Bluetooth 5.0 SoC with Audio Interface

General Description

The DA14585 is an ultra-low power SoC integrating a 2.4 GHz transceiver and an *ARM Cortex-M0*® microcontroller with 96kB of RAM and 64kB of One-Time Programmable (OTP) memory. It offers a very fast boot time (<50ms) and supports up to 8 BLE connections. It can be used as a standalone application processor or as a data pump in hosted systems.

The radio transceiver, the baseband processor and the qualified *Bluetooth® LE* stack is fully compliant with the *Bluetooth® Low Energy 5.0 standard*.

The DA14585 has dedicated hardware for the Link Layer implementation of *Bluetooth® Low Energy* and interface controllers for enhanced connectivity capabilities.

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 device is suitable for remote control units (RCU) requiring support for voice commands, wireless sensor nodes, Bluetooth Mesh applications, fitness trackers, toys and HID devices (keyboards, mice, etc.).

Key Features

- Complies with Bluetooth V5.0, ETSI EN 300 328 and EN 300 440 Class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan)
- Supports up to 8 Bluetooth LE connections
- Fast cold boot in less than 50 ms
- Processing power
  - 16 MHz 32 bit ARM Cortex-M0 with SWD interface
  - Dedicated Link Layer Processor
  - AES-128 bit encryption Processor
- Memories
  - 64 kB One-Time-Programmable (OTP) memory
  - 96 kB Data/Retention SRAM
  - 128 kB ROM
- Power management
  - Integrated Buck/Boost DCDC converter
  - P0, P1 and P2 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
  - 1.8 V cold boot support
  - 10-bit ADC for battery voltage measurement
- Digital controlled oscillators
  - 16 MHz crystal (±20 ppm max) and RC oscillator
  - 32 kHz crystal (±50 ppm, ±500 ppm max) and RCX oscillator
- Flexible Reset Circuitry
  - System & Power On Reset in a single pin
- General purpose, Capture and Sleep timers
- Digital interfaces
  - Gen. purpose I/Os: 14 (WLCSP34), 25 (QFN40), 32 (QFN48)
  - 2 x UARTs with hardware flow control up to 1 Mbps
  - 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, 2.40 mm x 2.66 mm
  - QFN 40 pins, 5 mm x 5 mm
  - QFN 48 pins, 6 mm x 6 mm

Applications

- Voice-controlled remote controls
- Beacons
- (Multi-sensor) Wearable devices
  - Fitness trackers
  - Consumer health
- Smartwatches
- Human interface devices
  - Keyboard
  - Mouse
- Toys
- Consumer appliances

Key Benefits

- Lowest power consumption
- Smallest system size
- Lowest system cost

System Diagram

Figure 1: System Diagram

Contents

Figures

Tables

1 The DA14585 Block Diagram

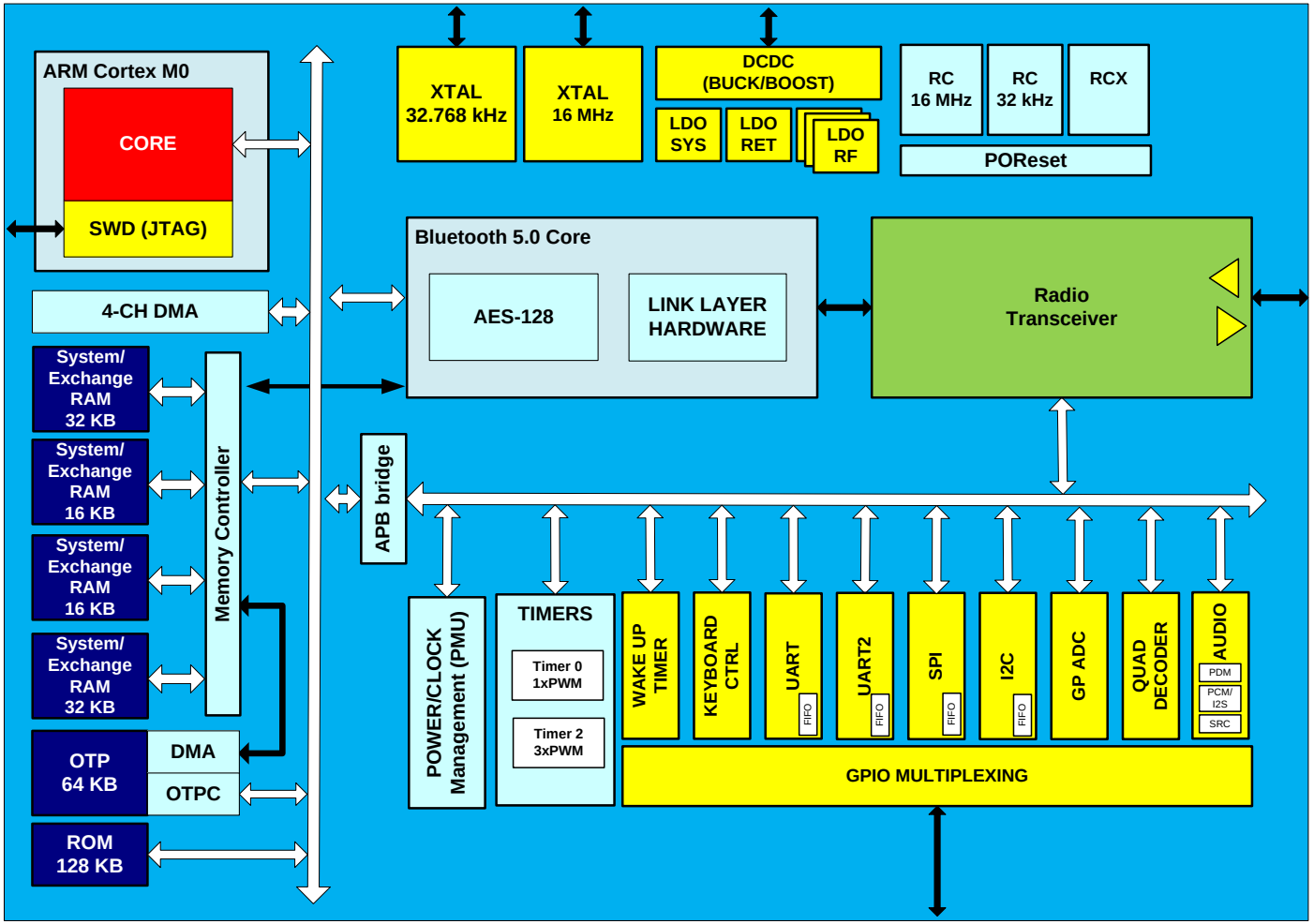


Figure 2: DA14585 Block Diagram

2 Pinout

The DA14585 comes in three packages:

- A Wafer Level Chip Scale Package (WLCSP) with 34 balls
- A Quad Flat Package No Leads (QFN) with 40 pins
- A Quad Flat Package No Leads (QFN) with 48 pins

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

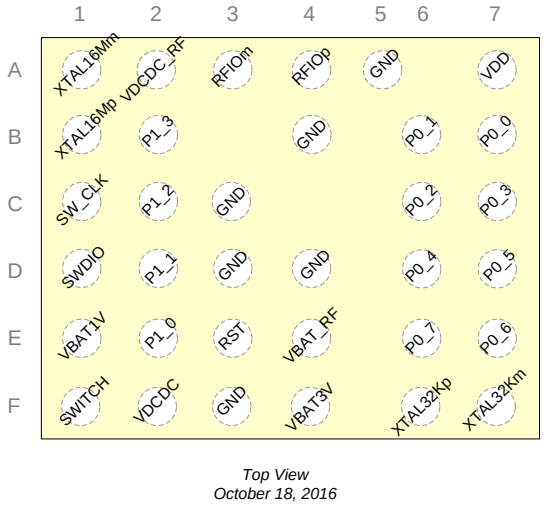


Figure 3: WLCSP34 Ball Assignment (Top View)

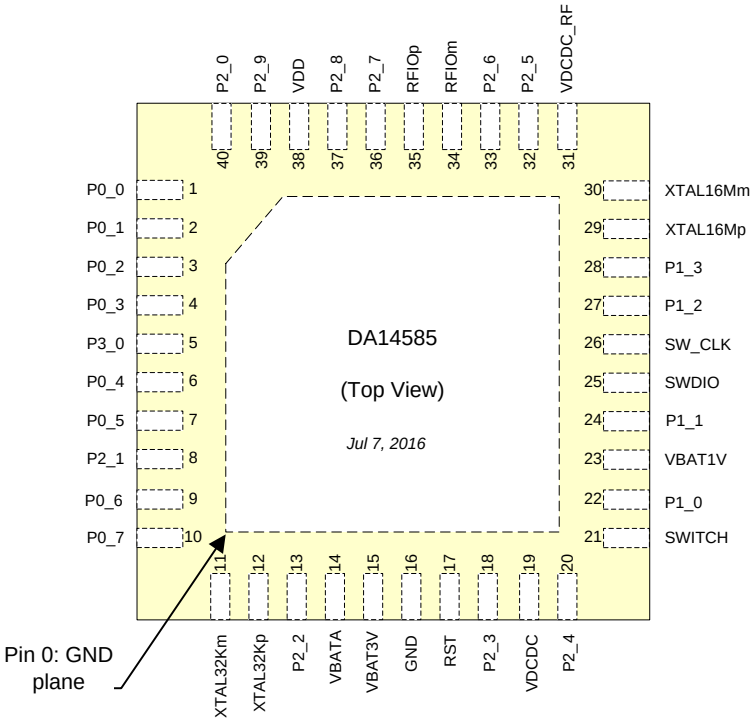


Figure 4: QFN40 Pin Assignment

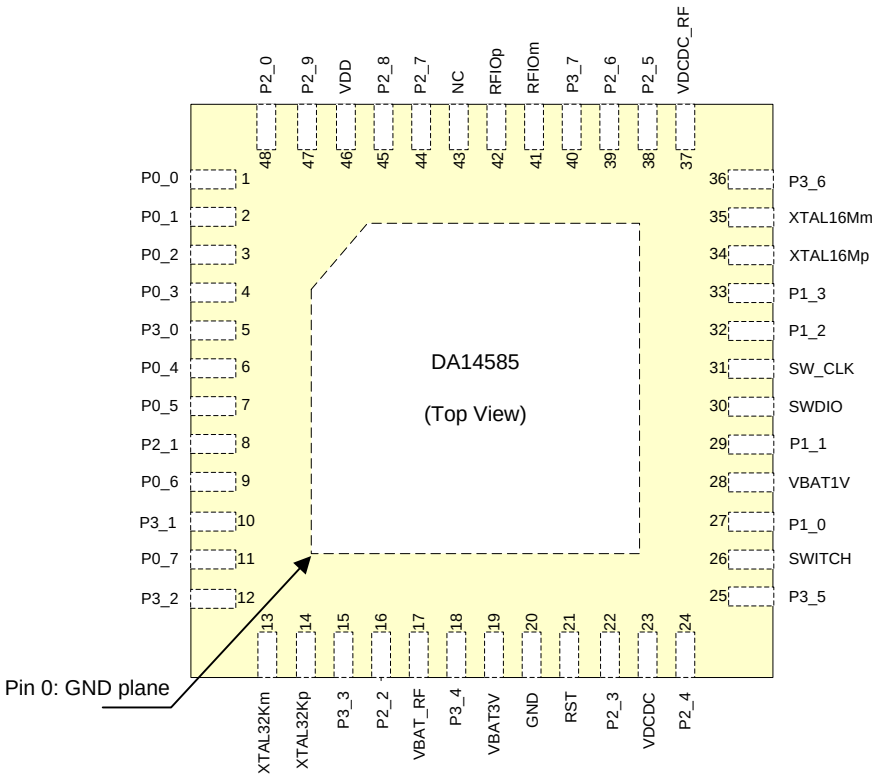


Figure 5: QFN48 Pin Assignment

Table 1: Pin Description

Pin Name	Type	Drive (mA)	Reset State	Description
General Purpose I/Os				
P0_0 P0_1 P0_2 P0_3 P0_4 P0_5 P0_6 P0_7	DIO DIO DIO DIO DIO DIO DIO DIO	3.5	I-PD I-PD I-PD I-PD I-PD I-PD I-PD 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 modes. Contains state retention mechanism during power down.
P1_0 P1_1 P1_2 (Note 1) P1_3 (Note 1) P1_4/SWCLK P1_5/SW_DIO	DIO DIO DIO DIO DIO DIO	3.5	I-PD I-PD I-PD I-PD I-PD 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 modes. Contains state retention mechanism during power down. This signal is the JTAG clock by default This signal is the JTAG data I/O by default

Pin Name	Type	Drive (mA)	Reset State	Description
P2_0 P2_1 P2_2 P2_3 P2_4 P2_5 P2_6 P2_7 P2_8 P2_9	DIO DIO DIO DIO DIO DIO DIO DIO DIO DIO	3.5	I-PD I-PD I-PD I-PD I-PD I-PD I-PD I-PD I-PD 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 modes. Contains state retention mechanism during power down. NOTE: This port is only available on the QFN40 and the QFN48 packages.
P3_0 P3_1 P3_2 P3_3 P3_4 P3_5 P3_6 P3_7	DIO DIO DIO DIO DIO DIO DIO DIO	3.5	I-PD I-PD I-PD I-PD I-PD I-PD I-PD 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 modes. Contains state retention mechanism during power down. NOTE: P3_0 is available on the QFN40 package. The full port is only available on the QFN48 package.
Debug Interface				
SWDIO/P1_5	DIO	3.5	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	3.5	I-PD	INPUT JTAG clock signal. Can also be used as a GPIO
Clocks				
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
Quadrature Decoder				
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
SPI Bus Interface				
SPI_CLK	DO			INPUT/OUTPUT. SPI Clock. Mapped on Px ports
SPI_DI	DI			INPUT. SPI Data input. Mapped on Px ports (Note 2)
SPI_DO	DO			OUTPUT. SPI Data output. Mapped on Px ports (Note 3)
SPI_EN	DO			OUTPUT. SPI Chip enable. Mapped on Px ports

Pin Name	Type	Drive (mA)	Reset State	Description
I2C Bus Interface				
SDA	DIO/DIOD			INPUT/OUTPUT. I2C bus Data with open drain port. Mapped on Px ports. The mapped Px pin is automatically configured with a pull up resistor (25kΩ) to the power supply configured by the register Px_PADPWR_CTRL_REG
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. The mapped Px pin is automatically configured with a pull up resistor (25kΩ) to the power supply configured by the register Px_PADPWR_CTRL_REG
UART Interface				
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
Analog Interface				
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]
Radio Transceiver				
RFIOp	AIO			RF input/output. Impedance 50 Ω
RFIOm	AIO			RF ground
Miscellaneous				
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
VDD	AI			INPUT. This pin is used for testing purposes only. Normal operation: leave VDD floating
Power Supply				
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.

Pin Name	Type	Drive (mA)	Reset State	Description
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

Note 1 An increased Packet Error rate might occur if GPIO P1-2 and P1-3 is toggling. The root cause of this problem is the close location of a few GPIO pins to the on-chip 16 MHz XTAL oscillator circuitry. Toggling of these GPIOs might corrupt the clock, which will be visible in the Packet Error Rate.

Note 2 Data input only. MOSI in SPI slave mode, MISO in SPI master mode

Note 3 Data output only. MISO in SPI slave mode, MOSI in SPI master mode

3 Specifications

All MIN/MAX specification limits are guaranteed by design, production testing and/or statistical characterization. Typical values are based on characterization results at default measurement conditions and are informative only.

Default measurement conditions (unless otherwise specified): V_{BAT}(VBAT3V) = 3.0 V (buck mode), V_{BAT}(VBAT1V) = 1.2 V (boost mode), T_A = 25 °C. All radio measurements are performed with standard RF measurement equipment providing a source/load impedance of 50 Ω.

The specifications in the following tables are valid for the reference circuits shown in Figure 6 (Boost mode) and Figure 7 (Buck mode).

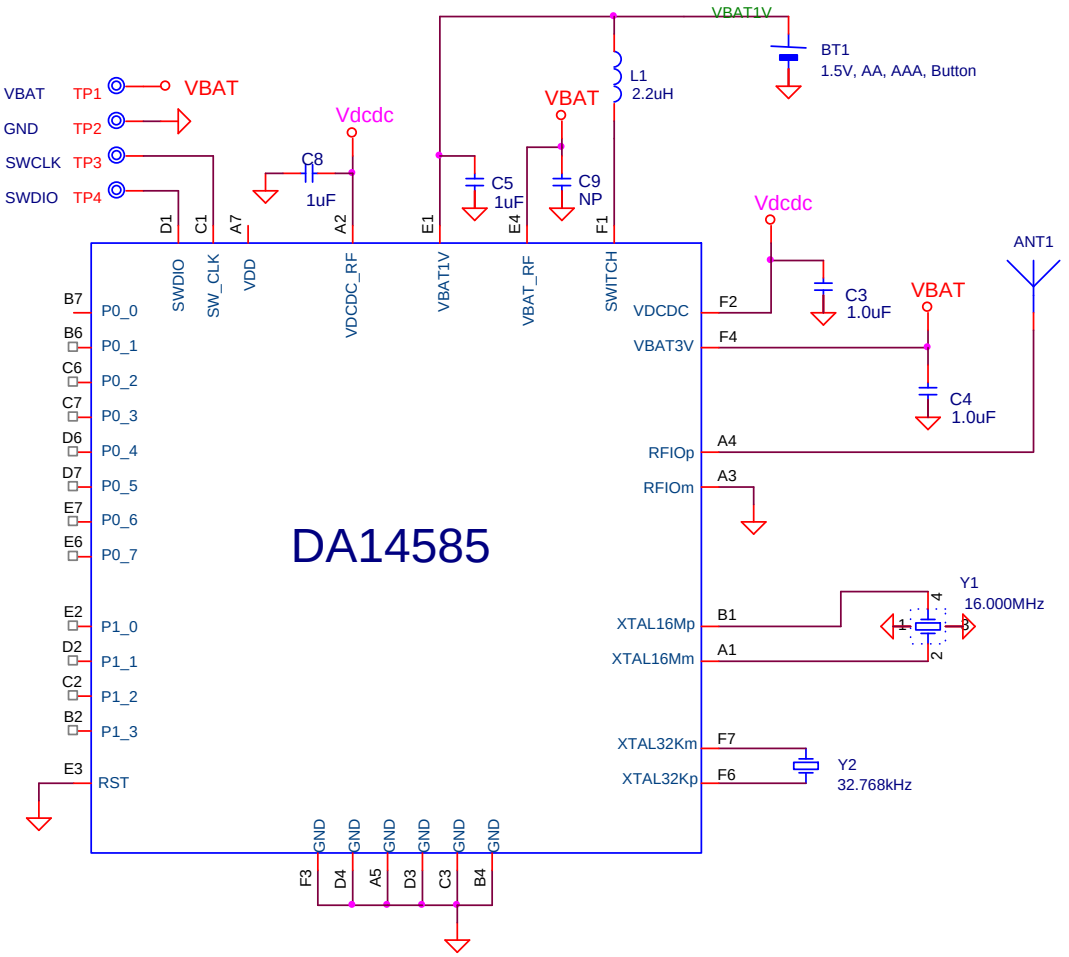


Figure 6: Alkaline Battery Cell Powered System Diagram (Boost mode)

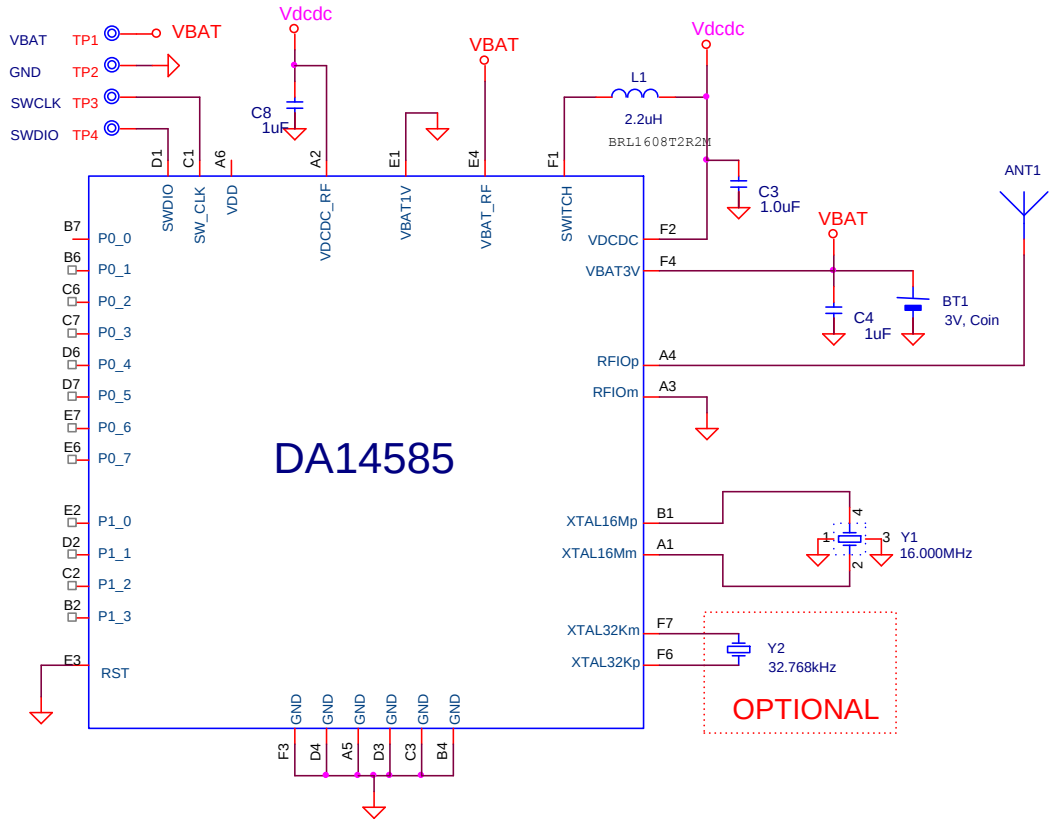


Figure 7: Lithium Coin Cell Powered System Diagram (Buck mode)

3.1 Absolute Maximum Ratings

Table 2: Absolute Maximum Ratings

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{PIN(LIM)(default)}$	limiting voltage on a pin	Voltage between pin and GND Note 1	-0.1		min{3.6, VBAT_RF+0.2}	V
T_{STG}	storage temperature		-50		150	°C
$t_{R(SUP)}$	supply rise time	Power supply rise time			100	ms
$V_{BAT(LIM)(VBAT1V)}$	limiting battery supply voltage	Supply voltage on VBAT1V in a boost converter application (VBAT3V is second output of boost-converter in this case) Note 1	-0.1		3.6	V
$V_{BAT(LIM)(VBAT3V)}$	limiting battery supply voltage	Supply voltage on VBAT3V and VBAT_RF in a buck-converter application, pin VBAT1V is connected to ground Note 1	-0.1		3.6	V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{PIN(LIM)(1V2)}$	limiting voltage on a pin	XTAL32Km, XTAL16Mp, XTAL16Mm Note 1	-0.2		min(1.2,VBAT_RF+0.2)	V
$V_{PIN(LIM)(VDCDC\_RF)}$	limiting voltage on the VDCDC_RF pin	Supply voltage on VDCDC_RF Note 1	-0.2		min(3.3,VBAT_RF+0.2)	V
$V_{PIN(LIM)(XTAL32Kp)}$	limiting voltage on a pin	XTAL32Kp	-0.2		min(1.5,VBAT_RF+0.2)	V
$V_{ESD(HBM)(WLCSP34)}$	electrostatic discharge voltage (Human Body Model)				2000	V
$V_{ESD(HBM)(QFN40)}$	electrostatic discharge voltage (Human Body Model)				4000	V
$V_{ESD(HBM)(QFN48)}$	electrostatic discharge voltage (Human Body Model)				4000	V
$V_{ESD(CDM)(WLCSP34)}$	electrostatic discharge voltage (Charged Device Model)				500	V
$V_{ESD(CDM)(QFN40)}$	electrostatic discharge voltage (Charged Device Model)				500	V
$V_{ESD(CDM)(QFN48)}$	electrostatic discharge voltage (Charged Device Model)				500	V

Note 1 The device should not be exposed for prolonged periods of time to voltages between the Recommended Operating Conditions and the Absolute Maximum Ratings range.

3.2 Recommended Operating Conditions

Table 3: Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{BAT(VBAT1V)}$	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
$V_{BAT(VBAT3V)}$	battery supply voltage	Supply voltage on VBAT3V and VBAT_RF in a buck-converter application, pin VBAT1V is connected to ground	1.8 Note 1		3.3	V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{BAT(OTP\_PROG)}$	battery supply voltage for otp programming	Supply voltage on VBAT3V for OTP programming	2.25			V
$V_{PIN(default)}$	voltage on a pin	Voltage between pin and GND	0		min(3.3,VBAT_RF+0.2)	V
$V_{PIN(1V2)}$	voltage on a pin	XTAL32Km, XTAL16Mp, XTAL16Mm	0		1.2	V
$V_{PIN(VDCDC\_RF)}$	voltage on a pin	Supply voltage on VDCDC_RF	0		3.3	V
T_A	ambient temperature		-40		85	°C

Note 1 Cold boot should not be performed if voltage is less than 1.8 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 >= 1.8 V.

3.3 DC Characteristics

Table 4: DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$I_{BAT(DP\_SLP)\_BOOST\_0kB}$	battery supply current	Boost configuration in deep-sleep with no RAM retained, running from XTAL32k oscillator Note 1		610		nA
$I_{BAT(DP\_SLP)\_BOOST\_16kB}$	battery supply current	Boost configuration in deep-sleep with 16 kB RAM retained, running from XTAL32k oscillator Note 1		1.4		µA
$I_{BAT(EXT\_SLP)\_BOOST\_32kB}$	battery supply current	Typical boost-application in extended-sleep with 32 kB RAM retained, running from XTAL32K oscillator Note 1		2.1		µA
$I_{BAT(EXT\_SLP)\_BOOST\_64KB}$	battery supply current	Typical boost-application in extended-sleep with 64 kB RAM retained, running from XTAL32K oscillator Note 1		3.3		µA
$I_{BAT(EXT\_SLP)\_BOOST\_96kB}$	battery supply current	Typical boost-application in extended-sleep with 96 kB RAM retained, running from XTAL32K oscillator Note 1		4.6		µA

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$I_{BAT(DP\_SLP)\_BUCK\_0kB}$	battery supply current	Buck configuration in deep-sleep with no RAM retained, running from XTAL32k oscillator Note 1		520		nA
$I_{BAT(DP\_SLP)\_BUCK\_16kB}$	battery supply current	Buck configuration in deep-sleep with 16 kB RAM retained, running from XTAL32k oscillator Note 1		1.2		µA
$I_{BAT(EXT\_SLP)\_BUCK\_32kB}$	battery supply current	Typical buck-application in extended-sleep mode with 32 kB RAM retained, running from XTAL32K oscillator Note 1		1.8		µA
$I_{BAT(EXT\_SLP)\_BUCK\_64KB}$	battery supply current	Typical buck-application in extended-sleep mode with 64 kB RAM retained, running from XTAL32K oscillator Note 1		2.9		µA
$I_{BAT(EXT\_SLP)\_BUCK\_96kB}$	battery supply current	Typical buck-application in extended-sleep mode with 96 kB RAM retained, running from XTAL32K oscillator Note 1		4		µA
$I_{BAT(ACT\_RX)\_BOOST}$	battery supply current	Typical application with Boost Converter(1.5V AA Battery) and Receiver Active		13.4		mA
$I_{BAT(ACT\_TX)\_BOOST}$	battery supply current	Typical application with Boost Converter(1.5V AA Battery) and Transmitter Active		12.4		mA
$I_{BAT(ACT\_RX)\_BUCK}$	battery supply current	Typical application with Buck Converter(3V Coin Cell) and Receiver Active		5.3		mA
$I_{BAT(ACT\_TX)\_BUCK}$	battery supply current	Typical application with Buck Converter(3V Coin Cell) and Transmitter Active		4.9		mA

Note 1 BANDGAP_REG[LDO_RET_TRIM] = 0x9

3.4 Timing Characteristics

Table 5: Timing Characteristics

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

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

3.5 16 MHz Crystal Oscillator Characteristics

Table 6: 16 MHz Crystal Oscillator - Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{XTAL}(16M)$	crystal oscillator frequency			16		MHz
ESR(16M)	equivalent series resistance				100	Ω
$C_L(16M)$	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 Note 1	-20		20	ppm
$\Delta f_{XTAL}(16M)UNT$	crystal frequency tolerance	Untrimmed; including aging and temperature drift Note 2	-40		40	ppm
$P_{DRV(MAX)}(16M)$	maximum drive power	Note 3	100			µW
$V_{CLK(EXT)}(16M)$	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
$\varphi_N(EXTERNAL)16M$	phase noise	f_C = 50 kHz in case of an external reference clock			-130	dBc/Hz

Note 1 Using the internal varicaps a wide range of crystals can be trimmed to the required tolerance.

Note 2 Maximum allowed frequency tolerance for compensation by the internal varicap trimming mechanism.

Note 3 Select a crystal which can handle a drive-level equal or more than this specification

Table 7: 16 MHz Crystal Oscillator - Timing Characteristics

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

3.6 32 kHz Crystal Oscillator Characteristics

Table 8: 32 kHz Crystal Oscillator - Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{CLK(EXT)}$(32K)	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_{XTAL}(32k)	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_L(32k)	load capacitance	no external capacitors are required for a 6 pF or 7 pF crystal	6	7	9	pF
C_0(32k)	shunt capacitance			1	2	pF
Δf_{XTAL}(32k)	crystal frequency tolerance (including aging)	Timing accuracy is dominated by crystal accuracy. A much smaller value is preferred	-250		250	ppm
$P_{DRV(MAX)}$(32k)	maximum drive power	Note 1	0.1			µW

Note 1 Select a crystal that can handle a drive-level of at least this specification.

Table 9: 32 kHz Crystal Oscillator - Timing Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{STA(XTAL)}$(32k)	crystal oscillator startup time	Typical application, time until 1000 clocks are detected Note 1		0.4		s

Note 1 This parameter is very much dependent on crystal parameters

3.7 Stable Low Frequency RCX Oscillator Characteristics

Table 10: Stable Low Frequency RCX Oscillator - Timing Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{RC}(RCX)	RCX oscillator frequency	default setting, buck mode only	5	10	40	kHz
Δf_{RC}(RCX)	RCX oscillator frequency drift	buck mode only Note 1	-500		500	ppm
$\Delta T_A/\Delta t$(RCX)100ms	ambient temperature gradient	buck mode only; connection interval 100 ms			0.66	°C/s
$\Delta T_A/\Delta t$(RCX)4s	ambient temperature gradient	buck mode only; connection interval 4 s			0.33	°C/s

Note 1 Maximum recommended connection interval (including slave latency) for the RCX usage is 2 s.

3.8 Digital Input/Output Characteristics

Table 11: Digital Input/Output - DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	HIGH level input voltage		0.84			V
V_{IL}	LOW level input voltage				0.36	V
V_{IH}(RST)	HIGH level input voltage	RST pin	0.84			V
V_{IL}(RST)	LOW level input voltage	RST pin			0.36	V
V_{OH}(VBAT1V)	HIGH level output voltage	Iout = -250 µA, VBAT3V = 2.35 V, VBAT1V = 0.9 V	0.72			V
V_{OH}(VBAT3V)	HIGH level output voltage	Iout = -3.5 mA, VBAT3V = 1.8 V, VBAT1V = 0 V Note 1	1.44			V
V_{OL}(VBAT1V)	LOW level output voltage	Iout = 250 µA, VBAT3V = 2.35 V, VBAT1V = 0.9 V			0.18	V
V_{OL}(VBAT3V)	LOW level output voltage	Iout = 3.5 mA, VBAT3V = 1.8 V, VBAT1V = 0 V Note 2			0.36	V
I_{IH}	HIGH level input current	Vin = VBAT3V = 2.5 V	-1		1	µA
I_{IL}	LOW level input current	Vin = VSS = 0 V	-1		1	µA
I_{IH}(PD)	HIGH level input current	Vin = VBAT3V = 2.5 V	50		150	µA
I_{IL}(PU)	LOW level input current	Vin = VSS = 0 V	-150		-50	µA
I_{IH}(RST)	HIGH level input current	RST pin, V(RST) = 1.2 V	25		75	µA

Note 1 In Boost mode the output source current is limited to Iout = -250 uA.

Note 2 In Boost mode the output sink current is limited to Iout = 250 uA.

3.9 General Purpose ADC Characteristics

Table 12: General Purpose ADC - Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
N_{BIT}(ADC)	number of bits (resolution)			10		bit

Table 13: General Purpose ADC - DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{I(ZS)}$	zero-scale input voltage	single-ended, calibrated at zero input	-2.5	0	2.5	mV
$V_{I(FS)}$	full-scale input voltage	single-ended, calibrated at zero input	1150	1180	1250	mV
$V_{I(FSN)}$	negative full-scale input voltage	differential, calibrated at zero input		-1180		mV
$V_{I(FSP)}$	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 14: General Purpose ADC - Timing Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{CONV}(ADC)	conversion time	Excluding initial settling time of the LDO and the 3x-attenuation (if used): LDO settling time is 20 µs (max), 3x-attenuation settling time = 1 µs (max) Using internal ADC-clock (~200 MHz)		0.25	0.4	µs

3.10 DC-DC Converter Characteristics

Table 15: DC-DC Converter - Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
L	effective inductance		1.5	2.2	3	µH

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C_{OUT}(VDCDC)	effective load capacitance	VDCDC and VDDCRF combined Note 1	0.5	1	10	µF
C_{OUT}(VBAT3V)	effective load capacitance	VBATRF and VBAT3V combined are the second output of the boost-converter Note 1	0.5	1	10	µF

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

Table 16: DC-DC Converter - DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_O(BUCK)	output voltage	default settings		1.42		V
V_O(BOOST)	output voltage	default settings, VDCDC		1.41		V
η_{CONV_MAX} (BUCK)	maximum conversion efficiency			86		%
η_{CONV_MAX} (BOOST)	maximum conversion efficiency			80		%
$\Delta V_O/\Delta V_I$(BUCK)	line regulation	2.35 V ≤ VBAT3V ≤ 3.3 V	-2	0.7	2	%/V
$\Delta V_O/\Delta V_I$(BOOST)	line regulation	0.9 V ≤ VBAT1V ≤ 1.2 V Note 1	-2	1	4	%/V
$\Delta V_O/\Delta I_L$(BUCK)	load regulation	VBAT3V = 2.5 V	-0.2	-0.02	0.2	%/mA
$\Delta V_O/\Delta I_L$(BOOST)	load regulation	VBAT1V = 1.2 V	-0.2	-0.07	0.2	%/mA
V_{RPL}(BUCK)	ripple voltage	buck mode; RMS ripple voltage		2.42		mV
V_{RPL}(BOOST)	ripple voltage	VBAT1V ≤ 1.2 V, boost mode; RMS ripple voltage Note 1		8		mV

Note 1 When VBAT1V > VDCDC_nominal, VDCDC will follow VBAT1V.

3.11 Radio Characteristics

Table 17: Radio - DC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{BAT}(RF)RX	battery supply current	receive mode; radio receiver and synthesizer active; DCDC converter assumed ideal; T_A = 25 °C Note 1		3.7		mA
I_{BAT}(RF)TX	battery supply current	transmit mode; radio transmitter and synthesizer active; DCDC converter assumed ideal; T_A = 25 °C Note 1		3.4		mA

Note 1 The DCDC-converter efficiency is assumed to be 100 % to enable benchmarking of the radio currents at battery supply domain (VBAT3V = 3 V).

Table 18: Radio - AC Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
P_{SENS_EPKT}	sensitivity level	Extended packet size (255 octets)		-91		dBm
P_{SENS}(CLEAN)	sensitivity level	DC-DC converter enabled; Dirty Transmitter disabled; T_A = 25 °C; PER = 30.8 % Note 1		-93		dBm
P_{SENS}	sensitivity level	Normal Operating Conditions; DC-DC converter enabled; T_A = 25 °C; PER = 30.8 % Note 1		-92.5		dBm
P_{SENS}(EOC)	sensitivity level	Extreme Operating Conditions; DC-DC converter enabled; PER = 30.8 %; -40 °C ≤ T_A ≤ +85 °C Note 2			-87	dBm
P_I(max)	input power level	DC-DC converter disabled; T_A = 25 °C; PER ≤ 30.8 % Note 1	10			dBm
CIR(0)	carrier to interferer ratio	n = 0; interferer @ $f_1 = f_0$ + n*1 MHz; T_A = 25 °C Note 3		7	21	dB
CIR(1)	carrier to interferer ratio	n = ±1; interferer @ $f_1 = f_0$ + n*1 MHz; T_A = 25 °C Note 3		-3	15	dB
CIR(P2)	carrier to interferer ratio	n = +2 (image frequency); interferer @ f1 = f0 + n*1 MHz; T_A = 25 °C Note 3		-20	-9	dB

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
CIR(M2)	carrier to interferer ratio	n = -2; interferer @ f1 = f0 + n*1 MHz; T_A = 25 °C Note 3		-30	-17	dB
CIR(P3)	carrier to interferer ratio	n = +3 (image frequency + 1 MHz); interferer @ f1 = f0 + n*1 MHz; T_A = 25 °C Note 3		-30	-15	dB
CIR(M3)	carrier to interferer ratio	n = -3; interferer @ f1 = f0 + n*1 MHz; T_A = 25 °C Note 3		-35	-27	dB
CIR(4)	carrier to interferer ratio	|n| >= 4 (any other BLE channel); interferer @ $f_1 = f_0$ + n*1 MHz; T_A = 25 °C Note 3		-37	-27	dB
P_{BL}(I)	blocker power level	30 MHz ≤ f_{BL} ≤ 2000 MHz; P_{WANTED} = -67 dBm; T_A = 25 °C Note 4	-5			dBm
P_{BL}(II)	blocker power level	2003 MHz ≤ f_{BL} ≤ 2399 MHz; P_{WANTED} = -67 dBm; T_A = 25 °C Note 5	-15			dBm
P_{BL}(III)	blocker power level	2484 MHz ≤ f_{BL} ≤ 2997 MHz; P_{WANTED} = -67 dBm; T_A = 25 °C Note 5	-15			dBm
P_{BL}(IV)	blocker power level	3000 MHz ≤ f_{BL} ≤ 12.75 GHz; P_{WANTED} = -67 dBm; T_A = 25 °C Note 5	-5			dBm
P_{RSSI}(min)	RSSI power level	absolute power level for RXRSSI[7:0] = 0; T_A = 25 °C Note 6	-115	-112	-109	dBm
$P_{INT(IMD)}$	intermodulation distortion interferer power level	worst case interferer level @ f_1, f_2 with $2f_1-f_2 = f_0$, $\|f_1-f_2\|$ = n MHz and n = 3,4,5; P_{WANTED} = -64 dBm @ f_0; PER = 30.8 %; T_A = 25 °C Note 7	-35	-31		dBm
P_{RSSI}(max)	RSSI power level	upper limit of monotonous range; T_A = 25 °C	-26	-19		dBm
L_{ACC}(RSSI)BOOST	level accuracy	tolerance of 5 % to 95 % confidence interval of P_{RF}: when RXRSSI[7:0] = X, 50 < X < 175; burst mode 1500 packets; T_A = 25 °C; DC-DC converter in BOOST mode		0	3	dB

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
L_{ACC}(RSSI)BUCK	level accuracy	tolerance of 5 % to 95 % confidence interval of P_{RF}: when RXRSSI[7:0] = X, 50 < X < 175; burst mode 1500 packets; T_A = 25 °C; DC-DC converter in BUCK mode		0	2	dB
L_{RES}(RSSI)	level resolution	gradient of monotonous range for RXRSSI[7:0] = X, 50 < X < 175; burst mode 1500 packets; T_A = 25 °C	0.46	0.474	0.485	dB/LSB
ACP(2M)	adjacent channel power level	f_{OFFSET} = 2 MHz; T_A = 25 °C Note 8		-53	-50	dBm
ACP(2M) (EOC)	adjacent channel power level	f_{OFFSET} = 2 MHz; -40 °C ≤ T_A ≤ +85 °C Note 8		-53	-47	dBm
ACP(3M)	adjacent channel power level	f_{OFFSET} ≥ 3 MHz; T_A = 25 °C Note 8		-57	-55	dBm
ACP(3M) (EOC)	adjacent channel power level	f_{OFFSET} ≥ 3 MHz; -40 °C ≤ T_A ≤ +85 °C Note 8		-57	-47	dBm
P_O	output power level	V_{DD} = 3 V; maximum gain; T_A = 25 °C	-2	-1	0	dBm
P_O(HD2)	output power level (second harmonic)	VDD = 3 V; maximum gain; T_A = 25 °C		-54	-40	dBm
P_O(HD3)	output power level (third harmonic)	VDD = 3 V; maximum gain; T_A = 25 °C		-56	-40	dBm
P_O(HD4)	output power level (fourth harmonic)	VDD = 3 V; maximum gain; T_A = 25 °C		-70	-40	dBm
P_O(HD5)	output power level (fifth harmonic)	VDD = 3 V; maximum gain; T_A = 25 °C		-70	-40	dBm
P_O(NFM)	output power level in 'Near Field Mode'	V_{DD} = 3 V; maximum gain; T_A = 25 °C Note 9	-25	-20	-15	dBm

Note 1 Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.4.1.

Note 2 Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.4.2.

Note 3 Measured according to Bluetooth® Core Technical Specification document, version 4.0, volume 6, section 4.2.

Note 4 Measured according to Bluetooth® Core Technical Specification document, version 4.0, volume 6, section 4.3. Due to limitations of the measurement equipment, levels of -5 dBm should be interpreted as > -5 dBm.

Note 5 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 dBm.

Note 6 PRF = PRSSI(min) + LRES(RSSI) x RXRSSI[7:0] ± LACC(RSSI). Thanks to constant gain biasing of RF part in the receiver, the RSSI can be used to estimate absolute power levels, not only level changes. Even across the full temperature range the variation is limited.

Note 7 Measured according to Bluetooth® Core Technical Specification document, version 4.0, volume 6, section 4.4. Published value is for n = IXIT = 4 . IXIT = 5 gives the same results, IXIT = 3 gives results that are 5 dB lower.

Note 8 Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.2.3.

Note 9 To activate the "Near Field Mode", program address 0x50002418 with the value 0x0030.

4 System Overview

4.1 Internal Blocks

The DA14585 contains the following blocks:

ARM Cortex M0 CPU with Wake-up Interrupt Controller (WIC). This processor provides 0.9 dMIPS/MHz and is used for assisting the Bluetooth LE protocol implementation, providing processing power for calculations or data fetches required by the application and finally housekeeping, including controlling of the power scheme of the system.

BLE core. This is the baseband hardware accelerator for the Bluetooth LE protocol.

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

OTP. This is a 64 kB One-Time Programmable memory array, used to store the application code as well as Bluetooth LE profiles. It also contains the system configuration and calibration data.

System RAM. These are 4 special low leakage SRAM cells, 96 kB in total, which is used for mirroring the program code from the OTP when the system wakes/powers up or mirroring from an external flash when the system powers up as well as to store various data of the Bluetooth LE protocol like the system's global variables and processor stack when the system goes into Extended 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 Extended Sleep mode.

Audio Interface. This interface comprises three separate blocks: a PDM block, a PCM/I2S block and a Sample Rate Converter (SRC block) with DMA support.

UART and UART2. These asynchronous serial interfaces implement hardware flow control with FIFO depths of 16 bytes each and DMA support.

SPI. This is the serial peripheral interface with master/slave capability, a FIFO of 2 16-bit words and DMA support.

I2C. This is Master/Slave I2C interface used for sensors and/or host MCUs communication. It comprises a 32 places 9-bits wide FIFO and DMA support.

General Purpose (GP) ADC. This is a 10-bits analog-to-digital converter with 4 external input channels.

Radio Transceiver. This block implements the RF part of the Bluetooth LE protocol.

Clock Generator. This block is responsible for the clocking of the system. It contains 2 XTAL oscillators: one running at 16 MHz (XTAL16M) which is used for the active mode of the system and one running at 32.768 kHz (XTAL32K) which is used for the sleep modes of the system. There are also three RC oscillators available: a 16 MHz and a 32 kHz oscillator (RC16M and RC32K) with low precision (> 500 ppm) and a 10 kHz oscillator (RCX) with high precision (< 500 ppm). The RCX oscillator can be used as a sleep clock replacing the XTAL32K oscillator to further improve the power dissipation of the system while reducing the bill of materials of the system. The RC16M oscillator is used to provide a clock used for the mirroring of the OTP code into the SysRAM while the XTAL16M oscillator is settling directly after power/wake up.

Software Timer. This block contains a 16-bit general purpose timer (Timer0) with PWM capability as well as a 14-bits timer (Timer2) which controls 3 PWM signals with respect to frequency and duty cycle.

Wake-Up Timer. This is a timer for capturing external events and it can be used as a wake-up trigger based on a programmable number of external events on any of the GPIO ports.

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.

Keyboard Controller. This circuit enables the reading and debouncing of a programmable number of GPIOs and generates an interrupt upon a configurable action.

AHB/APB Bus. Implements the AMBA Lite version of the AHB and APB specifications.

Power Management. A sophisticated power management circuit with a Buck/Boost DC-DC converter and several LDOs that can be turned on/off via software.

A more detailed description of each of the components of the DA14585 is presented in the following sections.

4.2 Functional Modes

The DA14585 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:

- **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 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.
- **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.
- **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 as described in section 4.3.1.

4.3 OTP Memory Layout

The One Time Programmable memory has to be programmed according to a specific layout, which structures information to be easily accessible from the BootROM code as well as the actual application. An overview of the layout scheme is presented in the following figure:

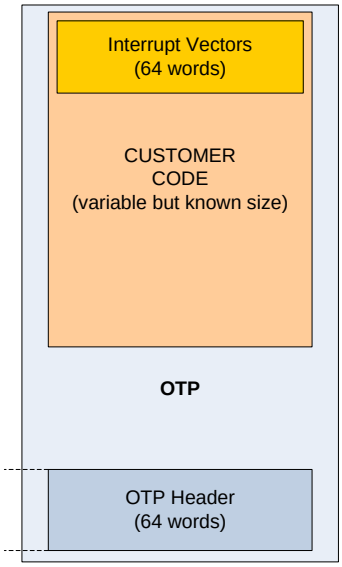


Figure 8: OTP Layout Scheme

The OTP memory comprises 8K of 64-bit words. The contents are described below:

- **Interrupt Vectors:** These are the vectors of the interrupt service routines always residing at address 0x0. This is part of the application (customer) code. The size of this vector list is 64 words.
- **Customer Code:** contains the application and the profiles that a customer has developed. The size is known and fixed before mass production and programming of the OTP.
- **OTP Header:** contains various information about the configuration of the system as well as Bluetooth LE specific data.

4.3.1 OTP Header

The OTP header breakdown is presented in the following table:

Table 19: OTP Header

Address	Size(words)	Description
0xFFF8	1	SWD enable flag: 0x00 = JTAG is enabled 0xAA = JTAG is disabled
0xFFF0	1	OTP DMA length (number of 32-bit words)
0xFFE8	1	Reserved. Keep to 0x0
0xFFA8	8	Bluetooth Device Address (8 64-bit words). It is handled as a string of bytes. Leftmost number will be programmed at 0x7FD4 etc.
0xFFA0	1	Signature Algorithm Selector: 0x00 = None 0xAA = MD5 0x55 = SHA-1 0xFF = CRC32
0xFF28	15	Signature of Customer Code (15 64-bit words)
0xFF20	1	Trim value for the VCO
0xFF18	1	Trim value for the XTAL16M oscillator
0xFF10	1	Trim value for the RC16M oscillator
0xFF08	1	Trim value for the BandGap
0xFF00	1	Trim value for the RFIO capacitance
0xFEF8	1	Trim value for the LNA

Address	Size(words)	Description
0xFEF0	1	Calibration Flags: Bit[63:32] - Reserved Bit[31:16] : 0xA5A5 = at least 1 calibration was done 0x0000 = no calibration was done Bit[15:6] - Reserved Bit[5] - 1 = VCO trim value is valid Bit[4] - Reserved Bit[3] - 1 = RC16M trim value is valid Bit[2] - 1 = BandGap trim value is valid Bit[1] - 1 = RFIO trim value is valid Bit[0] - 1 = LNA trim value is valid
0xFEE8	1	Sleep Clock Source Flag: 0x00 = External crystal (XTAL32K) 0xAA = Internal RC32k oscillator (RC32k)
0xFEE0	1	Device and Package Flag B[7-2]: Reserved B[1]: ● 0x00 = 585 ● 0xAA = 586 B[0]: ● 0x00 = WLCSP34 ● 0xAA = QFN40 ● 0x55 = QFN48 ● 0x99 = KGD
0xFEC8	3	Reserved
0xFEC0	1	Boot specific mapping B0[7:4] : SPI_CLK, Port number B0[3:0] : SPI_CLK, Pin number B1[7:4] : SPI_EN, Port number B1[3:0] : SPI_EN, Pin number B2[7:4] : SPI_DO, Port number B2[3:0] : SPI_DO, Pin number B3[7:4] : SPI_DI, Port number B3[3:0] : SPI_DI, Pin number B4 : ● 0xAA = Boot from SPI port at a specific location ● 0x00 = Normal sequence B5 : Wake up Command opcode B6 : Serial Speed Selection SPI DIV (0-3) B7 : Reserved
0xFEB8	1	UART STX Timeout (units of 4mS, default = 60 msec)
0xFEB0	1	OTP control value Bits[31:0] = 0xC0DEBABE Bits[63:32] = Reserved

Address	Size(words)	Description
0xFE20	18	Customer Specific Field (18 x 64-bit words)
0xFE18	1	CRC for Trim and Calibration values
0xFE10	1	IQ Trim value
0xFE08	1	Application Programmed Flag #2 0xA5A51234A5A51234 = Application is in OTP
0xFE00	1	Application Programmed Flag #1 0x1234A5A51234A5A5 = Application is in OTP

The first word (at address 0xFFF8) is a flag which defines whether SWD (JTAG) is mapped on pins or not. Default value is activating the SDW on the respective pins. The Length field (0xFFF0) specifies the number of 64-bit words to be copied to the SRAM.

