to Bluetooth® Core Technical Specification document, version 4.2, volume 6, section 4.3. Due to limitations of the measurement equipment, levels of -5 dBm should be interpreted as  $> -5 \text{ dBm}$ .

**Note 25:**  $\text{PRF} = \text{PRSSI}(\text{min}) + \text{LRES}(\text{RSSI}) \times \text{RXRSSI}[7:0] \pm \text{LACC}(\text{RSSI})$ . Thanks to constant gain biasing of RF part in the receiver, the RSSI can be used to estimate absolute power levels, rather than mere level changes. Even across the full temperature range the variation is limited.

**Note 26:** Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.2.3.

**Note 27:** To activate the "Near Field Mode", program address 0x50002418 with the value 0x0030.

Table 273: Stable Low Frequency RCX Oscillator: Timing Characteristics

| Parameter                     | Description                    | Conditions                                 | Min  | Typ | Max  | Unit |
|-------------------------------|--------------------------------|--|------|-----|------|------|
| f <sub>RC</sub> (RCX)         | RCX oscillator frequency       | default setting, buck mode only            | 5    | 10  | 40   | kHz  |
| Δf <sub>RC</sub> (RCX)        | RCX oscillator frequency drift | buck mode only<br>(Note 28)                | -500 |     | 500  | ppm  |
| ΔT <sub>A</sub> /Δt(RCX)100ms | ambient temperature gradient   | buck mode only; connection interval 100 ms |      |     | 0.66 | °C/s |
| ΔT <sub>A</sub> /Δt(RCX)4s    | ambient temperature gradient   | buck mode only; connection interval 4 s    |      |     | 0.33 | °C/s |

**Note 28:** Maximum recommended connection interval (including slave latency) for the RCX usage is 2 s.

## 7 Package Information

### 7.1 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.

WLCSP packages are qualified for MSL 1.

QFN 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 |

### 7.2 WLCSP HANDLING

Manual handling of WLCSP packages should be reduced to the absolute minimum. In cases where it is still necessary, a vacuum pick-up tool should be used. In extreme cases plastic tweezers could be used, but metal tweezers are not acceptable, since contact may easily damage the silicon chip.

Removal will cause damage to the solder balls and therefore a removed sample cannot be reused.

WLCSP is sensitive to visible and infrared light. Precautions should be taken to properly shield the chip in the final product.

### 7.3 SOLDERING INFORMATION

Refer to the JEDEC standard J-STD-020 for relevant soldering information.