The Bluetooth device address is stored at 0xFFA8 (8 64-bit words).

The next field (0xFFA0) identifies the algorithm to be used for creating a signature on the OTP payload. The actual signature value is stored in the next fields (if a signature algorithm is selected). This is an optional feature to guarantee trusted OTP images.

The next memory locations contain the trim values for the XTAL16M and the RC16M oscillators. Since every crystal is different, an extra calibration step is required in the production line process, to identify the correct trimming value for the XTALs so that they provide the precision required by the Bluetooth LE protocol. Since the crystal is an external component, this step has to be performed during the calibration phase.

A similar procedure is required for the trimming of the RC capacitance to keep the RC clock within a certain range of frequencies. However, this trimming is done during production tests by Dialog Semiconductor.

Trimming values for the VCO, Bandgap reference, the RFIO capacitance and the LNA of the Radio are also stored in the OTP header. These values are generated and programmed during production testing by Dialog Semiconductor.

The Calibration Flags define whether the chip has been already calibrated and if so, which trim values are valid.

The Sleep Clock Source Flag indicates to the application software whether an external 32 kHz crystal is used or not. This flag does not take care of enabling the XTAL32K clock if it is selected and as such the application should take care of successfully enabling the XTAL32K clock.

The Device and Package Flag reflects which the current device (DA14585) is and the package used: the 34 balls WLCSP, the 40 pins QFN or the 48 pins QFN.

Boot specific mapping value is used in order to define a specific configuration for the SPI interface when used for external booting (e.g. flash memory). Byte4 is the flag to instruct the BootROM to use the specific SPI pin mapping and skip the rest of the serial peripheral interfaces. The BootRom is taking care of waking up an external flash when the flash memory is in deep power-down state. Byte5 is used for the Wake up Command opcode that the flash memory responds to. If Byte5 is left unprogrammed, the BootROM will send the "0xAB" opcode by default. Furthermore, the BootROM is able to wake up the external flash by toggling the CS pin.

The UART STX Timeout field reflects the time that the BootROM waits until an external serial device initiates the communication procedure via UART. If this time expires the BootROM will try different baud rate settings or continue to the next available serial interface.

BootROM is responsible for making sure that the OTP memory operates correctly. The OTP control value is checked multiple times during the BootROM flow in order to ensure proper OTP functionality. This value is programmed during production testing by Dialog Semiconductor.

The IQ Trim value contains the value of respective radio configuration fields and is delivered at production testing.

Two more flags indicate if the application code has indeed been programmed (burned) into the OTP. Both flags are read by the BootROM software designating that the system is now in Normal mode and not in Development mode, as explained in sections 4.2 and 4.4.3.

4.4 System Start Procedure

The actual start procedure consists of the following distinct stages that are sequentially combined to form the preferred system start sequence:

1. The Power/Wake-up sequence
2. The OTP mirroring sequence
3. The BootROM booting sequence

The Power/Wake-up sequence is a hardwired state machine that enables the LDOs and prepares the system's internal voltages.

The OTP mirroring is a hardwired state machine which instructs the copying of the OTP contents into the SysRAM even before the ARM CPU starts.

The BootROM code will only be executed when the system is powered up or a hardware reset occurs.

The relation between the aforementioned sequences is presented in Figure 9.

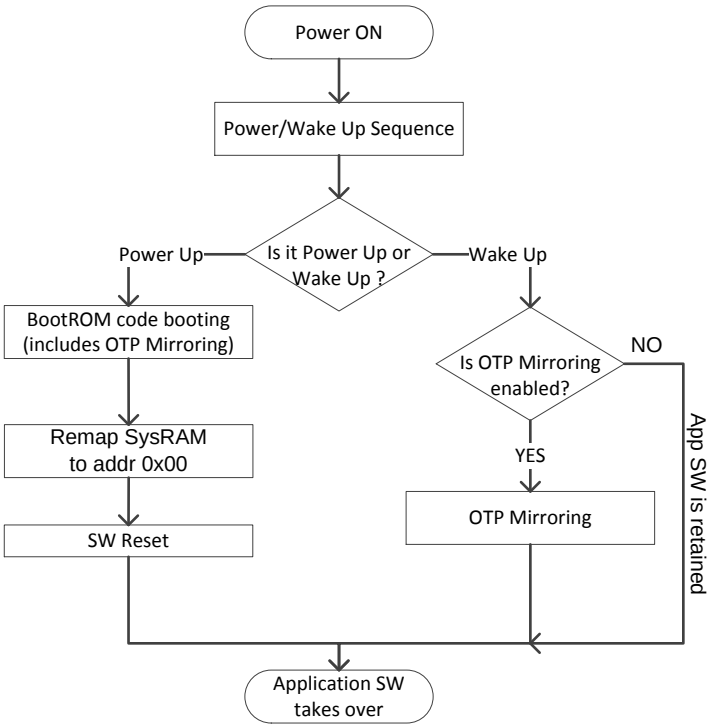


Figure 9: General Start-up Flow

The system will enter the Power/Wake-Up sequence after cold power up or after waking up from one of the Sleep modes, which are described in section 4.5.2. The distinction is done by means of a Flip-Flop which is set to '1' to indicate that the system has been just powered up (cold start). The whole process from power on up to the point that the system starts advertising (considering a typical application) takes up to 50 ms (depending on the OTP content size).

4.4.1 Power/Wake-Up Sequence

The Power/Wake-up sequence is a hardwired state machine that is activated every time the system is powered up (cold start) or woken up from one of the Sleep modes. This state machine uses

information from internal analog comparators to reliably identify whether the system is set up in BUCK (VBAT3V > 1.8 V) or BOOST (VBAT1B > 0.5 V) mode automatically.

Initially, it checks whether the voltage at the VBAT1V pin is above the threshold indicating that the system is in BOOST mode and if so, it activates the DC-DC converter accordingly. If not, it continues enabling the Bandgap and the digital LDO to provide a stable 1.2 V for the core and 1.8 V for the OTP cell. Next, the XTAL16M oscillator is started while a second check overwrites the programming of the DC-DC converter in the correct mode.

Since LDOs as well as the XTAL16M oscillator take some time to settle, the state machine is polling on signals indicating that the startup of these blocks has occurred successfully. However, if these signals are not set in a timely manner, the state machine ignores the status of these blocks and continues with the power-up sequence. The timeout can be disabled by software via SYS_CTRL_REG[TIMEOUT_DISABLE].

When no timeout occurs, the time required from the Power/Wake-up until the state that the chip is powered correctly and either the BootROM (in the case of a cold start) or the OTP mirroring (in the case of a programmed wake up) takes over, is ranging from 1.2 ms up to 1.5 ms depending on the supply voltage.

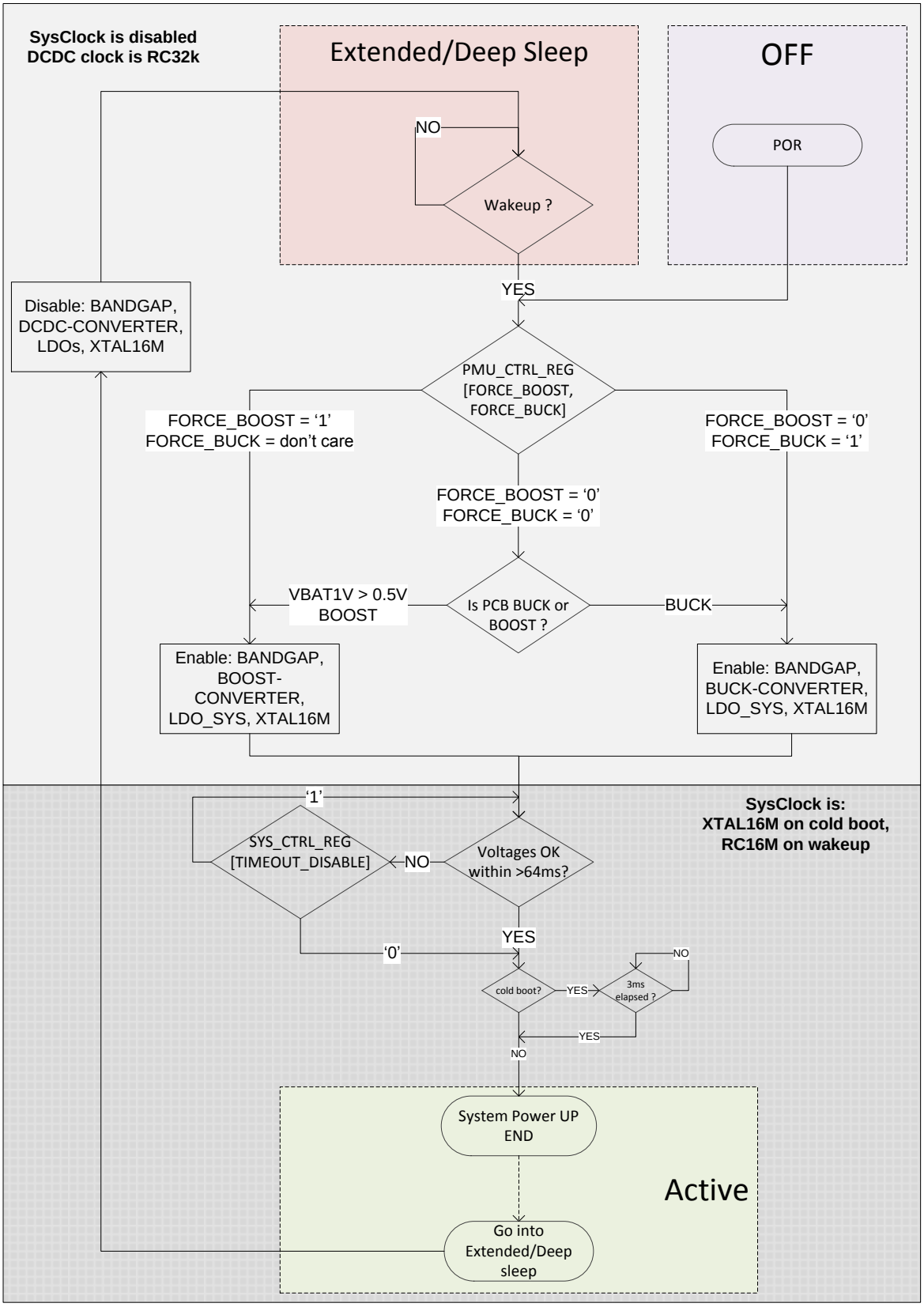


Figure 10: Power/Wake-Up Sequence

4.4.2 OTP Mirroring

This is one of the two branches of the decision regarding Power-up or Wake-up as illustrated in Figure 9. During the OTP mirroring process, the contents of the OTP memory are copied into the SysRAM, so that the ARM CPU can start executing code from there instead of the power hungry OTP memory. This task is programmable (via SYS_CTRL_REG[OTP_COPY]). In case of a Power Up, this task is performed by SW in the BootROM, i.e. the ARM CPU takes care of the mirroring of the application code into the SysRAM. However, in case of a Wake-up, the BootROM code is not executed and a small hardware state machine performs the mirroring as shown in Figure 11.

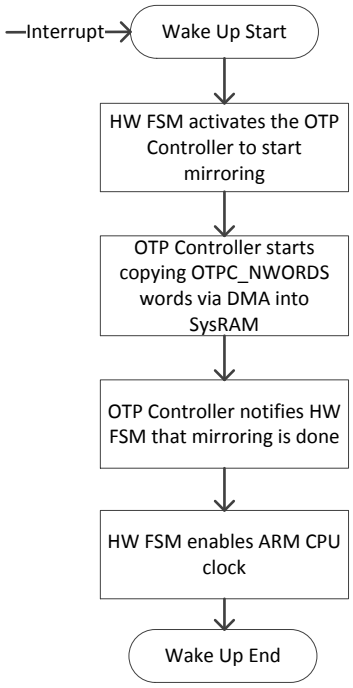


Figure 11: Wake-Up OTP Mirroring

The flow chart of Figure 11 assumes that the OTP controller is aware of the number of words that need to be copied from the OTP memory. This value resides in the OTP Header (see Table 19) and at power up will be copied by the BootROM code (i.e. the ARM CPU) into the OTPC_NWORDS register. This is done only once, since the OTPC_NWORDS register retains its value even when the system goes into any of the Sleep modes. In this way, the number of words to be transferred from the OTP memory into the SysRAM by the OTP controller DMA engine is always available.

4.4.3 BootROM Sequence

The BootROM code identifies whether the chip is in Development mode of Normal mode by reading the “Application Programmed” flags from the OTP header (see Table 19). The OTP contains all zeros when it is not programmed.

If the predefined value is identified this ensures that the OTP is functional and that the application code has been programmed. However, if the predefined value is not identified, either the OTP is not programmed (all zeros) or the OTP memory is not operational (random data).

In the first case, the system enters Development mode where the application can be developed and values can be calibrated. In the second case, the BootROM code recognizes the OTP to be malfunctioning due to power issues (e.g. battery life is ending and thus the LDO cannot generate the required voltages) and continues to activate the peripherals so that the system is still usable and can be debugged.

The booting process of the DA14585 is presented in Figure 12.

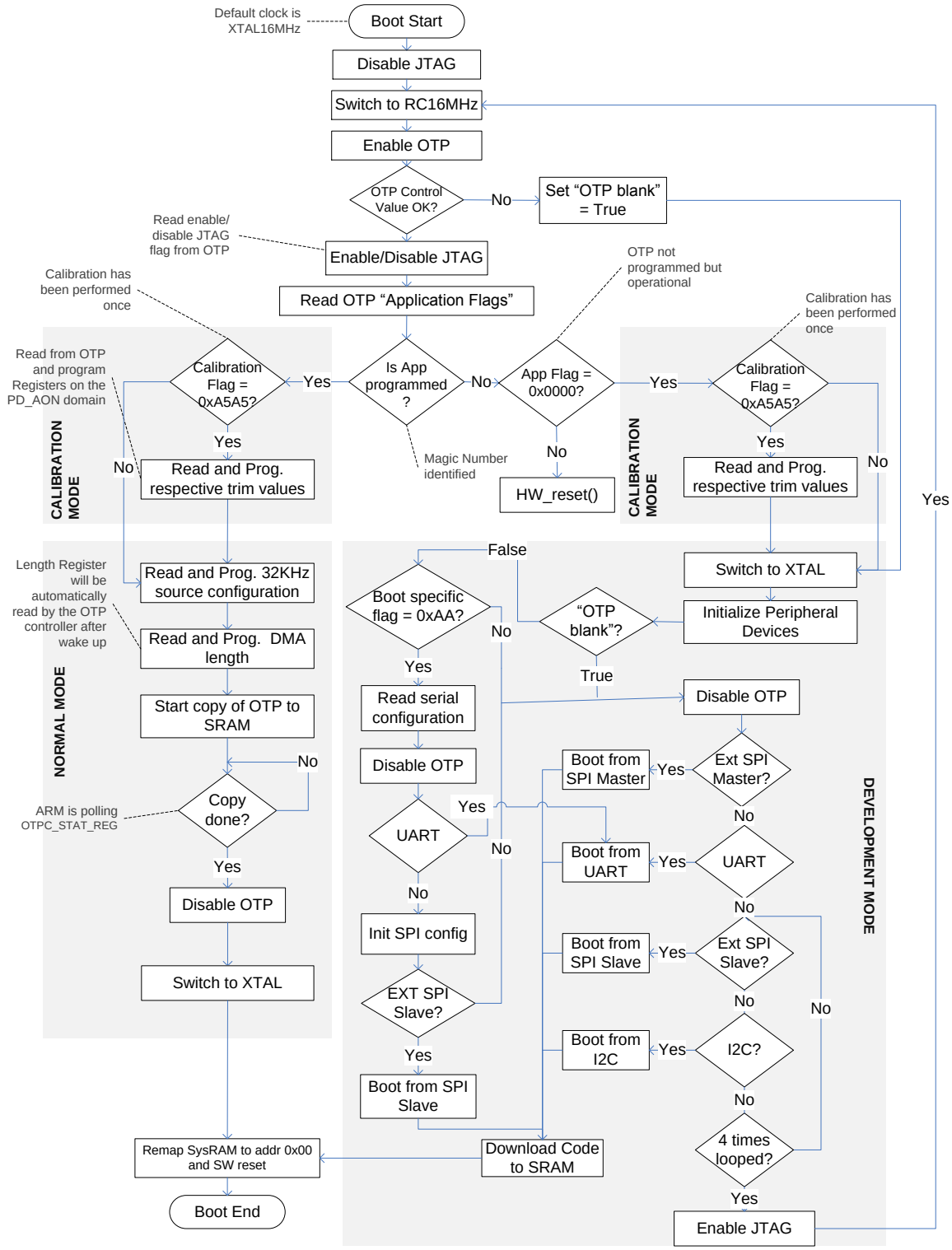


Figure 12: BootROM Sequence

When in Development mode, the BootROM code initializes all peripheral devices from which the DA14585 might download code: UART, SPI (both Master and Slave) and I2C. The code searches for a valid response on various combinations of I/Os one after the other. There is also the option that the user can define desired I/Os using a specific OTP field for the SPI interface. The step sequence and GPIO mapping is presented in the following tables:

Table 20: Development Mode Peripheral Pin Mapping

Interface	Signal	Step A	Step B	Step C	Step D
SPI Master (Note 1)	SCK	P0_0	P0_0		
	CS	P0_3	P0_1		
	DO (MISO)	P0_6	P0_2		
	DI (MOSI)	P0_5	P0_3		
UART	TX	P0_0	P0_2	P0_4	P0_6
	RX	P0_1	P0_3	P0_5	P0_7
	Baud Rate	57600 / 8-N-1	115200 / 8-N-1	57600 / 8-N-1	9600 / 8-N-1
SPI Slave (Note 2)	SCK	P0_0			
	CS	P0_3			
	DO (MOSI)	P0_6			
	DI (MISO)	P0_5			
I2C	SCL	P0_0	P0_2	P0_4	P0_6
	SDA	P0_1	P0_3	P0_5	P0_7

Note 1 DA14585 acts as slave device.

Note 2 DA14585 acts as master device.

Table 21: Development Mode Peripheral Search Sequence

Sequence	Action
0	SPI Master Step A
1	SPI Master Step B
2	UART Step A
3	UART Step B
4	UART Step C
5	UART Step D
6	SPI Slave Step A (Note 3)
7	I2C Step A (Note 4)
8	I2C Step B (Note 4)
9	I2C Step C (Note 4)
10	I2C Step D (Note 4)

Note 3 This step is performed in a 4 times' loop (see Figure 12). SCK in 1st iteration is 8MHz, in 2nd 4MHz, in 3rd 2MHz and in 4th 1MHz.

Note 4 SCL = 100Kbits/sec (Standard mode).

At each step, the BootROM code sends a certain character to the peripheral controller waiting for an answer. If no answer is received within a certain amount of time, a timeout instructs the code to continue to the next step. If a response is detected, a specific protocol is executed and communication between the DA14585 and the external device is established. The user has the ability to speed up the whole boot process by skipping all other boot paths except for the SPI (in case a specific SPI configuration has been selected). In that case, if the boot option fails, the BootROM will continue to use the standard development mode boot path.

When in Normal mode, the ARM CPU programs OTP header configuration information into actual registers, some of which retaining their value even if the system enters one of the Sleep modes (Calibration Flags Bits [31:16] should equal 0xA5A5). Next, the actual Application Code is mirrored

into the System RAM and the patching function is executed to ensure that SW updates are applied. Finally, the Remap register (SYS_CTRL_REG[REMAP_ADR0]) is programmed to ensure that address zero is now pointing to the System RAM.

A SW reset as a last step, instructs the ARM CPU to reboot running code from the SRAM.

4.5 Power Supply Configuration

4.5.1 Power Domains

The DA14585 comprises several different power domains that are controlled by power switching elements, thus eliminating leakage currents by totally powering them down. The partitioning of the DA14585's resources with respect to the various power domains is presented in Table 22.

Table 22: Power Domains

Power Domain	Description
PD_AON	Always ON: This power line connects to all the resources that must be powered constantly: the ARM/WIC, the LLP/Timer, the PMU/CRG, the Capture Timer, the Quadrature Decoder, the pad ring and various registers required for the Wake Up sequence.
PD_SYS	System: This power line connects to all the resources that should be powered only when the ARM M0 is running: the AHB bus, the OTP cell and controllers, the ROM, the SysRAM the Watchdog, the SW Timer and the GPIO port multiplexing.
PD_PER	Peripherals: This power line connects to the peripherals that can be switched off after completing their operation: the UARTs, the SPI, the I2C the Keyboard controller, the ADC and the Audio Unit.
PD_DBG	Debug: Powers the debug part of the ARM Cortex-M0 processor.
PD_RAD	Radio: This is the power domain that includes the digital part of the Radio (DPHY) and the BLE Core. The power management of the Radio (RF) subsystem is controlled via several dedicated LDOs.
PD_SR1	SysRAM1: This is a separate power line that only controls the first 32 kB SysRAM cell. If this memory cell is not needed, it should always be OFF.
PD_SR2	SysRAM2: This is a separate power line that only controls the first 16 kB SysRAM cell. If this memory cell is not needed, it should always be OFF
PD_SR3	SysRAM3: This is a separate power line that only controls the second 16 kB SysRAM cell. If this memory cell is not needed, it should always be OFF
PD_SR4	SysRAM4: This is a separate power line that only controls the second 32 kB SysRAM cell. If this memory cell is not needed, it should always be OFF

4.5.2 Power Modes

There are four different power modes in the DA14585:

- **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 and the programmed PD_SRx. OTP mirroring is required upon waking up the system when the application code and data are not retained. If the system code is mirrored from an external flash the corresponding SysRam cells are retained and no OTP mirroring is performed upon wake up.
- **Deep Sleep mode**: All power domains are off including all the SysRAM cells. A BLE connection cannot be maintained. The system will wake up only from an external interrupt (HW reset on

wakeup) or from a Power-On Reset source. In this mode the system consumes the minimum amount of power.

The first two power modes do not include any special power gating or manipulation of power domains. The entering into the Extended/Deep Sleep mode is summarized in Table 23.

Table 23: Activating Power Modes

Power mode	Activation steps

Extended Sleep	`PMU_CTRL_REG |= 0x6;`
	`SCB->SCR |= 1<<2;`
	enables the SLEEPDEEP bit on the System Control Register of the ARM CPU

Deep Sleep	Setup external interrupt or a POR source
	`PMU_CTRL_REG = (PMU_CTRL_REG & ~(0xF00)) | 0x6;`
	`SCB->SCR |= 1<<2;`
	enables the SLEEPDEEP bit on the System Control Register of the ARM CPU

However, the above Power modes which involve power gating do not turn off all power domains automatically. Certain power domains need to be powered off prior to activating the Extended Sleep or Deep Sleep mode. Table 24 summarizes the effect of the Extended/Deep Sleep mode activation with regards to specific power domains.

Table 24: Power Domain Manipulation When Activating Extended Sleep Mode

Power mode	PD_SYS	PD_PER	PD_DBG	PD_RAD	PD_SRx	Analog
Extended/Deep Sleep	Auto OFF	Programmable	Auto OFF	Programmable	Programmable.	Auto OFF

The “Analog” column in Table 24 refers to the Bandgap, the DC-DC converter, the XTAL16M oscillator, the RC oscillators, the ADC and the respective LDOs. These blocks are not part of a single power domain but the HW will turn them off upon activation of the Extended/Deep Sleep mode.

The lowest power consumption can be achieved when none of the SysRAM cells are retained during sleep (Deep Sleep). At that state, the device cannot sustain any BLE connection state data and thus a reset has to be performed after wakeup. This configuration can be used as the shipping mode of the product.

In order to retain the BLE state of the system (BLE connection data) but consume the bare minimum power, SysRAM4 cell has to be retained during sleep time. In this case, the application code is not retained and OTP mirroring needs to take place after wakeup. The benefit of retaining the RAM (needed for an application) during the Extended Sleep mode versus mirroring from OTP after wakeup can be determined by various parameters such as the Bluetooth LE application and its specific parameters (the required connection interval, the duty cycle and the amount of data transmitted or received per connection). Based on these parameters a time threshold for the connection interval can be calculated, above which mirroring from OTP performs better than retaining the needed RAM cells during sleep.

4.5.3 Retention Registers

A number of the registers in the DA14585 needs to retain their values when the system enters one of the Sleep modes described in the previous section. These registers and their power domains are described in Table 25.

Table 25: Retention Registers and Power Domains

Power Domain	Address Range		Retained Registers	Notes
	Start	End		
PD_SYS	0x07F4.0000	0x07F8.FFFF	OTPC_AHBADR_REG	Contains the target address of SysRAM in which the data from OTP will be copied to.
			OTPC_CELADR_REG	-
			OTPC_MODE_REG	-
			OTPC_NWORDS_REG	Contains the number of words to be copied from the OTP memory into the SysRAM.
			OTPC_TIM1_REG	-
			OTPC_TIM2_REG	-
			DEBUG_REG	Retained bit fields are: DEBUGS_FREEZE_EN
PD_RAD	0x4000.0000	0x4000.7FFF	BLE_CNTL2_REG	Used for BLE Core configuration.
PD_AON	0x5000.0000	0x5000.0FFF	All	
PD_PER	0x5000.1000	0x5000.1FFF	None	
	0x5000.4000	0x5000.4FFF	None	

4.5.4 Sleep LDO Voltage Trimming

To ensure proper operation of the device during sleep and across the application operating temperature range, the voltage level of the sleep LDO has to be configured according to Table 26.

Table 26: Sleep LDO Recommended Settings across Temperature

BANDGAP_REG[LDO_RET_TRIM]	Retained RAM	Indicative Application Operating Temperature Range
0x9	Any	< 40 °C
0x8	Any	-40 °C to 60 °C
0x7	up to 64 kB	-40 °C to 85 °C
0x6	up to 96 kB	-40 °C to 85 °C

5 Reset

The DA14585 comprises an RST pad which is active high. It contains an RC filter for spikes suppression with 400 kΩ and 2.8 pF for the resistor and the capacitor respectively. It also contains a 25 kΩ pull-down resistor. This pad should be connected to ground if not needed by the application. The response is illustrated in the Figure 13 which displays the voltage (V) on the vertical axis and the time (μs) on the horizontal axis:

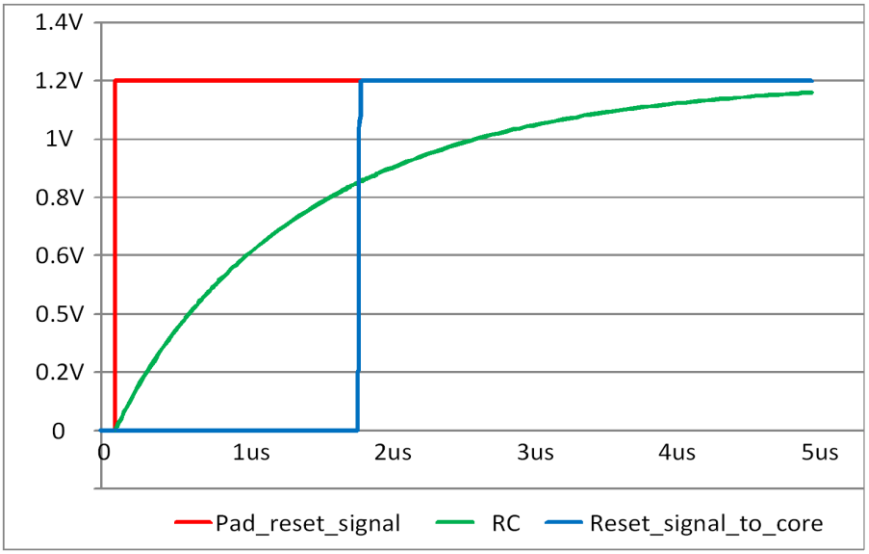


Figure 13: RST Pad Latency

The typical latency of the RST pad is in the range of 2 μs.

5.1 POR, HW and SW Reset

The Power-On Reset (POR) signal is generated:

- Internally and will release the system's flip flops as soon as the VDD voltage crosses the minimum threshold value.
- Externally by a Power-On Reset source (RST pad or GPIO).

There are three main reset signals in the DA14585.The PWR On reset which is triggered by a GPIO set as POR source with selectable polarity and/or the RST pad after a programmable time delay, the HW reset which is basically triggered by the RST pad when it becomes active for a short period of time (less than the programmable delay for POR) and the SW reset which is triggered by writing the SYS_CTRL_REG[SW_RESET] bit.

The HW reset can also be automatically activated upon waking up of the system from the Extended or Deep Sleep mode by programming bit PMU_CTRL_REG[RESET_ON_WAKEUP]. The PWR On reset as well as the HW reset will basically run the cold start-up sequence and the BootROM code will be executed.

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

The block diagram of the reset block is depicted in Figure 14.

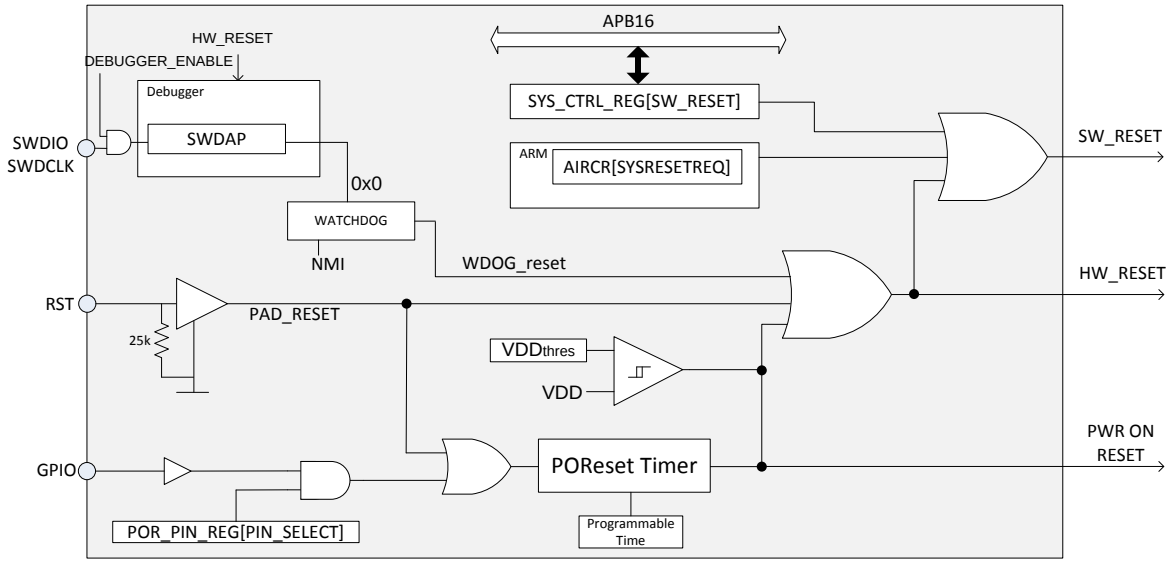


Figure 14: Reset Block Diagram

Certain registers are reset by POR only or by POR and the HW reset signal, but not by the SW reset. These registers are listed in Table 27.

Table 27: Reset Signals and Registers

Reset by POR Only	Reset by POR or HW Reset	Reset by POR, HW or SW Reset
BANDGAP_REG	OTPC_NWORDS_REG	The rest of the Register File
POR_PIN_REG	CLK_FREQ_TRIM_REG	
POR_TIMER_REG	CLK_RADIO_REG	
RF registers containing the trimming values for the VCO, the LNA and the I/O capacitance	All RF calibration registers	
	BLE_CNTL2_REG	
	CLK_CTRL_REG	
	PMU_CTRL_REG	
	SYS_CTRL_REG	
	CLK_32K_REG_I	
	CLK_16M_REG_I	
	CLK_RCX32K_REG	
	TRIM_CTRL_REG	
	DEBUG_REG[DEBUGS_FREEZE_EN]	
	GP_CONTROL_REG[EM_MAP]	

5.1.1 Power-On Reset Functionality

Power-On Reset functionality is available by two sources:

- Reset Pad: Reset pad is always capable of producing a Power-On Reset.

- GPIO Pin: A GPIO can be selected by the user application to act as POR source.

The time needed for the POReset pin to be active is stored in the POR_TIMER_REG. The register field POR_TIME is a 7-bits field which holds time factor that the total time for POR is calculated. The maximum value of the field is 0x7F. The total time for POR is calculated by the following formula:

Total time = POR_TIME x 4096 x RC32k clock period

where RC32k clock period = 31.25 µs at 25^{o}C .

The maximum time that a POR can be performed is ~16.2 seconds at 25^{o}C.

The RC32k clock is temperature dependent so based on the temperature span of -10^{o}C to 50^{o}C, clock frequency range is calculated to be 25kHz to 39kHz. Then,

$T_{PORcold}$ = 13 s

T_{PORhot} = 20.8 s

5.1.1.1 POR Timer Clock

The Power-On Reset timer is clocked by the RC32k clock. If the application disables the RC32k, then the hardware takes care of enabling the RC32k clock when the POR source (Reset pad or GPIO) is asserted. It should be noted that if POR is generated from the Reset pad the RC32 will operate with the reset trimming value. If a GPIO is used as POR source, the RC32 clock will be trimmed. The deviation between both cases in terms of timing is expected to be minor.

5.1.1.2 Reset Pad

The Reset pad will produce a HW Reset if the pin active time is less than the programmed value in the POR_TIMER_REG register and a Power-On Reset if it is greater or equal the value. Reset pad is always Active High.

5.1.1.3 POR from GPIO

When a GPIO is used as a Power-on Reset source, the selected pin retains its capability to act as GPIO. The POR_PIN_REG[PIN_SELECT] field holds the required GPIO pin number. If the value of the PIN_SELECT field equals to 0 the POR over GPIO functionality is disabled. The polarity of the pin can be configured by the POR_PIN_REG [POR_POLARITY] bit where 0 means Active Low and 1 Active High.

5.1.2 Power-On Reset Timing Diagram

The operation of the Power-On Reset for both Reset pad and GPIO is depicted in Figure 15.

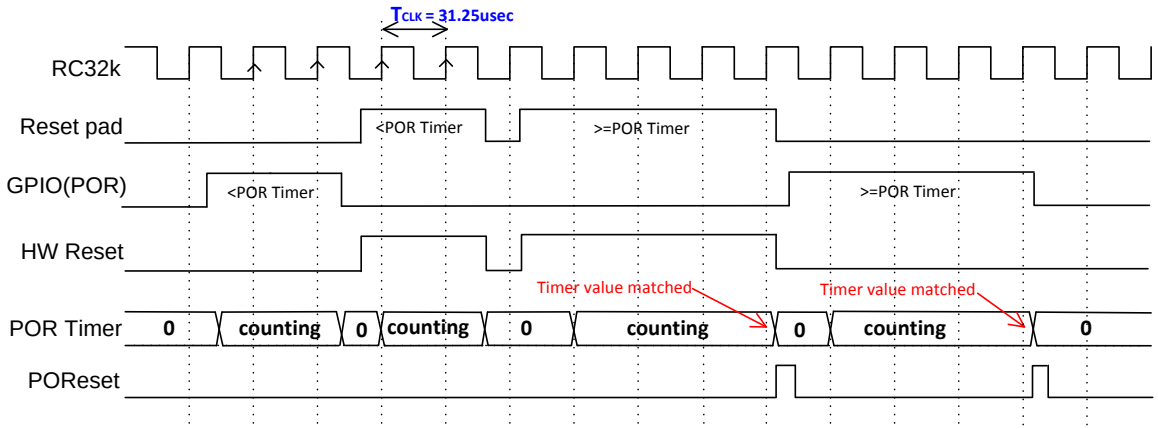


Figure 15: Power-On Reset Timing Diagram

5.1.3 Power-On Reset Considerations

If any of the POR sources is asserted then the POR timer starts to count. When a POR source is released before the timer has expired, POR timer will reset to 0. If a second source is asserted while the first is already asserted and the first is released after that point, POR will occur; assuming that the total time of both sources kept asserted is larger or equal than the POR_TIME.

The POR_PIN_REG[PIN_SELECT] field cannot survive any Reset (POR, HW, SW) and as such the user must take special care on setting up the GPIO POR source right after a reset. This also applies for the POR_TIMER_REG[POR_TIME] field after a Power-On Reset.

The user must also take into account that if a GPIO is used as POR source, the dynamic current of the system increases due to the dynamic current consumed by the RC32k oscillator. This increase is calculated to be 100 nA to 120 nA and it is also present during sleep time period. POR from Reset pin does not add this dynamic current consumption.

6 ARM Cortex-M0

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

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

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

A simplified block diagram of the Cortex-M0 is shown in Figure 16.

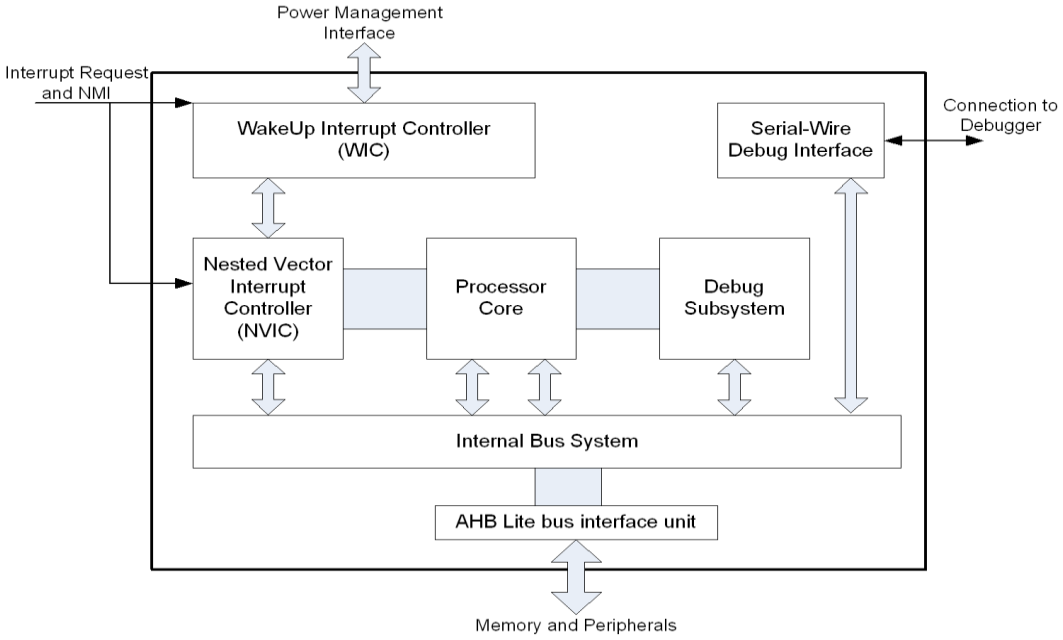


Figure 16: ARM Cortex-M0 Block Diagram

Features

- Thumb instruction set. Highly efficient, high code density and able to execute all Thumb instructions from the ARM7TDMI processor.
- High performance. Up to 0.9 DMIPS/MHz (Dhrystone 2.1) with fast multiplier.
- Built-in Nested Vectored Interrupt Controller (NVIC). This makes interrupt configuration and coding of exception handlers easy. When an interrupt request is taken, the corresponding interrupt handler is executed automatically without the need to determine the exception vector in software.
- Interrupts can have four different programmable priority levels. The NVIC automatically handles nested interrupts.
- The design is configured to respond to exceptions (e.g. interrupts) as soon as possible (minimum 16 clock cycles).
- Non maskable interrupt (NMI) input for safety critical systems.

- Easy to use and C friendly. There are only two modes (Thread mode and Handler mode). The whole application, including exception handlers, can be written in C without any assembler.
- Built-in System Tick timer for OS support. A 24-bit timer with a dedicated exception type is included in the architecture, which the OS can use as a tick timer or as a general timer in other applications without an OS.
- SuperVisor Call (SVC) instruction with a dedicated SVC exception and PendSV (Pendable SuperVisor service) to support various operations in an embedded OS.
- Architecturally defined sleep modes and instructions to enter sleep. The sleep features allow power consumption to be reduced dramatically. Defining sleep modes as an architectural feature makes porting of software easier because sleep is entered by a specific instruction rather than implementation defined control registers.
- Fault handling exception to catch various sources of errors in the system.
- Support for 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.

6.1 Interrupts

This section lists all 21 interrupt lines, except the NMI interrupt, and describes their source and functionality. The overview of the interrupts is illustrated in the following table:

Table 28: Interrupt list

IRQ number (inherent priority)	IRQ name	Description
0	BLE_WAKEUP_LP_IRQ	Wake-up from Low Power (Extended Sleep) interrupt from BLE.

IRQ number (inherent priority)	IRQ name	Description
1	BLE_GEN_IRQ	BLE Interrupt. Sources: - BLE_FINETGTIM_IRQn: Fine Target Timer interrupt generated when Fine Target timer expired. Timer resolution is 625µs base time reference - BLE_GROSSTGTIM_IRQn: Gross Target Timer interrupt generated when Gross Target timer expired. Timer resolution is 16 times 625µs base time reference - BLE_CSCNT_IRQn: 625µs base time reference interrupt, available in active modes - BLE_SLP_IRQn: End of Sleep mode interrupt - BLE_ERROR_IRQn: Error interrupt, generated when undesired behavior or bad programming occurs in the BLE Core - BLE_RX_IRQn: Receipt interrupt at the end of each received packets - BLE_EVENT_IRQn: End of Advertising / Scanning / Connection events interrupt - BLE_CRYPT_IRQn: Encryption / Decryption interrupt, generated either when AES and/or CCM processing is finished - BLE_SW_IRQn: SW triggered interrupt, generated on SW request
2	UART_IRQ	UART interrupt.
3	UART2_IRQ	UART2 interrupt.
4	I2C_IRQ	I2C interrupt.
5	SPI_IRQ	SPI interrupt.
6	ADC_IRQ	Analog-Digital Converter interrupt.
7	KEYBRD_IRQ	Keyboard interrupt.
8	BLE_RF_DIAG_IRQ	Baseband or Radio Diagnostics Interrupt. Triggered by internal events of the Radio or Baseband selected by the BLE_RF_DIAGIRQ_REG. For Debug purposes only.
9	RF_CAL_IRQ	RF Calibration Interrupt.
10	GPIO0_IRQ	GPIO interrupt through debounce.
11	GPIO1_IRQ	GPIO interrupt through debounce.
12	GPIO2_IRQ	GPIO interrupt through debounce.
13	GPIO3_IRQ	GPIO interrupt through debounce.
14	GPIO4_IRQ	GPIO interrupt through debounce.
15	SWTIM_IRQ	Software Timer interrupt.
16	WKUP_QUADEC_IRQ	Combines the Wake up Capture Timer interrupt, the GPIO interrupt and the QuadDecoder interrupt
17	PCM_IRQ	PCM interrupt.
18	SRC_IN_IRQ	Sample rate converter input interrupt.
19	SRC_OUT_IRQ	Sample rate converter output interrupt.
20	DMA_IRQ	DMA interrupt.

Interrupt priorities are programmable by the ARM Cortex-M0. The lower the priority number, the higher the priority level. The priority level is stored in a byte-wide register, which is set to 0x0 at reset. Interrupts with the same priority level follow a fixed priority order using the interrupt number listed in Table 28 (lower interrupt number has higher priority level).

To access the Cortex-M0 NVIC registers, CMSIS functions can be used. The input parameter IRQn of the CMSIS NVIC access functions is the IRQ number. This can be the IRQ number or (more convenient) the corresponding IRQ name listed in Table 28. The corresponding interrupt handler name in the vector table for IRQ#15 is e.g. SPI_Handler. For more information on the ARM Cortex-M0 interrupts and the corresponding CMSIS functions, see section 4.2 Nested Vectored Interrupt Controller on page 4-3 in the Cortex™-M0 User Guide Reference Material.

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

6.2 System Timer (systick)

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

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

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

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

6.3 Wake-Up Interrupt Controller

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

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

6.4 Reference

For more information on the ARM Cortex-M0, see the ARM documents listed in Table 29.

Table 29: ARM Documents List

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

7 AMBA Bus Overview

The DA14585 is based on an AMBA 2.0 AHB and APB components. The AHB is an AMBA Lite version which requires a single master on the system, i.e. in the DA14585 the ARM CPU. The APB interface implements 4 different decoding slaves which are grouped according to the power domain structure of the chip.

The AMBA bus organization is presented in Figure 17:

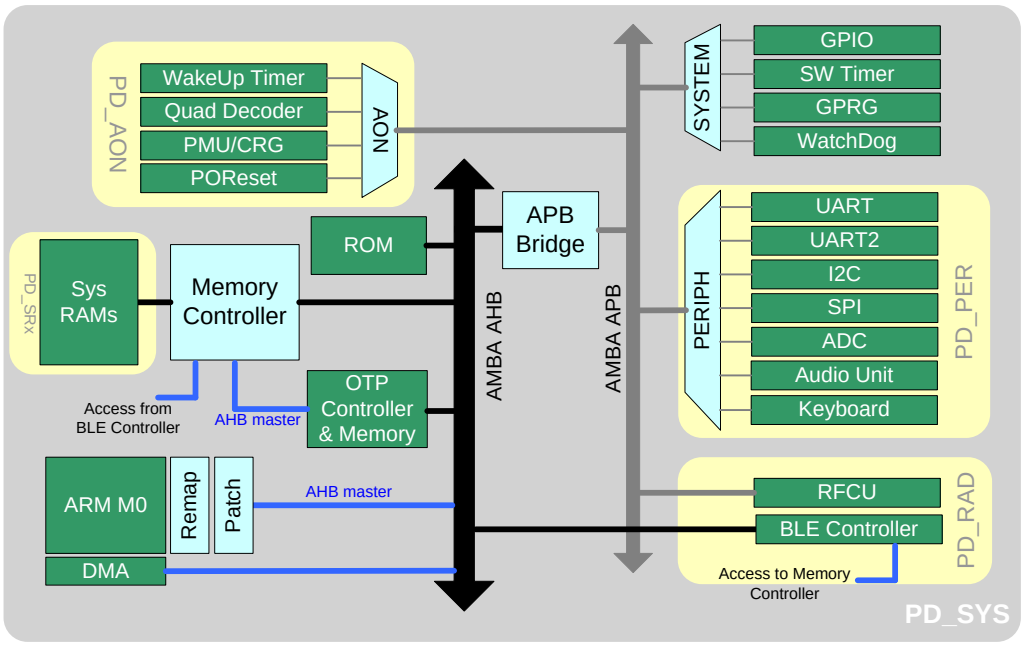


Figure 17: AMBA Bus Architecture and Power Domains

Since the DA14585 consists of several different power domains that are digitally controlled and can be shut down completely, various slave resources especially on the APB bus are grouped together to reduce signal isolation requirements.

On the AHB Lite bus, the CPU is the master while OTP, BLE Core, Memory and ROM controllers are the slaves. Furthermore, on the APB subsystem, the Always On power domain consists of the following blocks:

- Wake-up Timer
- Quadrature Decoder
- Power Management Unit / Clock Reset Generator including the Power-On Reset circuitry
- System RAMs (not mapped on APB but accessible through the Memory Controller)

The APB peripherals power domain comprises:

- UART and UART2
- SPI
- I2C
- General Purpose ADC
- Audio Unit (PDM, PCM, SRC)
- Keyboard Controller

The radio power domain contains both AHB and APB peripherals i.e. the Radio Frequency Control Unit (RFCU) and the Bluetooth Low Energy controller.

The rest of the system is mapped on the system power domain and contains APB blocks (the GPIO multiplexing, the Software Timer, the General Purpose Registers and the WatchDog timer) as well as AHB blocks (OTP, Memory and ROM controller).

8 Patch Block

The patch block provides a multiplexer on the data bus of the ROM Controller able to modify (patch) the contents of the data read from ROM. This feature provides a repair mechanism of a ROM image.

Features

- Data patching on ROM and OTP.
- Up-to 2 patch entries for data values
- Up-to 20 patch entries for ROM functions
- Enable/disable patch entries
- Priority scheme allows off-loads software
- Patch data can be stored internally in OTP or externally in EEPROM or SPI Flash memory

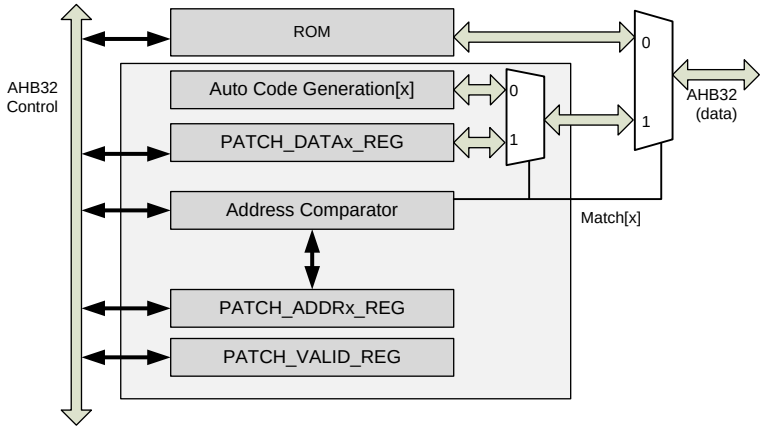


Figure 18: Patch Block

The patch block's registers are accessible via the AHB bus. The patch unit resides in the System Power domain. It is clocked by the system bus clock (HCLK) and must be initialized after each power-up or wake-up.

There are 2 data patching entries of 32-bit words each. A data patch entry consists of a 32-bit address register PATCH_ADDRx_REG and one 32-bit data register PATCH_DATAx_REG, implemented as sequential register pairs. The programmer must program the address register first followed by the data register. These patch entries are mostly used for patching a data value in the ROM image and are therefore designated as "data patch" entries.

Furthermore, the patch block contains an Auto Code Generation unit which automatically injects 5 16-bit instructions on the bus considered that the required patch involves a function replacement rather than a single data value correction. This short assembly code, seamlessly redirects the execution to a function with a base address stored in the patch table which resides right after the Interrupt Vector Table in SysRAM. Since the method replaces functions, it is considered to be a "function patch" entry in the patch address space.

Note: Special care must be taken when DMA is accessing patched data. The Patch block is attached to the CPU bus, patching only the CPU bus transactions. If the DMA tries to access a memory entry that is patched it will access the original rather than the patched data.

Patch Address Comparators

The patch unit has an address comparator to compare AHB address bus with PATCH_ADDRx_REG

With enabled patch entries, upon an address match, the data bus value is replaced with the corresponding PATCH_DATAx_REG. Matching uses priorities starting with highest priority at the last entry (PATCH_ADDR1_REG) and lowest priority at the first entry (PATCH_ADDR0_REG). This priority scheme helps SW to add more patches without having to check the previous entries with a lower entry number.

Only ROM and OTP memory areas are patchable.

Enabling or Disabling Patch Entries

Patch entries can be enabled or disabled by the corresponding bits at the PATCH_VALID_REG register.

To introduce new patches, SW must check the available patch entries, evaluate the priorities (e.g. if it will use two patch entries for the same address) and program the corresponding register pair and then enable the valid bit of this patch.

9 Memory Map

Table 30: Memory Map

Address	Description
0x00000000 0x0001FFFF	**Boot/BLE ROM/OTP/RAM** Remapped address space based on SYS_CTRL_REG[REMAP_ADR0].
0x00020000 0x04000000	**RESERVED**
0x07F00000 0x07F1FFFF	**Boot/BLE ROM** Contains 6 kB of Boot ROM code and 110 kB of Bluetooth LE protocol related code.
0x07F20000 0x07F3FFFF	**RESERVED**
0x07F40000 0x07F400FF	**OTP-Regs** Contains the control registers of the OTP Subsystem.
0x07F40100 0x07F7FFFF	**RESERVED**
0x07F80000 0x07F8FFFF	**OTP** Contains the OTP cell actual memory space.
0x07F90000 0x07FBFFFF	**RESERVED**
0x07FC0000 0x07FD7FFF	**System RAM** 96 kB. Contains Application code, data for the application. SysRAM1: 0x07FC0000 to 0x07FC7FFF SysRAM2: 0x07FC8000 to 0x07FCBFFF SysRAM3: 0x07FCC000 to 0x07FCFFFF SysRAM4: 0x07FD0000 to 0x07FD7FFF
0x07FD8000 0x3FFFFFFF	**RESERVED**
0x40000000 0x4001FFFF	**AHB/BLE-Regs** Contains the control registers of the Bluetooth LE Link Layer Processor.
0x40020000 0x4007FFFF	**RESERVED**
0x40080000 0x400803FF	**AHB/Patch-Regs** Contains the registers for the HW patching
0x40080400 0x4FFFFFFF	**RESERVED**
0x50000000 0x500000FF	**APB/PMU-CRG** Contains the control registers of the Power Management Unit and the Clock Generator.
0x50000100 0x500001FF	**APB/wake-up** Contains an event capture timers that can wake up the system.
0x50000200 0x500002FF	**APB/Quadrature Decoder** Contains Logic that implements a step counter for X and Y axis from a rotary encoder.
0x50000300 0x50000FFF	**RESERVED**

Address	Description
0x50001000 0x500010FF	**APB/UART** Contains the control registers of the UART.
0x50001100 0x500011FF	**APB/UART2** Contains the control registers of the UART2.
0x50001200 0x500012FF	**APB/SPI** Contains the control registers of the SPI interface.
0x50001300 0x500013FF	**APB/I2C** Contains the control registers of the I2C interface.
0x50001400 0x500014FF	**APB/Kbrd** Contains the registers of the Keyboard controller.
0x50001500 0x500015FF	**APB/ADC** Contains the registers of the 4 -channel ADC.
0x50001600 0x500016FF	**APB/AnaMisc** Contains registers for various analog blocks.
0x50001700 0x50001FFF	**RESERVED**
0x50002000 0x50002FFF	**APB/Radio** Contains the control registers of the Bluetooth Smart Radio.
0x50003000 0x500030FF	**APB/Ports** Contains the mode and direction registers of the GPIOs.
0x50003100 0x500031FF	**APB/Watchdog** Contains the control registers of the Watchdog timer.
0x50003200 0x500032FF	**APB/Version** Contains the version/revision of the chip.
0x50003300 0x500033FF	**APB/Gen Purpose** Contains general purpose control registers.
0x50003400 0x500034FF	**APB/Timer** Contains the control registers of the SW Timer.
0x50003500 0x500035FF	**APB/Radio Production Tool** Contains the control registers of the RFPT.
0x50003600 0x500036FF	**APB/DMA** Contains the control registers of the DMA.
0x50003700 0xDFFFFFFF	**RESERVED**
0xE0000000 0xE00FFFFF	**Internal Private Bus** Contains various registers of the ARM Cortex-M0.

Note 1 1 kB = 1024 bytes. 1 Mbit = 1024*1024 bits. 1 GB = 1024*1024*1024 bytes.

10 Memory Controller

The Memory controller is responsible for the interface of the memory cells with the masters of the system requesting for access. It comprises two arbiters which they use a fixed priority level scheme thus allow parallelization between the 3 main masters of the RAM. The memory controller also allows for the actual physical sequence of the RAM cells in a continuous memory space enabling activating only the required amount of SysRAM thus saving on power.

The block diagram is presented in Figure 19.

Features

- Two different capacities for the RAM cells with retention capability (16kB and 32kB).
- Arbitration between the AHB masters (CPU or DMAs), OTP and the BLE core with BLE to be highest priority by default.
- Retainable configuration of the RAM cells.
- Meets all timing constraints for any access to the physical cells.
- Transparently interfaces AHB busses to memory signaling
- Fixed arbitration algorithm with time sharing.

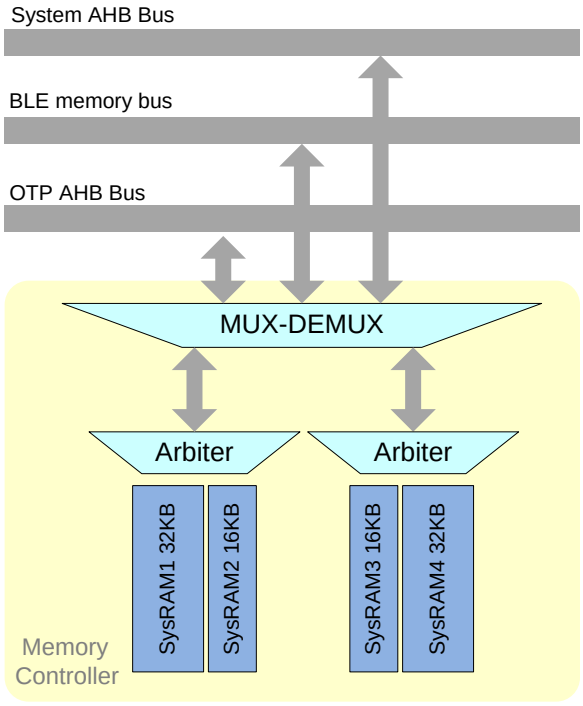


Figure 19: Memory Controller Block Diagram

The Memory Controller contains two Arbiters which connect through a Mux-Demux to the following busses:

- ICM which multiplexes the CPU or the DMA access. This interface is capable of operating at maximum 16MHz
- BLE Mem I/F: This is a memory interface from the Bluetooth 5.0 Core directly accessing the RAM used as exchange memory (TX/RX descriptors etc.). This interface is always operating at 16MHz.
- OTP Mem I/F: This is a memory interface from the OTP memory directly accessing the RAM used for copying data after power/wake up.

10.1 Arbitration

The arbitration is a mixture of highest priority and a fair use policy. If more masters request access to cells which reside under the same arbiter, time division is employed. This is to make sure none of the busses is able to stall another for a long period. The OTP and BLE accesses are handled as very critical and therefore they have highest priority. In general BLE and OTP accesses are mutual exclusive i.e. no BLE accesses occur while OTP requests. OTP has highest priority and will only occur during the OTP mirroring i.e. the system has not yet started executing application code. When OTP mirroring is done, BLE gets highest priority and only one every 3 BLE sequential cycles the CPU is allowed to get access. If none of the interfaces requests access, the IDLE or power down state is selected.

The Arbiter implements the priority scheme as depicted in the Figure 20:

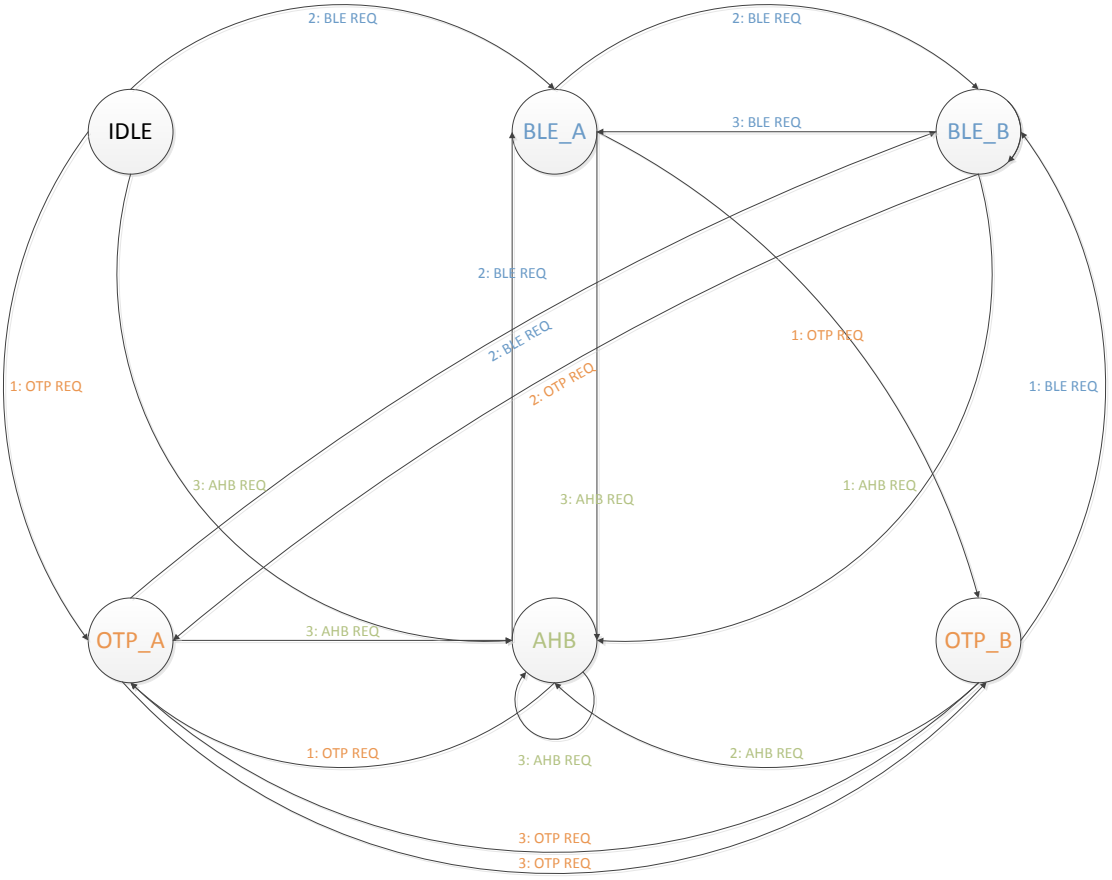


Figure 20: Arbitration Scheme

11 Clock Generation

11.1 Crystal Oscillators

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

The principle schematic of the two oscillators is shown in Figure 21. No external components to the DA14585 are required other than the crystal itself. If the crystal has a case connection, it is advised to connect the case to ground.

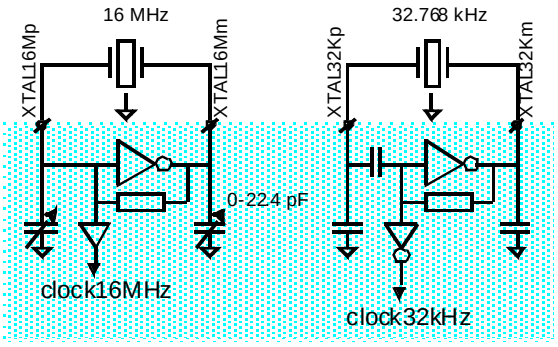


Figure 21: Crystal Oscillator Circuits

11.1.1 Frequency Control (16 MHz Crystal)

Register CLK_FREQ_TRIM_REG controls the trimming of the 16 MHz crystal oscillator. The frequency is trimmed by two on-chip variable capacitor banks. Both capacitor banks are controlled by the same register.

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

The eight least significant bits of CLK_FREQ_TRIM_REG directly control eight binary weighted capacitors, as shown in Figure 22. The three most significant bits are decoded according to Table 31. Each of the seven outputs of the decoder controls a capacitor (value is 256 times the value of the smallest capacitor).

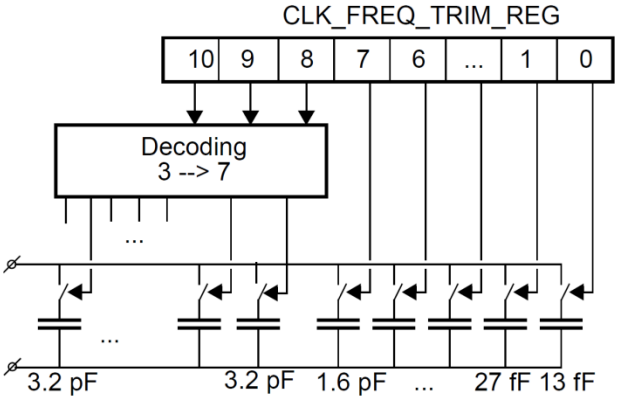


Figure 22: 16 MHz Crystal Oscillator Frequency Trimming

Table 31: CLK_FREQ_TRIM_REG Decoding 3 --> 7

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

Trimming might cause phase jumps in the oscillator signal. To reduce these phase jumps the user should only change one switch at a time (this especially applies to the seven larger capacitors). Use bits 10 to 8 for coarse adjustment and always increment or decrement this value by 1. Wait approximately 200 µs to allow the adjustment to settle.

Bits 7...0 are used for fine adjustment.

As an example, the recommended way to change the frequency trim register from 0x7FF to 0x100 is first to decrement the value of the three most significant bits by 1 at a time, and then change the least significant bits until the desired frequency is reached:
0x7FF --> 0x6FF -->0x5FF --> 0x4FF --> 0x3FF --> 0x2FF --> 0x1FF --> 0x100.

11.1.2 Automated Trimming Mechanism

There is provision in the DA14585 for automating the actual trimming of the 16 MHz crystal oscillator. This is a special hardware block that realizes the XTAL trimming in a single step.

Two programmable parameters are involved in the trimming of the XTAL16M oscillator:

- TRIM_CTRL_REG[TRIM_TIME]
  This field controls the time that elapses between the enabling of the XTAL and the application of the trim value specified by CLK_FREQ_TRIM_REG. The TRIM_TIME value counts cycles of 250 µs. As soon as TRIM_TIME*250 µs time has passed, the SYS_STAT_REG[XTAL16_TRIM_READY] bit becomes ‘1’. This is the point that the trim value will be applied at the oscillator. If TRIM_TIME = 0000 the trim value is applied immediately when XTAL16M starts up, to reduce time.
- TRIM_CTRL_REG[SETTLE_TIME]
  This field controls a counter that counts 250 µs intervals starting when XTAL16_TRIM_READY has been asserted. Upon completion of the SETTLE_TIME periods, the XTAL_SETTLED bit in the SYS_STAT_REG becomes ‘1’, indicating that XTAL16M clock is settled and trimmed and may be safely used as the system clock.

An initial manual frequency measuring action is required to store the wanted value in the OTP header (See section OTP Memory Layout).

11.2 RC Oscillators

There are three RC oscillators in the DA14585: one providing 16 MHz (RC16M), one providing 32 kHz (RC32K) and one providing a frequency 10 kHz (RCX).

The 16 MHz RC oscillator is powered by the Digital LDO i.e. the VDD = 1.2 V which is available for the core logic during Active or Sleep mode. The output clock is slightly slower than 16 MHz and is used to clock the system during the OTP mirroring procedure after wake-up, while the XTAL16M oscillator is settling. It is important to keep the 16 MHz RC oscillator at a frequency equal or below

16 MHz to guarantee correct behavior of the digital circuits. This is why the reset value of CLK_16M_REG[RC16M_TRIM] is equal to the minimum value.

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

The enhanced RC oscillator (RCX) provides a stable 10 kHz frequency and operates only if the external voltage is > 1.8 V (BUCK mode only).The RCX oscillator can be used to replace a 32.768 kHz crystal, since it has a precision of < 500 ppm, while its output frequency is quite stable over temperature. Although a different frequency is used, the accuracy is good enough to ensure that the correct number of slots (625 µs) is counted for the Sleep time.

11.2.1 Frequency Calibration

The output frequency of the 32 kHz crystal oscillator and the three RC-oscillators can be measured relative to the 16 MHz crystal oscillator using the on-chip reference counter.

The measurement procedure is as follows:

1. REF_CNT_VAL = N (the higher N, the more accurate and longer the calibration will be)
2. CLK_REF_SEL_REG = 0x0006 (XTAL32K) or
   CLK_REF_SEL_REG = 0x0005 (RC16M) or
   CLK_REF_SEL_REG = 0x0004 (RC32K) or
   CLK_REF_SEL_REG = 0x0007 (RCX)
3. Wait until
   CLK_REF_SEL_REG[REF_CAL_START] = 0
4. Read CLK_REF_VAL_H_REG and CLK_REF_VAL_H_REG = M (32-bits values)
5. Frequency = (N/M) * 16 MHz

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

11.3 System Clock Generation

The DA14585 clock generation overview is presented in the following figure. External clock sources of the device are pins XTAL16Mp/XTAL16Mm and pins XTAL32Kp/XTAL32Km for the 16 MHz and 32.768 kHz oscillators respectively.

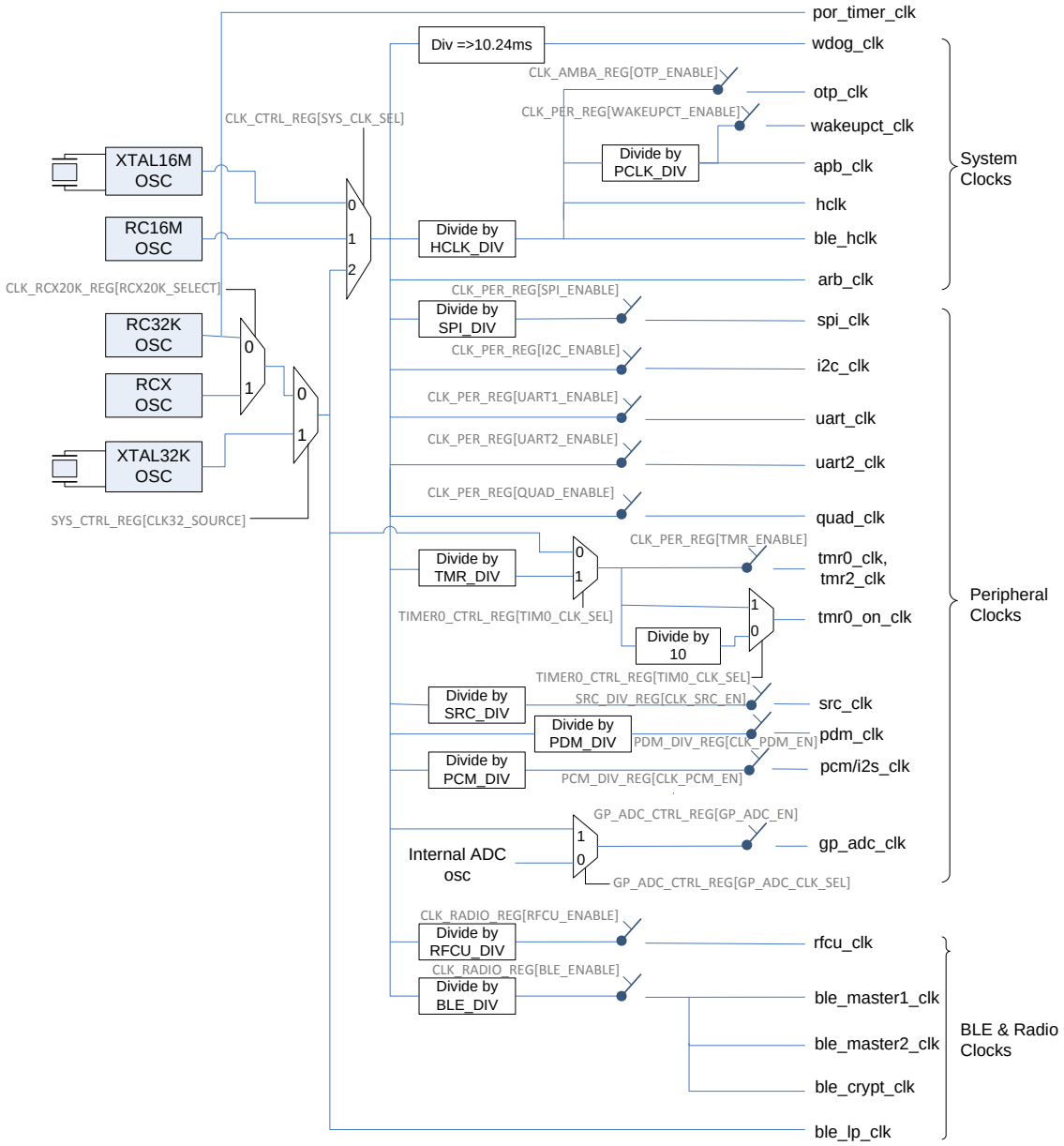


Figure 23: Clock Generation Block Diagram

Each of the clocks is generated by either a divider or a clock gate. Both dividers and clock gates are able to start/stop the clock of the module they drive. The control of the peripheral clocks is provided by CLK_PER_REG.

The clock names depicted in Figure 23: Clock Generation Block Diagram are explained in Table 32:

Table 32: Generated Clocks Description

Clock Name	Description
wakeupct_clk	Wake-up capture timer clock.
wdog_clk	Watchdog Clock.
apb_clk	AMBA APB interface clock
otp_clk	OTP controller clock. This clock is also used for the OTP macro cell and should not be higher than 16 MHz
hclk	AMBA AHB interface clock

Clock Name	Description
ble_hclk	AMBA AHB clock for the BLE core
wdog_clk	Watchdog clock.
por_timer_clk	Clock for the Power-On Reset circuitry. This clock is always enabled if POR from GPIO functionality is enabled
spi_clk	Clock for the SPI controller. This clock is further divided by 2, 4, 8 or 14 as defined by SPI_CTRL_REG[SPI_CLK]
i2c_clk	Clock for the I2C controller. This clock is further divided to provide 100 kHz or 400 kHz as defined by I2C_CON_REG[I2C_SPEED]
uart_clk	Clock for the UART
uart2_clk	Clock for the UART2
quad_clk	Clock for the quadrature decoders
rfcu_clk	Clock for the RF control unit of the Radio
tmr0_clk, tmr2_clk	Timer0/2 clocks
tmr0_on_clk	Timer0 ON counter clock
src_clk	Sample Rate Converter Clock
pdm_clk	PDM clock
pcm/i2s_clk	PCM/I2S clock
gp_adc_clk	General Purpose ADC conversion clock
ble_crypt_clk	Clock for the Crypto block of the BLE core
ble_master1_clk	Internal clock for the BLE core
ble_master2_clk	Internal clock for the BLE core
arb_clk	Clock for the memory controller arbiter
ble_lp_clk	BLE core low power clock

11.4 General Clock Constraints

There are certain constraints on various clocks regarding their frequency relations or the effectiveness. This section summarizes these rules:

- Minimum hclk clock have to be 8 MHz when BLE is utilized. This is also the clock of the ARM CPU and ensures the required MIPS for the Bluetooth LE Protocol handling.
- hclk should always be greater or equal to the ble_*_clks. This is required for the proper operation of the protocol. For example, hclk at 16 MHz and BLE clocks at 8 MHz is an acceptable combination but not the other way around.
- When the SPI block is set to Slave mode, then the spi_clk (as depicted in Figure 23) should be at least 4x the clock frequency provided by the external SPI Master. This is required for proper sampling of the data line.

12 OTP Controller

12.1 Introduction

The OTP controller realizes all functions of the OTP macro cell in an automated and transparent way. The controller facilitates all data transfers (reading and programming), while implementing the required OTP test modes in hardware. The controller comprises a DMA engine which connects to the AHB-DMA bus and has maximum priority to copy code from OTP into SysRAM in mirrored mode. The block diagram is presented in Figure 24.

Features

- Implements all timing constraints for any access to the physical memory cells.
- Automatic single Error Code Correction (ECC) and double error detection.
- Built-In Self Repair (BISR) mechanism for programming and reading.
- 64-bits read in a single clock cycle from the OTP cell.
- Embedded DMA engine for fast mirroring of the OTP contents into the SysRAM.
- Embedded DMA supports reading in bursts of 8 32-bit words.
- Transparent random address access to the OTP memory cells via the AHB slave memory interface.
- Hardwired handshaking with the PMU to realize the mirroring procedure.

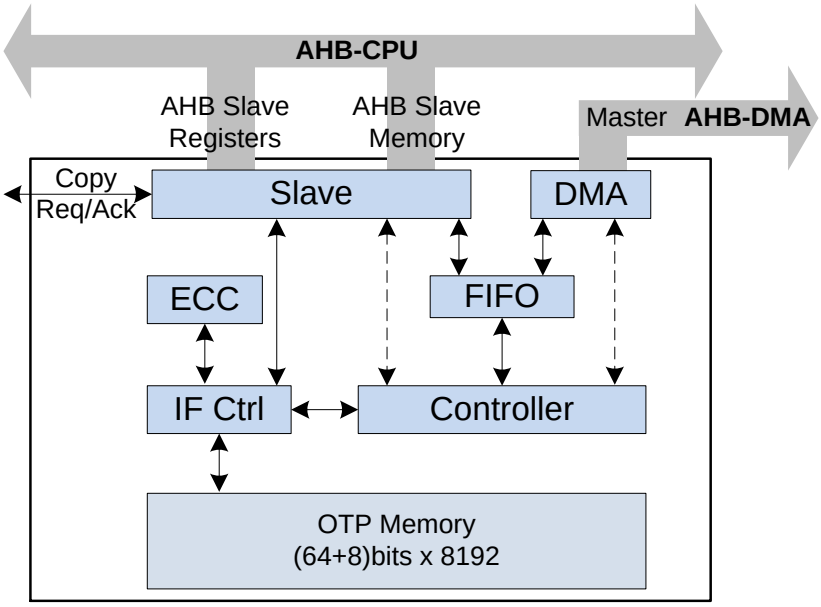


Figure 24: OTP Controller Block Diagram

12.2 Operating Modes

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

The **MREAD** mode enables the use of the memory slave interface. By activating this mode the contents of the macro cell are transparently mapped onto the specific AHB slave address space. The controller runs the SECDED algorithm on the fly to correct single bit errors and notify for dual bit errors by means of the status register (OTPC_STAT_REG[OTPC_STAT_RERROR]). This mode can be used for execution of software in place (XIP). In a case of a dual bit error the controller can also produce a bus error, which causes a Hard Fault exception in the CPU. The configuration bit OTPC_MODE_REG[OTPC_MODE_ERR_RESP_DIS] controls this feature.

The **AREAD** mode provides the ability for reading data from the macro cell in bursts, without the use of the slave interface. This mode is used for copying large data blocks from the macro cell, as in the case of the OTP mirroring into the SysRAM. As in the MREAD mode, SECDED is also applied on every burst and the OTPC_STAT_REG OTPC_STAT_RERROR] indicates the existence of a dual bit error. However, a transfer will occur even if an error has been identified in the status register and there is no generation of Hard Fault exception.

The **MPROG** mode provides the functionality for programming a 64 bit word. The controller expands the 64 bit word by calculating and appending an 8-bit checksum, implementing SECDED (Single Error Correction Double Error Detection). In this way a complete 72 bits word is constructed, which is stored at a selected OTP position.
Programming is performed in a single step. In the case that one or more bits have failed to be programmed correctly, software should trigger re-programming. If a re-programming is attempted, the OTP controller will try to write the bits that had not programmed in the last attempt. The total attempts for programming a word should not exceed the number of 10.
The 64 bit data word as well as the address is defined through configuration registers. The controller applies the corresponding control sequence for the programming and the result of the verification step is indicated in a status register. Note that the BISR capability of the OTP controller is not performed by the hardware in case of the MPROG.

The **APROG** mode gives the ability for programming large data blocks into the macro cell. The programming is an automated procedure, during which it is only necessary to feed the controller with the required data. Data blocks can be fetched in two ways:

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

In the latter case, data are pumped into the OTP controller through a register, which acts as a port providing access to a FIFO. The controller expands each 64 bit word by calculating and adding automatically an 8-bit checksum, in order to provide SECDED functionality. The programming attempts that may be required for the writing of each word are performed by the OTP controller automatically. The BISR capability of the controller is active and will be applied by the hardware automatically whenever is required.

In both programming methods (MPROG and APROG), the expected to be programmed addresses must be blank. The programming of an OTP address that contains already a value will result in a corrupted OTP word.

12.3 AHB Master Interface

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

The DMA engine internally supports the following burst types:

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

12.4 AHB Slave Interface

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

12.5 Error Correcting Code (ECC)

The error correcting code is based on the Hamming code, for a single bit error correction. The functionality of the Hamming code has been enhanced with the addition of a parity bit. The presence of the parity bit enables the detection of a double bit error. The redundancy that is provided by the use of the two algorithms (Hamming and parity generation) is stored together with the actual data at each OTP position consisting of a 72 bits word. The exact layout of the OTP word is presented in the following figure:

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

Figure 25: OTP Word Layout

12.6 BUILD-IN Self Repair (BISR)

The repair mechanism is available during programming (APROG mode only) or reading (Both AREAD and MREAD). The main purpose of the BISR mechanism is to repair the OTP addresses that are corrupted, by replacing the corrupted word with a spare word that is available in the OTP memory. The spare words are placed into the spare rows of the OTP memory. A total number of the 8 words can be repaired by the BISR logic. The BISR is utilized by the OTP controller automatically when the APROG mode is enabled.

The BISR can also be enabled during the production test of the OTP memory. It can repair OTP addresses that were corrupted during the production process of the chip and are not blank (occupied spare rows).

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

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

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

If the address, of the read requested, is found in one of the repair records, the data are not retrieved from the normal memory array of the OTP, but from the repair records. The ECC is, in this case, bypassed. If there is no match to a repair record, data are retrieved from the normal memory array. ECC is then activated.

13 DMA controller

The DMA controller has 4 Direct Memory Access (DMA) channels for fast data transfers from/to SPI, UART, I2C, PDM and PCM to/from any on-chip RAM. The DMA controller off-loads the ARM interrupt rate if an interrupts is given after a number of transfers. More peripherals DMA requests are multiplexed on the 4 available channels, to increase utilization of the DMA. The block diagram of the DMA controller is depicted in Figure 21.

Features

- 4 channels with optional peripheral trigger
- Full 32 bit source and destination pointers.
- Flexible interrupt generation.
- Programmable length.
- Flexible peripheral request per channel.
- Option to initialize memory.
- Programmable Edge-Sensitive request support(suggested for UART and I2C service)

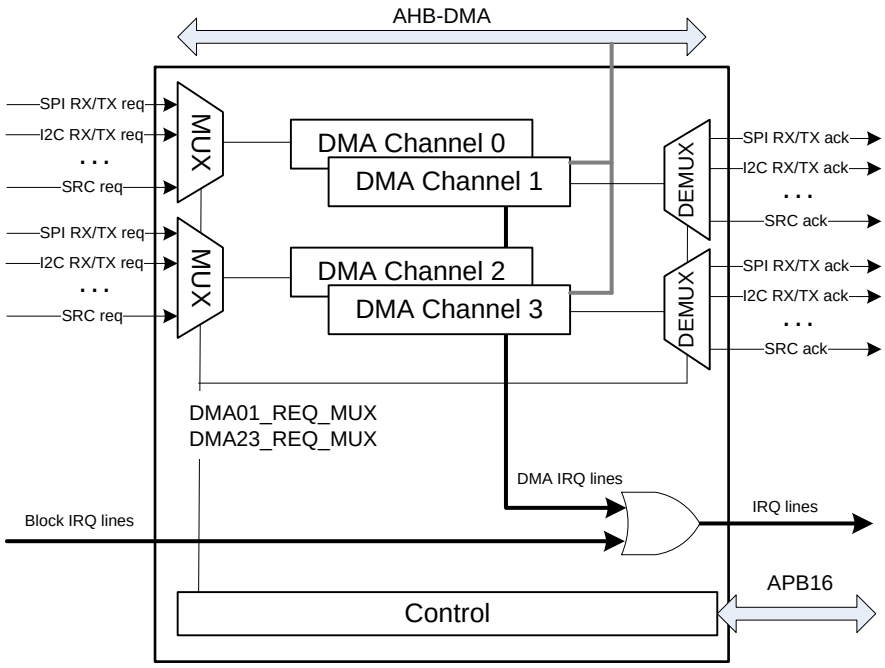


Figure 26: DMA Controller Block Diagram

13.1 DMA Peripherals

The list of peripherals that can request for a DMA service is presented in the Table 33.

Table 33: DMA Served Peripherals

Name	Direction
SPI	RX
SPI	TX
UART	RX
UART	TX
UART2	RX
UART2	TX
I2C	RX

Name	Direction
I2C	TX
PCM	RX
PCM	TX
SRC	RX
SRC	TX

13.2 Input/Output Multiplexer

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

Similarly, an acknowledging de-multiplexing mechanism is applied.

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

13.3 DMA Channel Operation

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

If DMA starts, data is transferred from address DMAx_A_START_REG to address DMAx_B_START_REG for a length of DMAx_LEN_REG, which can be 8, 16 or 32 bits wide. The address increment is realized with an internal 13 bits counter DMAx_IDX_REG, which is set to 0 if the DMA transfer starts and is compared with the DMA_LEN_REG after each transfer. The register value is multiplied according to the AINC and BINC and BW values before it is added to DMA_B_START_REG and DMA_B_START_REG. AINC or BINC must be 0 for register access.

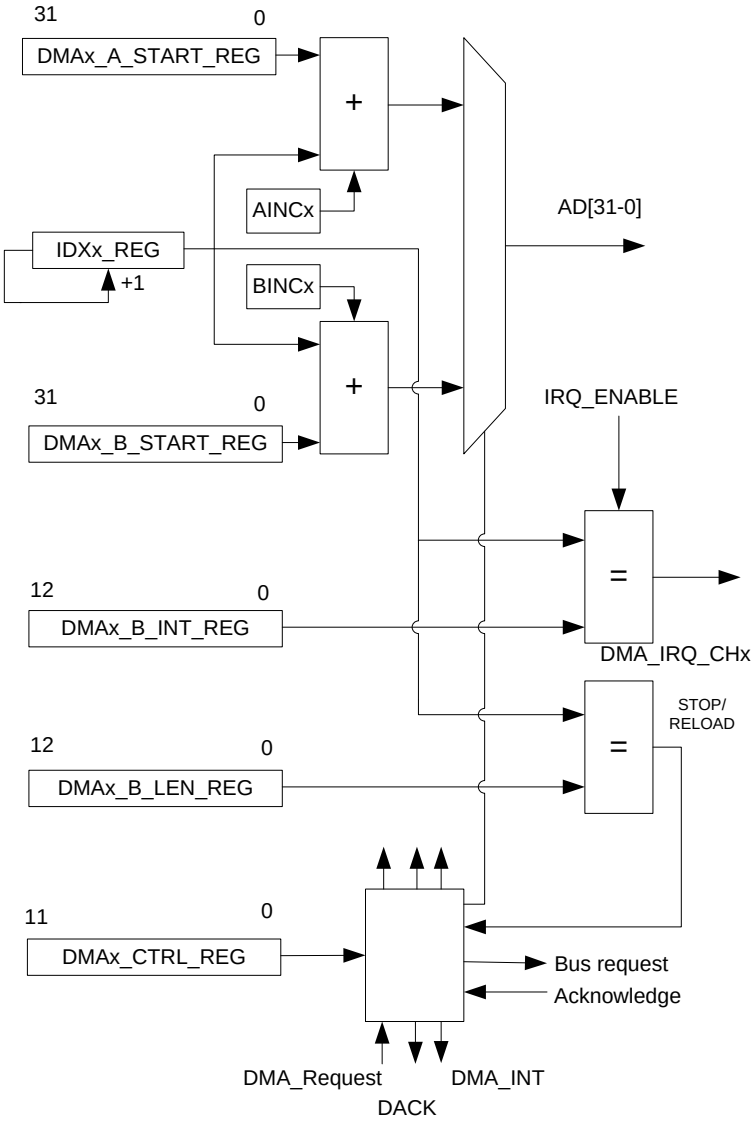


Figure 27: DMA Channel Diagram

If at the end of a DMA cycle, the DMA start condition is still true, the DMA continues. The DMA stops if DREQ_MODE is low or if DMAx_LEN_REG is equal to the internal index register. This condition also clears the DMA_ON bit.

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

Each DMA channel can generate an interrupt if DMAx_INT_REG if equal to DMAx_IDX_REG. After the transfer and before DMAx_IDX_REG is incremented, the interrupt is generated.

Example: if DMA_x_INT_REG=0 and DMA_x_LEN_REG=0, there will be one transfer and an interrupt.

13.4 DMA Arbitration

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

With DREQ_MODE = 0, a DMA can be interrupted by a channel with a higher priority if the DMA_IDLE bit is set.

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

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

13.5 Freezing DMA channels

Each channel of the DMA controller can be temporarily disabled by writing a 1 to freeze all channels at SET_FREEZE_REG.

To enable the channels again, a 1 to bits at the RESET_FREEZE_REG must be written.

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

It is noted that the on-going Memory-to-Memory transfers (DREQ_MODE='0') cannot be interrupted. Thus, in that case, the corresponding DMA channels will be frozen after any on-going Memory-to-memory transfer is completed.

14 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
- 32B deep transmit/receive FIFOs
- Master transmit, Master receive operation
- 7-bit or 10-bit addressing
- 7-bit or 10-bit combined format transfers
- Bulk transmit mode
- Default slave address of 0x055
- Interrupt or polled-mode operation
- Handles Bit and Byte waiting at both bus speeds
- Programmable SDA hold time
- DMA support

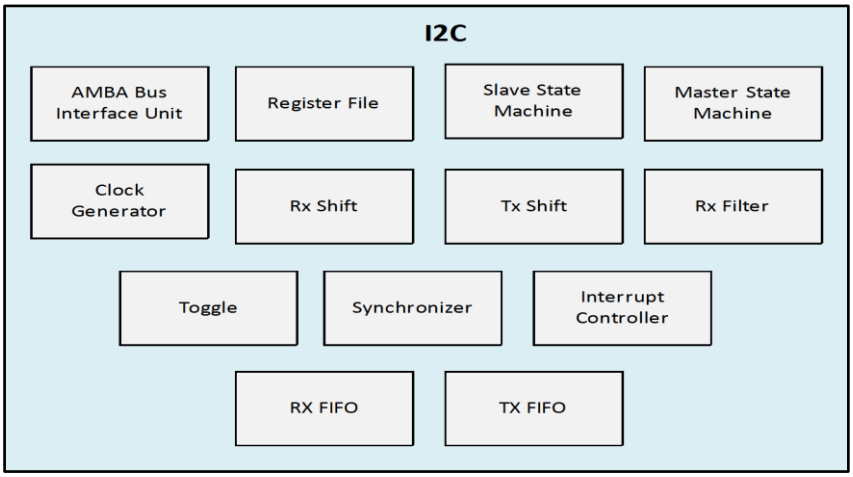


Figure 28: I2C Controller Block Diagram

The I2C Controller block diagram is shown in Figure 28. It contains the following sub-blocks:

- AMBA Bus Interface Unit. Interfacing via the APB interface to access the register file.
- Register File. Contains configuration registers and is the interface with software.
- Master State Machine. Generates the I2C protocol for the master transfers.
- Clock Generator. Calculates the required timing to do the following:
  - Generate the SCL clock when configured as a master
  - Check for bus idle
  - Generate a START and a STOP
  - Setup the data and hold the data

- Rx Shift. Takes data into the design and extracts it in byte format.
- Tx Shift. Presents data supplied by CPU for transfer on the I2C bus.
- Rx Filter. Detects the events in the bus; for example, start, stop and arbitration lost.
- Toggle. Generates pulses on both sides and toggles to transfer signals across clock domains.
- Synchronizer. Transfers signals from one clock domain to another.
- Interrupt Controller. Generates the raw interrupt and interrupt flags, allowing them to be set and cleared.
- RX FIFO/TX. Holds the RX FIFO and TX FIFO register banks and controllers, along with their status levels.

14.1 I2C Bus Terms

The following terms relate to how the role of the I2C device and how it interacts with other I2C devices on the bus.

- Transmitter. The device that sends data to the bus. A transmitter can either be a device that initiates the data transmission to the bus (a master-transmitter) or responds to a request from the master to send data to the bus (a slave-transmitter).
- Receiver. The device that receives data from the bus. A receiver can either be a device that receives data on its own request (a master-receiver) or in response to a request from the master (a slave-receiver).
- Master. The component that initializes a transfer (START command), generates the clock (SCL) signal and terminates the transfer (STOP command). A master can be either a transmitter or a receiver.
- Slave. The device addressed by the master. A slave can be either receiver or transmitter.

These concepts are illustrated in Figure 29.

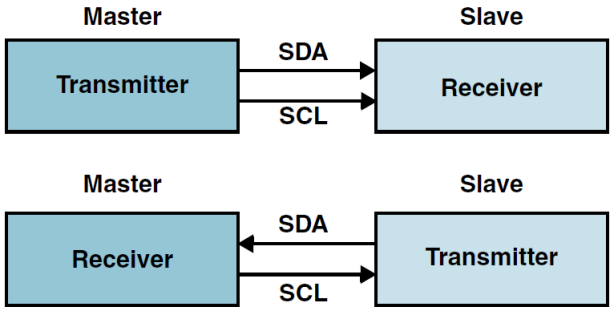


Figure 29: Master/Slave and Transmitter/Receiver Relationships

- Multi-master. The ability for more than one master to co-exist on the bus at the same time without collision or data loss.
- Arbitration. The predefined procedure that authorizes only one master at a time to take control of the bus. For more information about this behavior, refer to Multiple Master Arbitration section.
- Synchronization. The predefined procedure that synchronizes the clock signals provided by two or more masters. For more information about this feature, refer to Clock Synchronization s.
- SDA. Data signal line (Serial Data)
- SCL. Clock signal line (Serial Clock)

14.1.1 Bus Transfer Terms

The following terms are specific to data transfers that occur to/from the I2C bus.

- START (RESTART). Data transfer begins with a START or RESTART condition. The level of the SDA data line changes from high to low, while the SCL clock line remains high. When this occurs, the bus becomes busy.
- STOP. Data transfer is terminated by a STOP condition. This occurs when the level on the SDA data line passes from the low state to the high state, while the SCL clock line remains high. When the data transfer has been terminated, the bus is free or idle once again. The bus stays busy if a RESTART is generated instead of a STOP condition.

Note: START and RESTART conditions are functionally identical.

14.2 I2C Behavior

The I2C can be only be controlled via software to be an I2C master only, communicating with other I2C slaves;

The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The acknowledgement of data is sent by the device that is receiving data, which can be either a master or a slave. As mentioned previously, the I2C protocol also allows multiple masters to reside on the I2C bus and uses an arbitration procedure to determine bus ownership.

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledge pulse (ACK) after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behavior is illustrated in Figure 30.

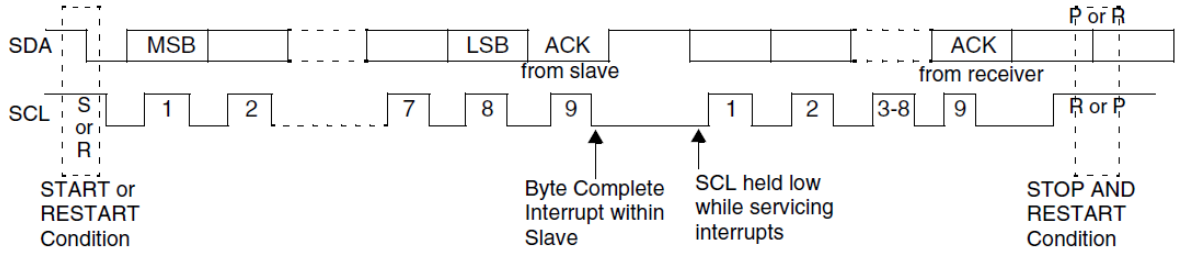


Figure 30: Data Transfer on the I2C Bus

The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only while the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wire-AND functions on the bus. The maximum number of devices on the bus is limited by only the maximum capacitance specification of 400 pF. Data is transmitted in byte packages.

14.2.1 START and STOP Generation

When operating as an I2C master, putting data into the transmit FIFO causes the I2C Controller to generate a START condition on the I2C bus. Allowing the transmit FIFO to empty causes the I2C Controller to generate a STOP condition on the I2C bus.

When operating as a slave, the I2C Controller does not generate START and STOP conditions, as per the protocol. However, if a read request is made to the I2C Controller, it holds the SCL line low

until read data has been supplied to it. This stalls the I2C bus until read data is provided to the slave I2C Controller, or the I2C Controller slave is disabled by writing a 0 to I2C_ENABLE.

14.2.2 Combined Formats

The I2C Controller supports mixed read and write combined format transactions in both 7-bit and 10-bit addressing modes.

The I2C Controller does not support mixed address and mixed address format - that is, a 7-bit address transaction followed by a 10-bit address transaction or vice versa - combined format transactions.

To initiate combined format transfers, I2C_CON.I2C_RESTART_EN should be set to 1. With this value set and operating as a master, when the I2C Controller completes an I2C transfer, it checks the transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the transmit FIFO is empty when the current I2C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

14.3 I2C Protocols

The I2C Controller has the following protocols:

- START and STOP Conditions
- Addressing Slave Protocol
- Transmitting and Receiving Protocol
- START BYTE Transfer Protocol

14.3.1 START and STOP Conditions

When the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master wants to terminate the transmission, the master issues a STOP condition. This is defined to be a low-to-high transition of the SDA line while SCL is 1. Figure 31 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when SCL is 1.