This document can be downloaded from <http://www.jedec.org>

7.4 PACKAGE OUTLINES

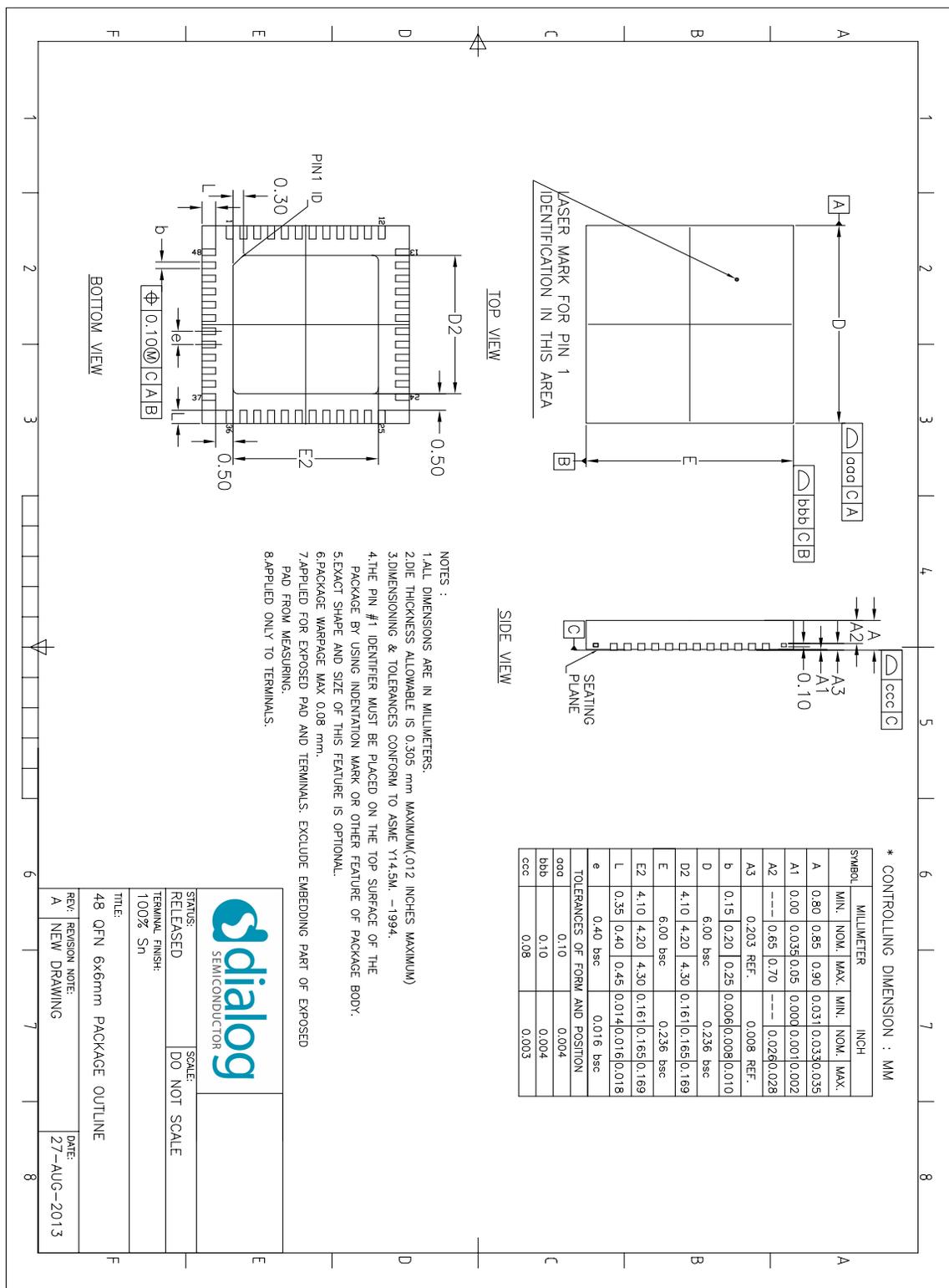


Figure 13: QFN48 Package Outline Drawing



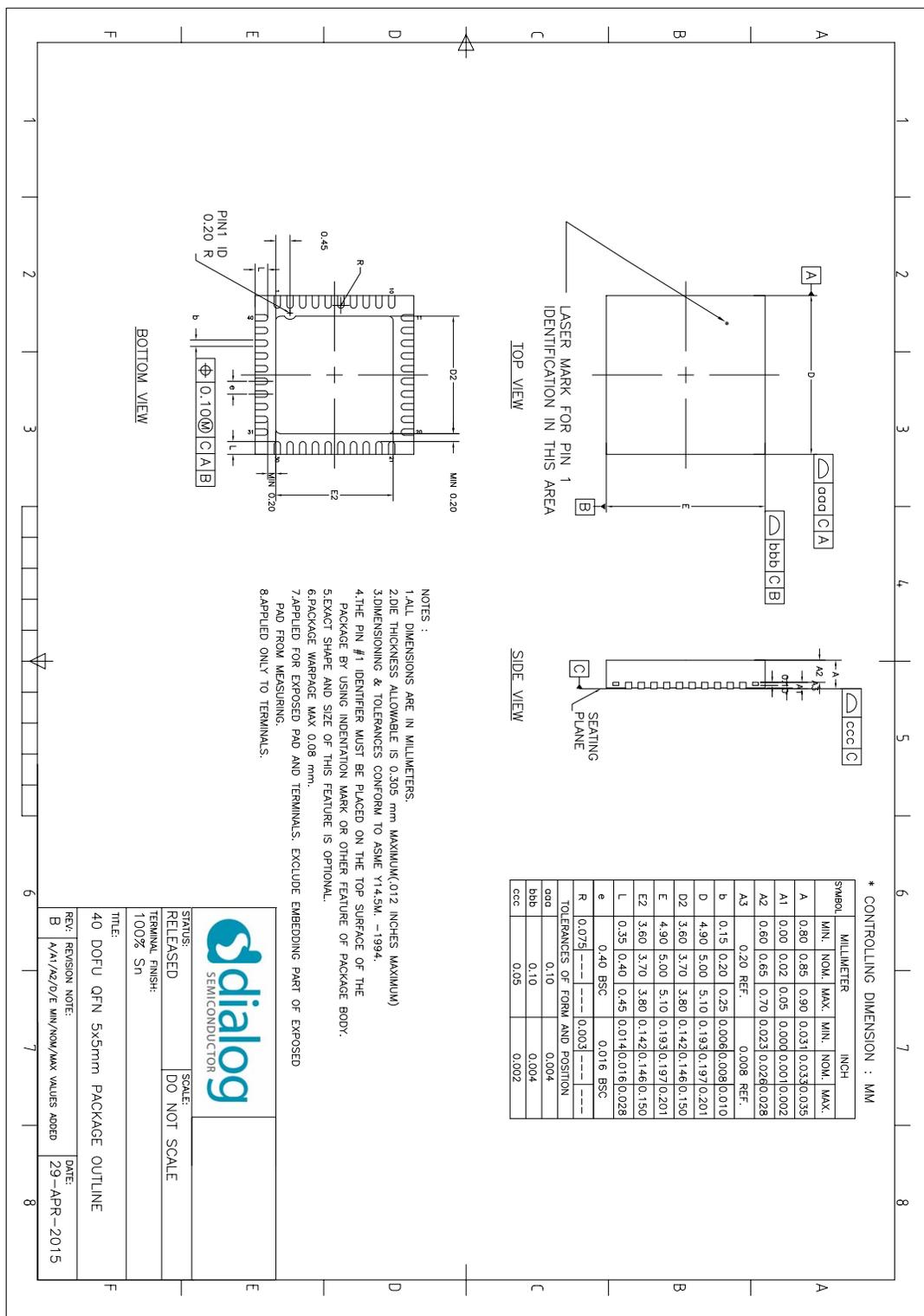


Figure 14: QFN40 Package Outline Drawing

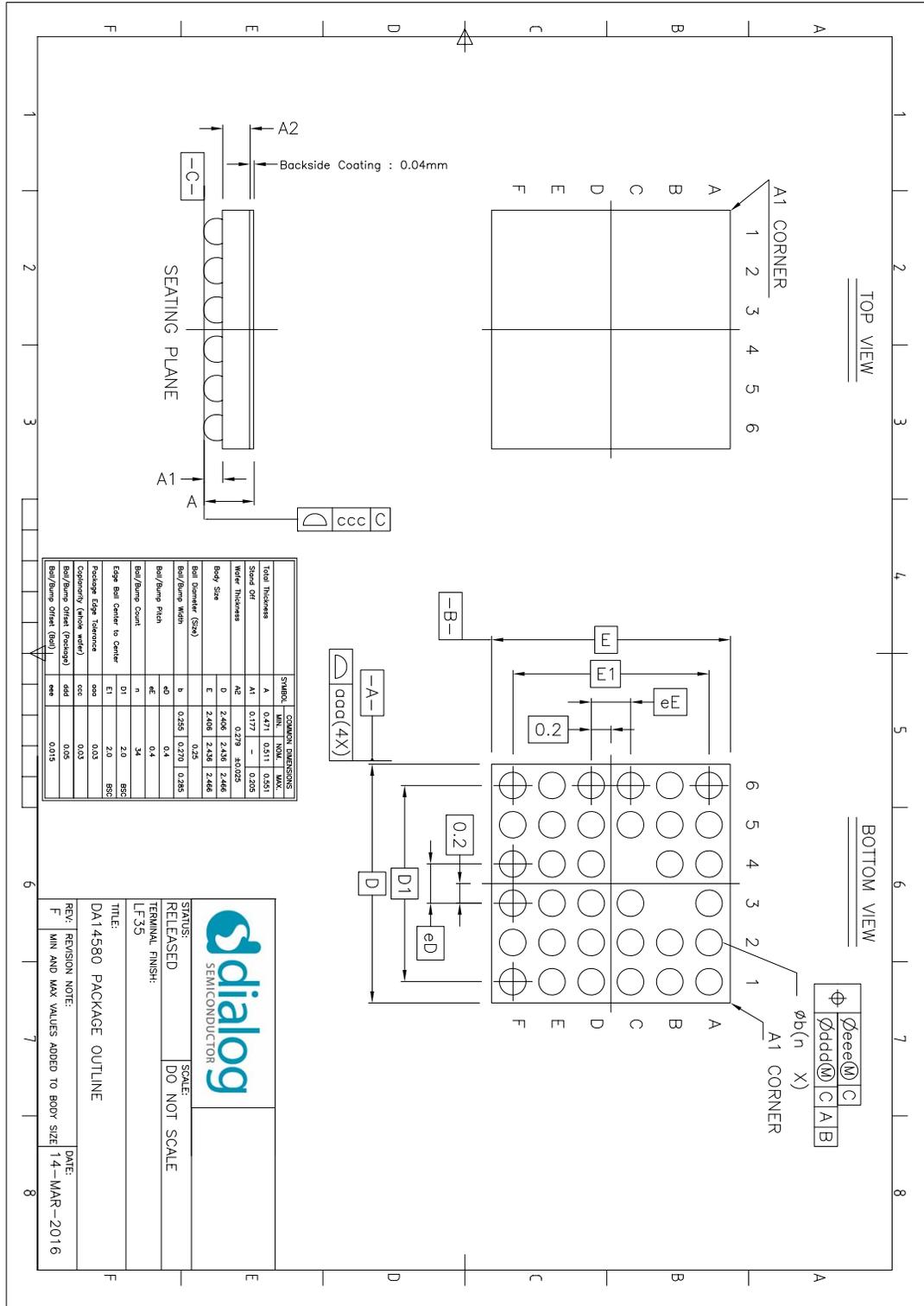


Figure 15: WLCSP34 Package Outline Drawing

## 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 characterization 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. Major specification changes are communicated via Customer Product Notifications. Datasheet changes are communicated via <a href="http://www.dialog-semiconductor.com">www.dialog-semiconductor.com</a> . |
| 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](http://www.dialog-semiconductor.com), available on the company website ([www.dialog-semiconductor.com](http://www.dialog-semiconductor.com)) unless otherwise stated.

Dialog and the Dialog logo are trademarks of Dialog Semiconductor plc or its subsidiaries. All other product or service names are the property of their respective owners.

© 2014 Dialog Semiconductor. All rights reserved.

## RoHS Compliance

Dialog Semiconductor's suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. 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

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Dialog Semiconductor:](#)

[DA14580-01AT2](#) [DA14580-01UNA](#) [DA14580-01A31](#) [DA14580-01AT1](#) [DA14580-01A32](#)