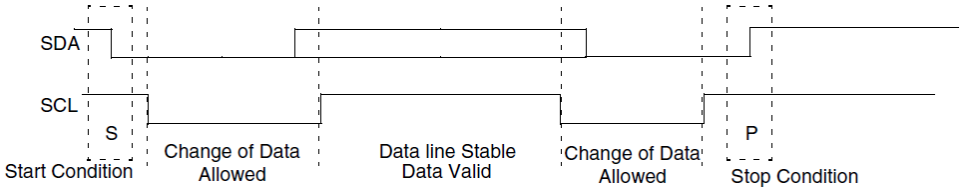


Figure 31: START and STOP Conditions

Note: The signal transitions for the START/STOP conditions, as depicted in Figure 31, reflect those observed at the output signals of the Master driving the I2C bus. Care should be taken when observing the SDA/SCL signals at the input signals of the Slave(s), because unequal line delays may result in an incorrect SDA/SCL timing relationship.

14.3.2 Addressing Slave Protocol

There are two address formats: 7-bit address format and 10-bit address format.

7-bit Address Format

During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit as shown in Figure 32. When bit 0 (R/W) is set to 0, the master writes to the slave. When bit 0 (R/W) is set to 1, the master reads from the slave.

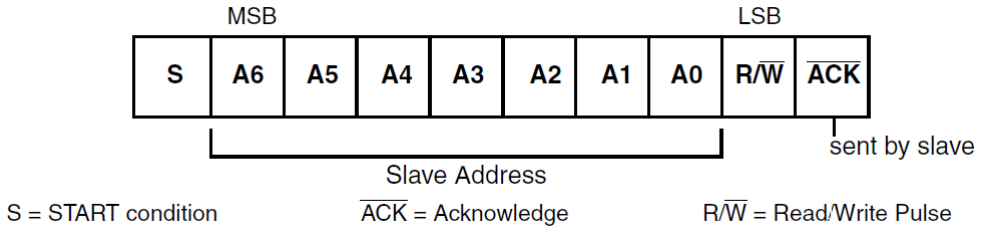


Figure 32: 7-bit Address Format

10-bit Address Format

During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slaves address bits 9:8, and the LSB bit (bit 0) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address. Figure 33 shows the 10-bit address format, and Table 34 defines the special purpose and reserved first byte addresses.

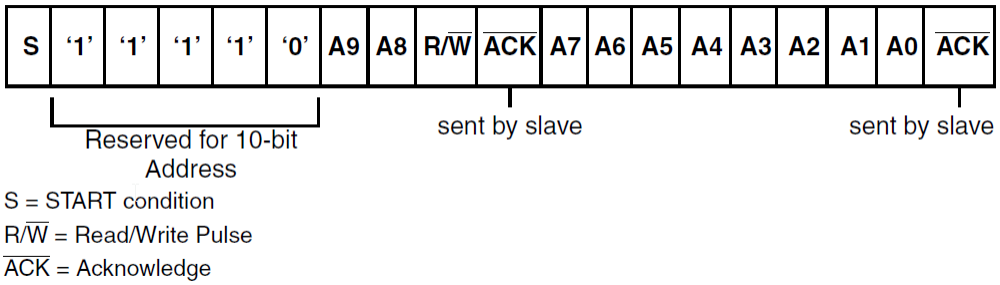


Figure 33: 10-bit Address Format

Table 34: I2C Definition of Bits in First Byte

Slave Address	R/W Bits	Description
0000 000	0	General Call Address. I2C Controller places the data in the receive buffer and issues a General Call interrupt.
0000 000	1	START byte. For more details, refer to “START BYTE Transfer Protocol” 0000
0000 001	X	CBUS address. I2C Controller ignores these accesses
0000 010	X	Reserved
0000 011	X	Reserved
0000 1XX	X	High-speed master code (for more information, refer to “Multiple Master Arbitration”
1111 1XX	X	Reserved
1111 0XX	X	10-bit slave addressing

The I2C Controller does not restrict you from using these reserved addresses. However, if you use these reserved addresses, you may run into incompatibilities with other I2C components.

14.3.3 Transmitting and Receiving Protocols

The master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively.

Master-Transmitter and Slave-Receiver

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal (ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

If the master-transmitter is transmitting data as shown in Figure 34, then the slave-receiver responds to the master-transmitter with an acknowledge pulse after every byte of data is received.

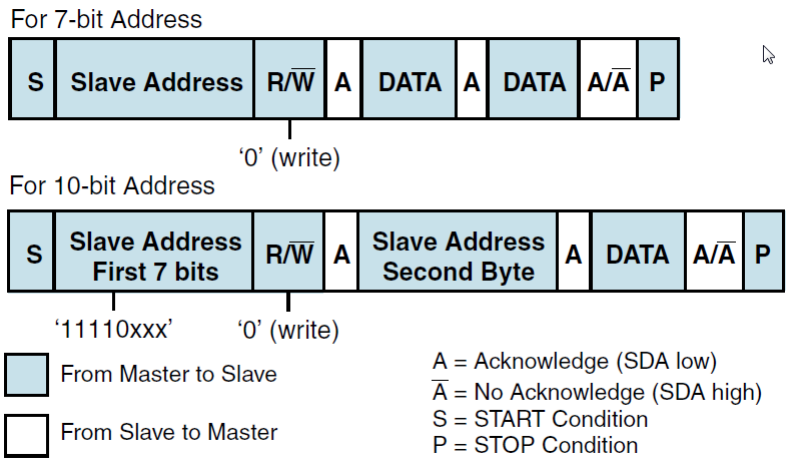


Figure 34: Master-Transmitter Protocol

Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 35 then the master responds to the slave-transmitter with an acknowledge pulse after a byte of data has been received, except for the last byte. This is the way the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

When a master does not want to relinquish the bus with a STOP condition, the master can issue a RESTART condition. This is identical to a START condition except it occurs after the ACK pulse. The master can then communicate with the same slave or a different slave.

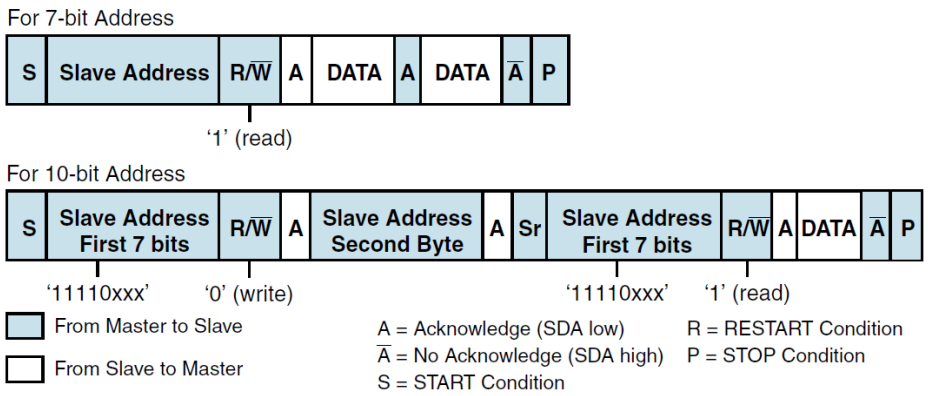


Figure 35: Master-Receiver Protocol

START BYTE Transfer Protocol

The START BYTE transfer protocol is set up for systems that do not have an on-board dedicated I2C hardware module. When the I2C Controller is addressed as a slave, it always samples the I2C bus at the highest speed supported so that it never requires a START BYTE transfer. However, when I2C

Controller is a master, it supports the generation of START BYTE transfers at the beginning of every transfer in case a slave device requires it. This protocol consists of seven zeros being transmitted followed by a 1, as illustrated in Figure 36. This allows the processor that is polling the bus to under-sample the address phase until 0 is detected. Once the microcontroller detects a 0, it switches from the under sampling rate to the correct rate of the master.

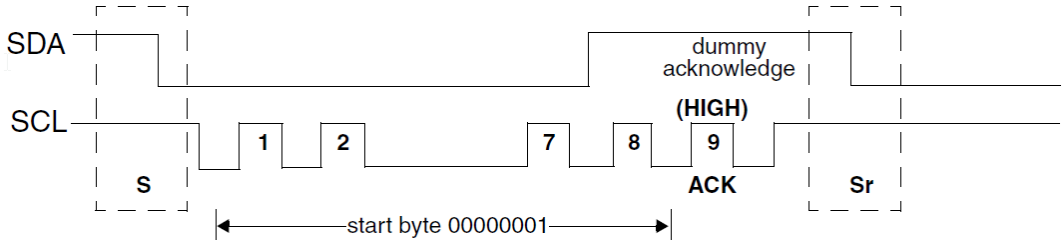


Figure 36: START BYTE Transfer

The START BYTE procedure is as follows:

1. Master generates a START condition.
2. Master transmits the START byte (0000 0001).
3. Master transmits the ACK clock pulse. (Present only to conform with the byte handling format used on the bus)
4. No slave sets the ACK signal to 0.
5. Master generates a RESTART (R) condition.

A hardware receiver does not respond to the START BYTE because it is a reserved address and resets after the RESTART condition is generated.

14.4 Multiple Master Arbitration

The I2C Controller bus protocol allows multiple masters to reside on the same bus. If there are two masters on the same I2C-bus, there is an arbitration procedure if both try to take control of the bus at the same time by generating a START condition at the same time. Once a master (for example, a microcontroller) has control of the bus, no other master can take control until the first master sends a STOP condition and places the bus in an idle state.

Arbitration takes place on the SDA line, while the SCL line is 1. The master, which transmits a 1 while the other master transmits 0, loses arbitration and turns off its data output stage. The master that lost arbitration can continue to generate clocks until the end of the byte transfer. If both masters are addressing the same slave device, the arbitration could go into the data phase. Figure 37 illustrates the timing of when two masters are arbitrating on the bus.

For high-speed mode, the arbitration cannot go into the data phase because each master is programmed with a unique high-speed master code. This 8-bit code is defined by the system designer and is set by writing to the High Speed Master Mode Code Address Register, I2C_HS_MADDR. Because the codes are unique, only one master can win arbitration, which occurs by the end of the transmission of the high-speed master code.

Control of the bus is determined by address or master code and data sent by competing masters, so there is no central master or any order of priority on the bus.

Arbitration is not allowed between the following conditions:

- A RESTART condition and a data bit
- A STOP condition and a data bit
- A RESTART condition and a STOP condition

Slaves are not involved in the arbitration process.

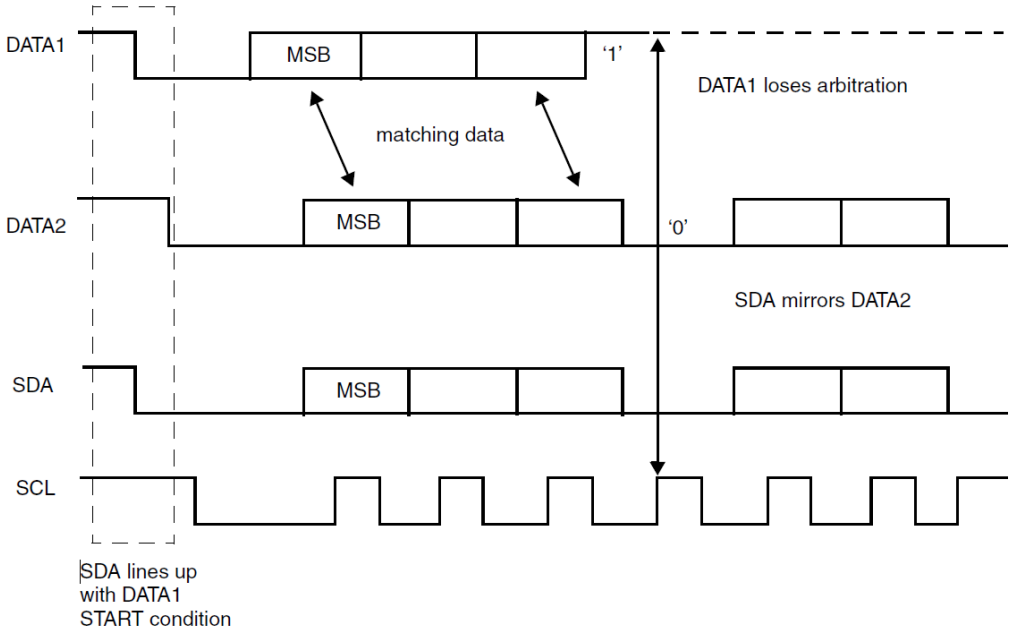


Figure 37: Multiple Master Arbitration

14.5 Clock Synchronization

When two or more masters try to transfer information on the bus at the same time, they must arbitrate and synchronize the SCL clock. All masters generate their own clock to transfer messages. Data is valid only during the high period of SCL clock. Clock synchronization is performed using the wired-AND connection to the SCL signal. When the master transitions the SCL clock to 0, the master starts counting the low time of the SCL clock and transitions the SCL clock signal to 1 at the beginning of the next clock period. However, if another master is holding the SCL line to 0, then the master goes into a HIGH wait state until the SCL clock line transitions to 1.

All masters then count off their high time and the master with the shortest high time transitions the SCL line to 0. The masters then counts out their low time and the one with the longest low time forces the other master into a HIGH wait state. Therefore, a synchronized SCL clock is generated, which is illustrated in Figure 38. Optionally, slaves may hold the SCL line low to slow down the timing on the I2C bus.

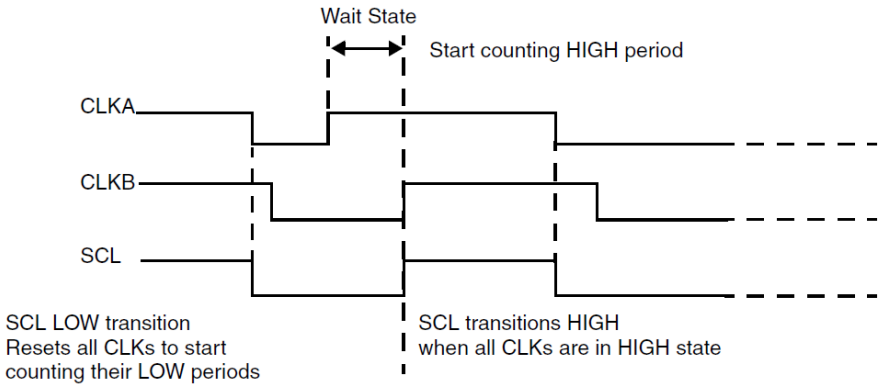


Figure 38: Multiple Master Clock Synchronization.

14.6 Operation Modes

This section provides information on the following topics:

- Slave mode operation

- Master mode operation

Note: It is important that the I2C Controller should only be set to operate as an I2C Master, or I2C Slave, but not both simultaneously. This is achieved by ensuring that bit 6 (I2C_SLAVE_DISABLE) and bit 0 (I2C_MASTER_MODE) of the I2C_CON register are never set to 0 and 1, respectively.

14.6.1 Slave Mode Operation

This section includes the following procedures:

- Initial Configuration
- Slave-Transmitter Operation for a Single Byte
- Slave-Receiver Operation for a Single Byte
- Slave-Transfer Operation For Bulk Transfers

Initial Configuration

To use the I2C Controller as a slave, perform the following steps:

1. Disable the I2C Controller by writing a ‘0’ to bit 0 of the I2C_ENABLE register.
2. Write to the I2C_SAR register (bits 9:0) to set the slave address. This is the address to which the I2C Controller responds.
3. Write to the I2C_CON register to specify which type of addressing is supported (7-bit or 10-bit by setting bit 3). Enable the I2C Controller in slave-only mode by writing a ‘0’ into bit 6 (I2C_SLAVE_DISABLE) and a ‘0’ to bit 0 (MASTER_MODE).
   Note: Slaves and masters do not have to be programmed with the same type of addressing 7-bit or 10-bit address. For instance, a slave can be programmed with 7-bit addressing and a master with 10-bit addressing, and vice versa.
4. Enable the I2C Controller by writing a ‘1’ in bit 0 of the I2C_ENABLE register.
   Note: Depending on the reset values chosen, steps 2 and 3 may not be necessary because the reset values can be configured. For instance, if the device is only going to be a master, there would be no need to set the slave address because you can configure I2C Controller to have the slave disabled after reset and to enable the master after reset. The values stored are static and do not need to be reprogrammed if the I2C Controller is disabled.

Slave-Transmitter Operation for a Single Byte

When another I2C master device on the bus addresses the I2C Controller and requests data, the I2C Controller acts as a slave-transmitter and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the slave address in the I2C_SAR register of the I2C Controller.
2. The I2C Controller acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a slave-transmitter.
3. The I2C Controller asserts the RD_REQ interrupt (bit 5 of the I2C_RAW_INTR_STAT register) and holds the SCL line low. It is in a wait state until software responds. If the RD_REQ interrupt has been masked, due to I2C_INTR_MASK[5] register (M_RD_REQ bit field) being set to 0, then it is recommended that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the I2C_RAW_INTR_STAT register.
   a. Reads that indicate I2C_RAW_INTR_STAT[5] (R_RD_REQ bit field) being set to 1 must be treated as the equivalent of the RD_REQ interrupt being asserted.
   b. Software must then act to satisfy the I2C transfer.
   c. The timing interval used should be in the order of 10 times the fastest SCL clock period the I2C Controller can handle. For example, for 400 kb/s, the timing interval is 25us.
      Note: The value of 10 is recommended here because this is approximately the amount of time required for a single byte of data transferred on the I2C bus.
4. If there is any data remaining in the TX FIFO before receiving the read request, then the I2C Controller asserts a TX_ABRT interrupt (bit 6 of the I2C_RAW_INTR_STAT register) to flush the old data from the TX FIFO.
   Note: Because the I2C Controller’s TX FIFO is forced into a flushed/reset state whenever a

TX_ABRT event occurs, it is necessary for software to release the I2C Controller from this state by reading the I2C_CLR_TX_ABRT register before attempting to write into the TX FIFO. See register I2C_RAW_INTR_STAT for more details.

5. If the TX_ABRT interrupt has been masked, due to of I2C_INTR_MASK[6] register (M_TX_ABRT bit field) being set to 0, then it is recommended that re-using the timing routine (described in the previous step), or a similar one, be used to read the I2C_RAW_INTR_STAT register.
   a. Reads that indicate bit 6 (R_TX_ABRT) being set to 1 must be treated as the equivalent of the TX_ABRT interrupt being asserted.
   b. There is no further action required from software.
   c. The timing interval used should be similar to that described in the previous step for the I2C_RAW_INTR_STAT[5] register.
6. Software writes to the I2C_DATA_CMD register with the data to be written (by writing a '0' in bit 8).
7. Software must clear the RD_REQ and TX_ABRT interrupts (bits 5 and 6, respectively) of the I2C_RAW_INTR_STAT register before proceeding.
8. If the RD_REQ and/or TX_ABRT interrupts have been masked, then clearing of the I2C_RAW_INTR_STAT register will have already been performed when either the R_RD_REQ or R_TX_ABRT bit has been read as 1.
9. The I2C Controller releases the SCL and transmits the byte.
10. The master may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

Slave-Receiver Operation for a Single Byte

When another I2C master device on the bus addresses the I2C Controller and is sending data, the I2C Controller acts as a slave-receiver and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the I2C Controller's slave address in the I2C_SAR register.
2. The I2C Controller acknowledges the sent address and recognizes the direction of the transfer to indicate that the I2C Controller is acting as a slave-receiver.
3. I2C Controller receives the transmitted byte and places it in the receive buffer.
4. If the RX FIFO is completely filled with data when a byte is pushed, then an overflow occurs and the I2C Controller continues with subsequent I2C transfers. Because a NACK is not generated, software must recognize the overflow when indicated by the I2C Controller (by the R_RX_OVER bit in the I2C_INTR_STAT register) and take appropriate actions to recover from lost data. Hence, there is a real time constraint on software to service the RX FIFO before the latter overflow as there is no way to re-apply pressure to the remote transmitting master. You must select a deep enough RX FIFO depth to satisfy the interrupt service interval of their system.
5. I2C Controller asserts the RX_FULL interrupt (I2C_RAW_INTR_STAT[2] register).
6. If the RX_FULL interrupt has been masked, due to setting I2C_INTR_MASK[2] register to 0 or setting I2C_TX_TL to a value larger than 0, then it is recommended that a timing routine (described in "Slave-Transmitter Operation for a Single Byte") be implemented for periodic reads of the I2C_STATUS register. Reads of the I2C_STATUS register, with bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX_FULL interrupt being asserted.
7. Software may read the byte from the I2C_DATA_CMD register (bits 7:0).
8. The other master device may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

Slave-Transfer Operation for Bulk Transfers

In the standard I2C protocol, all transactions are single byte transactions and the programmer responds to a remote master read request by writing one byte into the slave's TX FIFO. When a slave (slave-transmitter) is issued with a read request (RD_REQ) from the remote master (master-receiver), at a minimum there should be at least one entry placed into the slave-transmitter's TX FIFO.

The I2C Controller is designed to handle more data in the TX FIFO so that subsequent read requests can take that data without raising an interrupt to get more data. Ultimately, this eliminates the possibility of significant latencies being incurred between raising the interrupt for data each time had there been a restriction of having only one entry placed in the TX FIFO.

This mode only occurs when I2C Controller is acting as a slave-transmitter. If the remote master acknowledges the data sent by the slave-transmitter and there is no data in the slave's TX FIFO, the I2C Controller holds the I2C SCL line low while it raises the read request interrupt (RD_REQ) and waits for data to be written into the TX FIFO before it can be sent to the remote master.

If the RD_REQ interrupt is masked, due to bit 5 (M_RD_REQ) of the I2C_INTR_STAT register being set to 0, then it is recommended that a timing routine be used to activate periodic reads of the I2C_RAW_INTR_STAT register. Reads of I2C_RAW_INTR_STAT that return bit 5 (R_RD_REQ) set to 1 must be treated as the equivalent of the RD_REQ interrupt referred to in this section. This timing routine is similar to that described in "Slave-Transmitter Operation for a Single Byte"

The RD_REQ interrupt is raised upon a read request, and like interrupts, must be cleared when exiting the interrupt service handling routine (ISR). The ISR allows you to either write 1 byte or more than 1 byte into the TX FIFO. During the transmission of these bytes to the master, if the master acknowledges the last byte, then the slave must raise the RD_REQ again because the master is requesting for more data.

If the programmer knows in advance that the remote master is requesting a packet of n bytes, then when another master addresses I2C Controller and requests data, the TX FIFO could be written with n number bytes and the remote master receives it as a continuous stream of data. For example, the I2C Controller slave continues to send data to the remote master as long as the remote master is acknowledging the data sent and there is data available in the TX FIFO. There is no need to hold the SCL line low or to issue RD_REQ again.

If the remote master is to receive n bytes from the I2C Controller but the programmer wrote a number of bytes larger than n to the TX FIFO, then when the slave finishes sending the requested n bytes, it clears the TX FIFO and ignores any excess bytes.

The I2C Controller generates a transmit abort (TX_ABRT) event to indicate the clearing of the TX FIFO in this example. At the time an ACK/NACK is expected, if a NACK is received, then the remote master has all the data it wants. At this time, a flag is raised within the slave's state machine to clear the leftover data in the TX FIFO. This flag is transferred to the processor bus clock domain where the FIFO exists and the contents of the TX FIFO is cleared at that time.

14.6.2 Master Mode Operation

This section includes the following topics:

- Initial Configuration
- Master Transmit and Master Receive

Initial Configuration

The procedures are very similar and are only different with regard to where the I2C_10BITADDR_MASTER bit is set (either bit 4 of I2C_CON register or bit 12 of I2C_TAR register).

To use the I2C Controller as a master perform the following steps:

1. Disable the I2C Controller by writing 0 to the I2C_ENABLE register.
2. Write to the I2C_CON register to set the maximum speed mode supported (bits 2:1) and the desired speed of the I2C Controller master-initiated transfers, either 7-bit or 10-bit addressing (bit 4).
3. Ensure that bit 6 I2C_SLAVE_DISABLE = 1 and bit 0 MASTER_MODE =1
   Note: Slaves and masters do not have to be programmed with the same type of addressing 7-bit or 10-bit address. For instance, a slave can be programmed with 7-bit addressing and a master with 10-bit addressing, and vice versa.
4. Write to the I2C_TAR register the address of the I2C device to be addressed (bits 9:0). This register also indicates whether a General Call or a START BYTE command is going to be performed by I2C.

5. Only applicable for high-speed mode transfers. Write to the I2C_HS_MADDR register the desired master code for the I2C Controller. The master code is programmer-defined.
6. Enable the I2C Controller by writing a 1 in bit 0 of the I2C_ENABLE register.
7. Now write transfer direction and data to be sent to the I2C_DATA_CMD register. If the I2C_DATA_CMD register is written before the I2C Controller is enabled, the data and commands are lost as the buffers are kept cleared when I2C Controller is disabled.
8. This step generates the START condition and the address byte on the I2C Controller. Once I2C Controller is enabled and there is data in the TX FIFO, I2C Controller starts reading the data.

Note: Depending on the reset values chosen, steps 2, 3, 4, and 5 may not be necessary because the reset values can be configured. The values stored are static and do not need to be reprogrammed if the I2C Controller is disabled, with the exception of the transfer direction and data.

Master Transmit and Master Receive

The I2C Controller supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the I2C_DATA_CMD_REG register. The CMD bit [8] should be written to 0 for I2C write operations.

Subsequently, a read command may be issued by writing "don't cares" to the lower byte of the I2C_DATA_CMD_REG register, and a 1 should be written to the CMD bit.

The I2C Controller master continues to initiate transfers as long as there are commands present in the transmit FIFO. If the transmit FIFO becomes empty, the master inserts a STOP condition after completing the current transfer.

14.7 Disabling the I2C Controller

The register I2C_ENABLE_STATUS is added to allow software to unambiguously determine when the hardware has completely shut down in response to the I2C_ENABLE register being set from 1 to 0. Only one register is required to be monitored.

Procedure

1. Define a timer interval (ti2c_poll) equal to the 10 times the signalling period for the highest I2C transfer speed used in the system and supported by I2C Controller. For example, if the highest I2C transfer mode is 400 kbit/s, then this ti2c_poll is 25 µs.
2. Define a maximum time-out parameter, MAX_T_POLL_COUNT, such that if any repeated polling operation exceeds this maximum value, an error is reported.
3. Execute a blocking thread/process/function that prevents any further I2C master transactions to be started by software, but allows any pending transfers to be completed.
   Note: This step can be ignored if I2C Controller is programmed to operate as an I2C slave only.
4. The variable POLL_COUNT is initialized to zero.
5. Set I2C_ENABLE to 0.
6. Read the I2C_ENABLE_STATUS register and test the I2C_EN bit (bit 0). Increment POLL_COUNT by one. If POLL_COUNT >= MAX_T_POLL_COUNT, exit with the relevant error code.
7. If I2C_ENABLE_STATUS[0] is 1, then sleep for ti2c_poll and proceed to the previous step.

Otherwise, exit with a relevant success code.

15 UART

The DA14585 contains two identical instances of this block, i.e. UART and UART2.

The UART is compliant to the industry-standard 16550 and is used for serial communication with a peripheral. 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 also DMA support on the UART block thus the internal FIFOs can be used. Both UARTs support hardware flow control signals (RTS, CTS).

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)/(16 * divisor).

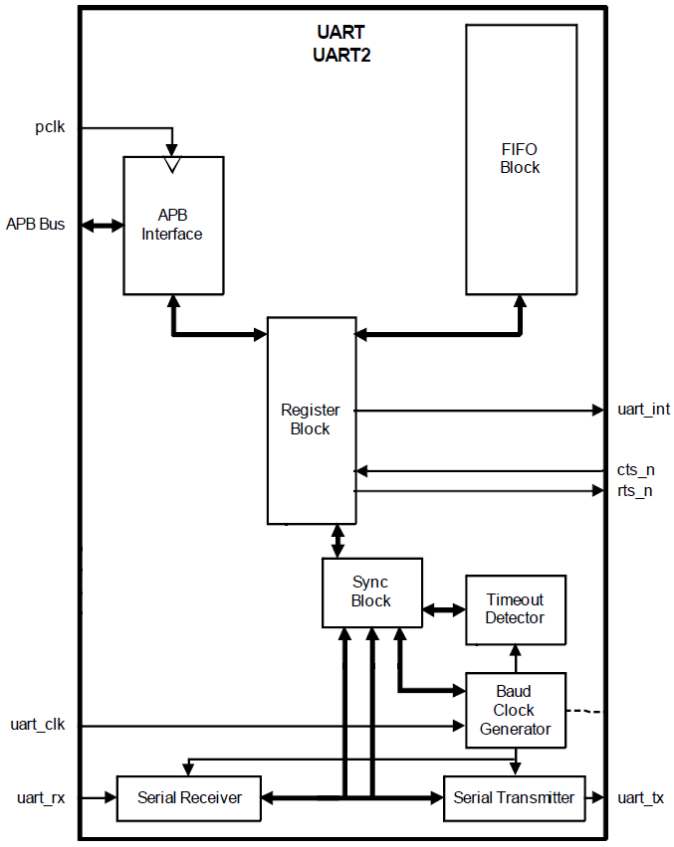


Figure 39: UART Block Diagram

15.1 UART (RS232) Serial Protocol

Because the serial communication between the UART and the selected device is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. Utilizing these bits allows two devices to be synchronized. This structure of serial data accompanied by start and stop bits is referred to as a character, as shown in Figure 40.

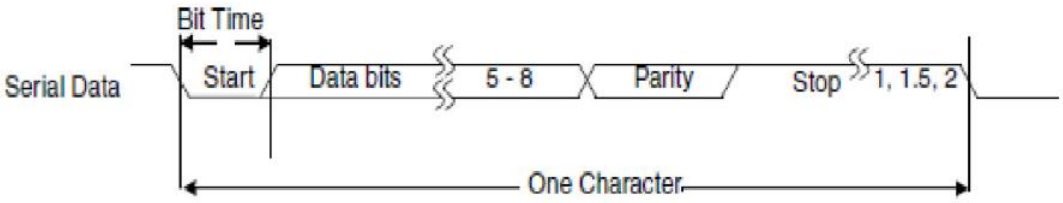


Figure 40: Serial Data Format

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

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

All the bits in the transmission (with exception to the half stop bit when 1.5 stop bits are used) are transmitted for exactly the same time duration. This is referred to as a Bit Period or Bit Time. One BitTime equals 16 baud clocks. To ensure stability on the line the receiver samples the serial input data at approximately the mid-point of the Bit Time once the start bit has been detected. As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid-point sample of the start bit. Figure 41 shows the sampling points of the first couple of bits in a serial character.

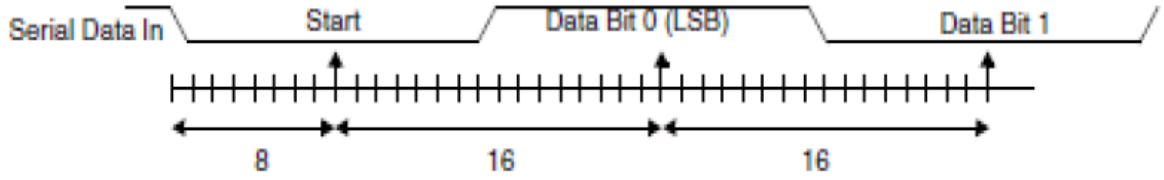


Figure 41: Receiver Serial Data Sampling Points

As part of the 16550 standard an optional baud clock reference output signal (baudout_n) is supplied to provide timing information to receiving devices that require it. The baud rate of the UART is controlled by the serial clock (*sclk* or *pclk* in a single clock implementation) and the Divisor Latch Register (DLH and DLL). The registers settings for common baud rate values are presented in the following table:

Table 35: UART Baud Rate Generation

Baud Rate	Divider	Divisor Latch	DLH Reg	DLL Reg	DLF Reg	Actual BR	Error %
1200	833,333	833,3125	3	65	5	1200,03	0,00
2400	416,667	416,6875	1	160	11	2399,88	0,00
4800	208,333	208,3125	0	208	5	4800,48	0,01
9600	104,167	104,1875	0	104	3	9598,08	0,02
19200	52,083	52,0625	0	52	1	19207,68	0,04
38400	26,042	26,0625	0	26	1	38369,30	0,08

Baud Rate	Divider	Divisor Latch	DLH Reg	DLL Reg	DLF Reg	Actual BR	Error %
57600	17,361	17,375	0	17	6	57553,96	0,08
115200	8,681	8,6875	0	8	11	115107,91	0,08
230400	4,340	4,3125	0	4	5	231884,06	0,64
460800	2,170	2,1875	0	2	3	457142,86	0,79
921600	1,085	1,0625	0	1	1	941176,47	2,12
1000000	1	1	0	1	0	1000000	0,00

15.2 IrDA 1.0 SIR Protocol

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

NOTE

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

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

Trying to adjust the number of data bits sent or enable parity with the Line Control Register (LCR) has no effect. When the UART is configured to support IrDA 1.0 SIR it can be enabled with Mode Control Register (MCR) bit 6. When the UART is not configured to support IrDA SIR mode, none of the logic is implemented and the mode cannot be activated, reducing total gate counts. When SIR mode is enabled and active, serial data is transmitted and received on the *sir_out_n* and *sir_in* ports, respectively.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16th of a normal serial bit time. Thus, each new character begins with an infrared pulse for the start bit. However, received data is inverted from transmitted data due to the infrared pulses energizing the photo transistor base of the IrDA receiver, pulling its output low. This inverted transistor output is then fed to the UART *sir_in* port, which then has correct UART polarity. Figure 42 shows the timing diagram for the IrDA SIR data format in comparison to the standard serial format.

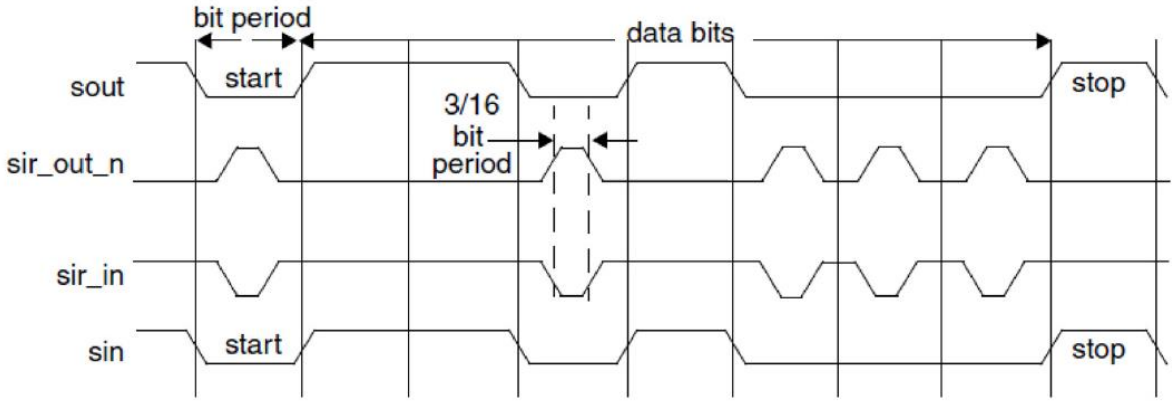


Figure 42: IrDA SIR Data Format

As detailed in the IrDA 1.0 SIR, the UART can be configured to support a low-power reception mode. When the UART is configured in this mode, the reception of SIR pulses of 1.41 µs (minimum pulse duration) is possible, as well as nominal 3/16th of a normal serial bit time. Using this low-power reception mode requires programming the Low Power Divisor Latch (LPDLL/LPDLH) registers.

It should be noted that for all *sclk* frequencies greater than or equal to 7.37 MHz (and obey the requirements of the Low Power Divisor Latch registers), pulses of 1.41 µs are detectable. However there are several values of *sclk* that do not allow the detection of such a narrow pulse and these are as follows:

Table 36: Low Power Divisor Latch Register Values

SCLK	Low Power Divisor Latch Register Value	Min Pulse Width for Detection
1.84 MHz	1	3.77 µs
3.69 MHz	2	2.086 µs
5.33 MHz	3	1.584 µs

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

15.3 Clock Support

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

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

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

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

15.4 Interrupts

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

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

When an interrupt occurs the master accesses the UART_IIR_REG to determine the source of the interrupt before dealing with it accordingly. These interrupt types are described in more detail in Table 37.

Table 37: UART Interrupt Priorities

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

15.5 Programmable THRE Interrupt

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

When Programmable THRE Interrupt mode is selected it can be enabled via the Interrupt Enable Register (IER[7]). When FIFOs and the THRE Mode are implemented and enabled, THRE Interrupts are active at, and below, a programmed transmitter FIFO empty threshold level, as opposed to empty, as shown in the flowchart in Figure 43.

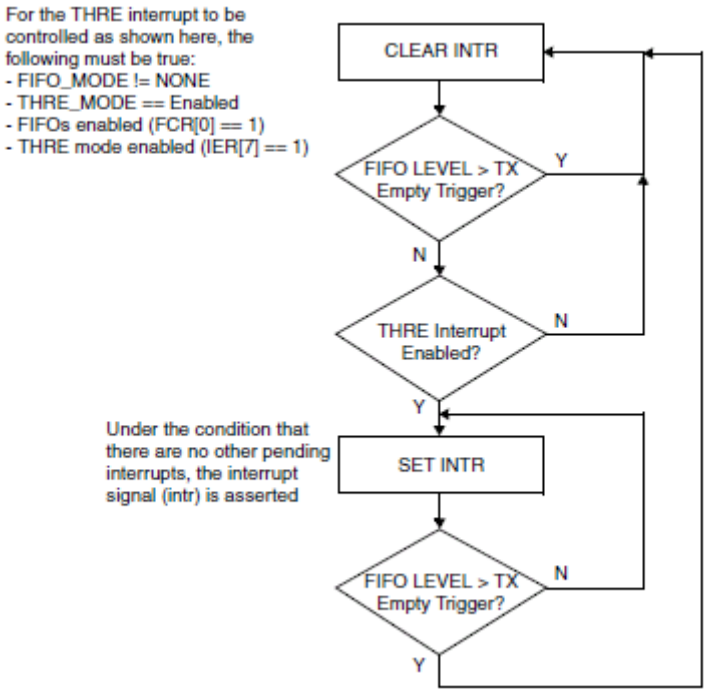


Figure 43: Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode

This threshold level is programmed into FCR[5:4]. The available empty thresholds are: empty, 2, ¼ and ½. See UART_FCR_REG for threshold setting details. Selection of the best threshold value depends on the system's ability to begin a new transmission sequence in a timely manner. However, one of these thresholds should prove optimum in increasing system performance by preventing the transmitter FIFO from running empty.

In addition to the interrupt change, Line Status Register (LSR[5]) also switches function from indicating transmitter FIFO empty, to FIFO full. This allows software to fill the FIFO each transmit sequence by polling LSR[5] before writing another character. Instead of waiting until the FIFO is completely empty, the flow becomes, “fill transmitter FIFO whenever an interrupt occurs and there is data to transmit”. Waiting until the FIFO is empty causes a performance hit whenever the system is too busy to respond immediately.

Even if everything else is selected and enabled, if the FIFOs are disabled via FCR[0], the Programmable THRE Interrupt mode is also disabled. When not selected or disabled, THRE interrupts and LSR[5] function normally (both reflecting an empty THR or FIFO). The flowchart of THRE interrupt generation when not in programmable THRE interrupt mode is shown in Figure 44.

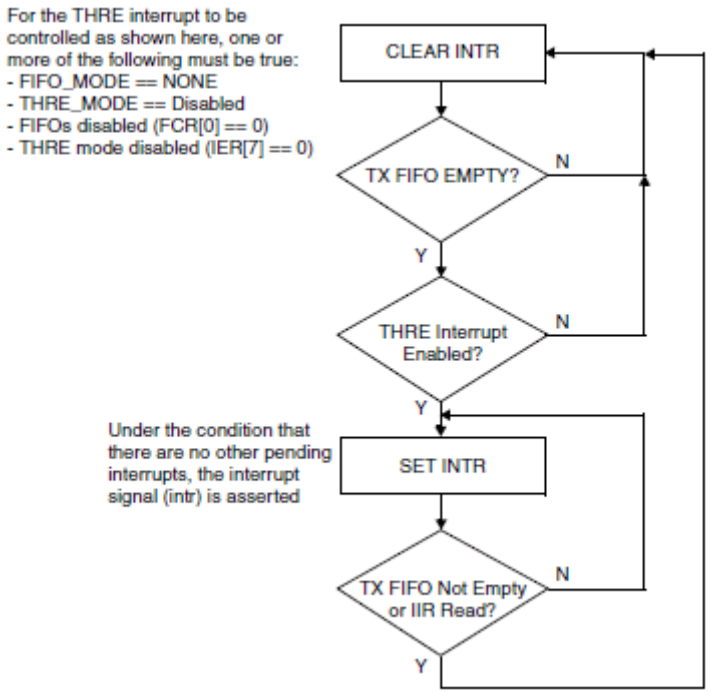


Figure 44: Flowchart of Interrupt Generation When Not in Programmable THRE Interrupt Mode

15.6 Shadow Registers

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

- UART_SRBR_REG support a host burst mode where the host increments its address but still accesses the same Receive buffer register.
- UART_STHR support a host burst mode where the host increments its address but still accesses the same transmit holding register.
- UART_SFE_REG accesses the FCR[0] register without accessing the other UART_FCR_REG bits.
- UART_SRT_REG accesses the FCR[7-6] register without accessing the other UART_FCR_REG bits.
- UART_STER_REG accesses the FCR[5-4] register without accessing the other UART_FCR_REG bits.

15.7 Direct Test Mode

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

16 SPI+ Interface

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

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.
- DMA support

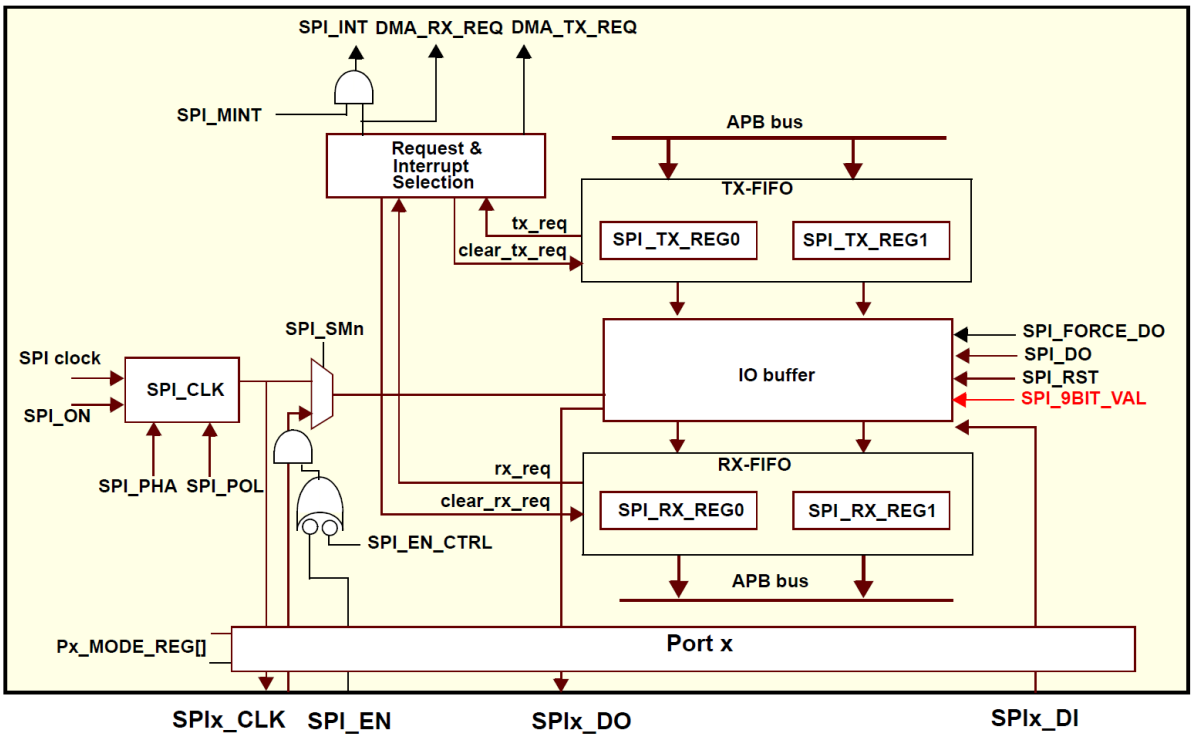


Figure 45: SPI Block Diagram

16.1 Operation without FIFOs

This mode is the default mode.

Master Mode

To enable SPITM operation, first the individual port signal must be enabled. Next the SPI must be configured in SPI_CTRL_REG, for the desired mode. Finally bit SPI_ON must be set to 1.

A SPI transfer cycle starts after writing to the SPI_RX_TX_REG0. In case of 32 bits mode, the SPI_RX_TX_REG1 must be written first. Writing to SPI_RX_TX_REG0 also sets the SPI_TXH. As soon as the holding register is copied to the IO buffer, the SPI_TXH is reset and a serial transfer cycle of 8/9/16/32 clock-cycles is started which causes 8/9/16/32 bits to be transmitted on SPI_DO. Simultaneously, data is received on SPI_DI and shifted into the IO buffer. The transfer cycle finishes after the 8th/9th/16th/32nd clock cycle and SPI_INT_BIT bit is set in the SPI_CTRL_REG and SPI_INT_PEND bit in (RE)SET_INT_PENDING_REG is set. The received bits in the IO buffer are copied to the SPI_RX_TX_REG0 (and SPI_RX_TX_REG1 in case of 32 bits mode) were they can be read by the CPU.

Interrupts to the CPU can be disabled using the SPI_MINT bit. To clear the SPI interrupt source, any value to SPI_CLEAR_INT_REG must be written. Note however that SPI_INT will be set as long as the RX-FIFO contains unread data.

Slave Mode

The slave mode is selected with SPI_SMn set to 1 and the Px_MODE_REG must also select SPI_CLK as input. The functionality of the IO buffer in slave and master mode is identical. The SPI module clocks data in on SPI_DI and out on SPI_DO on every active edge of SPI_CLK. As shown in Figure 46 to Figure 49Figure 49: SPI Master/slave, Mode 3: SPI_POL=1 and SPI_PHA=1. The SPI has an active low clock enable SPI_EN, which can be enabled with bit SPI_EN_CTRL=1.

In slave mode the internal SPI clock must be more than four times the SPI_CLK

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

SPI_POL and SPI_PHA

The phase and polarity of the serial clock can be changed with bits SPI_POL and SPI_PHA in the SPI_CTRL_REG.

SPI_DO Idle Levels

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

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

In SPI modes 1 and 3 however there is no clock edge after the sampled LSB and pin SPI_DO gets the LSB value of the IO buffer. If required, the SPI_DO can be forced to the SPI_DO bit level by resetting the SPI to the idle state by shortly setting bit SPI_RST to 1. (Optionally SPI_FORCE_DO can be set, but this does not reset the IO buffer). The following diagrams show the timing of the SPI™ interface.

Writes Only Mode

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

SPI_INT acts as a tx_request signal, indicating that there is still place in the FIFO. It will be ‘0’ when the FIFO is full or else ‘1’ when it’s not full. This is also indicated in the SPI_CTRL_REG[SPI_TXH], which is ‘1’ if the TX-FIFO is full. Writing to the FIFO if this bit is still 1, will result in transmission of undefined data. If all data has been transferred, SPI_CTRL_REG1 [SPI_BUSY] will become ‘0’.

Reads Only Mode

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

In Slave mode transfers only take place if the external master initiates them, but in master mode this means that transfers will continue until the RX-FIFO is full. If this happens SPI_CTRL_REG1[SPI_BUSY] will become ‘0’. If exactly N words need to be read from SPI device, first read (N - fifosize+1) words. Then wait until the SPI_BUSY becomes ‘0’, set SPI_FIFO_MODE to “00” and finally read the remaining (fifosize +1) words. Here fifosize is 4/2/1 words for 8/16/32 bits mode respectively.

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

Bidirectional transfers with FIFO

If SPI_FIFO_MODE is “00“, both registers are used as a FIFO. SPI_TXH indicates that TX-FIFO is full, SPI_INT indicates that there is data in the RX-FIFO.

DMA Operation Requirements

If the DMA operation is needed when the TX-FIFO is disabled (SPI_CTRL_REG1[SPI_FIFO_MODE] = 0x3), the CPU must write one byte to the SPI_RX_TX_REG0 register (and SPI_RX_TX_REG1 in 16bit mode) before the DMA takes control of the data transfer.

16.2 9 Bits Mode

The 9 bits mode can be used to support 9 bits displays and is selected with SPI_CTRL_REG[SPI_WORD] set to ‘11’. The value of the 9th bit, set in the SPI_CTRL_REG1[SPI_9BIT_VAL] and is used to determine if the next 8 bits form a command word or data word. Because the 9th bit is not part of the data, the FIFOs are still used in the 8 bits mode. The 9th bit is received but not saved because it is shifted out of the 8 bits shift register upon reception.

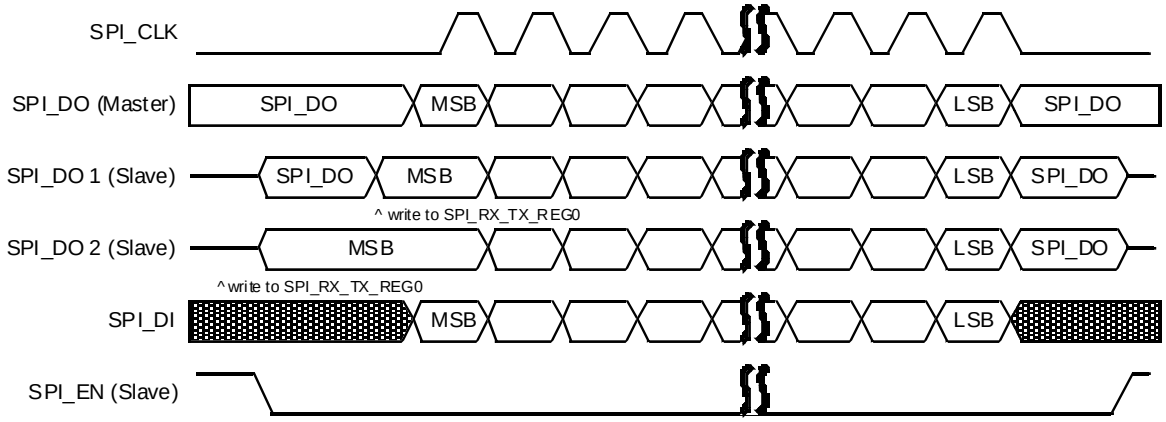


Figure 46: SPI Master/Slave, Mode 0: SPI_POL=0 and SPI_PHA=0

Note 1 If 9 bits SPI mode, the MSB bit in transmit direction is determined by bit SPI_CTRL_REG[SPI_9BIT_VAL]. In receive direction, the MSB is received but not stored.

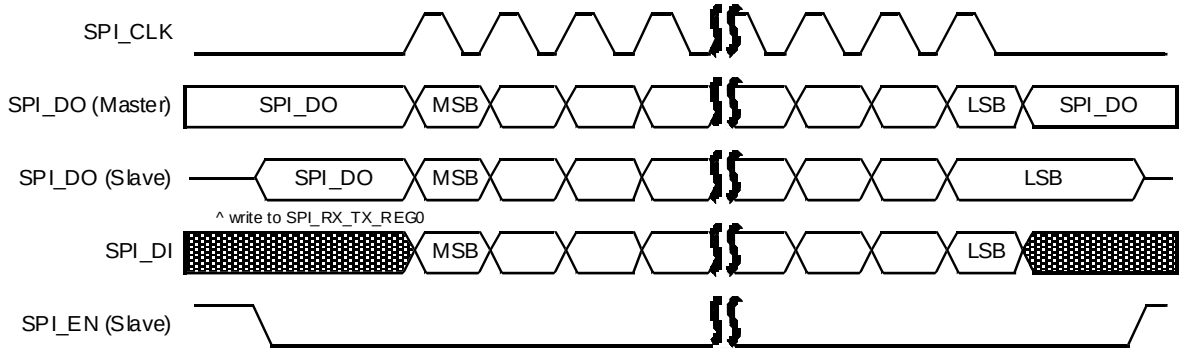


Figure 47: SPI Master/Slave, Mode 1: SPI_POL=0 and SPI_PHA=1

For the MSB bit refer to Note 1.

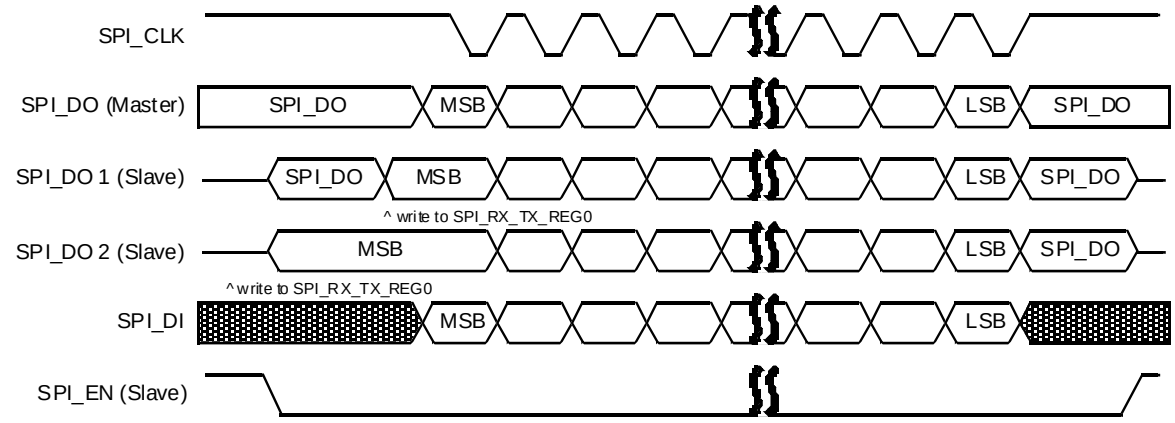


Figure 48: SPI Master/Slave, Mode 2: SPI_POL=1 and SPI_PHA=0

For the MSB bit refer to Note 1.

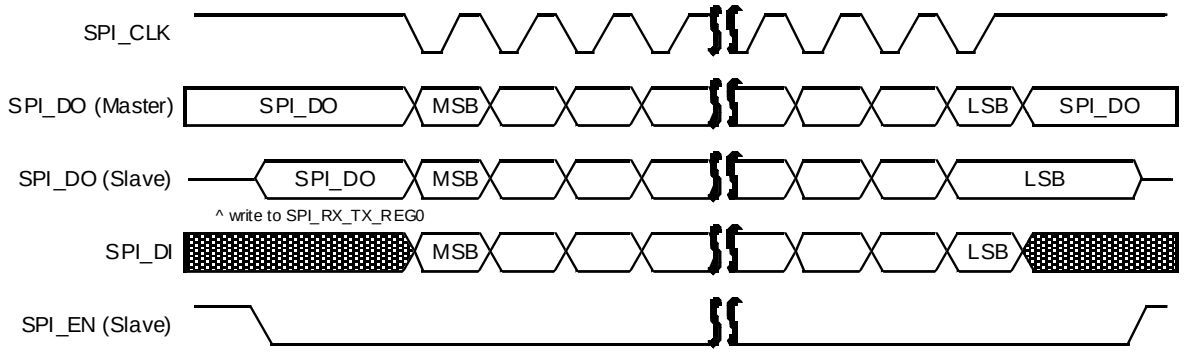


Figure 49: SPI Master/slave, Mode 3: SPI_POL=1 and SPI_PHA=1

For the MSB bit refer to Note 1.

16.3 SPI Timing

The timing of the SPI interface when the SPI controller is in Slave mode is presented in Figure 50.

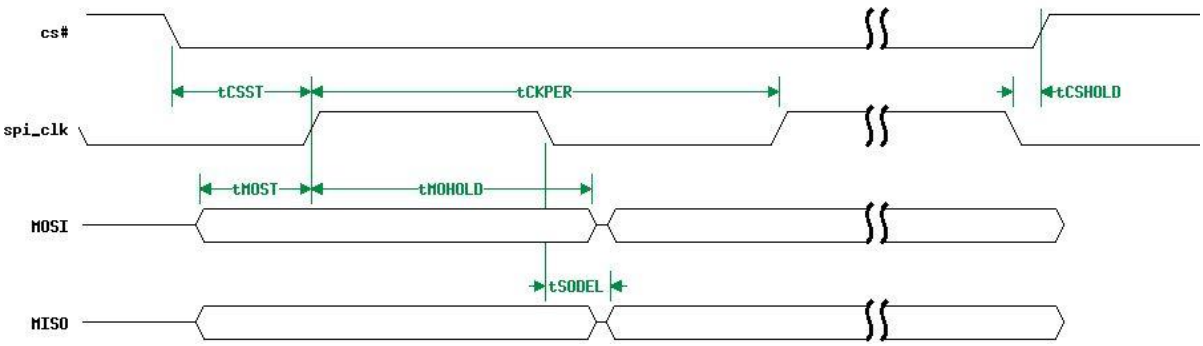


Figure 50: SPI Slave Mode Timing (CPOL = 0, CPHA = 0)

Table 38: SPI Timing Parameters

Parameter	Description	Min	Typ	Max	Unit
t_{CKPER}	spi_clk clock period			0.25 * T_{SPI_CLK}	ns
t_{CSST}	CS active time before rising edge of spi_clk	10.6 +T_{INT} (Note 2)	5.2 + T_{INT} (Note 2)	3.1 + T_{INT} (Note 2)	ns
t_{CSHOLD}	CS active time after falling edge of spi_clk	0	0	0	ns
t_{MOST}	Input data latching setup time	2.7	1.5	0.9	ns
t_{MOHOLD}	Input data hold time	0	0	0	ns
t_{SODEL}	Output data hold time	17.2	8.6	5.5	ns

Note 2 T_{INT} represents the internal SPI clock period and is equal to 1.5 * spi_clk period.

17 Quadrature Decoder

The DA14585 has an integrated Quadrature decoder that can automatically decode the signals for the X, Y and Z axes of a HID input device, reporting step count and direction. This block can be used for waking up the chip as soon as there is any kind of movement from the external device connected to it. The block diagram of the quadrature decoder is presented in Figure 51.

Features

- Three 16-bit signed counters that provide the step count and direction on each of the axes (X, Y and Z)
- Programmable system clock sampling at maximum 16 MHz.
- APB interface for control and programming
- Programmable source from P0, P1 and P2 ports
- Digital filter on the channel inputs to avoid spikes

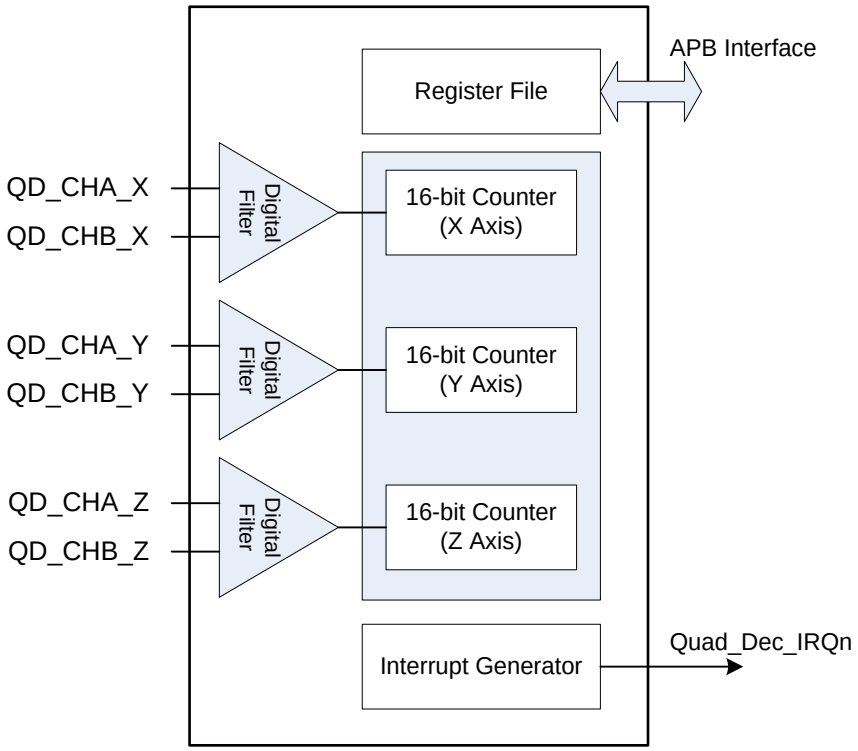


Figure 51: Quadrature Decoder Block Diagram

Channels are expected to provide a pulse train with 90 degrees rotation as displayed in the following figures:

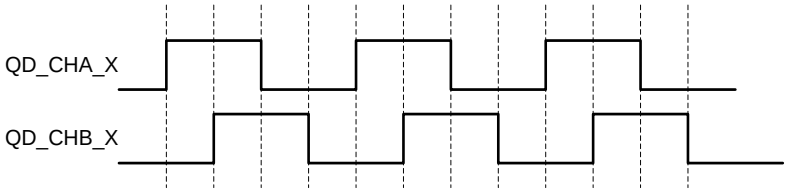


Figure 52: Moving Forward on Axis X

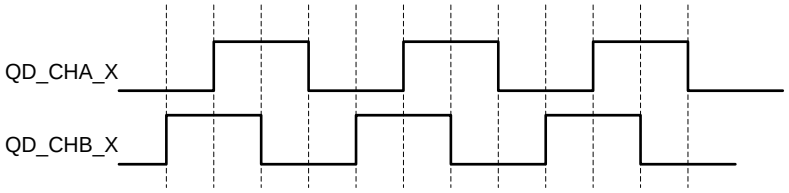


Figure 53: Moving Backwards on Axis X

Depending on whether channel A or channel B is leading in phase, the quadrature decoding block calculates the direction on the related axis. Furthermore, the signed counter value represents the number of steps moved.

Since six channels are required (two for each axis), all P0 and P1 signals and some of the P2 port can be mapped onto this block. The user can choose which GPIOs to use for the channels by programming the QDEC_CTRL2_REG register.

The digital filter eliminates any spike shorter than two clock periods. The counter holds the movement events of the channel. When a channel is disabled the counter is reset. The counters are accessible via the APB bus.

The quadrature decoder operates on the system clock. The QDEC_CLOCKDIV register defines the number of clock cycles of the period at which the decoding logic samples the data on the channel inputs.

18 Wake-Up Timer

The Wake-up timer can be programmed to wake up the DA14585 from power down mode after a pre-programmed number of GPIO events.

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

The input signal edge can be selected by programming the WKUP_POL_Px_REG register.

The block diagram illustrating the Wake-up function is shown in Figure 54.

Features

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

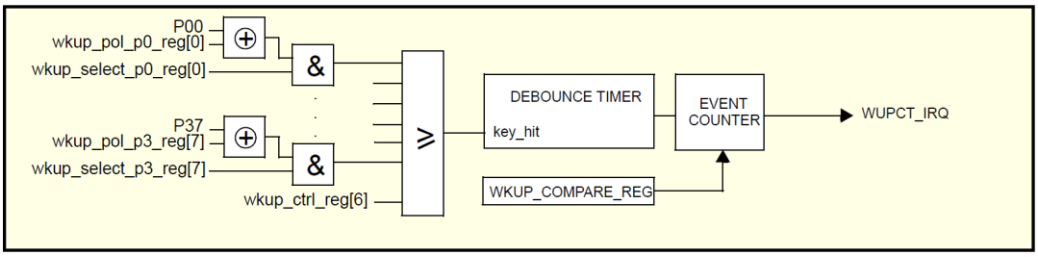


Figure 54: Wake-Up Timer Block Diagram

A LOW to HIGH level transition on the selected input port, while WKUP_POL_Px_REG[y] = 0, sets internal signal “key_hit” to ‘1’. This signal triggers the event counter state machine as shown in Figure 55. The debounce timer is loaded with value WKUP_CTRL_REG[WKUP_DEB_VALUE]. The timer counts down every 1 ms. If the timer reaches 0 and the “key_hit” signal is still ‘1’, the event counter will be incremented.

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

If the event counter is equal to the value set in the WKUP_COMPARE_REG register, the counter will be reset and an interrupt will be generated, if it was enabled by WKUP_CTRL_REG[ENABLE_IRQ].

The interrupt can be cleared by writing any value to register WKUP_RESET_IRQ_REG.
The event counter can be reset by writing any value to register WKUP_RESET_CNTR_REG.
The value of the event counter can be read at any time by reading register WKUP_COUNTER_REG.

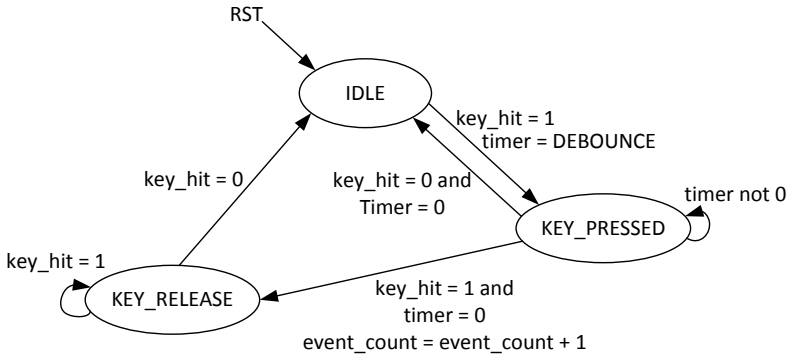


Figure 55: Event Counter State Machine for the Wake-Up Interrupt Generator

19 General Purpose Timers

The Timer block contains two timer modules that are software controlled, programmable and can be used for various tasks. Timer 0 is a 16-bit general purpose timer with a PWM output capability. Timer 2 is a 14-bit counter that generates three identical PWM signals in a quite flexible manner.

19.1 Timer 0

Timer 0 is a 16-bit general purpose software programmable timer, which has the ability of generating Pulse Width Modulated signals, namely PWM0 and PWM1. It also generates the SWTIM_IRQ interrupt to the ARM Cortex-M0. It can be configured in various modes regarding output frequency, duty cycle and the modulation of the PWM signals.

Features

- 16-bit general purpose timer
- Ability to generate 2 Pulse Width Modulated signals (PWM0 and PWM1)
- Programmable output frequency: $f = \frac{(16,\ 8,\ 4,\ 2\ MHz\ or\ 32\ kHz)}{(M+1)+(N+1)}$ with N = 0 to (2^{16}-1), M = 0 to (2^{16}-1)
- Programmable duty cycle: $\delta = \frac{M+1}{(M+1)+(N+1)} \times 100\ \%$
- Separately programmable interrupt timer: $T = \frac{(16,\ 8,\ 4,\ 2\ MHz\ or\ 32\ kHz)}{(ON+1)}$

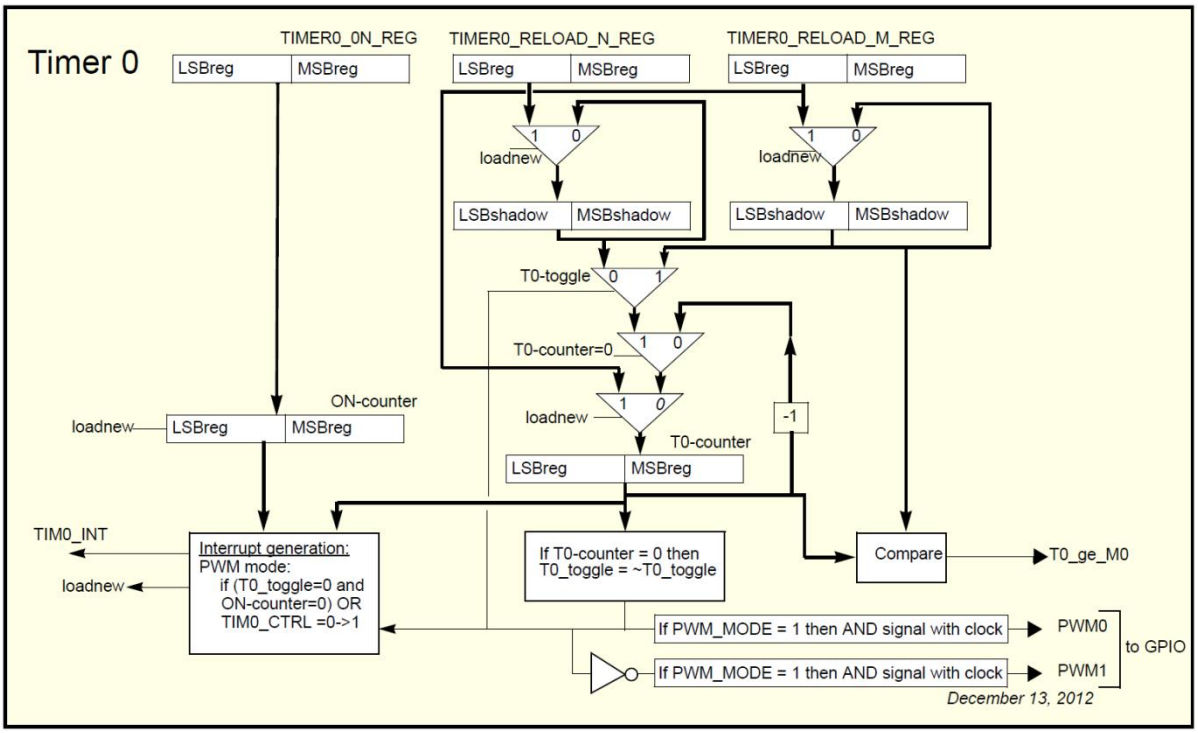


Figure 56: Timer 0 Block Diagram

Figure 56 shows the block diagram of Timer 0. The 16 bits timer consists of two counters: T0-counter and ON-counter, and three registers: TIMER0_RELOAD_M_REG, TIMER0_RELOAD_N_REG and TIMER0_ON_REG. Upon reset, the counter and register values are 0x0000. Timer 0 will generate a Pulse Width Modulated signal PWM0. The frequency and duty cycle of PWM0 are determined by the contents of the TIMER0_RELOAD_N_REG and the TIMER0_RELOAD_M_REG registers.

The timer can run at five different clocks: 16 MHz, 8 MHz, 4 MHz, 2 MHz or 32 kHz. The 32 kHz clock is selected by default with bit TIM0_CLK_SEL in the TIMER0_CTRL_REG register. This ‘slow’

clock has no enabling bit. The other four options can be selected by setting the TIM0_CLK_SEL bit and the TMR_ENABLE bit in the CLK_PER_REG (default disabled). This register also controls the frequency via the TMR_DIV bits. An extra clock divider is available that can be activated via bit TIM0_CLK_DIV of the timer control register TIMER0_CTRL_REG. This clock divider is only used for the ON-counter and always divides by 10.

Timer 0 operates in PWM mode. The signals PWM0 and PWM1 can be mapped to any GPIOs.

Timer 0 PWM Mode

If bit TIM0_CTRL in the TIMER0_CTRL_REG is set, Timer 0 will start running. SWTIM_IRQ will be generated and the T0-counter will load its start value from the TIMER0_RELOAD_M_REG register, and will decrement on each clock. The ON-counter also loads its start value from the TIMER0_ON_REG register and decrements with the selected clock.

When the T0-counter reaches zero, the internal signal T0-toggle will be toggled to select the TIMER0_RELOAD_N_REG whose value will be loaded in the T0-counter. Each time the T0-counter reaches zero it will alternately be reloaded with the values of the M0- and N0-shadow registers respectively. PWM0 will be high when the M0-value decrements and low when the N0-value decrements. For PWM1 the opposite is applicable since it is inverted. If bit PWM_MODE in the TIMER0_CTRL_REG register is set, the PWM signals are not HIGH during the ‘high time’ but output a clock in that stage. The frequency is based on the clock settings defined in the CLK_PER_REG register (also in 32 kHz mode), but the selected clock frequency is divided by two to get a 50 % duty cycle.

If the ON-counter reaches zero it will remain zero until the T0-counter also reaches zero, while decrementing the value loaded from the TIMER0_RELOAD_N_REGregister (PWM0 is low). The counter will then generate an interrupt (SWTIM_IRQ). The ON-counter will be reloaded with the value of the TIMER0_ON_REG register. The T0-counter as well as the M0-shadow register will be loaded with the value of the TIMER0_RELOAD_M_REG register. At the same time, the N0-shadow register will be loaded by the TIMER0_RELOAD_N_REG register. Both counters will be decremented on the next clock again and the sequence will be repeated.

Note that it is possible to generate interrupts at a high rate, when selecting a high clock frequency in combination with low counter values. This could result in missed interrupt events.

During the time that the ON-counter is non-zero, new values for the ON-register, M0-register and N0-register can be written, but they are not used by the T0-counter until a full cycle is finished. More specifically, the newly written values in the TIMER0_RELOAD_M_REG and TIMER0_RELOAD_N_REG registers are only stored into the shadow registers when the ON-counter and the T0-counter have both reached zero and the T0-counter was decrementing the value loaded from the TIMER0_RELOAD_N_REG register (see Figure 57).

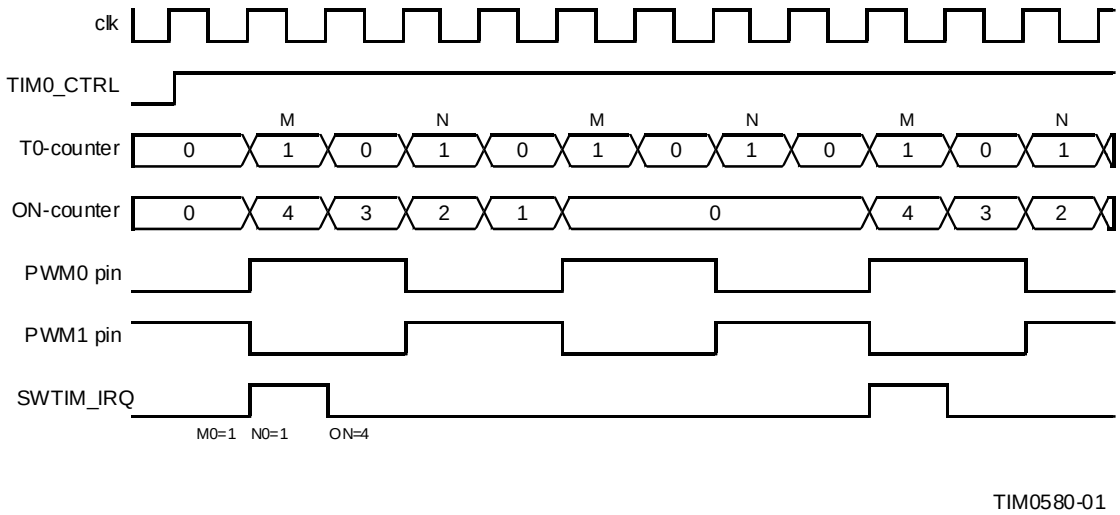


Figure 57: Timer 0 PWM Mode

At start-up both counters and the PWM0 signal are LOW so also at start-up an interrupt is generated. If Timer 0 is disabled all flip-flops, counters and outputs are in reset state except for the ON-register, the TIMER0_RELOAD_N_REG register and the TIMER0_RELOAD_M_REG register.

The timer input registers ON-register, TIMER0_RELOAD_N_REGand TIMER0_RELOAD_M_REG can be written and the counter registers ON-counter and T0-counter can be read. When reading from the address of the ON-register, the value of the ON-counter is returned. Reading from the address of either the TIMER0_RELOAD_N_REG or the TIMER0_RELOAD_M_REG register, returns the value of the T0-counter.

It is possible to freeze Timer 0 with bit FRZ_SWTIM of the register SET_FREEZE_REG. When the timer is frozen the timer counters are not decremented. This will freeze all the timer registers at their last value. The timer will continue its operation again when bit FRZ_SWTIM is cleared via register RESET_FREEZE_REG.

19.2 Timer 2

Timer 2 has three Pulse Width Modulated (PWM) outputs. The block diagram is shown in Figure 58.

Features

- 14-bit general purpose timer
- Ability to generate 3 Pulse Width Modulated signals (PWM2, PWM3 and PWM4)
- Input clock frequency:
  $f_{IN} = \frac{sys\_clk}{N}$ with N = 1, 2, 4 or 8
  and sys_clk = 16 MHz or 32 kHz
  $$f_{OUT} = \left(\frac{f_{IN}}{2}\right) to \left(\frac{f_{IN}}{2^{14}-1}\right)$$
- Programmable output frequency:
- Three outputs with programmable duty cycle from 0 % to 100 %
- Used for white LED intensity (on/off) control

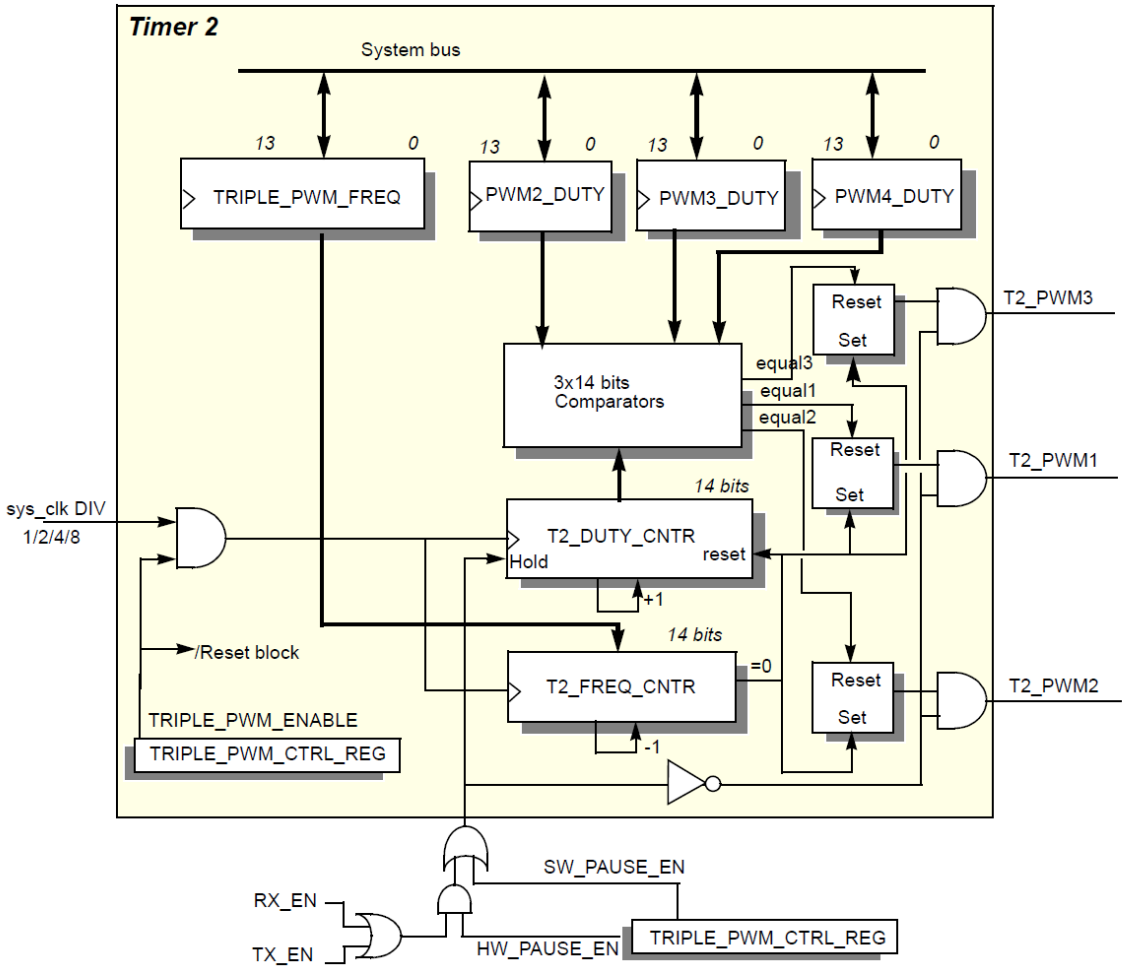


Figure 58: Timer 2 PWM Block Diagram

The Timer 2 is clocked with the system clock divided by TMR_DIV (1, 2, 4 or 8) and can be enabled with TRIPLE_PWM_CTRL_REG[TRIPLE_PWM_ENABLE].

T2_FREQ_CNTR determines the output frequency of the T2_PMWn output. This counter counts down from the value stored in register TRIPLE_PWM_FREQUENCY. At counter value 0, T2_FREQ_CNTR sets the T2_PWMn output to '1' and the counter is reloaded again.

T2_DUTY_CNTR is an up-counter that determines the duty cycle of the T2_PWMn output signal. After the block is enabled, the counter starts from 0. If T2_DUTY_CNTR is equal to the value stored in the respective PWMn_DUTY_CYCLE register, this resets the T2_PWMn output to 0. T2_DUTY_CNTR is reset when TRIPLE_PWM_FREQUENCY is 0.

Note that the value of PWMn_DUTY_CYCLE must be less or equal than TRIPLE_PWM_FREQUENCY.

The Timer 2 is enabled/disabled by programming the TRIPLE_PWM_CTRL_REG[TRIPLE_PWM_EN] bit.

The timing diagram of Timer 2 is shown in Figure 59.

Freeze function

During RF activity it may be desirable to temporarily suppress the PWM switching noise. This can be done by setting TRIPLE_PWM_CTRL_REG[HW_PAUSE_EN] = 1. The effect is that whenever there is a transmission or a reception process from the Radio, T2_DUTY_CNTR is frozen and T2_PWMx output is switched to '0' to disable the selected T2_PWM1, T2_PWM2, T2_PWM3. As soon as the Radio is idle (i.e. RX_EN or TX_EN signals are zero), T2_DUTY_CNTR resumes counting and finalizes the remaining part of the PWM duty cycle.

TRIPLE_PWM_CTRL_REG[SW_PAUSE_EN] can be set to '0' to disable the automatic, hardware driven freeze function of the duty counter and keep the duty cycle constant.

Note that the RX_EN and TX_EN signals are not software driven but controlled by the BLE core hardware.

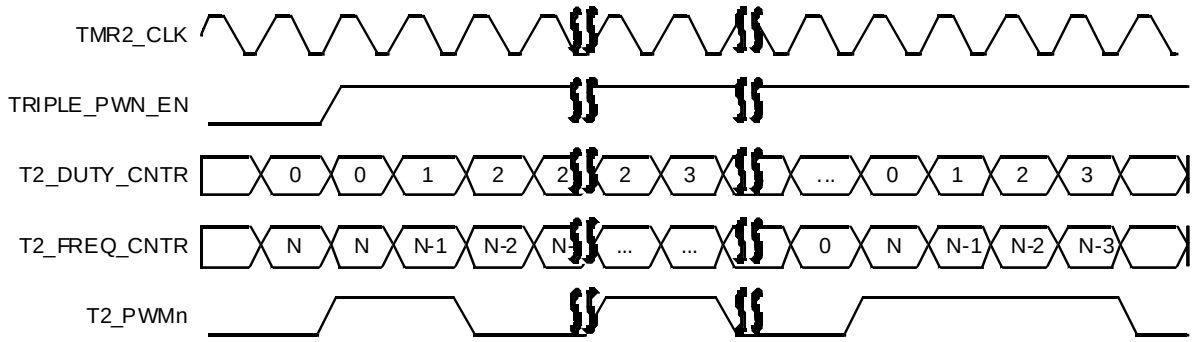


Figure 59: Timer 2 PWM Timing Diagram

20 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.

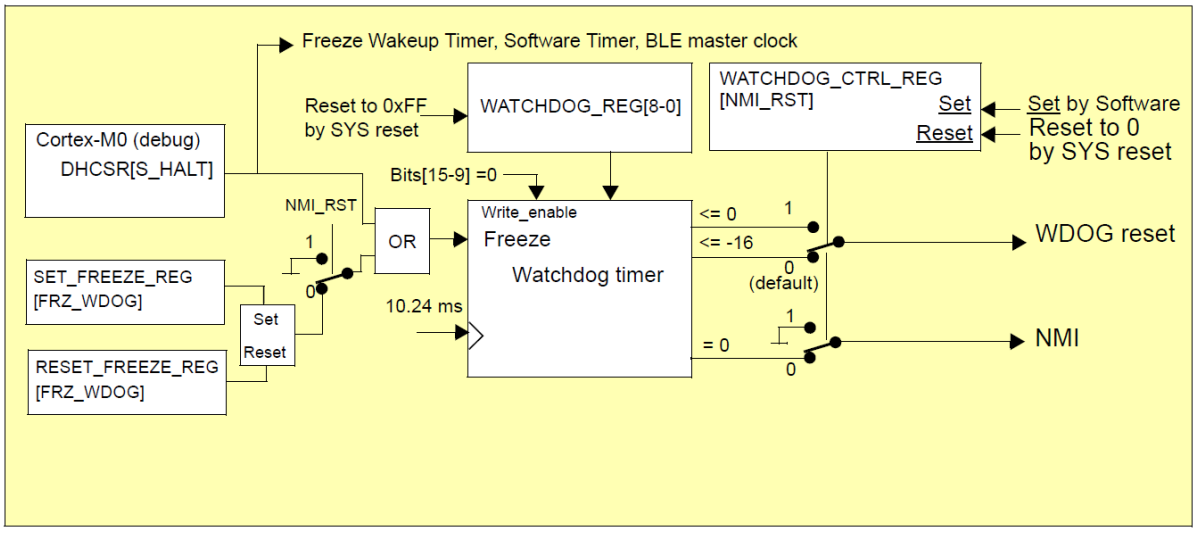


Figure 60: Watchdog Timer Block Diagram

The 8 bits watchdog timer is decremented by 1 every 10.24 ms. The timer value can be accessed through the WATCHDOG_REG register which is set to 255 (FF_{16}) at reset. This results in a maximum watchdog time-out of ~2.6 s. During write access the WATCHDOG_REG[WDOG_WEN] bits must be 0. This provides extra filtering for a software run-away writing all ones to the WATCHDOG_REG. If the watchdog counter reaches 0, the counter value will get a negative value by setting bit 8. The counter sequence becomes 1, 0, $1FF_{16}$ (-1), $1FE_{16}$(-2),...$1F0_{16}$ (-16).

If WATCHDOG_CTRL_REG[NMI_RST] = 0, the watchdog timer will generate an NMI if the watchdog timer reaches 0 and a WDOG reset if the counter becomes less or equal to -16 ($1F0_{16}$). The NMI handler must write any value > -16 to the WATCHDOG_REG to prevent the generation of a WDOG reset at counter value -16 after 16*10.24 = 163.8 ms.

If WATCHDOG_CTRL_REG[NMI_RST] = 1, the watchdog timer generates a WDOG reset if the timer becomes less or equal than 0.

The WDOG reset is one of the SYS (system) reset sources and resets the whole device, including setting the WATCHDOG_REG register to 255, except for the RST pin, the Power On reset, the HW reset and the DBG (debug module) reset. Since the HW reset is not triggered, the SYS_CTRL_REG[REMAP_ADR0] bits will retain their value and the Cortex-M0 will start executing again from the current selected memory at address zero. Refer to the POR, HW and SW Reset section for an overview of the complete reset circuit and conditions.

For debugging purposes, the Cortex-M0 Debug module can always freeze the watchdog by setting the DHCSR[DBGKEY | C_HALT | C_DEBUGEN] control bits (reflected by the status bit S_HALT). This is automatically done by the debug tool, e.g. during step-by-step debugging. Note that this bit also freezes the Wake-up Timer, the Software Timer and the BLE master clock. For additional information also see the DEBUG_REG[DEBUGS_FREEZE_EN] mask register. The C_DEBUGEN bit is not accessible by the user software to prevent freezing the watchdog.

In addition to the S_HALT bit, the watchdog timer can also be frozen if NMI_RST=0 and SET_FREEZE_REG[FRZ_WDOG] is set to ‘1’. The watchdog timer resumes counting when RESET_FREEZE_REG[FRZ_WDOG] is set to ‘1’. The WATCHDOG_CTRL_REG[NMI_RST] bit can only be set by software and will only be reset on a SYS reset. 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.

21 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 (14 in the WLCSP package, 25 in the QFN40 and 32 in the QFN48)
- Generates a keyboard interrupt on key press or key release
- Implements debouncing time from 0 upto 63 ms
- Supports five separate interrupt generation lines from GPIO toggling

The block diagram of the Keyboard Controller is presented in the Figure 61: Keyboard Controller Block Diagram.

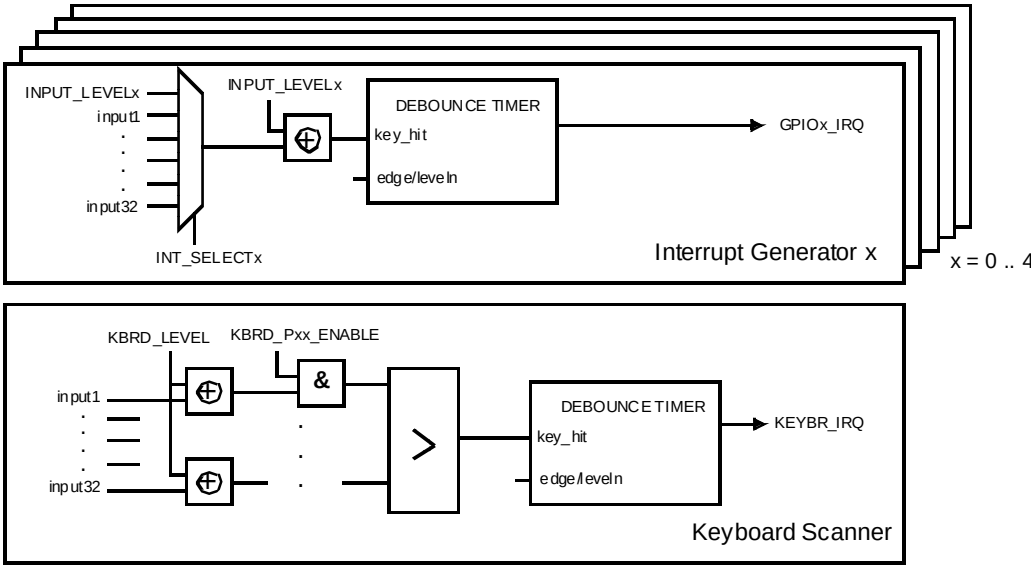


Figure 61: Keyboard Controller Block Diagram

21.1 Keyboard Scanner

A HIGH-to-LOW transition on one of the port inputs, while KBRD_IRQ_IN_SEL0_REG[KBRD_LEVEL] = 0 and KBRD_IRQ_IN_SELx_REG[KBRD_Pyy_EN] = 1 sets internal signal “key_hit” to 1. This signal triggers the keyboard interface state machine as shown in Figure 62: Keyboard Scanner State Machine. The debounce timer is loaded with value GPIO_DEBOUNCE_REG[DEB_VALUE]. The timer counts down every 1 ms. When the timer reaches 0 and the “key_hit” signal is still ‘1’, the timer is loaded with KBRD_IRQ_IN_SEL0_REG[KEY_REPEAT], generating a repeating sequence of interrupts every time the timer reaches 0.

When the key is released (key_hit = 0) and bit KBRD_REL (key release) is set to ‘1’, a new debounce sequence is started and a KEYBR_IRQ interrupt is generated after the debounce time.

The debounce timer can be disabled with GPIO_DEBOUNCE_REG[DEB_ENABLE_KBRD] = 0. The key repeat function can be disabled by setting KEY_REPEAT to ‘0’.

The level for generating an interrupt is programmable via bit KBRD_IRQ_IN_SEL0_REG[KBRD_LEVEL]. The key release function can be disabled by setting bit KBRD_IRQ_IN_SEL0_REG[KBRD_REL] to ‘0’. The inputs for the keyboard interface can be selected by setting the corresponding bits KBRD_IRQ_IN_SEL0_REG[KBRD_Pxx_EN] to ‘1’.

The keyboard interrupt service routine can distinguish which input has caused the interrupt by reading the Px_DATA_REG registers.

21.2 GPIO Interrupt Generator

Five identical GPIO interrupt generators support the generation of up to five interrupts (GPIO0_IRQ to GPIO_4_IRQ). One of the GPIO inputs can be selected to generate an interrupt by programming the corresponding GPIO_IRQx_IN_SEL_REG register. The input level can be selected by GPIO_INT_LEVEL_CTRL_REG[INPUT_LEVELx].

A LOW-to-HIGH level transition on one of the port inputs, while bit INPUT_LEVELx = 0, sets internal signal “key_hit” to ‘1’. This signal triggers the GPIO Interrupt Generator state machine as shown in Figure 62: Keyboard Scanner State Machine. The debounce timer is loaded with value GPIO_DEBOUNCE_REG[DEB_VALUE]. The timer counts down every 1 ms. If the timer reaches 0 and the “key_hit” signal is still ‘1’, an interrupt will be generated. The debounce timer for each interrupt can be disabled with GPIO_DEBOUNCE_REG[DEB_ENABLEx].

The interrupt flag will remain set until it is reset by writing to the corresponding bit in the GPIO_RESET_IRQ_REG register. When the GPIO interrupt is edge sensitive, selected with bit GPIO_INT_LEVEL_CTRL_REG[EDGE_LEVELNx], the state machine will progress to state WAIT_FOR_RELEASE when the interrupt is reset. It will progress to the IDLE state only after detecting the non-active edge.

For detecting both signal edges the edge polarity INPUT_LEVELx must be inverted in the WAIT_FOR_RELEASE state. This will result in “key_hit” = 0 and will advance the state machine to the Idle state, allowing detection of the next inverted edge.

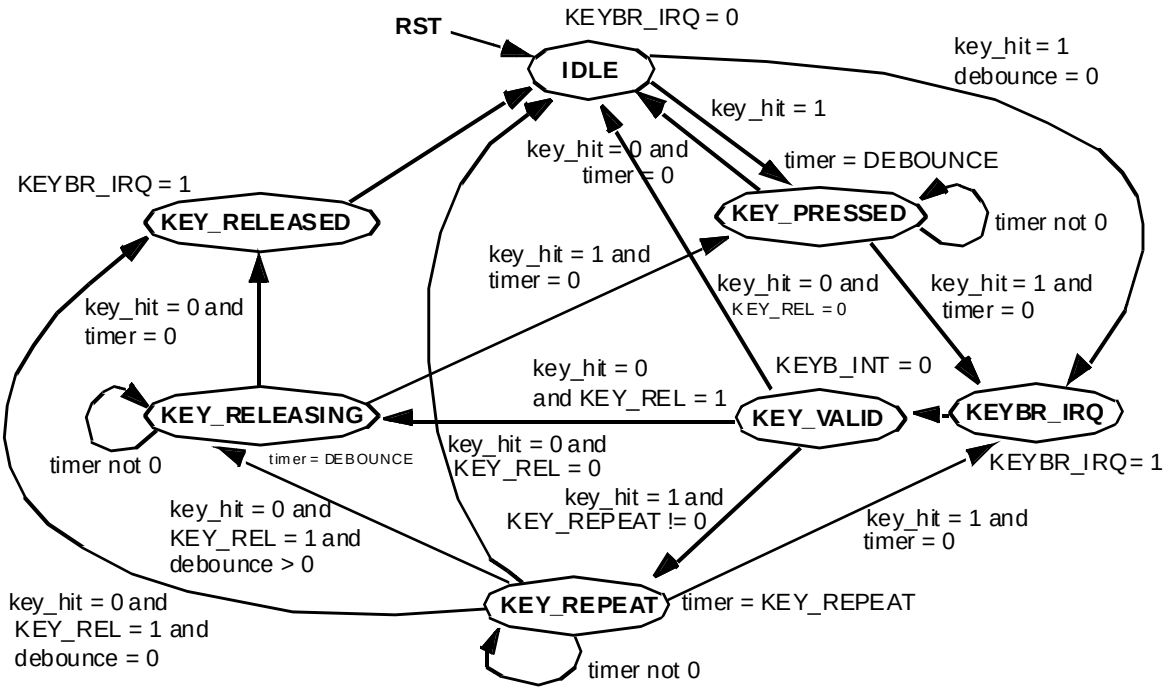


Figure 62: Keyboard Scanner State Machine

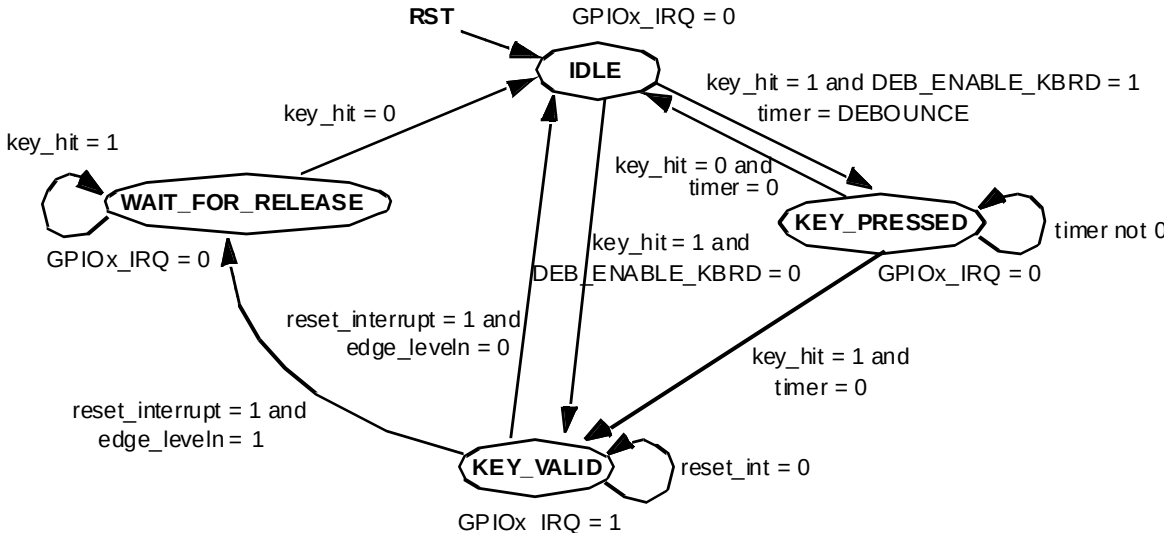


Figure 63: GPIO Interrupt Generator State Machine

22 Input/Output Ports

The DA14585 has software-configurable I/O pin assignment, organized into ports Port 0, Port 1 and Port 2. Port 2 is only available in the QFN40 and QFN48 package. Pin 0 of Port 3 is also available in the QFN40 package, whereas the full Port 3 is available on the QFN48 package.

Features

- Port 0: 8 pins, Port 1: 6 pins (including SW_CLK and SWDIO), Port 2: 10 pins, Port 3: 1-8 pins
- Fully programmable pin assignment
- Selectable 25 kΩ 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 can retain their last state when system enters the Extended or Deep Sleep mode.

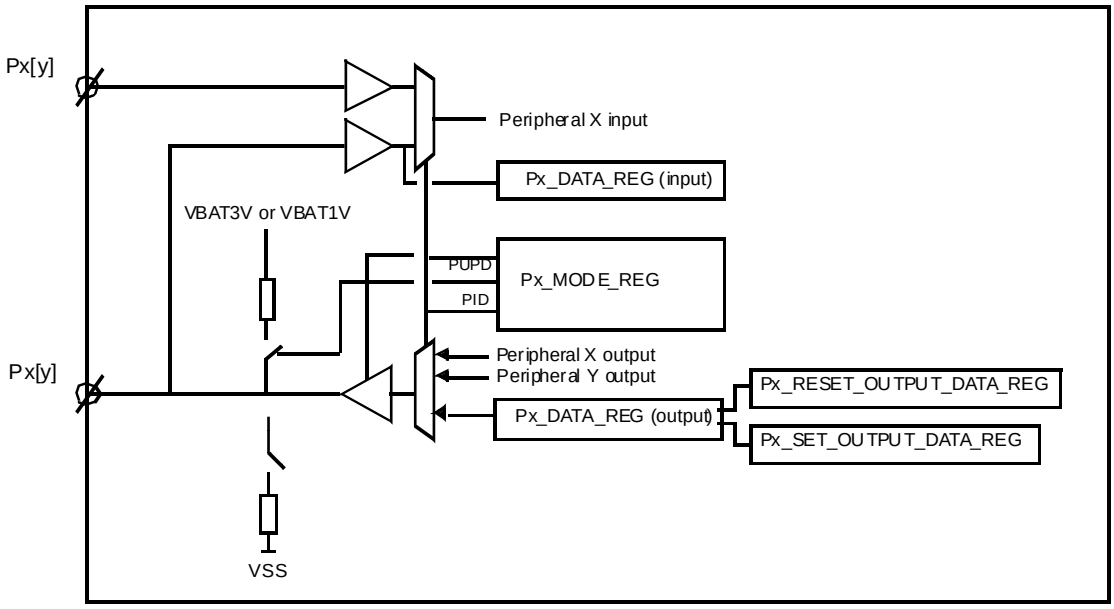


Figure 64: Port P0, P1, P2 and P3 with Programmable Pin Assignment

22.1 Programmable Pin Assignment

The Programmable Pin Assignment (PPA) provides a multiplexing function to the I/O pins of on-chip peripherals. Any peripheral input or output signal can be freely mapped to any I/O port bit by setting Pxy_MODE_REG[4-0]:

```
0x00 to 0x1F: Peripheral IO ID (PID)
```

Refer to the Px_MODE_REGs for an overview of the available PIDs. Analog ADC has fixed pin assignment in order to limit interference with the digital domain. The SWD interface (JTAG) is mapped on P1_4 and P1_5.

Priority

The firmware has the possibility to assign the same peripheral output to more than one pin. It is the responsibility of the user to make a unique assignment.

In case more than one input signal is assigned to a peripheral input, the left most pin in the lowest port pin number has priority. (e.g P00_MODE_REG has priority over P01_MODE_REG).

Direction Control

The port direction is controlled by setting:

Pxy_MODE_REG[9-8]

```
00 = Input, no resistors selected
01 = Input, pull-up selected
10 = Input, Pull-down selected
11 = Output, no resistors selected
```

In output mode and analog mode the pull-up/down resistors are automatically disabled.

22.2 General Purpose Port Registers

The general purpose ports are selected with PID=0. The port function is accessible through registers:

- Px_DATA_REG: Port data input/output register
- Px_SET_OUTPUT_DATA_REG: Port set output register
- Px_RESET_OUTPUT_DATA_REG: Port reset output register

22.2.1 Port Data Register

The registers input Px_DATA_REG and output Px_DATA_REG are mapped on the same address.

The data input register (Px_DATA_REG) is a read-only register that returns the current state on each port pin even if the output direction is selected, regardless of the programmed PID, unless the analog function is selected (in this case it reads 0). The ARM CPU can read this register at any time even when the pin is configured as an output.

The data output register (Px_DATA_REG) holds the data to be driven on the output port pins. In this configuration, writing to the register changes the output value.

22.2.2 Port Set Data Output Register

Writing a 1 in the set data output register (Px_SET_DATA_REG) sets the corresponding output pin. Writing a 0 is ignored.

22.2.3 Port Reset Data Output Register

Writing a 1 in the reset data output register (Px_RESET_DATA_REG) resets the corresponding output pin. Writing a 0 is ignored.

22.3 Fixed Assignment Functionality

There are certain signals that have a fixed mapping on specific general purpose IOs. This assignment is illustrated in the following table:

Table 39: Fixed Assignment of Specific Signals

GPIO	GPIO (Note 1)	QUAD DEC (Note 2)	ADC (Note 3)
P0_0		CHY_A/CHX_A/CHZ_A	ADC_0
P0_1		CHY_B/CHX_B/CHZ_B	ADC_1
P0_2		CHY_A/CHX_A/CHZ_A	ADC_2
P0_3		CHY_B/CHX_B/CHZ_B	ADC_3
P0_4		CHY_A/CHX_A/CHZ_A	
P0_5		CHY_B/CHX_B/CHZ_B	
P0_6		CHY_A/CHX_A/CHZ_A	

GPIO	GPIO (Note 1)	QUAD DEC (Note 2)	ADC (Note 3)
P0_7		CHY_B/CHX_B/CHZ_B	
P1_0		CHY_A/CHX_A/CHZ_A	
P1_1		CHY_B/CHX_B/CHZ_B	
P1_2		CHY_A/CHX_A/CHZ_A	
P1_3		CHY_B/CHX_B/CHZ_B	
P1_4	SW_CLK	CHY_A/CHX_A/CHZ_A	
P1_5	SWDIO	CHY_B/CHX_B/CHZ_B	
P2_0		CHY_A/CHX_A/CHZ_A	
P2_9		CHY_B/CHX_B/CHZ_B	

Note 1 The SWD signals mapping is defined by SYS_CTRL_REG[DEBUGGER_ENABLE]. However, these signals are mapped on the ports by default

Note 2 The mapping of the Quad Decoder signals on the respective pins, is overruled by the QDEC_CTRL2_REG[CHx_PORT_SEL] register.

Note 3 The ADC case can be selected by the PID bit field on the respective Px port.

23 General Purpose ADC

The DA14585 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 µA typical supply current at 100 ksample/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

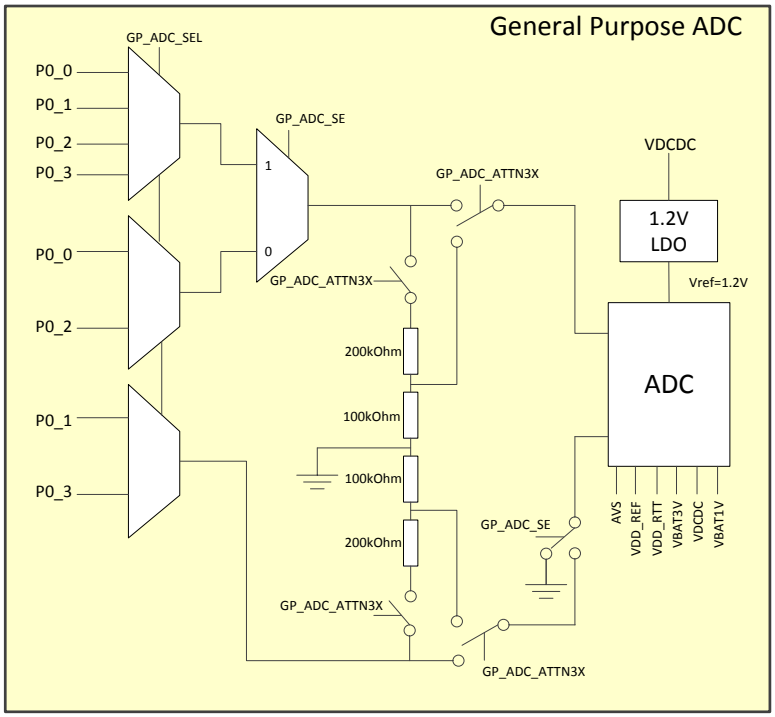


Figure 65: Block Diagram of the General Purpose ADC

23.1 Input Channels and Input Scale

The DA14585 has a multiplexer between the ADC and four specific GPIO ports (P0_0 to P0_3). Furthermore, the ADC can also be used to monitor the battery voltage and several internal voltages of the system (see GP_ADC_CTRL_REG).

Single-ended or differential operation is selected via bit GP_ADC_CTRL_REG[GP_ADC_SE]. In differential mode the voltage difference between two GPIO input ports will be converted. Via bit GP_ADC_CTRL2_REG[GP_ADC_ATTN3X] the input scale can be enlarged by a factor of three, as summarized in Table 40.

Table 40: GPADC Input Channels and Voltage Scale

GP_ADC_ATTN3X	GP_ADC_SE	Input Channels	Input Scale	Input Limits
0	1	P0_0, P0_1, P0_2, P0_3	0 V to +1.2 V	-0.1 V to +1.3 V
0	0	[P0_0, P0_1], [P0_2, P0_3]	-1.2 V to +1.2 V	-1.3 V to +1.3 V
1	1	P0_0, P0_1, P0_2, P0_3	0 V to +3.6 V	-0.1 V to +3.45 V
1	0	[P0_0, P0_1], [P0_2, P0_3]	-3.6 V to +3.6 V	-3.45 V to +3.45 V

23.2 Starting the ADC and Sampling Rate

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

Enabling/disabling of the ADC is triggered by configuring bit GP_ADC_CTRL_REG[GP_ADC_LDO_EN]. After enabling the LDO, a settling time of 20 µs is required before an AD-conversion can be started.

Each conversion has two phases: the sampling phase and the conversion phase. When bit GP_ADC_CTRL_REG[GP_ADC_EN] is set to ‘1’, the ADC continuously tracks (samples) the selected input voltage. Writing a '1' at bit GP_ADC_CTRL_REG[GP_ADC_START] ends the sampling phase and triggers the conversion phase. When the conversion is ready, the ADC resets bit GP_ADC_START to ‘0’ and returns to the sampling phase.

The conversion itself is fast and takes approximately one clock cycle of 16 MHz, though the data handling will require several additional clock cycles, depending on the software code style. The fastest code can handle the data in four clock cycles of 16 MHz, resulting to a highest sampling rate of 16 MHz/5 = 3.3 Msample/s.

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

23.3 Non-Ideal Effects

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

The ADC result will also include some noise. If the input signal itself is noise free (inductive effects included), the average noise level will be ±1 LSB. Taking more samples and calculating the average value will reduce the noise and increase the resolution.

With a 'perfect' input signal (e.g. if a filter capacitor is placed close to the input pin) most of the noise comes from the low-power voltage regulator (LDO) of the ADC. Since the DA14585 is targeted for ultra-compact applications, there is no pin available to add a capacitor at this voltage regulator output.

The dynamic current of the ADC causes extra noise at the regulator output. This noise can be reduced by setting bits GP_ADC_CTRL2_REG[GP_ADC_I20U] and GP_ADC_CTRL2_REG[GP_ADC_IDYN] to ‘1’. Bit GP_ADC_I20U enables a constant 20 µA load current at the regulator output so that the current will not drop to zero. Bit GP_ADC_IDYN enables a 10 µA load current during sampling phase so that the load current during sampling and conversion phase becomes approximately the same.

23.4 Chopping

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

Chopping is enabled by setting bit GP_ADC_CTRL_REG[GP_ADC_CHOP] to ‘1’.

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

If bit GP_ADC_CTRL2_REG[GP_ADC_ATTN3X] is set to ‘1’, the zero point is 3 times higher (1.8 V single-ended and 0.0 V differential).

With bit GP_ADC_CTRL_REG[GP_ADC_MUTE], the ADC input is switched to the center scale input level, so the ADC result ideally is 511.5. If instead a value of 515 is observed, the output offset is +3.5 (adc_off_p = 3.5).

With bit GP_ADC_CTRL_REG[GP_ADC_SIGN] the sign of the ADC input and output is changed. Two sign changes have no effect on the signal path, though the sign of the ADC offset will change.

If adc_off_p = 3.5 the ADC_result with opposite GP_ADC_SIGN will be 508. The sum of these equals 515 + 508 = 1023. This is the mid-scale value of an 11-bit ADC, so one extra bit due to the over-sampling by a factor of two.

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

23.5 Offset Calibration

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

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

To calibrate the ADC follow the steps in Table 41.

Table 41: GPADC Calibration Procedure for Single-Ended and Differential Modes

Step	Single-Ended Mode (GP_ADC_SE = 1)	Differential Mode (GP_ADC_SE = 0)
1	Set GP_ADC_OFFP = GP_ADC_OFFN=0x200; GP_ADC_MUTE = 0x1; GP_ADC_SIGN = 0x0	Set GP_ADC_OFFP=GP_ADC_OFFN = 0x200; GP_ADC_MUTE = 0x1; GP_ADC_SIGN = 0x0
2	Start conversion	Start conversion
3	adc_off_p = GP_ADC_RESULT - 0x200	adc_off_p = GP_ADC_RESULT - 0x200
4	Set GP_ADC_SIGN = 0x1	Set GP_ADC_SIGN = 0x1
5	Start conversion	Start conversion
6	adc_off_n = GP_ADC_RESULT - 0x200	adc_off_n = GP_ADC_RESULT - 0x200
7	GP_ADC_OFFP = 0x200 - 2*adc_off_p GP_ADC_OFFN = 0x200 - 2*adc_off_n (Note 1)	GP_ADC_OFFP = 0x200 - adc_off_p GP_ADC_OFFN = 0x200 - adc_off_n (Note 1)

Note 1 The average of GP_ADC_OFFP and GP_ADC_OFFN should be 0x200 (with a margin of 20 LSB).

It is recommended to implement the above calibration routine during the initialization phase of the DA14585. To verify the calibration results, check whether the GP_ADC_RESULT value is close to 0x200 while bit GP_ADC_MUTE = 1.

23.6 Zero-Scale Adjustment

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

23.7 Common Mode Adjustment

The common mode level of the differential signal must be 0.6 V (or 1.8 V with GP_ADC_ATTN3X = 1). If the common mode input level of 0.6 V cannot be achieved, the common mode level of the GP_ADC can be adjusted (the GP_ADC can tolerate a common mode margin up to 50 mV) according to Table 42.

Table 42: Common Mode Adjustment

CM Voltage (Vcmm)	GP_ADC_OFFP = GP_ADC_OFFN
0.3 V	0x300
0.6 V	0x200
0.9 V	0x100

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

Note: The input voltage limits for the ADC in differential mode are: -1.3 V to +1.3 V (for GP_ADC_ATTN3X = 0, see Table 40: GPADC Input Channels and Voltage Scale). The differential input range of the ADC is: -1.2 V < V[P0_0,P0_1] < +1.2 V. Therefore, if Vcmm < 0.5 V or Vcmm > 0.7 V, the input can no longer cover the whole ADC range.

23.8 Input Impedance, Inductance and Input Settling

The GPADC has no input buffer stage. During sampling phase a capacitor of 0.2 pF is switched to the input line. The precharge of this capacitor is at mid-scale level so the input impedance is infinite.

At 100 ksample/s, zero or full-scale single-ended input signal, this sampling capacitor will load the input with:
ILOAD = V * C * fS = ±0.6 V * 0.2 pF * 100 kHz = ±12 nA (differential: ±1.2 V * 0.2 pF * 100 kHz = ±24 nA at both pins).

During sampling phase a certain settling time is required. A 10-bit accuracy requires at least 7 time constants of the output impedance of the input signal source and the 0.2 pF sampling capacitor. The conversion time is approximately one clock cycle of 16 MHz (62.5 ns).

7 * ROUT * 0.2 pF - 62.5 ns < 1/fS
=> ROUT < (1 + 62.5 ns * fS) / (7 * 0.2 pF * fS)

Examples:

ROUT < 7.2 MΩ at fS = 100 kHz
ROUT < 760 kΩ at fS = 1 MHz

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

To observe the noise level of the ADC and the voltage regulator, bit GP_ADC_CTRL_REG[GP_ADC_MUTE] must be set to ‘1’. The noise should be less than ±1 LSB on average, with occasionally a ±2 LSB peak value. If a higher noise level is observed on the input channel(s), applying filter capacitor(s) will reduce the noise.

The 3x input attenuator is realized with a resistor divider network. When bit GP_ADC_CTRL_REG2[GP_ADC_ATTN3X] is set to ‘1’, the input impedance of the selected ADC input channel becomes 300 kΩ (typical) instead of infinite. In addition, the resistor divider network will require more settling time in the sampling phase. The general guideline with bit GP_ADC_ATTN3X = 1 is: select the input channel, then wait 1 µs (16 clock cycles) before starting the conversion. Only the required sampling time is affected by the attenuator, the conversion time remains approximately one clock cycle of 16 MHz (62.5 ns).

Note: Selecting the battery measurement channel automatically activates the 3x input attenuator (bit GP_ADC_ATTN3X = 1). Therefore the 1 µs waiting time also applies when measuring the battery voltage, otherwise the resulting Vbat level will be too low.

23.9 Delay Counter

The GPADC has a delay counter that can be used to add delays to several ADC control signals. A delay of up to 32 µs can be added for the bits GP_ADC_LDO_EN, GP_ADC_START and GP_ADC_EN via registers GP_ADC_DELAY_REG and GP_ADC_DELAY2_REG.

The reset values of these two registers are the recommended values for a correct start-up, since it is not allowed to activate all signals at once.

To make use of the delay counter for a certain signal, the corresponding bit has to be set in register GP_ADC_CTRL_REG and the delay counter must be enabled via bit GP_ADC_DELAY_EN in register GP_ADC_CTRL2_REG. The delay counter starts counting when the GP_ADC_START bit is programmed while the GP_ADC_DELAY_EN bit is set. The counter is stopped after the conversion is finished.

The delay counter must be reset before reuse, which is typically only required after the LDO was disabled. Bit GP_ADC_DELAY_EN must be made zero to reset the counter. It is recommended to check that this bit is zero before (re)activating it.

24 Audio Unit (AU)

24.1 Introduction

The Audio Unit comprises 2 digital interfaces namely a PDM and PCM and a Sampling Rate Converter (SRC) which is used for adjusting the sampling rate of audio samples between the two interfaces and memory.

The PDM interface provides a serial connection for 1 stereo or 2 mono input devices (e.g. MEMS microphones) or output devices. The interface has a single clock PDM_CLK and one input/output PDM_DI/PDM_DO which is capable of carrying two channels in a time divided manner.

The PCM controller is implementing an up-to 192 kHz synchronous interface to external audio devices, ISDN circuits and serial data interfaces. It enables master and slave modes and also supports I2S and TDM formats.

The AU has dedicated DMA channels for the PCM and PDM streams.

Features

- Supported conversions:
  - SRC_IN (24 bits) to SRC_OUT (24 bits)
  - PDM_IN (1bit) to SRC_OUT (24 bits)
  - SRC_IN (24 bits) to PDM_OUT (1 bit)
- SRC_IN, SRC_OUT Sample rates 8 kHz to 192 kHz
- SNR > 100 dB
- Single Buffer I/O with DMA support
- Automatic mode to adjust sample rate to the applied frame sync (e.g. PCM_FSC)
- Manual mode to generate interrupts at the programmed sample rate. Adjustment is done by SW based on buffer pointers
- PCM (Master/Slave) interface
  - PCM_FSC
  - Master/slave 4 kHz to 96 kHz
  - Strobe Length 1, 8, 16, 24, 32, 40, 48 and 64 bits
  - PCM_FSC before or on the first bit. (In Master mode)
  - 2x24 channels
  - Programmable slot delay up-to 31*8 bits
  - Formats
  - PCM mode
  - I2S mode (Left/Right channel selection) with N*8 for Left and N*8 for Right
  - IOM2 mode (double clock per bit)
  - Programmable clock and frame sync inversion
- PDM interface
  - PDM_CLK frequency 62.5 kHz - 4 MHz
  - Down-sampling to 24 bits in SRC
  - PDM_CLK on/off to support Sleep mode
  - PDM_DATA
  - (input): 1 Channel in stereo format
  - (output): 2 Channels in mono format, 1 Channel in stereo format
  - Programmable Left/Right channel selection
  -

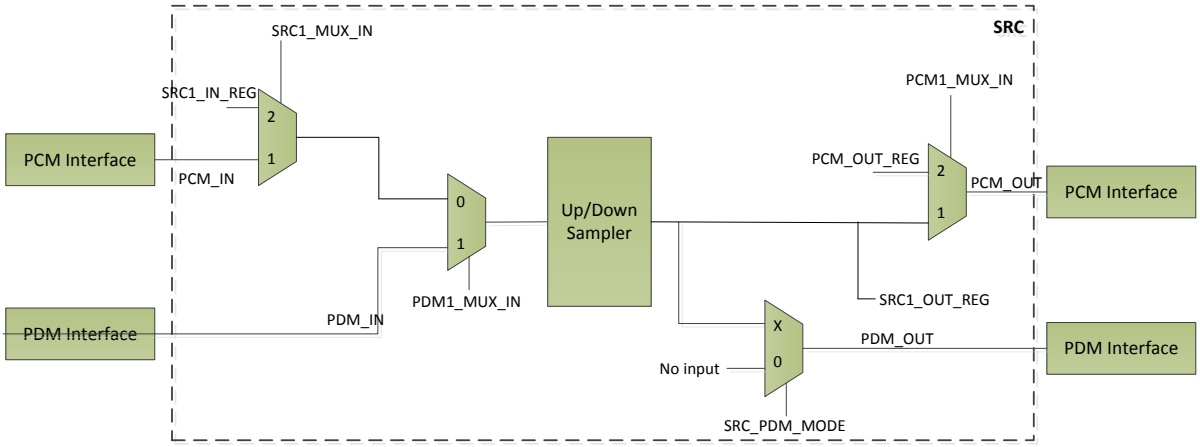


Figure 66: Audio Unit Block Diagram

24.2 Architecture

24.2.1 Data Paths

The SRC block converts two 24-bit channels either as a stereo pair or as two mono streams. PCM linear data pairs are transferred to SRC1_IN1/2 and the output is 2x24-bit left aligned on SRC1_OUT1/2. The two 1 bit PDM data inputs are received on PDM_IN and are converted to 2x24 bits, left aligned to SRC_OUT.

The SRCx_IN input multiplexer (Figure 66) is controlled by APU_MUX_REG[PDM1_MUX_IN]. The input of these multiplexers comes either from the audio interfaces or from SRC1_IN1/2_REG. The data to these register is left aligned, bits 31-8 are mapped on bits 23-0 of the SRC.

The 24 bits SRC outputs can be read in SRC1_OUT1_REG and SRC1_OUT2_REG and can also be routed to the PCM interface. This input selection of these multiplexers is also controlled by APU_MUX_REG[PCM1_MUX_IN].

The SRC can be configured to operate in two different modes of operation:

- Manual mode
- Automatic mode

In **manual mode,** the input/output sampling rate is determined by SRC1_IN_FS_REG / SRC1_OUT_FS_REG registers.

In **automatic mode**, the input/output sampling rate is automatically derived from the external synchronization signals and can only be read back at SRC1_IN_FS_REG / SRC1_OUT_FS_REG registers.

When PDM is used (input/output), SRC operates in automatic mode. The sampling rate reported in SRC1_IN_FS_REG register is PDM_CLK/64, 64 being the default oversampling ratio.

24.2.2 Up/Down Sampler

The Up/Down Sampler performs the required arbitrary resampling by polynomial interpolation at an 8x oversampled input rate and 16x oversampled output rate.

A two stage upsampler proves the input to the polynomial interpolator from a PCM input. The first stage (ups_1to4) supports 1x, 2x and 4x oversampled input/output and 4x oversampled output.

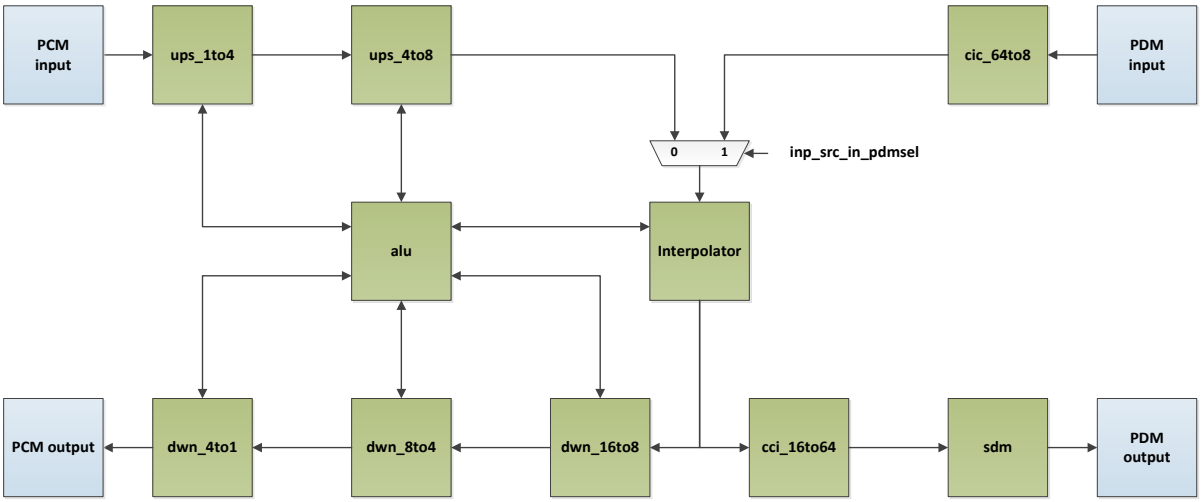


Figure 67: Audio Unit Up/Down Sampler Block Diagram

The second stage (ups_4to8) consist of a halfband IIR interpolation filter, implemented in a polyphase structure based on allpass lattice filters.

A three stage downsampler processes the output of the polynomial interpolator in order to generate the PCM output. The first two stages (dwn_16to8, dwn_8to4) are halfband IIR decimation filters, based on allpass lattice filters. The last stage (dwn_4to1) supports 1x, 2x and 4x oversampled output.

A CIC type downsampler (CIC_64to8) provides input to the polynomial interpolator from a PDM input, and CCI type upsampler (CCI_16to64) is used to provide input to a sigma delta modulator (sdm) that provides the PDM output.

For maximum flexibility, a generic single cycle multiplier facilitates variable coefficient multiplications. The multiplier is combined with optional pre and post adders into an arithmetic unit (alu).

24.2.3 PCM Interface

24.2.3.1 Channel Access and Delay

The PCM interface has two 32-bit registers for TX and RX, namely PCM1_IN1/OUT1_REG and PCM1_IN1/OUT2_REG for input and output directions respectively. These registers can be arranged as 8 channels of 8 bits each, named channel 1 to channel 8. By a configurable clock inversion, channel delay and strobe length adjustment, various formats like PCM, I2S, TDM and IOM2 can be supported.

The 8 PCM channels can be delayed with a maximum delay of 31x8bits by configuring PCM1_CTRL_REG[PCM_CH_DEL]. Note that a high delay count in combination with a slow clock, can lead to the PCM_FSC sync occurring before all channels are shifted in or out. The received bits of the current channel may not be properly aligned in that case.

24.2.3.2 Clock Generation

The PCM clock (PCM_CLK) has to be generated according to the required sample rate. There are 2 ways of generating the clock:

1. The Fractional Option. Dividing the system clock by an integer and a fractional part (i.e. inserting jitter in the clock pulse train). This is programmed in the PCM_DIV_REG and PCM_FDIV_REG respectively.
2. The Integer Only option. Approximate the sample rate by adding more clock pulses than bits required. These extra pulses are ignored. This approach is used when external slave devices cannot tolerate the inserted jitter on the clock line. It is configured in PCM_DIV_REG.

The PCM_DIV_REG[PCM_DIV] is a 12 bits field which holds the integer part of the desired clock divider. The fractional part of the divider is stored in the 16 bits PCM_FDIV_REG register. The value of the register is calculated in the following way:

- The position of the left most '1' of the value in binary format defines the denominator
- The amount of '1's define the numerator of the fraction as explained in the example of Table 43.

Table 43: PCM_FDIV_REG programming example

PCM_FDIV_REG (Hex)	PCM_FDIV_REG (Binary)	Numerator	Denominator	Fraction
0x0110	0b100010000	2	9	2/9
0x0101	0b100000001	2	9	2/9
0x1ABC	0b1101010111100	8	13	8/13
0xBEEF	0b1011111011101111	13	16	13/16
0xFEEE	0b1111111011101110	13	16	13/16

The FSC pulse is generated from the PCM_CLK by further dividing it by PCM_FSC_DIV.

Both clock generation options are explained in the following table, with 8 bits, 16 bits, 32 bits and 48 bits in various sample rates.

Table 44: Integer only and Fractional Clock Divisors for various Sample Rates

			XTAL: 16000 kHz				
Sample Rate (kHz)	Bits	Desired Bit Clock (kHz)	Desired Divider (Note 1)	Integer Only Option		Fractional Option	
				PCM_DIV_REG	Actual Word Size (Bits) (Note 2)	PCM_DIV_REG	PCM_FDIV_REG
8	1*8	64	250	250	8	250	
8	1*16	128	125	125	16	125	
8	1*24	192	83,33333333	80	25	83	1/3
8	1*32	256	62,5	50	40	62	1/2
8	2*8	128	125	125	8	125	
8	2*16	256	62,5	50	20	622	1/2
8	2*24	384	41,66666667	40	25	41	2/3
8	2*32	512	31,25	25	40	31	1/4
16	1*8	128	125	125	8	125	
16	1*16	256	62,5	50	20	62	1/2
16	1*24	384	41,66666667	40	25	41	2/3
16	1*32	512	31,25	25	40	31	1/4
16	2*8	256	62,5	50	10	62	1/2
16	2*16	512	31,25	25	20	31	1/4
16	2*24	768	20,83333333	20	25	20	5/6
16	2*32	1024	15,625	10	50	15	5/8
32	1*8	256	62,5	50	10	62	1/2
32	1*16	512	31,25	25	20	31	1/4
32	1*24	768	20,83333333	20	25	20	5/6
32	1*32	1024	15,625	10	50	15	5/8

32	2*8	512	31,25	25	10	31	1/4
32	2*16	1024	15,625	10	25	15	5/8
32	2*24	1536	10,41666667	10	25	10	5/12
32	2*32	2048	7,8125	5	50	7	13/16
48	1*8	384	41,66666667	N/A	N/A	41	2/3
48	1*16	768	20,83333333	N/A	N/A	20	5/6
48	1*24	1152	13,88888889	N/A	N/A	13	8/9
48	1*32	1536	10,41666667	N/A	N/A	10	5/12
48	2*8	768	20,83333333	N/A	N/A	20	5/6
48	2*16	1536	10,41666667	N/A	N/A	10	5/12
48	2*24	2304	6,944444444	N/A	N/A	N/A	N/A
48	2*32	3072	5,208333333	N/A	N/A	N/A	N/A

Note 1 Desired Divider = XTAL Frequency (kHz) / Desired Bit Clock (kHz).

Note 2 Actual Word Size = XTAL Frequency (kHz) / (Actual Divider / Sample Rate).

The yellow marked fields designate that the achieved actual word size in the Integer Only option is larger than the required bits. The last clock pulses will be ignored in this case. For example, to get 24 bits at 8 KHz sampling rate, a clock of 192 KHz is required. In the Integer Only option this is not feasible. A higher clock will be generated (200 KHz) which results in a word of 25 bits. In this case, the last bit will be ignored.

24.2.3.3 External Synchronization

With the PCM interface in slave mode, the PCM interface supports direct routing through the sample rate converter (SRC). Any drift in PCM_FSC or other frame sync frequencies like 44.1 kHz can be directly resampled to e.g. 48 kHz internal sample rate.

24.2.3.4 Data Formats

PCM Master Mode

Master mode is selected if PCM1_CTRL_REG[PCM_MASTER] = 1.

In master mode PCM_FSC is output and falls always over Channel 0. The duration of PCM_FSC is programmable with PCM1_CTRL_REG[PCM_FSCLEN]= 1 or 8,16, 24, 32 clock pulses high. The start position is programmable with PCM1_CTRL_REG[PCM_FSCDEL] and can be placed before or on the first bit of channel 0. The repetition frequency of PCM_FSC is programmable in PCM1_CTRL_REG[PCM_FSC_DIV] to from 8-192kHz.

If master mode selected, PCM_CLK is output and provides one or two clocks per data bit programmable in PCM1_CTRL_REG[PCM_CLK_BIT].

The polarity of the signal can be inverted with bit PCM1_CTRL_REG[PCM_CLKINV].

The PCM_CLK frequency selection is described in Section 24.2.3.2.

PCM Slave Mode

In slave mode (bit MASTER = 0) PCM_FSC is input and determines the starting point of channel 0. The repetition rate of PCM_FSC must be equal to PCM_SYNC and must be high for at least one PCM_CLK cycle. Within one frame, PCM_FSC must be low for at least PCM_CLK cycle. Bit PCM_FSCDEL sets the start position of PCM_FSC before or on the first bit (MSB).

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

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

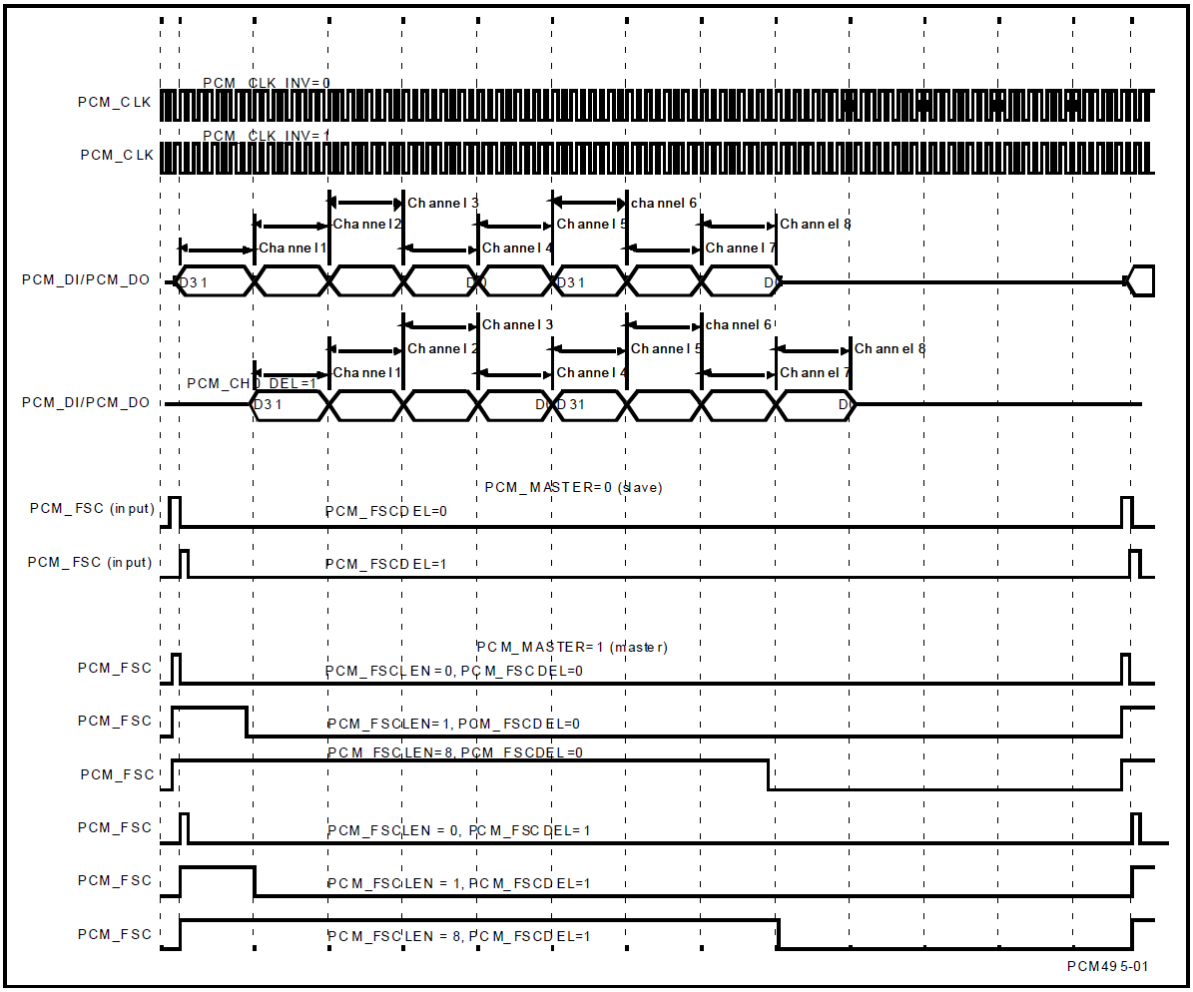


Figure 68: PCM Interface Formats

I2S Formats

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

I2S mode

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

Settings for I2S mode:

- PCM_FSC_EDGE: 1 (all after PCM_FSC)
- PCM_FSCLEN: 4 (4x8 High, 4x8 Low)
- PCM_FSC_DEL: 0 (one bit delayed)
- PCM_CLK_INV: 1 (output on falling edge)
- PCM_CH0_DEL: 0 (no channel delay)

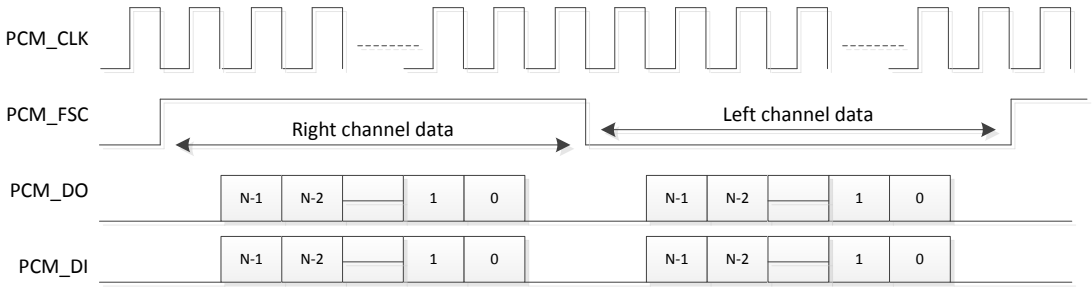


Figure 69: I2S Mode

TDM mode

A time is specified from the normal 'start of frame' condition using register bits PCM_CH_DEL. In the left-justified TDM example illustrated in Figure 70, the left channel data is valid PCM_CH_DEL clock cycles after the rising edge of the PCM_FSC, and the right channel data is valid the same PCM_CH_DEL number of clock cycles after the falling edge of the PCM_FSC.

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

Settings for TDM mode:

- PCM_FSC_EDGE: 1 (rising and falling PCM_FSC)
- PCM_FSCLEN: Master 1 to 4
  Slave waiting for edge.
- PCM_FSC_DEL: 1 (no bit delay)
- PCM_CLK_INV: 1 (output on falling edge)
- PCM_CH0_DEL: Slave 0-31 (channel delay)
  Master 1-3

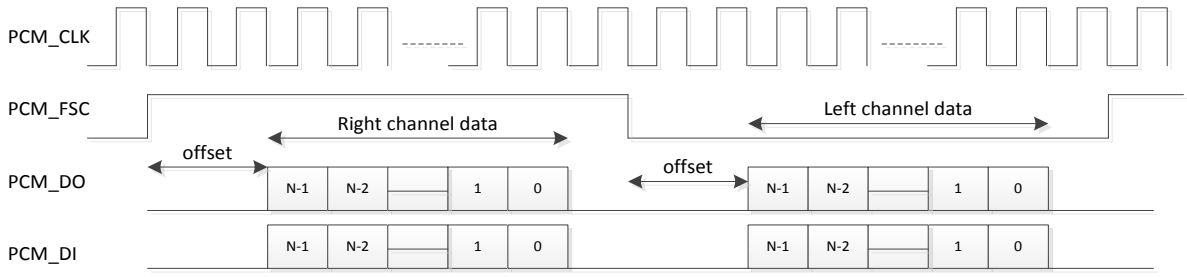


Figure 70: I2S TDM mode (left justified mode)

IOM Mode

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

Settings for IOM mode:

- PCM_FSC_EDGE: 0 (rising edge PCM_FSC)
- PCM_FSCLEN: 0 (one cycle)
- PCM_FSC_DEL: 0 (no bit delay)
- PCM_CLK_INV: 0 (output on rising edge)
- PCM_CH0_DEL: 0 (no delay)
- PCM_CLK_BIT: 1

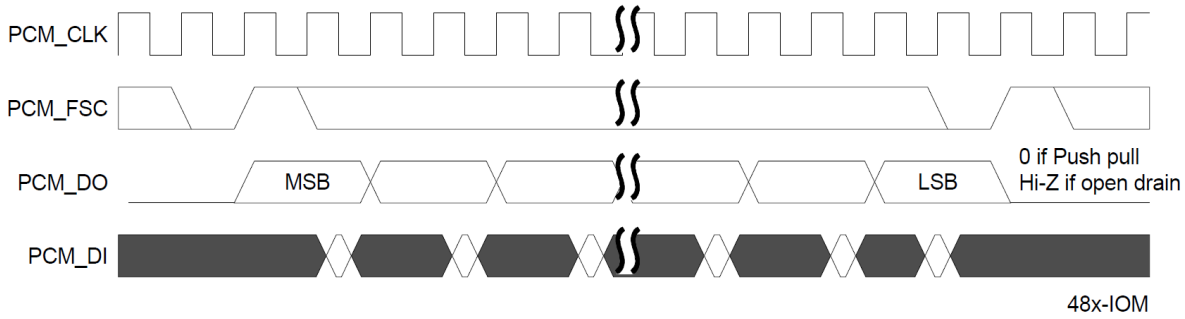


Figure 71: IOM format

24.2.4 PDM Interface

The PDM comprises 2 signals, namely the DATA and the CLK and supports stereo streams. PDM_DATA is encoded so that the left channel is clocked in on the falling edge of CLK and the right channel is clocked on the rising edge of PDM_CLK as shown in Figure 72.

The interface supports MEMS microphone sleep mode by disabling the PDM_CLK. The PDM interface signals can be mapped on any GPIO by programming PID=30 and PID=31 for DATA and CLK respectively in the Pxy_MODE_REG.

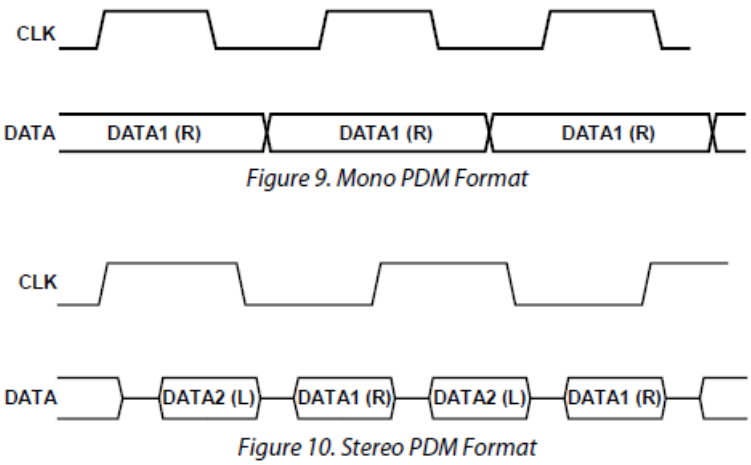


Figure 72: PDM Mono/Stereo formats

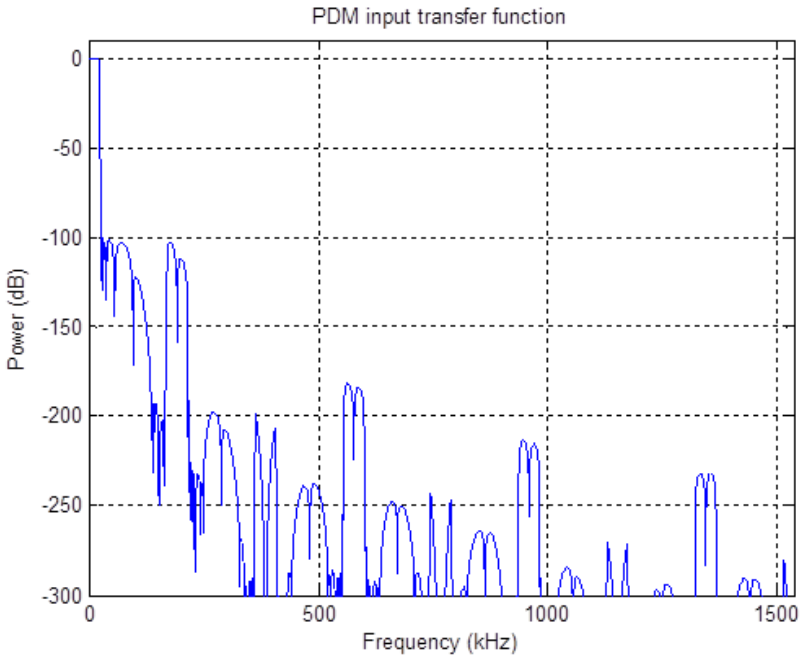


Figure 73: SRC PDM Input Transfer Function

It should be noted that the audio quality degrades when the oversampling ratio is less than 64. For an 8 kHz sample rate the minimum recommended PDM clock rate is 64 x 8 kHz = 512 kHz.

24.2.5 DMA Support

If more than one sample must be transfer to/from the CPU or the sample rate is so high that it interrupts the CPU too often, the DMA controller must be engaged to perform the transactions. Four channels are reserved in the DMA to support the PCM (IN), PCM (OUT), the SRC (IN) and the SRC (OUT) directions.

24.2.6 Interrupts

After a Sample Rate Conversion the input up-sampler and output down-sampler generate edge triggered interrupts on SRC_IN_SYNC and SRC_OUT_SYNC to the CPU which do not have to be cleared. Note that only one sample shall be read from or written to a single register at a time (i.e. there are no FIFOs included).

25 Power Management

The DA14585 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. The system diagram of the analog power block is presented in Figure 74.

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 two Alkaline batteries down to 1.8 V.
- Battery voltage measurement ADC (multiplexed input from general purpose ADC)
- Use of small external components (2.2 µH inductor and 1 µF capacitor)

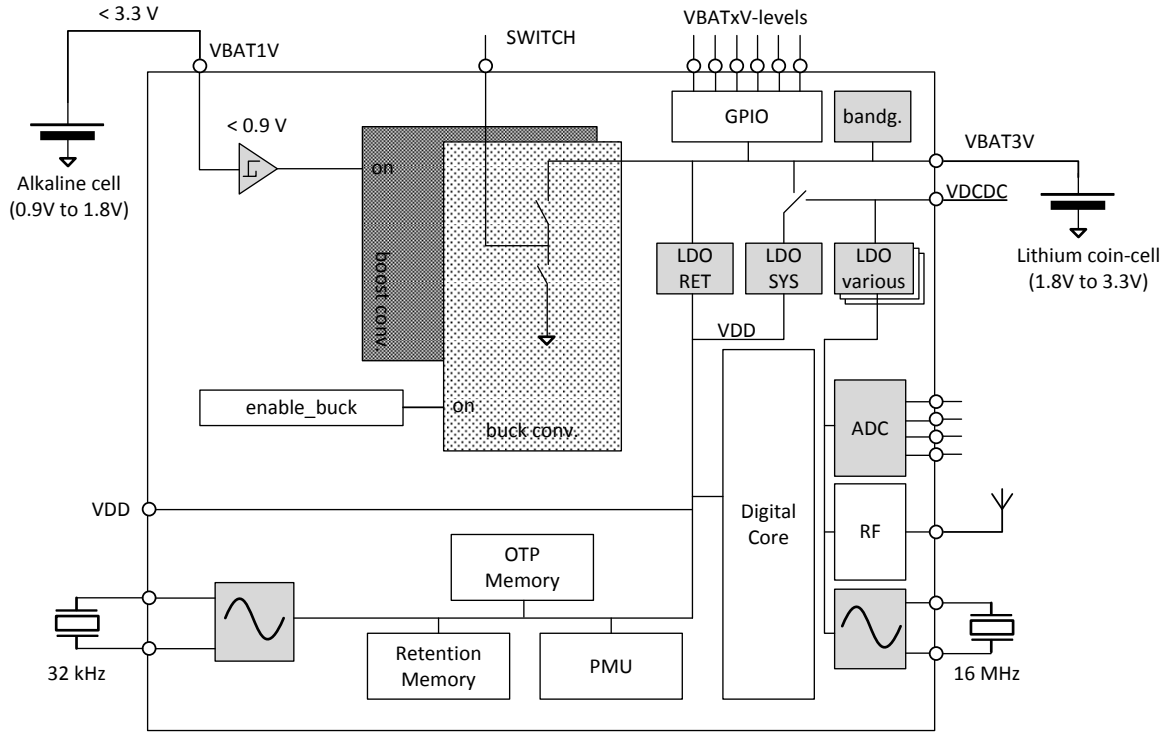


Figure 74: Block Diagram of the Analog Power Block and Internal Interconnections

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 SysRAMs when used as retention memory and the digital part which is always on.
2. 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 Extended or Deep Sleep mode. In the latter case, a low voltage is applied to the PD_AON power domain to further reduce leakage.
3. LDO (various): This is 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 DA14585. They depend on the specific battery cell used and its voltage range. Battery cells are distinguished into Lithium coin cells (1.8 V to 3.3 V) and Alkaline cells (1.0 V to 1.8 V). The connection diagrams are presented in Figure 75 and Figure 76 respectively:

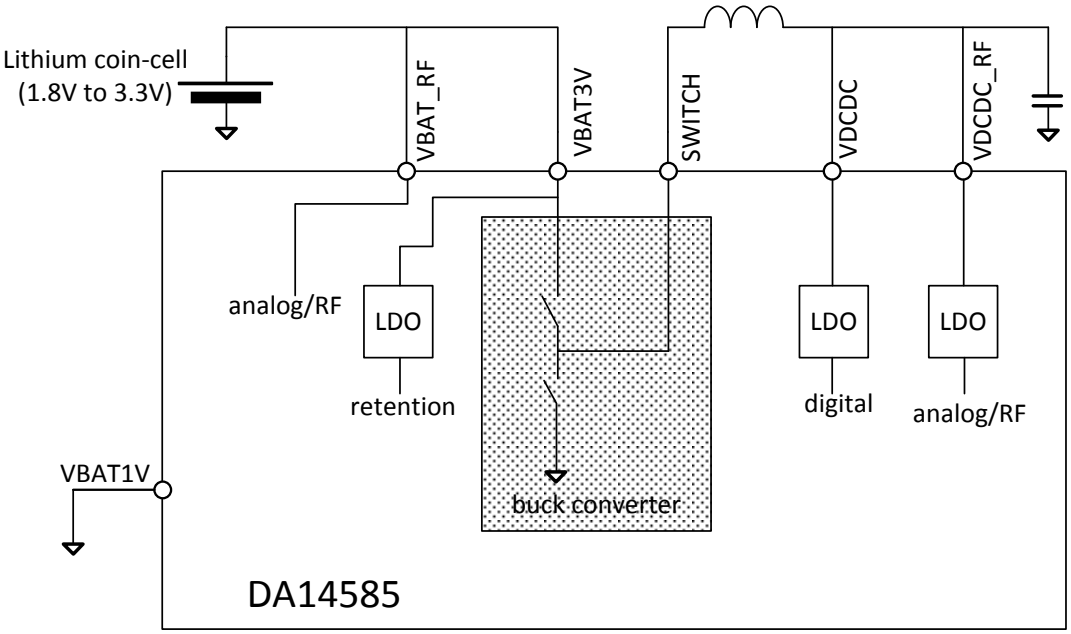


Figure 75: Supply Overview, Coin-Cell Application

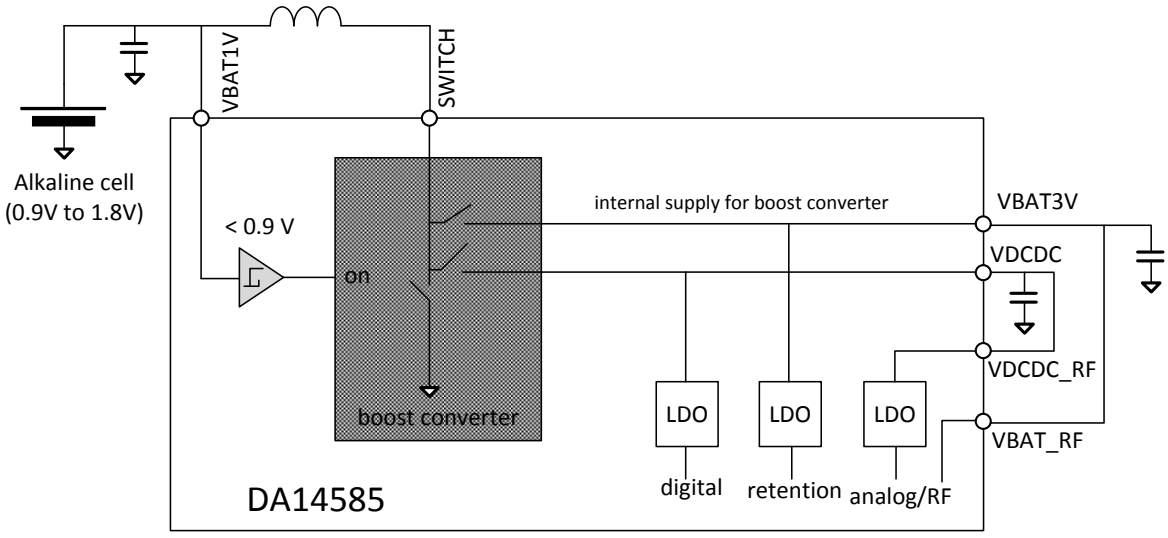


Figure 76: Supply Overview, Alkaline-Cell Application

The usage of Boost or Buck mode with respect to the provided voltage ranges is shown in Figure 77 which also illustrates the efficiency of the engine assuming a 10 mA constant load.

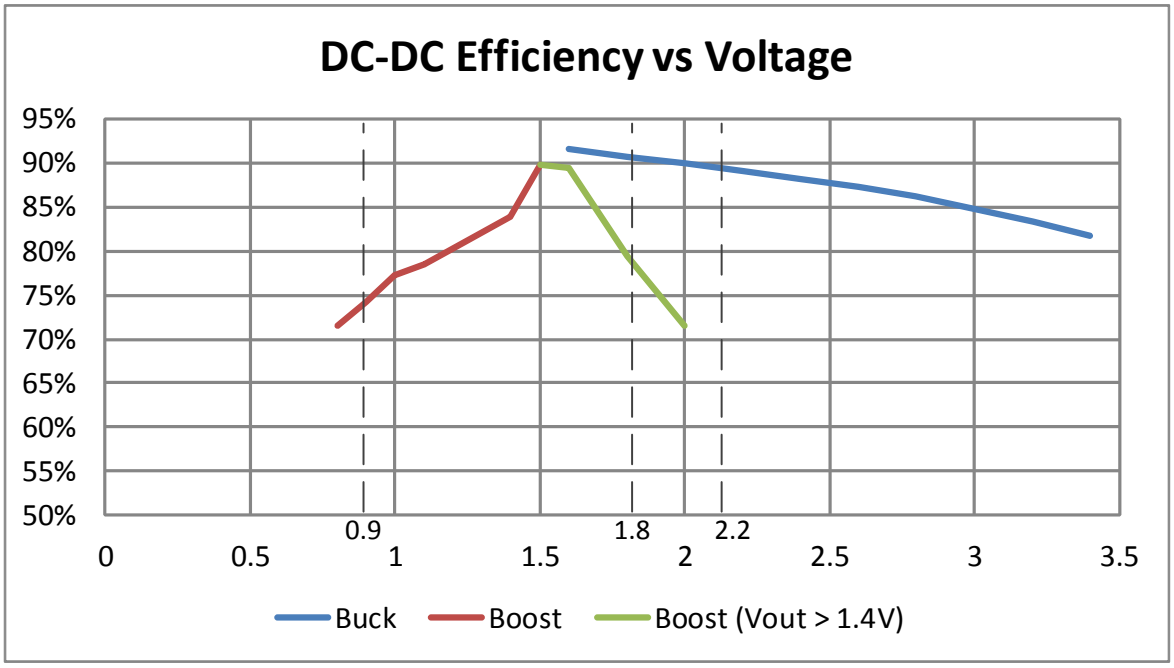


Figure 77: DC-DC Efficiency in Buck/Boost Mode at Various Voltage Levels

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 1.8 V to 3.3 V.

There are two voltage areas in Figure 77 designated by dashed lines. The first one (0 V to 0.9 V) indicates that the DA14585 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 (0 V to 1.8 V) indicates that Deep Sleep 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 (0.9 V to 1.8 V) 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.

26 BLE Core

The BLE (Bluetooth Low Energy) core is a qualified Bluetooth 5.0 baseband controller compatible with Bluetooth Smart specification and it is in charge of packet encoding/decoding and frame scheduling.

The block diagram is presented in Figure 78.

Features

- Bluetooth Smart Specifications compliant according to the Specification of the Bluetooth System, v5.0, Bluetooth SIG.
  - Dual Topology
  - Low duty cycle advertising
  - L2CAP connection oriented channels
- All device classes support (Broadcaster, Central, Observer, Peripheral)
- All packet types (Advertising / Data / Control)
- Dedicated Encryption (AES / CCM)
- Bit stream processing (CRC, Whitening)
- FDMA / TDMA / events formatting and synchronization
- Frequency Hopping calculation
- Operating clock 16 MHz or 8 MHz.
- Low power modes supporting 32.0 kHz, 32.768 kHz or 10 kHz
- Supports power down of the baseband during the protocol's idle periods.
- AHB Slave interface for register file access.
- AHB Slave interface for Exchange Memory access of CPU via BLE core.
- AHB Master interface for direct access of BLE core to Exchange Memory space

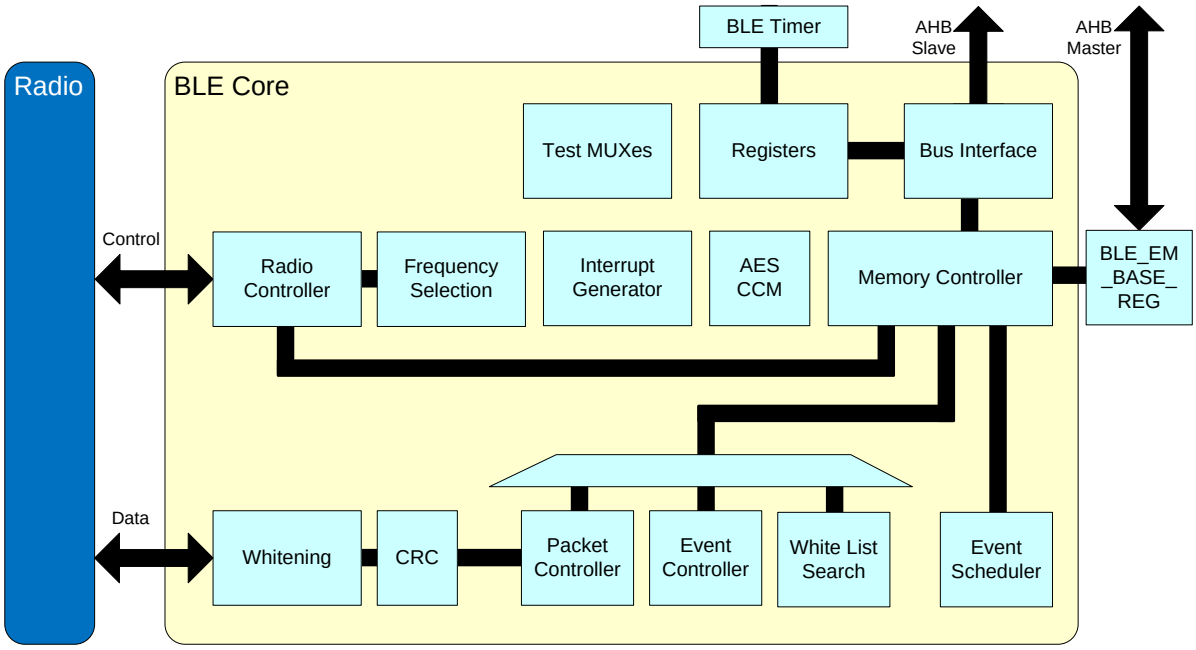


Figure 78: BLE Core Block Diagram

26.1 Architecture

26.1.1 Exchange Memory

The BLE Core requires access to a memory space named “Exchange Memory” to store control structures and frame buffers. The access to Exchange Memory is performed via the AHB Master interface. The base address of the Exchange Memory is programmable by means of the BLE_EM_BASE register. The maximum addressable size of the BLE core is 64 kB.

26.2 Programming

26.2.1 Wake-Up IRQ

Once BLE core switches to “BLE Deep Sleep mode” the only way to correctly exit from this state is by initially generating the BLE_WAKEUP_LP_IRQ and consecutively the BLE_SLP_IRQ. This sequence must be followed regardless of the cause of the termination of the “BLE Deep Sleep mode”, i.e. regardless if the BLE Timer expired or BLE Timer has been stopped due to the assertion of BLE_WAKEUP_REQ.

The assertion and de-assertion of BLE_WAKEUP_LP_IRQ is fully controlled via the BLE_ENBPRESET_REG bit fields. Detailed description is following:

TWIRQ_SET: Number of “ble_lp_clk” cycles before the expiration of the BLE Timer, when the BLE_WAKEUP_LP_IRQ must be asserted. It is recommended to select a TWIRQ_SET value larger than the time required for the XTAL 16MHz trimming to complete (refer to XTAL16_TRIM_READY), plus the IRQ Handler execution time. If the programmed value of TWIRQ_SET is less than the minimum recommended value, then the system will wake up but the actual BLE sleep duration (refer to BLE_DEEPSLSTAT_REG) will be larger than the programmed sleep duration (refer to BLE_DEEPSLWKUP_REG).

TWIRQ_RESET: Number of “ble_lp_clk” cycles before the expiration of the sleep period, when the BLE_WAKEUP_LP_IRQ will be de-asserted. It is recommended to always set to “1”.

TWEXT: Determines the high period of BLE_WAKEUP_LP_IRQ, in the case of an external wake up event (refer to GP_CONTROL_REG[BLE_WAKEUP_REQ]). Minimum value is "TWIRQ_RESET + X", where X is the number of “ble_lp_clk” clock cycles that BLE_WAKEUP_LP_IRQ will be held high. Recommended value is "TWIRQ_RESET + 1". Note that as soon as GP_CONTROL_REG[BLE_WAKEUP_REQ] is set to “1” the BLE_WAKEUP_LP_IRQ will be asserted.

Minimum BLE Sleep Duration: The minimum value of BLE_DEEPSLWKUP_REG[DEEPSLTIME] time, measured in “ble_lp_clk” cycles, is the maximum of (a) “TWIRQ_SET + 1” and (b) the SW execution time from setting BLE_DEEPSLCNTL_REG[DEEP_SLEEP_ON] up to preparing CPU to accept the BLE_WAKEUP_LP_IRQ (e.g. to call the ARM instruction WFI). If programmed DEEPSLTIME is less than the aforementioned minimum value, then BLE_WAKEUP_LP_IRQ Handler may execute sooner than the call of ARM WFI instruction for example, causing SW instability.

26.2.2 Switch from BLE Active Mode to BLE Deep Sleep Mode

Software can set the BLE core into the “BLE Deep Sleep mode”, by first programming the timing of BLE_WAKEUP_LP_IRQ generation, then program the desired sleep duration at BLE_DEEPSLWKUP_REG and finally set the register bit BLE_DEEPSLCNTL_REG[DEEP_SLEEP_ON].

The BLE Core will switch to the “ble_lp_clk” (32.0 kHz or 32.768 kHz) in order to maintain its internal 625 µs timing reference. Software must poll the state of BLE_CNTL2_REG[RADIO_PWRDN_ALLOW] to detect the completion of this mode transition. Once the “ble_lp_clk” is used for base time reference, SW must disable the BLE clocks (“ble_master1_clk”, “ble_master2_clk” and “ble_crypt_clk”) by setting to “0” the CLK_RADIO_REG[BLE_ENABLE] register bit.

Finally, SW can optionally power down the Radio Subsystem by using the PMU_CTRL_REG[RADIO_SLEEP] and the Peripheral and System power domains as well.

Figure 79 presents the waveforms while entering in BLE Deep Sleep mode. In this case, SW, as soon as it detects that RADIO_PWRDOWN_ALLOW is “1”, it sets the PMU_CTRL_REG[RADIO_SLEEP] to power down the Radio Subsystem. At the following figures, the corresponding BLE Core signals are marked with red while Radio Subsystem is in power down state.

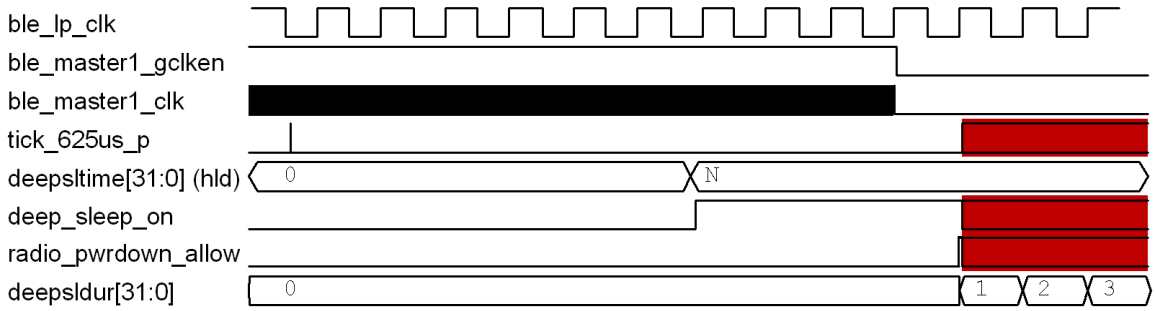


Figure 79: Entering into BLE Deep Sleep mode

26.2.3 Switch from BLE Deep Sleep Mode to BLE Active Mode

There are two possibilities for BLE Core to terminate the BLE Deep Sleep mode:

1. Termination at the end of a predetermined time.
2. Termination on software wake-up request, due to an external event.

26.2.4 Switching at Anchor Points

Figure 82 shows a typical BLE deep sleep phase that is terminated at predetermined time. After a configurable time before the scheduled wake up time (configured via BLE_ENBPRESET_REG register bit fields), the BLE Timer asserts the BLE_WAKEUP_LP_IRQ in order to wake-up the CPU (powering up the System Power Domain). The BLE_WAKEUP_LP_IRQ Interrupt Handler will prepare the code environment and the XTAL16M oscillator stabilization (refer to SYS_STAT_REG[XTAL16_SETTLED]) and will decide when the BLE Core will be ready to exit from the BLE Deep Sleep mode.

Once the SW decides that BLE Core can wake up, it must enable the BLE clocks (via CLK_RADIO_REG[BLE_ENABLE]) and power up the Radio Power Domain (refer to PMU_CTRL_REG[RADIO_SLEEP] and SYS_STAT_REG[RAD_IS_UP]).

After the expiration of the sleep period (as specified in BLE_DEEPSLWKUP_REG[DEEPSLTIME]) the BLE Timer will not exit the BLE Deep Sleep mode until it will detect that BLE Core is powered up. That means that if the SW requires more time to power up the BLE Core, then the final sleep duration (provided by BLE_DEEPSLSTAT_REG) will be larger than the preprogrammed value.

When BLE Timer is expired, BLE clocks are enabled and BLE Core (Radio Subsystem) is powered up, the BLE Core exists the “BLE Core Deep Sleep mode” and asserts the BLE_SLP_IRQ.

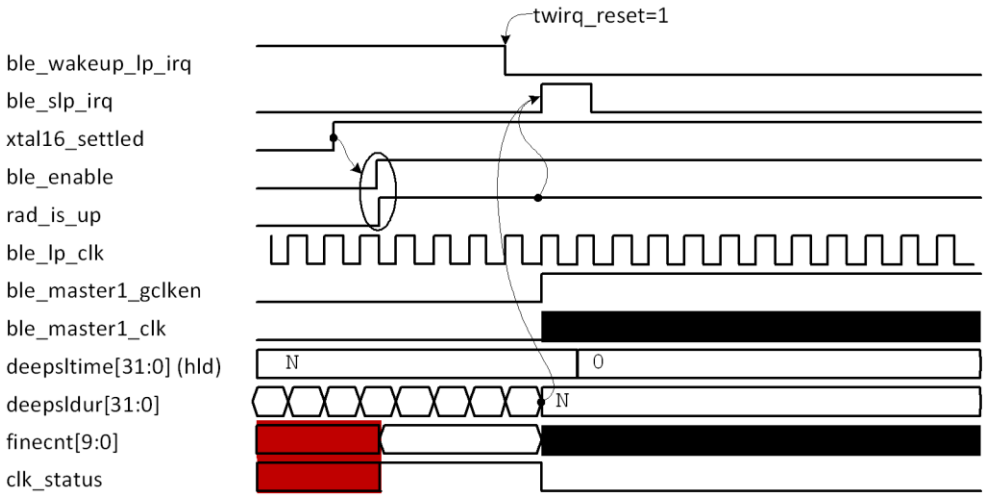


Figure 80: Exit BLE Deep Sleep Mode at Predetermined Time (Zoom In)

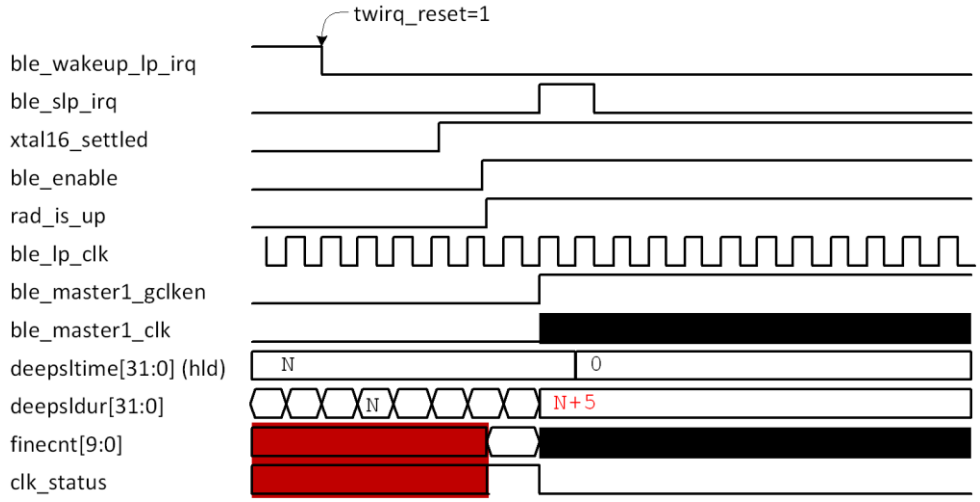


Figure 81: Exit BLE Deep Sleep Mode after Predetermined Time (Zoom In)

Figure 82: Exit BLE Deep Sleep Mode at Predetermined Time (Zoom Out)

26.2.5 Switching Due to an External Event

Figure 83 shows a wake up from a BLE deep sleep period forced by the assertion of register bit GP_CONTROL_REG[BLE_WAKEUP_REQ].

Assume that the system is in Extended Sleep state with all Power Domains switched off, and both the Wake-up Timer and Wake-up Controller programmed appropriately. Then assume that an event is detected at one of the GPIOs, causing the System Power Domain to wake-up due to WKUP_QUADDEC_IRQ. In that case, the SW will decide to wake-up the BLE core, then it should set to “1” the GP_CONTROL_REG[BLE_WAKEUP_REQ] in order to force the wake up sequence.

At Figure 83 the BLE_WAKEUP_REQ is by the software raised as soon as possible, causing BLE_WAKEUP_LP_IRQ Handler to be executed as soon as possible. It is also possible to raise BLE_WAKEUP_REQ after the detection of XTAL16_TRIM_READY, causing both BLE_WAKEUP_LP_IRQ and BLE_SLP_IRQ Handlers to execute sequentially. The decision depends on the software structure and the application.

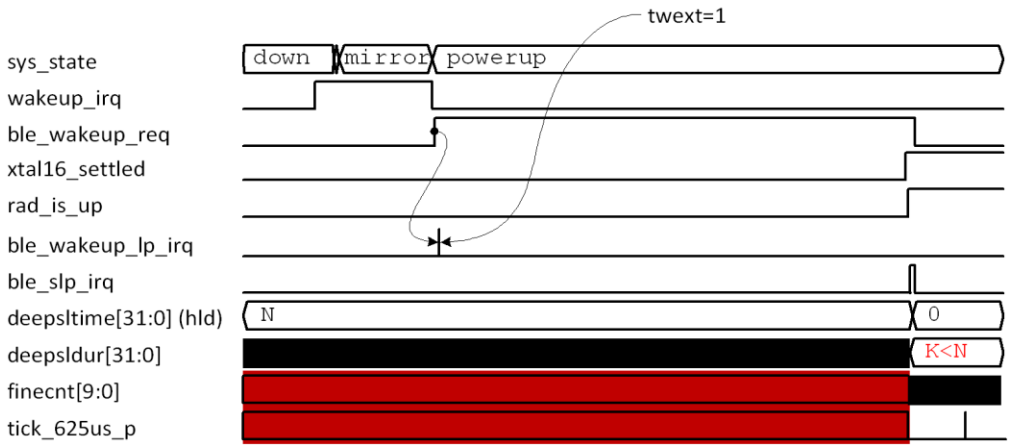


Figure 83: Exit BLE Deep Sleep Mode Due to External Event

As soon as bit field BLE_WAKEUP_REQ is set to “1” the BLE_WAKEUP_LP_IRQ will be asserted. In that case, the high period of BLE_WAKEUP_LP_IRQ is controlled via TWEXT. The recommended value of TWEXT is "TWIRQ_RESET + 1", meaning that BLE_WAKEUP_LP_IRQ will remain high for one “ble_llp_clk” period.

As long as the BLE_WAKEUP_REQ is high, entering the sleep mode is prohibited. Please note that BLE_WAKEUP_REQ event can be disabled by setting BLE_DEEPSLCNTL_REG[EXTWKUPDSB].

27 Radio

The Radio Transceiver implements the RF part of the Bluetooth LE protocol. Together with the Bluetooth 5.0 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 block diagram is given in Figure 84. It comprises the Receiver, Transmitter, Synthesizer, Rx/Tx combiner block, and Biasing LDOs.

Features

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

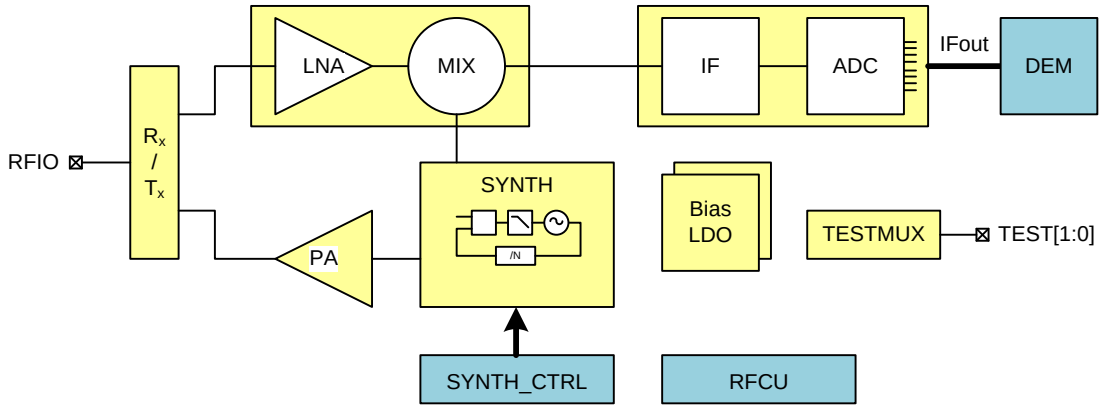


Figure 84: Bluetooth Radio block diagram

27.1 Receiver

The RX frontend consists of a selective matching network, a low noise amplifier (LNA) and an image rejection down conversion mixer. The LNA gain is controlled by the AGC.

The intermediate frequency (IF) part of the receiver comprises a complex filter and two variable gain amplifiers. This provides the necessary signal conditioning prior to digitalization. The digital demodulator block (DEM) provides a synchronous bit stream.

27.2 Synthesizer

The RF Synthesizer generates the quadrature LO signal for the mixer, but also generates the modulated TX output signal. The VCO runs at twice the required frequency and a dedicated divide-by-2 circuit generates the 2.4 GHz signals in the required phase relations. Its frequency is controlled by a classic 3rd order type II PLL with a passive loop filter, operated in fractional-*N* mode. The reference frequency is the 16 MHz crystal clock. The multi-modulus divider has a nominal divide ratio of 153 which is varied by a ΣΔ modulator. The modulation of the TX frequency is performed by 2-point modulation. The fractional divide ratio also contains the shaped TX data stream. A second modulation path feeds the TX data stream directly to the VCO. The latter path is automatically calibrated from time to time to align the low and high frequency parts of the 2-point modulation scheme.

27.3 Transmitter

The RF power amplifier (RFPA) is an extremely efficient Class-D structure, providing typically 0 dBm to the antenna. It is fed by the VCO's divide-by-2 circuit and delivers its TX power to the antenna pin through the combined RX/TX matching circuit.

27.4 RFIO

The RX/TX combiner block is a unique feature of the DA14585. It makes sure that the received power is applied to the LNA with minimum losses towards the RFPA. In TX mode, the LNA poses a minimal load for the RFPA and its input pins are protected from the RFPA. In both modes, the single ended RFIO port is matched to 50 Ω, in order to provide the simplest possible interfacing to the antenna on the printed circuit board.

27.5 Biasing

All RF blocks are supplied by on-chip low-drop out-regulators (LDOs). The bias scheme is programmable per block and optimized for minimum power consumption.

27.6 Control

The radio control unit (RFCU) controls the block timing and configuration registers. The BLE interfaces directly with the RFCU. The DA14585 can be put in test mode using a standard Bluetooth tester (e.g. Rohde & Schwarz CBT with K57 option) by connecting the antenna terminal for the RF link and the UART as described in section Direct Test Mode.

28 Registers

This section contains a detailed view of the DA14585 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).

28.1 Analog Miscellaneous Registers

Table 45: Register map ANAMISC

Address	Register	Description
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

Table 46: CLK_REF_SEL_REG (0x50001600)

Bit	Mode	Symbol	Description	Reset
15:3	R	-		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: 0x0 : RC32K oscillator 0x1 : RC16M oscillator 0x2 : XTAL32K oscillator 0x3 : RCX oscillator	0x0

Table 47: 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 48: 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 49: CLK_REF_VAL_H_REG (0x50001606)

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

28.2 Audio Unit Registers

Table 50: Register map APU

Address	Register	Description
0x50004000	SRC1_CTRL_REG	SRC1 control register
0x50004004	SRC1_IN_FS_REG	SRC1 Sample input rate
0x50004008	SRC1_OUT_FS_REG	SRC1 Sample output rate
0x5000400C	SRC1_IN1_REG	SRC1 data in 1
0x50004010	SRC1_IN2_REG	SRC1 data in 2
0x50004014	SRC1_OUT1_REG	SRC1 data out 1
0x50004018	SRC1_OUT2_REG	SRC1 data out 2
0x5000401C	APU_MUX_REG	APU mux register
0x50004020	COEF10_SET1_REG	SRC coefficient 1,0 set 1
0x50004024	COEF32_SET1_REG	SRC coefficient 3,2 set 1
0x50004028	COEF54_SET1_REG	SRC coefficient 5,4 set 1
0x5000402C	COEF76_SET1_REG	SRC coefficient 7,6 set 1
0x50004030	COEF98_SET1_REG	SRC coefficient 9,8 set 1
0x50004034	COEF0A_SET1_REG	SRC coefficient 10 set 1
0x50004100	PCM1_CTRL_REG	PCM1 Control register
0x50004104	PCM1_IN1_REG	PCM1 data in 1
0x50004108	PCM1_IN2_REG	PCM1 data in 2
0x5000410C	PCM1_OUT1_REG	PCM1 data out 1
0x50004110	PCM1_OUT2_REG	PCM1 data out 2

Table 51: SRC1_CTRL_REG (0x50004000)

Bit	Mode	Symbol	Description	Reset
31:30	R/W	SRC_PDM_DO_DEL	PDM_DO output delay line 0: no delay 1: 14 ns 2: 20 ns 3: 26 ns	0
29:28	R/W	SRC_PDM_MODE	PDM Output mode selection on PDM_DO1 00: No output 01: Right channel (falling edge of PDM_CLK) 10: Left channel (rising edge of PDM_CLK) 11: Left and Right channel	0
27:26	R/W	SRC_PDM_DI_DEL	PDM_DI input delay line 0: no delay 1: 6 ns 2: 12 ns 3: 18 ns	0
25	W	SRC_OUT_FLOWCLR	Writing a 1 clears the SRC1_OUT Overflow/underflow bits 23-22. No more over/underflow indications while bit is 1. Keep 1 until the over/under flow bit is cleared	0

Bit	Mode	Symbol	Description	Reset
24	W	SRC_IN_FLOWCLR	Writing a 1 clears the SRC1_IN Overflow/underflow bits 21-20. No more over/underflow indications while bit is 1. Keep 1 until the over/under flow bit is cleared	0
23	R	SRC_OUT_UNFLOW	1 = SRC1_OUT Underflow occurred	0
22	R	SRC_OUT_OVFLOW	1 = SRC1_OUT Overflow occurred	0
21	R	SRC_IN_UNFLOW	1 = SRC1_IN Underflow occurred	0
20	R	SRC_IN_OVFLOW	1 = SRC1_IN Overflow occurred	0
19	R0/W	SRC_RESYNC	1 = SRC will restart synchronisation	0
18	R	SRC_OUT_OK	SRC1_OUT Status 0: acquisition in progress 1: acquisition ready (In manual mode this bit is always 1)	0
17:16	R/W	SRC_OUT_US	SRC1_OUT UpSampling IIR filters setting 00: for sample rates up-to 48kHz 01: for sample rates of 96kHz 10: reserved 11: for sample rates of 192kHz	0
15	R	-		0
14	R/W	SRC_OUT_CAL_BYPASS	SRC1_OUT1 upsampiling filter bypass 0:Do not bypass 1:Bypass filter	0
13	R/W	SRC_OUT_AMODE	SRC1_OUT1 Automatic Conversion mode 0:Manual mode 1:Automatic mode	0
12:10	R	-		0
9:8	R/W	-		0
7	R/W	SRC_DITHER_DISABLE	Dithering feature 0: Enable 1: Disable	0
6	R	SRC_IN_OK	SRC1_IN status 0: Acquisition in progress 1: Acquisition ready	0
5:4	R/W	SRC_IN_DS	SRC1_IN UpSampling IIR filters setting 00: for sample rates up-to 48kHz 01: for sample rates of 96kHz 10: reserved 11: for sample rates of 192kHz	0
3	R	-		0
2	R/W	SRC_IN_CAL_BYPASS	SRC1_IN upsampeling filter bypass 0: Do not bypass 1: Bypass filter	0
1	R/W	SRC_IN_AMODE	SRC1_IN Automatic conversion mode 0: Manual mode 1: Automatic mode	0
0	R/W	SRC_EN	SRC1_IN and SRC1_OUT enable 0: disabled 1: enabled	0

Table 52: SRC1_IN_FS_REG (0x50004004)

Bit	Mode	Symbol	Description	Reset
31:24	-	-		0
23:0	R/W	SRC_IN_FS	SRC_IN Sample rate SRC_IN_FS = 8192*Sample_rate/100 Sample_rate upper limit is 192kHz. For 96kHz and 192kHz SRC_CTRLx_REG[SRC_IN_DS] must be set as shown below: Sample_rate SRC_IN_FS SRC_IN_DS Audio bandwidth 8000 Hz 0xA0000 0 4000 Hz 11025 Hz 0x0DC800 0 5512 Hz 16000 Hz 0x140000 0 8000 Hz 22050 Hz 0x1B9000 0 11025 Hz 32000 Hz 0x280000 0 16000 Hz 44100 Hz 0x372000 0 22050 Hz 48000 Hz 0x3C0000 0 24000 Hz 96000 Hz 0x3C0000 1 24000 Hz 192000 Hz 0x3C0000 3 24000 Hz In manual SRC mode, SRC_IN_FS can be set and adjusted to the desired sample rate at any time. In automatic mode the SRC returns the final sample rate as soon as SRC_IN_OK. Note that SRC_DS is not calculated in automatic mode and must be set manually automatic mode with Sample_rate of 96 and 192kHz.	0

Table 53: SRC1_OUT_FS_REG (0x50004008)

Bit	Mode	Symbol	Description	Reset
31:24	-	-		0

Bit	Mode	Symbol	Description	Reset
23:0	R/W	SRC_OUT_FS	SRC_OUT Sample rate SRC_OUT_FS = 8192*Sample_rate/100 Sample_rate upper limit is 192kHz. For 96kHz and 192kHz SRC_CTRLx_REG[SRC_DS] must be set as shown below: Sample_rate SRC_OUT_FS SRC_OUT_DS Audio bandwidth 8000 Hz 0xA0000 0 4000 Hz 11025 Hz 0x0DC800 0 5512 Hz 16000 Hz 0x140000 0 8000 Hz 22050 Hz 0x1B9000 0 11025 Hz 32000 Hz 0x280000 0 16000 Hz 44100 Hz 0x372000 0 22050 Hz 48000 Hz 0x3C0000 0 24000 Hz 96000 Hz 0x3C0000 1 24000 Hz 192000 Hz 0x3C0000 3 24000 Hz In manual SRC mode, SRC_OUT_FS can be set and adjusted to the desired sample rate at any time. In automatic mode the SRC returns the final sample rate as soon as SRC_OUT_OK. Note that SRC_DS is not calculated in automatic mode and must be set manually automatic mode with Sample_rate of 96 and 192kHz.	0

Table 54: SRC1_IN1_REG (0x5000400C)

Bit	Mode	Symbol	Description	Reset
31:8	R/W	SRC_IN	SRC1_IN1	0

Table 55: SRC1_IN2_REG (0x50004010)

Bit	Mode	Symbol	Description	Reset
31:8	R/W	SRC_IN	SRC1_IN2	0

Table 56: SRC1_OUT1_REG (0x50004014)

Bit	Mode	Symbol	Description	Reset
31:8	R	SRC_OUT	SRC1_OUT1	0

Table 57: SRC1_OUT2_REG (0x50004018)

Bit	Mode	Symbol	Description	Reset
31:8	R	SRC_OUT	SRC1_OUT2	0

Table 58: APU_MUX_REG (0x5000401C)

Bit	Mode	Symbol	Description	Reset
6	R/W	PDM1_MUX_IN	PDM1 input mux 0 = SRC1_MUX_IN 1 = PDM input	0x0
5:3	R/W	PCM1_MUX_IN	PCM1 input mux 0 = off 1 = SRC1 output 2 = PCM output registers	0x0
2:0	R/W	SRC1_MUX_IN	SRC1 input mux 0 = off 1 = PCM output 2 = SRC1 input registers	0x0

Table 59: COEF10_SET1_REG (0x50004020)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	SRC_COEF1	coefficient 1	0x79A9
15:0	R/W	SRC_COEF0	coefficient 0	0x9278

Table 60: COEF32_SET1_REG (0x50004024)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	SRC_COEF3	coefficient 3	0x6D56
15:0	R/W	SRC_COEF2	coefficient 2	0x8B41

Table 61: COEF54_SET1_REG (0x50004028)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	SRC_COEF5	coefficient 5	0x9BC5
15:0	R/W	SRC_COEF4	coefficient 4	0xBE15

Table 62: COEF76_SET1_REG (0x5000402C)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	SRC_COEF7	coefficient 7	0x8C28

Bit	Mode	Symbol	Description	Reset
15:0	R/W	SRC_COEF6	coefficient 6	0x7E1A

Table 63: COEF98_SET1_REG (0x50004030)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	SRC_COEF9	coefficient 9	0x92D7
15:0	R/W	SRC_COEF8	coefficient 8	0x75E6

Table 64: COEF0A_SET1_REG (0x50004034)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	SRC_COEF10	coefficient 10	0x41F2

Table 65: PCM1_CTRL_REG (0x50004100)

Bit	Mode	Symbol	Description	Reset
31:20	R/W	PCM_FSC_DIV	PCM Framesync divider, Values 7-0xFFF. To divide by N, write N-1. (Minimum value N-1=7 for 8 bits PCM_FSC) Note if PCM_CLK_BIT=1, N must always be even	0x0
19:17	R	-		0x0
16	R/W	PCM_FSC_EDGE	0: shift channels 1, 2, 3, 4, 5, 6, 7, 8 after PCM_FSC edge 1: shift channels 1, 2, 3, 4 after PCM_FSC edge shift channels 5, 6, 7, 8 after opposite PCM_FSC edge	0x0
15:11	R/W	PCM_CH_DEL	Channel delay in multiples of 8 bits	0x0
10	R/W	PCM_CLK_BIT	0:One clock cycle per data bit 1:Two cloc cycles per data bit	0x0
9	R/W	PCM_FSCINV	0: PCM FSC 1: PCM FSC inverted	0x0
8	R/W	PCM_CLKINV	0:PCM CLK 1:PCM CLK inverted	0x0
7	R/W	PCM_PPOD	0:PCM DO push pull 1:PCM DO open drain	0x0
6	R/W	PCM_FSCDEL	0:PCM FSC starts one cycle before MSB bit 1:PCM FSC starts at the same time as MSB bit	0x0
5:2	R/W	PCM_FSCLEN	0:PCM FSC length equal to 1 data bit N:PCM FSC length equal to N*8	0x0
1	R/W	PCM_MASTER	0:PCM interface in slave mode 1:PCM interface in master mode	0x0
0	R/W	PCM_EN	0:PCM interface disabled 1:PCM interface enabled	0x0

Table 66: PCM1_IN1_REG (0x50004104)

Bit	Mode	Symbol	Description	Reset
31:0	R	PCM_IN	PCM1_IN1 bits 31-0	0x0

Table 67: PCM1_IN2_REG (0x50004108)

Bit	Mode	Symbol	Description	Reset
31:0	R	PCM_IN	PCM1_IN2 bits 31-0	0x0

Table 68: PCM1_OUT1_REG (0x5000410C)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	PCM_OUT	PCM1_OUT1 bits 31-0	0xFFFF FFFF

Table 69: PCM1_OUT2_REG (0x50004110)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	PCM_OUT	PCM1_OUT2 bits 31-0	0xFFFF FFFF

28.3 BLE Core Registers

Table 70: Register map BLE

Address	Register	Description
0x40000000	BLE_RWBLECNTL_REG	BLE Control register
0x40000004	BLE_VERSION_REG	Version register
0x40000008	BLE_RWBLECONF_REG	Configuration register
0x4000000C	BLE_INTCNTL_REG	Interrupt controller register
0x40000010	BLE_INTSTAT_REG	Interrupt status register
0x40000014	BLE_INTRAWSTAT_REG	Interrupt raw status register
0x40000018	BLE_INTACK_REG	Interrupt acknowledge register
0x4000001C	BLE_BASETIMECNT_REG	Base time reference counter
0x40000020	BLE_FINETIMECNT_REG	Fine time reference counter
0x40000024	BLE_BDADDRL_REG	BLE device address LSB register
0x40000028	BLE_BDADDRU_REG	BLE device address MSB register
0x4000002C	BLE_CURRENTRXDESCPTR_REG	Rx Descriptor Pointer for the Receive Buffer Chained List
0x40000030	BLE_DEEPSLCNTL_REG	Deep-Sleep control register
0x40000034	BLE_DEEPSLWKUP_REG	Time (measured in Low Power clock cycles) in Deep Sleep Mode before waking-up the device
0x40000038	BLE_DEEPSLSTAT_REG	Duration of the last deep sleep phase register
0x4000003C	BLE_ENBPRESET_REG	Time in low power oscillator cycles register
0x40000040	BLE_FINECNTCORR_REG	Phase correction value register
0x40000044	BLE_BASETIMECNTCORR_REG	Base Time Counter
0x40000050	BLE_DIAGCNTL_REG	Diagnostics Register
0x40000054	BLE_DIAGSTAT_REG	Debug use only
0x40000058	BLE_DEBUGADDMAX_REG	Upper limit for the memory zone
0x4000005C	BLE_DEBUGADDMIN_REG	Lower limit for the memory zone
0x40000060	BLE_ERRORTYPESTAT_REG	Error Type Status registers
0x40000064	BLE_SWPROFILING_REG	Software Profiling register
0x40000070	BLE_RADIOCNTL0_REG	Radio interface control register

Address	Register	Description
0x40000074	BLE_RADIOCNTL1_REG	Radio interface control register
0x40000078	BLE_RADIOCNTL2_REG	Radio interface control register
0x4000007C	BLE_RADIOCNTL3_REG	Radio interface control register
0x40000080	BLE_RADIOPWRUPDN_REG	RX/TX power up/down phase register
0x40000090	BLE_ADVCHMAP_REG	Advertising Channel Map
0x400000A0	BLE_ADVTIM_REG	Advertising Packet Interval
0x400000A4	BLE_ACTSCANSTAT_REG	Active scan register
0x400000B0	BLE_WLPUBADDPTR_REG	Start address of public devices list
0x400000B4	BLE_WLPRIVADDPTR_REG	Start address of private devices list
0x400000B8	BLE_WLNBDEV_REG	Devices in white list
0x400000C0	BLE_AESCNTL_REG	Start AES register
0x400000C4	BLE_AESKEY31_0_REG	AES encryption key
0x400000C8	BLE_AESKEY63_32_REG	AES encryption key
0x400000CC	BLE_AESKEY95_64_REG	AES encryption key
0x400000D0	BLE_AESKEY127_96_REG	AES encryption key
0x400000D4	BLE_AESPTR_REG	Pointer to the block to encrypt/decrypt
0x400000D8	BLE_TXMICVAL_REG	AES / CCM plain MIC value
0x400000DC	BLE_RXMICVAL_REG	AES / CCM plain MIC value
0x400000E0	BLE_RFTESTCNTL_REG	RF Testing Register
0x400000E4	BLE_RFTESTTXSTAT_REG	RF Testing Register
0x400000E8	BLE_RFTESTRXSTAT_REG	RF Testing Register
0x400000F0	BLE_TIMGENCNTL_REG	Timing Generator Register
0x400000F4	BLE_GROSSTIMTGT_REG	Gross Timer Target value
0x400000F8	BLE_FINETIMTGT_REG	Fine Timer Target value
0x400000FC	BLE_SAMPLECLK_REG	Samples the Base Time Counter
0x40000100	BLE_COEXIFCNTL0_REG	Coexistence interface Control 0 Register

Address	Register	Description
0x40000104	BLE_COEXIFCNTL1_REG	Coexistence interface Control 1 Register
0x40000108	BLE_BLEMPRIO0_REG	Coexistence interface Priority 0 Register
0x4000010C	BLE_BLEMPRIO1_REG	Coexistence interface Priority 1 Register
0x40000200	BLE_CNTL2_REG	BLE Control Register 2
0x40000208	BLE_EM_BASE_REG	Exchange Memory Base Register
0x4000020C	BLE_DIAGCNTL2_REG	Debug use only
0x40000210	BLE_DIAGCNTL3_REG	Debug use only

Table 71: BLE_RWBLECNTL_REG (0x40000000)

Bit	Mode	Symbol	Description	Reset
31	R0/W	MASTER_SOFT_RST	Reset the complete BLE Core except registers and timing generator, when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.	0x0
30	R0/W	MASTER_TGSOFT_RST	Reset the timing generator, when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.	0x0
29	R/W	REG_SOFT_RST	Reset the complete register block, when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0. Note that INT STAT will not be cleared, so the user should also write to BLE_INTACK_REG after the SW Reset	0x0
28	R0/W	SWINT_REQ	Forces the generation of ble_sw_irq when written with a 1, and proper masking is set. Resets at 0 when action is performed. No action happens if it is written with 0.	0x0
26	R0/W	RFTEST_ABORT	Abort the current RF Testing defined as per CS-FORMAT when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0. Note that when RFTEST_ABORT is requested: 1) In case of infinite Tx, the Packet Controller FSM stops at the end of the current byte in process, and processes accordingly the packet CRC. 2) In case of Infinite Rx, the Packet Controller FSM either stops as the end of the current Packet reception (if Access address has been detected), or simply stop the processing switching off the RF.	0x0
25	R0/W	ADVERT_ABORT	Abort the current Advertising event when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.	0x0
24	R0/W	SCAN_ABORT	Abort the current scan window when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.	0x0

Bit	Mode	Symbol	Description	Reset
22	R/W	MD_DSB	value forced by SW from Tx Descriptor value just saved in Rx Descriptor during reception	0x0
21	R/W	SN_DSB	value forced by SW from Tx Descriptor value ignored in Rx, where no SN error reported.	0x0
20	R/W	NESN_DSB	value forced by SW from Tx Descriptor value ignored in Rx, where no NESN error reported.	0x0
19	R/W	CRYPT_DSB	0: Normal operation. Encryption / Decryption enabled. 1: Encryption / Decryption disabled. Note that if CS-CRYPT_EN is set, then MIC is generated, and only data encryption is disabled, meaning data sent are plain data.	0x0
18	R/W	WHIT_DSB	0: Normal operation. Whitening enabled. 1: Whitening disabled.	0x0
17	R/W	CRC_DSB	0: Normal operation. CRC removed from data stream. 1: CRC stripping disabled on Rx packets, CRC replaced by 0x000 in Tx.	0x0
16	R/W	HOP_REMAP_DSB	0: Normal operation. Frequency Hopping Remapping algorithm enabled. 1: Frequency Hopping Remapping algorithm disabled	0x0
13:12	R/W	-		0x0
9	R/W	ADVERTFILT_EN	Advertising Channels Error Filtering Enable control 0: RW-BLE Core reports all errors to RW-BLE Software 1: RW-BLE Core reports only correctly received packet, without error to RW-BLE Software	0x0
8	R/W	RWBLE_EN	0: Disable RW-BLE Core Exchange Table pre-fetch mechanism. 1: Enable RW-BLE Core Exchange table pre-fetch mechanism.	0x0
7:4	R/W	RXWINSZDEF	Default Rx Window size in us. Used when device: is master connected performs its second receipt. 0 is not a valid value. Recommended value is 10 (in decimal).	0x0
2:0	R/W	SYNCERR	Indicates the maximum number of errors allowed to recognize the synchronization word.	0x0

Table 72: BLE_VERSION_REG (0x40000004)

Bit	Mode	Symbol	Description	Reset
31:24	R	TYP	BLE Core Type	0x7

Bit	Mode	Symbol	Description	Reset
23:16	R	REL	BLE Core version Major release number.	0x1
15:8	R	UPG	BLE Core upgrade Upgrade number.	0x0
7:0	R	BUILD	BLE Core Build Build number.	0x0

Table 73: BLE_RWBLECONF_REG (0x40000008)

Bit	Mode	Symbol	Description	Reset
29:24	R	ADD_WIDTH	Value of the RW_BLE_ADDRESS_WIDTH parameter concerted into binary.	0x10
22:16	R	RFIF	Radio Interface ID	0x2
13:8	R	CLK_SEL	Operating Frequency (in MHz)	0x0
6	R	DECIPHER	0: AES deciphering not present	0x0
5	R	DMMODE	0: RW-BLE Core is used as a standalone BLE device	0x0
4	R	INTMODE	1: Interrupts are trigger level generated, i.e. stays active at 1 till acknowledgement	0x1
3	R	COEX	1: WLAN Coexistence mechanism present	0x1
2	R	USEDBG	1: Diagnostic port instantiated	0x1
1	R	USECRYPT	1: AES-CCM Encryption block present	0x1
0	R	BUSWIDTH	Processor bus width: 1: 32 bits	0x1

Table 74: BLE_INTCNTL_REG (0x4000000C)

Bit	Mode	Symbol	Description	Reset
15	R/W	CSCNTDEVMSK	CSCNT interrupt mask during event. This bit allows to enable CSCNT interrupt generation during events (i.e. advertising, scanning, initiating, and connection) 0: CSCNT Interrupt not generated during events. 1: CSCNT Interrupt generated during events.	0x1
14:10	R	-		0x0
9	R/W	SWINTMSK	SW triggered interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x0
8	R/W	EVENTAPFAINTMSK	End of event / anticipated pre-fetch abort interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x1
7	R/W	FINETGTIMINTMSK	Fine Target Timer Mask 0: Interrupt not generated 1: Interrupt generated	0x0
6	R/W	GROSSTGTIMINTMSK	Gross Target Timer Mask 0: Interrupt not generated 1: Interrupt generated	0x0

Bit	Mode	Symbol	Description	Reset
5	R/W	ERRORINTMSK	Error Interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x0
4	R/W	CRYPTINTMSK	Encryption engine Interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x1
3	R/W	EVENTINTMSK	End of event Interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x1
2	R/W	SLPINTMSK	Sleep Mode Interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x1
1	R/W	RXINTMSK	Rx Interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x1
0	R/W	CSCNTINTMSK	625us Base Time Interrupt Mask 0: Interrupt not generated 1: Interrupt generated	0x1

Table 75: BLE_INTSTAT_REG (0x40000010)

Bit	Mode	Symbol	Description	Reset
9	R	SWINTSTAT	SW triggered interrupt status 0: No SW triggered interrupt. 1: A SW triggered interrupt is pending	0x0
8	R	EVENTAPFAINTSTAT	End of event / Anticipated Pre-Fetch Abort interrupt status 0: No End of Event interrupt. 1: An End of Event interrupt is pending.	0x0
7	R	FINETGTIMINTSTAT	Masked Fine Target Timer Error interrupt status 0: No Fine Target Timer interrupt. 1: A Fine Target Timer interrupt is pending.	0x0
6	R	GROSSTGTIMINTSTAT	Masked Gross Target Timer interrupt status 0: No Gross Target Timer interrupt. 1: A Gross Target Timer interrupt is pending.	0x0
5	R	ERRORINTSTAT	Masked Error interrupt status 0: No Error interrupt. 1: An Error interrupt is pending.	0x0
4	R	CRYPTINTSTAT	Masked Encryption engine interrupt status 0: No Encryption / Decryption interrupt. 1: An Encryption / Decryption interrupt is pending.	0x0
3	R	EVENTINTSTAT	Masked End of Event interrupt status 0: No End of Advertising / Scanning / Connection interrupt. 1: An End of Advertising / Scanning / Connection interrupt is pending.	0x0
2	R	SLPINTSTAT	Masked Sleep interrupt status 0: No End of Sleep Mode interrupt. 1: An End of Sleep Mode interrupt is pending.	0x0

Bit	Mode	Symbol	Description	Reset
1	R	RXINTSTAT	Masked Packet Reception interrupt status 0: No Rx interrupt. 1: An Rx interrupt is pending.	0x0
0	R	CSCNTINTSTAT	Masked 625us base time reference interrupt status 0: No 625us Base Time interrupt. 1: A 625us Base Time interrupt is pending.	0x0

Table 76: BLE_INTRAWSTAT_REG (0x40000014)

Bit	Mode	Symbol	Description	Reset
9	R	SWINTRAWSTAT	SW triggered interrupt raw status 0: No SW triggered interrupt. 1: A SW triggered interrupt is pending.	0x0
8	R	EVENTAPFAINTRAWSTAT	End of event / Anticipated Pre-Fetch Abort interrupt raw status 0: No End of Event interrupt. 1: An End of Event interrupt is pending.	0x0
7	R	FINETGTIMINTRAWSTAT	Fine Target Timer Error interrupt raw status 0: No Fine Target Timer interrupt. 1: A Fine Target Timer interrupt is pending.	0x0
6	R	GROSSTGTIMINTRAWSTAT	Gross Target Timer interrupt raw status 0: No Gross Target Timer interrupt. 1: A Gross Target Timer interrupt is pending.	0x0
5	R	ERRORINTRAWSTAT	Error interrupt raw status 0: No Error interrupt. 1: An Error interrupt is pending.	0x0
4	R	CRYPTINTRAWSTAT	Encryption engine interrupt raw status 0: No Encryption / Decryption interrupt. 1: An Encryption / Decryption interrupt is pending.	0x0
3	R	EVENTINTRAWSTAT	End of Event interrupt raw status 0: No End of Advertising / Scanning / Connection interrupt. 1: An End of Advertising / Scanning / Connection interrupt is pending.	0x0
2	R	SLPINTRAWSTAT	Sleep interrupt raw status 0: No End of Sleep Mode interrupt. 1: An End of Sleep Mode interrupt is pending.	0x0
1	R	RXINTRAWSTAT	Packet Reception interrupt raw status 0: No Rx interrupt. 1: An Rx interrupt is pending.	0x0
0	R	CSCNTINTRAWSTAT	625us base time reference interrupt raw status 0: No 625us Base Time interrupt. 1: A 625us Base Time interrupt is pending.	0x0

Table 77: BLE_INTACK_REG (0x40000018)

Bit	Mode	Symbol	Description	Reset
9	R0/W	SWINTACK	SW triggered interrupt acknowledgement bit Software writing 1 acknowledges the SW triggered interrupt. This bit resets SWINTSTAT and SWINTRAWSTAT flags. Resets at 0 when action is performed	0x0
8	R0/W	EVENTAPFAINTACK	End of event / Anticipated Pre-Fetch Abort interrupt acknowledgement bit Software writing 1 acknowledges the End of event / Anticipated Pre-Fetch Abort interrupt. This bit resets EVENTAPFAINTSTAT and EVENTAPFAINTRAWSTAT flags. Resets at 0 when action is performed	0x0
7	R0/W	FINETGTIMINTACK	Fine Target Timer interrupt acknowledgement bit Software writing 1 acknowledges the Fine Timer interrupt. This bit resets FINETGTIMINTSTAT and FINETGTIMINTRAWSTAT flags. Resets at 0 when action is performed	0x0
6	R0/W	GROSSTGTIMINTACK	Gross Target Timer interrupt acknowledgement bit Software writing 1 acknowledges the Gross Timer interrupt. This bit resets GROSSTGTIMINTSTAT and GROSSTGTIMINTRAWSTAT flags. Resets at 0 when action is performed	0x0
5	R0/W	ERRORINTACK	Error interrupt acknowledgement bit Software writing 1 acknowledges the Error interrupt. This bit resets ERRORINTSTAT and ERRORINTRAWSTAT flags. Resets at 0 when action is performed	0x0
4	R0/W	CRYPTINTACK	Encryption engine interrupt acknowledgement bit Software writing 1 acknowledges the Encryption engine interrupt. This bit resets CRYPTINTSTAT and CRYPTINTRAWSTAT flags. Resets at 0 when action is performed	0x0
3	R0/W	EVENTINTACK	End of Event interrupt acknowledgment bit Software writing 1 acknowledges the End of Advertising / Scanning / Connection interrupt. This bit resets SLPINTSTAT and SLPINTRAWSTAT flags. Resets at 0 when action is performed	0x0
2	R0/W	SLPINTACK	End of Deep Sleep interrupt acknowledgment bit Software writing 1 acknowledges the End of Sleep Mode interrupt. This bit resets SLPINTSTAT and SLPINTRAWSTAT flags. Resets at 0 when action is performed	0x0
1	R0/W	RXINTACK	Packet Reception interrupt acknowledgment bit Software writing 1 acknowledges the Rx interrupt. This bit resets RXINTSTAT and RXINTRAWSTAT flags. Resets at 0 when action is performed	0x0
0	R0/W	CSCNTINTACK	625us base time reference interrupt acknowledgment bit Software writing 1 acknowledges the CLKN interrupt. This bit resets CLKINTSTAT and CLKINTRAWSTAT flags. Resets at 0 when action is performed	0x0

Table 78: BLE_BASETIMECNT_REG (0x4000001C)

Bit	Mode	Symbol	Description	Reset
26:0	R	BASETIMECNT	Value of the 625us base time reference counter. Updated each time SAMPCLK is written. Used by the SW in order to synchronize with the HW	0x0

Table 79: BLE_FINETIMECNT_REG (0x40000020)

Bit	Mode	Symbol	Description	Reset
9:0	R	FINECNT	Value of the current s fine time reference counter. Updated each time SAMPCLK is written. Used by the SW in order to synchronize with the HW, and obtain a more precise sleep duration	0x0

Table 80: BLE_BDADDRL_REG (0x40000024)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	BDADDRL	Bluetooth Low Energy Device Address. LSB part.	0x0

Table 81: BLE_BDADDRU_REG (0x40000028)

Bit	Mode	Symbol	Description	Reset
16	R/W	PRIV_NPUB	Bluetooth Low Energy Device Address privacy indicator 0: Public Bluetooth Device Address 1: Private Bluetooth Device Address	0x0
15:0	R/W	BDADDRU	Bluetooth Low Energy Device Address. MSB part.	0x0

Table 82: BLE_CURRENTRXDESCPTR_REG (0x4000002C)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	ETPTR	Exchange Table Pointer that determines the starting point of the Exchange Table	0x0
14:0	R/W	CURRENTRXDESCPTR	Rx Descriptor Pointer that determines the starting point of the Receive Buffer Chained List	0x0

Table 83: BLE_DEEPSLCNTL_REG (0x40000030)

Bit	Mode	Symbol	Description	Reset
31	R/W	EXTWKUPDSB	External Wake-Up disable 0: RW-BLE Core can be woken by external wake-up 1: RW-BLE Core cannot be woken up by external wake-up	0x0
15	R	DEEP_SLEEP_STAT	Indicator of current Deep Sleep clock mux status: 0: RW-BLE Core is not yet in Deep Sleep Mode 1: RW-BLE Core is in Deep Sleep Mode (only low_power_clk is running)	0x0
4	R/W	SOFT_WAKEUP_REQ	Wake Up Request from RW-BLE Software. Applies when system is in Deep Sleep Mode. It wakes up the RW-BLE Core when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.	0x0
3	R0/W	DEEP_SLEEP_CORR_EN	625us base time reference integer and fractional part correction. Applies when system has been woken-up from Deep Sleep Mode. It enables Fine Counter and Base Time counter when written with a 1. Resets at 0 when action is performed. No action happens if it is written with 0.	0x0
2	R0/W	DEEP_SLEEP_ON	0: RW-BLE Core in normal active mode 1: Request RW-BLE Core to switch in deep sleep mode. This bit is reset on DEEP_SLEEP_STAT falling edge.	0x0
1:0	R/W	DEEP_SLEEP_IRQ_EN	Always set to "3" when DEEP_SLEEP_ON is set to "1". It controls the generation of BLE_WAKEUP_LP_IRQ.	0x0

Table 84: BLE_DEEPSLWKUP_REG (0x40000034)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	DEEPSLTIME	Determines the time in low_power_clk clock cycles to spend in Deep Sleep Mode before waking-up the device. This ensures a maximum of 37 hours and 16mn sleep mode capabilities at 32kHz. This ensures a maximum of 36 hours and 16mn sleep mode capabilities at 32.768kHz	0x0

Table 85: BLE_DEEPSLSTAT_REG (0x40000038)

Bit	Mode	Symbol	Description	Reset
31:0	R	DEEPSLDUR	Actual duration of the last deep sleep phase measured in low_power_clk clock cycle. DEEPSLDUR is set to zero at the beginning of the deep sleep phase, and is incremented at each low_power_clk clock cycle until the end of the deep sleep phase.	0x0

Table 86: BLE_ENBPRESET_REG (0x4000003C)

Bit	Mode	Symbol	Description	Reset
31:21	R/W	TWEXT	Minimum and recommended value is "TWIRQ_RESET + 1". In the case of wake-up due to an external wake-up request, TWEXT specifies the time delay in low power oscillator cycles to deassert BLE_WAKEUP_LP_IRQ. Refer also to GP_CONTROL_REG[BLE_WAKEUP_REQ]. Range is [0...64 ms] for 32kHz; [0...62.5 ms] for 32.768kHz	0x0
20:10	R/W	TWIRQ_SET	Minimum value is "TWIRQ_RESET + 1". Time in low power oscillator cycles to set BLE_WAKEUP_LP_IRQ before the BLE sleep timer expiration. Refer also to BLE_DEEPSLWKUP_REG[DEEPSLTIME]. Range is [0...64 ms] for 32kHz; [0...62.5 ms] for 32.768kHz	0x0
9:0	R/W	TWIRQ_RESET	Recommended value is 1. Time in low power oscillator cycles to reset BLE_WAKEUP_LP_IRQ before the BLE sleep timer expiration. Refer also to BLE_DEEPSLWKUP_REG[DEEPSLTIME]. Range is [0...32 ms] for 32kHz; [0...31.25 ms] for 32.768kHz.	0x0

Table 87: BLE_FINECNTCORR_REG (0x40000040)

Bit	Mode	Symbol	Description	Reset
9:0	R/W	FINECNTCORR	Phase correction value for the 625us reference counter (i.e. Fine Counter) in us.	0x0

Table 88: BLE_BASETIMECNTCORR_REG (0x40000044)

Bit	Mode	Symbol	Description	Reset
26:0	R/W	BASETIMECNTCOR R	Base Time Counter correction value.	0x0

Table 89: BLE_DIAGCNTL_REG (0x40000050)

Bit	Mode	Symbol	Description	Reset
31	R/W	DIAG3_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0

Bit	Mode	Symbol	Description	Reset
29:24	R/W	DIAG3	Only relevant when DIAG3_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG3.	0x0
23	R/W	DIAG2_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
21:16	R/W	DIAG2	Only relevant when DIAG2_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG2.	0x0
15	R/W	DIAG1_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
13:8	R/W	DIAG1	Only relevant when DIAG1_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG1.	0x0
7	R/W	DIAG0_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
5:0	R/W	DIAG0	Only relevant when DIAG0_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG0.	0x0

Table 90: BLE_DIAGSTAT_REG (0x40000054)

Bit	Mode	Symbol	Description	Reset
31:24	R	DIAG3STAT	Directly connected to ble_dbg3[7:0] output. Debug use only.	0x0
23:16	R	DIAG2STAT	Directly connected to ble_dbg2[7:0] output. Debug use only.	0x0
15:8	R	DIAG1STAT	Directly connected to ble_dbg1[7:0] output. Debug use only.	0x0
7:0	R	DIAG0STAT	Directly connected to ble_dbg0[7:0] output. Debug use only.	0x0

Table 91: BLE_DEBUGADDMAX_REG (0x40000058)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	REG_ADDMAX	Upper limit for the Register zone indicated by the reg_inzone flag	0x0
15:0	R/W	EM_ADDMAX	Upper limit for the Exchange Memory zone indicated by the em_inzone flag	0x0

Table 92: BLE_DEBUGADDMIN_REG (0x4000005C)

Bit	Mode	Symbol	Description	Reset
31:16	R/W	REG_ADDMIN	Lower limit for the Register zone indicated by the reg_inzone flag	0x0
15:0	R/W	EM_ADDMIN	Lower limit for the Exchange Memory zone indicated by the em_inzone flag	0x0

Table 93: BLE_ERRORTYPESTAT_REG (0x40000060)

Bit	Mode	Symbol	Description	Reset
17	R	CONCEVTIRQ_ERROR	Indicates whether two consecutive and concurrent ble_event_irq have been generated, and not acknowledged in time by the RW-BLE Software. 0: No error 1: Error occurred	0x0
16	R	RXDATA_PTR_ERROR	Indicates whether Rx data buffer pointer value programmed is null: this is a major programming failure. 0: No error 1: Error occurred	0x0
15	R	TXDATA_PTR_ERROR	Indicates whether Tx data buffer pointer value programmed is null during Advertising / Scanning / Initiating events, or during Master / Slave connections with non-null packet length: this is a major programming failure. 0: No error 1: Error occurred	0x0
14	R	RXDESC_EMPTY_ERROR	Indicates whether Rx Descriptor pointer value programmed in register is null: this is a major programming failure. 0: No error 1: Error occurred	0x0
13	R	TXDESC_EMPTY_ERROR	Indicates whether Tx Descriptor pointer value programmed in Control Structure is null during Advertising / Scanning / Initiating events: this is a major programming failure. 0: No error 1: Error occurred	0x0
12	R	CSFORMAT_ERROR	Indicates whether CS-FORMAT has been programmed with an invalid value: this is a major software programming failure. 0: No error 1: Error occurred	0x0
11	R	LLCHMAP_ERROR	Indicates Link Layer Channel Map error, happens when actual number of CS-LLCHMAP bit set to one is different from CS-NBCHGOOD at the beginning of Frequency Hopping process 0: No error 1: Error occurred	0x0
10	R	ADV_UNDERRUN	Indicates Advertising Interval Under run, occurs if time between two consecutive Advertising packet (in Advertising mode) is lower than the expected value. 0: No error 1: Error occurred	0x0

Bit	Mode	Symbol	Description	Reset
9	R	IFS_UNDERRUN	Indicates Inter Frame Space Under run, occurs if IFS time is not enough to update and read Control Structure/Descriptors, and/or White List parsing is not finished and/or Decryption time is too long to be finished on time 0: No error 1: Error occurred	0x0
8	R	WHITELIST_ERROR	Indicates White List Timeout error, occurs if White List parsing is not finished on time 0: No error 1: Error occurred	0x0
7	R	EVT_CNTL_APFM_ERROR	Indicates Anticipated Pre-Fetch Mechanism error: happens when 2 consecutive events are programmed, and when the first event is not completely finished while second pre-fetch instant is reached. 0: No error 1: Error occured	0x0
6	R	EVT_SCHDL_APFM_ERROR	Indicates Anticipated Pre-Fetch Mechanism error: happens when 2 consecutive events are programmed, and when the first event is not completely finished while second pre-fetch instant is reached. 0: No error 1: Error occured	0x0
5	R	EVT_SCHDL_ENTRY_ERROR	Indicates Event Scheduler faced Invalid timing programing on two consecutive ET entries (e.g first one with 624s offset and second one with no offset) 0: No error 1: Error occurred	0x0
4	R	EVT_SCHDL_EMACC_ERROR	Indicates Event Scheduler Exchange Memory access error, happens when Exchange Memory accesses are not served in time, and blocks the Exchange Table entry read 0: No error 1: Error occurred	0x0
3	R	RADIO_EMACC_ERROR	Indicates Radio Controller Exchange Memory access error, happens when Exchange Memory accesses are not served in time and data are corrupted. 0: No error 1: Error occurred	0x0
2	R	PKTCNTL_EMACC_ERROR	Indicates Packet Controller Exchange Memory access error, happens when Exchange Memory accesses are not served in time and Tx/Rx data are corrupted 0: No error 1: Error occurred	0x0
1	R	RXCRYPT_ERROR	Indicates real time decryption error, happens when AES-CCM decryption is too slow compared to Packet Controller requests. A 16-bytes block has to be decrypted prior the next block is received by the Packet Controller 0: No error 1: Error occurred	0x0

Bit	Mode	Symbol	Description	Reset
0	R	TXCRYPT_ERROR	Indicates Real Time encryption error, happens when AES-CCM encryption is too slow compared to Packet Controller requests. A 16-bytes block has to be encrypted and prepared on Packet Controller request, and needs to be ready before the Packet Controller has to send ti 0: No error 1: Error occurred	0x0

Table 94: BLE_SWPROFILING_REG (0x40000064)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	SWPROFVAL	Software Profiling register: used by RW-BLE Software for profiling purpose: this value is copied on Diagnostic port	0x0

Table 95: BLE_RADIOCNTL0_REG (0x40000070)

Bit	Mode	Symbol	Description	Reset
31:24	-	-		0x0
23:18	R/W	-		0x0
17:7	-	-		0x0
6:5	R/W	-		0x0
4:2	-	-		0x0
1	R/W	-		0x1
0	R0/W	-		0x0

Table 96: BLE_RADIOCNTL1_REG (0x40000074)

Bit	Mode	Symbol	Description	Reset
31:21	-	-		0x0
20:16	R/W	XRFSEL	Extended radio selection field, Must be set to "2".	0x0

Table 97: BLE_RADIOCNTL2_REG (0x40000078)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	-		0x0

Table 98: BLE_RADIOCNTL3_REG (0x4000007C)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	-		0x40

Table 99: BLE_RADIOPWRUPDN_REG (0x40000080)

Bit	Mode	Symbol	Description	Reset
30:24	R/W	RTRIP_DELAY	Defines round trip delay value. This value correspond to the addition of data latency in Tx and data latency in Rx. Value is in us	0x0
23:16	R/W	RXPWRUP	This register holds the length in s of the RX power up phase for the current radio device. Default value is 210us (reset value). Operating range depends on the selected radio.	0xD2
11:8	R/W	TXPWRDN	This register extends the length in s of the TX power down phase for the current radio device. Default value is 3us (reset value). Operating range depends on the selected radio.	0x3
7:0	R/W	TXPWRUP	This register holds the length in s of the TX power up phase for the current radio device. Default value is 210us (reset value). Operating range depends on the selected radio.	0xD2

Table 100: BLE_ADVCHMAP_REG (0x40000090)

Bit	Mode	Symbol	Description	Reset
2:0	R/W	ADVCHMAP	Advertising Channel Map, defined as per the advertising connection settings. Contains advertising channels index 37 to 39. If ADVCHMAP[i] equals: 0: Do not use data channel i+37. 1: Use data channel i+37.	0x7

Table 101: BLE_ADVTIM_REG (0x400000A0)

Bit	Mode	Symbol	Description	Reset
13:0	R/W	ADVINT	Advertising Packet Interval defines the time interval in between two ADV_xxx packet sent. Value is in us. Value to program depends on the used Advertising Packet type and the device filtering policy.	0x0

Table 102: BLE_ACTSCANSTAT_REG (0x400000A4)

Bit	Mode	Symbol	Description	Reset
24:16	R	BACKOFF	Active scan mode back-off counter initialization value.	0x1

Bit	Mode	Symbol	Description	Reset
8:0	R	UPPERLIMIT	Active scan mode upper limit counter value.	0x1

Table 103: BLE_WLPUBADDPTR_REG (0x400000B0)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	WLPUBADDPTR	Start address pointer of the public devices white list.	0x0

Table 104: BLE_WLPRIVADDPTR_REG (0x400000B4)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	WLPRIVADDPTR	Start address pointer of the private devices white list.	0x0

Table 105: BLE_WLNBDEV_REG (0x400000B8)

Bit	Mode	Symbol	Description	Reset
15:8	R/W	NBPRIVDEV	Number of private devices in the white list.	0x0
7:0	R/W	NBPUBDEV	Number of public devices in the white list.	0x0

Table 106: BLE_AESCNTL_REG (0x400000C0)

Bit	Mode	Symbol	Description	Reset
1	R/W	AES_MODE	0: Cipher mode 1: Decipher mode	0x0
0	R0/W	AES_START	Writing a 1 starts AES-128 ciphering/deciphering process. This bit is reset once the process is finished (i.e. ble_crypt_irq interrupt occurs, even masked)	0x0

Table 107: BLE_AESKEY31_0_REG (0x400000C4)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	AESKEY31_0	AES encryption 128-bit key. Bit 31 down to 0	0x0

Table 108: BLE_AESKEY63_32_REG (0x400000C8)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	AESKEY63_32	AES encryption 128-bit key. Bit 63 down to 32	0x0

Table 109: BLE_AESKEY95_64_REG (0x400000CC)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	AESKEY95_64	AES encryption 128-bit key. Bit 95 down to 64	0x0

Table 110: BLE_AESKEY127_96_REG (0x400000D0)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	AESKEY127_96	AES encryption 128-bit key. Bit 127 down to 96	0x0

Table 111: BLE_AESPTR_REG (0x400000D4)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	AESPTR	Pointer to the memory zone where the block to cipher/decipher using AES-128 is stored.	0x0

Table 112: BLE_TXMICVAL_REG (0x400000D8)

Bit	Mode	Symbol	Description	Reset
31:0	R	TXMICVAL	AES-CCM plain MIC value. Valid on when MIC has been calculated (in Tx)	0x0

Table 113: BLE_RXMICVAL_REG (0x400000DC)

Bit	Mode	Symbol	Description	Reset
31:0	R	RXMICVAL	AES-CCM plain MIC value. Valid on once MIC has been extracted from Rx packet.	0x0

Table 114: BLE_RFTESTCNTL_REG (0x400000E0)

Bit	Mode	Symbol	Description	Reset
31	R/W	INFINITERX	Applicable in RF Test Mode only 0: Normal mode of operation 1: Infinite Rx window	0x0
27	R/W	RXPKTCNTEN	Applicable in RF Test Mode only 0: Rx packet count disabled 1: Rx packet count enabled, and reported in CS-RXCCMPKTCNT and RFTESTRXSTAT-RXPKTCNT on RF abort command	0x0
15	R/W	INFINITETX	Applicable in RF Test Mode only 0: Normal mode of operation. 1: Infinite Tx packet / Normal start of a packet but endless payload	0x0

Bit	Mode	Symbol	Description	Reset
14	R/W	TXLENGTHSRC	Applicable only in Tx/Rx RF Test mode 0: Normal mode of operation: TxDESC-TXADVLEN controls the Tx packet payload size 1: Uses RFTESTCNTL-TXLENGTH packet length (can support up to 512 bytes transmit)	0x0
13	R/W	PRBSTYPE	Applicable only in Tx/Rx RF Test mode 0: Tx Packet Payload are PRBS9 type 1: Tx Packet Payload are PRBS15 type	0x0
12	R/W	TXPLDSRC	Applicable only in Tx/Rx RF Test mode 0: Tx Packet Payload source is the Control Structure 1: Tx Packet Payload are PRBS generator	0x0
11	R/W	TXPKTCNTEN	Applicable in RF Test Mode only 0: Tx packet count disabled 1: Tx packet count enabled, and reported in CS-TXCCMPKTCNT and RFTESTTXSTAT-TXPKTCNT on RF abort command	0x0
8:0	R/W	TXLENGTH	Applicable only for Tx/Rx RF Test mode, and valid when RFTESTCNTL-TXLENGTHSRC = 1 Tx packet length in number of byte	0x0

Table 115: BLE_RFTESTTXSTAT_REG (0x400000E4)

Bit	Mode	Symbol	Description	Reset
31:0	R	TXPKTCNT	Reports number of transmitted packet during Test Modes. Value is valid if RFTESTCNTL-TXPKTCNTEN is set	0x0

Table 116: BLE_RFTESTRXSTAT_REG (0x400000E8)

Bit	Mode	Symbol	Description	Reset
31:0	R	RXPKTCNT	Reports number of correctly received packet during Test Modes (no sync error, no CRC error). Value is valid if RFTESTCNTL-RXPKTCNTEN is set	0x0

Table 117: BLE_TIMGENCNTL_REG (0x400000F0)

Bit	Mode	Symbol	Description	Reset
31	R/W	APFM_EN	Controls the Anticipated pre-Fetch Abort mechanism 0: Disabled 1: Enabled	0x1
25:16	R/W	PREFETCHABORT_TIME	Defines the instant in s at which immediate abort is required after anticipated pre-fetch abort	0x1FE
8:0	R/W	PREFETCH_TIME	Defines Exchange Table pre-fetch instant in us	0x96

Table 118: BLE_GROSSTIMTGT_REG (0x400000F4)

Bit	Mode	Symbol	Description	Reset
22:0	R/W	GROSSTARGET	Gross Timer Target value on which a ble_grosstgtim_irq must be generated. This timer has a precision of 10ms: interrupt is generated only when GROSSTARGET[22:0] = BASETIMECNT[26:4] and BASETIMECNT[3:0] = 0.	0x0

Table 119: BLE_FINETIMTGT_REG (0x400000F8)

Bit	Mode	Symbol	Description	Reset
26:0	R/W	FINETARGET	Fine Timer Target value on which a ble_finetgtim_irq must be generated. This timer has a precision of 625us: interrupt is generated only when FINETARGET = BASETIMECNT	0x0

Table 120: BLE_SAMPLECLK_REG (0x400000FC)

Bit	Mode	Symbol	Description	Reset
31:1	-	-		0x0
0	R0/W	SAMP	Writing a 1 samples the Base Time Counter value in BASETIMECNT register. Resets at 0 when action is performed.	0x0

Table 121: BLE_COEXIFCNTL0_REG (0x40000100)

Bit	Mode	Symbol	Description	Reset
21:20	R/W	WLCRXPRIOMODE	Defines Bluetooth Low Energy packet ble_rx mode behavior. 00: Rx indication excluding Rx Power up delay (starts when correlator is enabled) 01: Rx indication including Rx Power up delay 10: Rx High priority indicator 11: n/a	0x0
17:16	R/W	WLCTXPRIOMODE	Defines Bluetooth Low Energy packet ble_tx mode behavior 00: Tx indication excluding Tx Power up delay 01: Tx indication including Tx Power up delay 10: Tx High priority indicator 11: n/a	0x0
7:6	R/W	WLANTXMSK	Determines how wlan_tx impact BLE Tx and Rx 00: wlan_tx has no impact (default mode) 01: wlan_tx can stop BLE Tx, no impact on BLE Rx 10: wlan_tx can stop BLE Rx, no impact on BLE Tx 11: wlan_tx can stop both BLE Tx and BLE Rx	0x0

Bit	Mode	Symbol	Description	Reset
5:4	R/W	WLANRXMSK	Determines how wlan_rx impact BLE Tx and Rx 00: wlan_rx has no impact 01: wlan_rx can stop BLE Tx, no impact on BLE Rx (default mode) 10: wlan_rx can stop BLE Rx, no impact on BLE Tx 11: wlan_rx can stop both BLE Tx and BLE Rx	0x1
1	R/W	SYNCGEN_EN	Determines whether ble_sync is generated or not. 0: ble_sync pulse not generated 1: ble_sync pulse generated	0x0
0	R/W	COEX_EN	Enable / Disable control of the MWS/WLAN Coexistence control 0: Coexistence interface disabled 1: Coexistence interface enabled	0x0

Table 122: BLE_COEXIFCNTL1_REG (0x40000104)

Bit	Mode	Symbol	Description	Reset
28:24	R/W	WLCPRXTHR	Applies on ble_rx if WLCRXPRIOMODE equals 10 Determines the threshold for Rx priority setting. If ble_pti[3:0] output value is greater than WLCPRXTHR, then Rx Bluetooth Low Energy priority is considered as high, and must be provided to the WLAN coexistence interface	0x0
20:16	R/W	WLCPTXTHR	Applies on ble_tx if WLCTXPRIOMODE equals 10 Determines the threshold for priority setting. If ble_pti[3:0] output value is greater than WLCPTXTHR, then Tx Bluetooth Low Energy priority is considered as high, and must be provided to the WLAN coexistence interface	0x0
14:8	R/W	WLCPDURATION	Applies on ble_tx if WLCTXPRIOMODE equals 10 Applies on ble_rx if WLCRXPRIOMODE equals 10 Determines how many s the priority information must be maintained Note that if WLCPDURATION = 0x00, then Tx/Rx priority levels are maintained till Tx/Rx EN are de-asserted.	0x0
6:0	R/W	WLCPDELAY	Applies on ble_tx if WLCTXPRIOMODE equals 10. Applies on ble_rx if WLCRXPRIOMODE equals 10. Determines the delay (in us) in Tx/Rx enables rises the time Bluetooth Low energy Tx/Rx priority has to be provided .	0x0

Table 123: BLE_BLEMPRIO0_REG (0x40000108)

Bit	Mode	Symbol	Description	Reset
31:28	R/W	BLEM7	Set Priority value for Passive Scanning	0x3
27:24	R/W	BLEM6	Set Priority value for Non-Connectable Advertising	0x4
23:20	R/W	BLEM5	Set Priority value for Connectable Advertising BLE message	0x8
19:16	R/W	BLEM4	Set Priority value for Active Scanning BLE message	0x9

Bit	Mode	Symbol	Description	Reset
15:12	R/W	BLEM3	Set Priority value for Initiating (Scanning) BLE message	0xA
11:8	R/W	BLEM2	Set Priority value for Data Channel transmission BLE message	0xD
7:4	R/W	BLEM1	Set Priority value for LLCP BLE message	0xE
3:0	R/W	BLEM0	Set Priority value for Initiating (Connection Request Response) BLE message	0xF

Table 124: BLE_BLEMPRIO1_REG (0x4000010C)

Bit	Mode	Symbol	Description	Reset
31:28	R/W	BLEMDEFAULT	Set default priority value for other BLE message than those defined above	0x3

Table 125: BLE_CNTL2_REG (0x40000200)

Bit	Mode	Symbol	Description	Reset
31:22	R	-		0x0
21	R/W	BLE_RSSI_SEL	0: Select Peak-hold RSSI value (default). 1: Select current Average RSSI value.	0x0
20	R	WAKEUPLPSTAT	The status of the BLE_WAKEUP_LP_IRQ. The Interrupt Service Routine of BLE_WAKEUP_LP_IRQ should return only when the WAKEUPLPSTAT is cleared. Note that BLE_WAKEUP_LP_IRQ is automatically acknowledged after the power up of the Radio Subsystem, plus one Low Power Clock period.	0x0
19	R/W	SW_RPL_SPI	Keep to 0.	0x0
18	R/W	BB_ONLY	Keep to 0.	0x0
17	R/W	BLE_PTI_SOURCE_SEL	0: Provide to COEX block the PTI value indicated by the Control Structure. Recommended value is "0". 1: Provide to COEX block the PTI value generated dynamically by the BLE core, which is based on the PTI of the Control Structure.	0x0
16:15	R	-		0x0

Bit	Mode	Symbol	Description	Reset
14:9	R/W	BLE_CLK_SEL	BLE Clock Select. Specifies the BLE master clock absolute frequency in MHz. Typical values are 16 and 8. Value depends on the selected XTAL frequency and the value of CLK_RADIO_REG[BLE_DIV] bitfield. For example, if XTAL oscillates at 16MHz and CLK_RADIO_REG[BLE_DIV] = 1 (divide by 2), then BLE master clock frequency is 8MHz and BLE_CLK_SEL should be set to value 8. The selected BLE master clock frequency (affected by BLE_DIV and BLE_CLK_SEL) must be modified and set only during the initialization time, i.e. before setting BLE_RWBLECNTL_REG[RWBLE_EN] to 1. Refer also to BLE_RWBLECONF_REG[CLK_SEL].	0x0
8	R	RADIO_PWRDN_ALLOW	This active high signal indicates when it is allowed for the BLE core (embedded in the Radio sub-System power domain) to be powered down. After the assertion of the BLE_DEEPSLCNTL_REG[DEEP_SLEEP_ON] a hardware sequence based on the Low Power clock will cause the assertion of RADIO_PWRDN_ALLOW. The RADIO_PWRDN_ALLOW will be cleared to "0" when the BLE core exits from the sleep state, i.e. when the BLE_SLP_IRQ will be asserted.	0x0
7	R	MON_LP_CLK	The SW can only write a "0" to this bit. Whenever a positive edge of the low power clock used by the BLE Timers is detected, then the HW will automatically set this bit to "1". This functionality will not work if BLE Timer is in reset state (refer to CLK_RADIO_REG[BLE_LP_RESET]). This bit can be used for SW synchronization, to debug the low power clock, etc.	0x0
6	R	BLE_CLK_STAT	0: BLE uses low power clock 1: BLE uses master clock	0x0
5:4	R/W	-		0x0
3	R/W	BLE_DIAG_OVR	1: Overrule BLE_DIAG. 0: BLE_DIAG is not overruled.	0x0
2	R/W	EMACCERRMSK	Exchange Memory Access Error Mask: When cleared to "0" the EM_ACC_ERR will not cause an BLE_ERROR_IRQ interrupt. When set to "1" an BLE_ERROR_IRQ will be generated as long as EM_ACC_ERR is "1".	0x1
1	R0/W	EMACCERRACK	Exchange Memory Access Error Acknowledge. When the SW writes a "1" to this bit then the EMACCERRSTAT bit will be cleared. When the SW writes "0" it will have no affect. The read value is always "0".	0x0
0	R	EMACCERRSTAT	Exchange Memory Access Error Status: The bit is read-only and can be cleared only by writing a "1" at EMACCERRACK bitfield. This bit will be set to "1" by the hardware when the controller will access an EM page that is not mapped according to the EM_MAPPING value. When this bit is "1" then the BLE_ERROR_IRQ will be asserted as long as EMACCERRMSK is "1".	0x0

Table 126: BLE_EM_BASE_REG (0x40000208)

Bit	Mode	Symbol	Description	Reset
31:17	R	-		0x0
16:10	R/W	BLE_EM_BASE_16_10	The physical address on the system memory map of the base of the Exchange Memory.	0x0
9:0	R	-		0x0

Table 127: BLE_DIAGCNTL2_REG (0x4000020C)

Bit	Mode	Symbol	Description	Reset
31	R/W	DIAG7_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
30	R	-		0x0
29:24	R/W	DIAG7	Only relevant when DIAG7_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG7.	0x0
23	R/W	DIAG6_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
22	R	-		0x0
21:16	R/W	DIAG6	Only relevant when DIAG6_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG6.	0x0
15	R/W	DIAG5_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
14	R	-		0x0
13:8	R/W	DIAG5	Only relevant when DIAG5_EN= 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG5.	0x0
7	R/W	DIAG4_EN	0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
6	R	-		0x0
5:0	R/W	DIAG4	Only relevant when DIAG4_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG4.	0x0

Table 128: BLE_DIAGCNTL3_REG (0x40000210)

Bit	Mode	Symbol	Description	Reset
31	R/W	DIAG7_INV	If set, then the specific diagnostic bit will be inverted.	0x0

Bit	Mode	Symbol	Description	Reset
30:28	R/W	DIAG7_BIT	Selects which bit from the DIAG7 word will be forwarded to bit 7 of the BLE DIagnostic Port.	0x0
27	R/W	DIAG6_INV	If set, then the specific diagnostic bit will be inverted.	0x0
26:24	R/W	DIAG6_BIT	Selects which bit from the DIAG6 word will be forwarded to bit 6 of the BLE DIagnostic Port.	0x0
23	R/W	DIAG5_INV	If set, then the specific diagnostic bit will be inverted.	0x0
22:20	R/W	DIAG5_BIT	Selects which bit from the DIAG5 word will be forwarded to bit 5 of the BLE DIagnostic Port.	0x0
19	R/W	DIAG4_INV	If set, then the specific diagnostic bit will be inverted.	0x0
18:16	R/W	DIAG4_BIT	Selects which bit from the DIAG4 word will be forwarded to bit 4 of the BLE DIagnostic Port.	0x0
15	R/W	DIAG3_INV	If set, then the specific diagnostic bit will be inverted.	0x0
14:12	R/W	DIAG3_BIT	Selects which bit from the DIAG3 word will be forwarded to bit 3 of the BLE DIagnostic Port.	0x0
11	R/W	DIAG2_INV	If set, then the specific diagnostic bit will be inverted.	0x0
10:8	R/W	DIAG2_BIT	Selects which bit from the DIAG2 word will be forwarded to bit 2 of the BLE DIagnostic Port.	0x0
7	R/W	DIAG1_INV	If set, then the specific diagnostic bit will be inverted.	0x0
6:4	R/W	DIAG1_BIT	Selects which bit from the DIAG1 word will be forwarded to bit 1 of the BLE DIagnostic Port.	0x0
3	R/W	DIAG0_INV	If set, then the specific diagnostic bit will be inverted.	0x0
2:0	R/W	DIAG0_BIT	Selects which bit from the DIAG0 word will be forwarded to bit 0 of the BLE DIagnostic Port.	0x0

28.4 Clock Generation and Reset Registers

Table 129: Register map CRG

Address	Register	Description
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	RCX-oscillator control register
0x50000028	BANDGAP_REG	Bandgap trimming
0x5000002A	ANA_STATUS_REG	Status bit of analog (power management) circuits
0x50000040	POR_PIN_REG	Selects a GPIO pin for POR generation
0x50000042	POR_TIMER_REG	Time for POR to happen
0x50001C40	PCM_DIV_REG	PCM divider and enables
0x50001C42	PCM_FDIV_REG	PCM fractional division register
0x50001C44	PDM_DIV_REG	PDM divider and enables
0x50001C46	SRC_DIV_REG	SRC divider and enables

Table 130: CLK_AMBA_REG (0x50000000)

Bit	Mode	Symbol	Description	Reset
7	R/W	OTP_ENABLE	Clock enable for OTP controller	0x0
6	R/W	-		0x0
5:4	R/W	PCLK_DIV	APB interface clock (PCLK). Divider is cascaded with HCLK_DIV. PCLK is HCLK divided by: 0x0: divide by 1 0x1: divide by 2 0x2: divide by 4 0x3: divide by 8	0x0
3:2	R/W	-		0x0
1:0	R/W	HCLK_DIV	AHB interface and microprocessor clock (HCLK). HCLK is source clock divided by: 0x0: divide by 1 0x1: divide by 2 0x2: divide by 4 0x3: divide by 8	0x0

Table 131: CLK_FREQ_TRIM_REG (0x50000002)

Bit	Mode	Symbol	Description	Reset
15:11	-	-		0x0
10:8	R/W	COARSE_ADJ	Xtal frequency course trimming register. 0x0: lowest frequency 0x7: highest frequencyIncrement 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. 0x00: lowest frequency 0xFF: highest frequency	0x0

Table 132: CLK_PER_REG (0x50000004)

Bit	Mode	Symbol	Description	Reset
15	R/W	QUAD_ENABLE	Enable the Quadrature clock	0x0
14:12	R/W	-		0x0
11	R/W	SPI_ENABLE	Enable SPI clock	0x0
10	R/W	-		0x0
9:8	R/W	SPI_DIV	Division factor for SPI 0x0: divide by 1 0x1: divide by 2 0x2: divide by 4 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 CaptureTimer clock	0x0
3	R/W	TMR_ENABLE	Enable TIMER0 and TIMER2 clock	0x0
2	R/W	-		0x0
1:0	R/W	TMR_DIV	Division factor for TIMER0 0x0: divide by 1 0x1: divide by 2 0x2: divide by 4 0x3: divide by 8	0x0

Table 133: CLK_RADIO_REG (0x50000008)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		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

Bit	Mode	Symbol	Description	Reset
5:4	R/W	BLE_DIV	Division factor for BLE core blocks 0x0: divide by 1 0x1: divide by 2 0x2: divide by 4 0x3: divide by 8 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	R/W	-		0x0
1:0	R/W	RFCU_DIV	Division factor for RF Control Unit 0x0: divide by 1 0x1: divide by 2 0x2: divide by 4 0x3: divide by 8 The programmed frequency must be exactly 8 MHz.	0x0

Table 134: CLK_CTRL_REG (0x5000000A)

Bit	Mode	Symbol	Description	Reset
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	R/W	-		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. 0x0: XTAL16M (check the XTAL16_SETTLED and XTAL16_TRIM_READY bits!!) 0x1: RC16M 0x2/0x3: either RC32k or XTAL32k is used	0x0

Table 135: PMU_CTRL_REG (0x50000010)

Bit	Mode	Symbol	Description	Reset
11:8	R/W	RETENTION_MODE	Select the retainability of the 4 RAM macros. '1' is retainable, '0' is power gated. (3) is SysRAM4 (2) is SysRAM3 (1) is SysRAM2 (0) is SysRAM1	0x0
7	R/W	FORCE_BOOST	Force the DCDC into boost mode at next wakeup. Setting this bit reduces the deepsleep current. FORCE_BOOST has highest priority. 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. Setting this bit reduces the deepsleep current. FORCE_BOOST has highest priority. 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
3	R/W	-		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_WAKE UP	Perform a Hardware Reset after waking up. Booter will be started.	0x0

Table 136: 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: SYS_CTRL_REG CLK_FREQ_TRIM_REG ...	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	R/W	-		0x0
7	R/W	DEBUGGER_ENAB LE	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. 0: Control signals are retained 1: Latch is transparant, pad can be recontrolled	0x1
4	R/W	OTP_COPY	Enables OTP to SysRAM copy action after waking up PD_SYS	0x0

Bit	Mode	Symbol	Description	Reset
3	R/W	CLK32_SOURCE	Sets the clock source of the 32 kHz clock 0 = RC-oscillator 1 = 32 kHz crystal oscillator	0x0
2	R/W	DEV_PHASE	Sets the development phase mode. If this bit is set, in combination with the OTP_COPY bit, the OTP DMA will emulate the OTP mirroring to System RAM. No actual writing to RAM is done, but the exact same amount of time is spend as if the mirroring would take place. This is to mimic the behavior as if the System Code is already in OTP, and the mirroring takes place after waking up, but the (development) code still resides in an external source. If this bit is set to '0' and OTP_COPY='1', then the OTP DMA will actually do the OTP mirroring at wakeup.	0x0
1:0	R/W	REMAP_ADR0	Controls which memory is located at address 0x0000 for execution. 0x0: ROM 0x1: OTP 0x2: RAM (SysRAM1) 0x3: RAM (SysRAM4, 64 kBytes offset)	0x0

Table 137: SYS_STAT_REG (0x50000014)

Bit	Mode	Symbol	Description	Reset
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 138: 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. 0x0: apply directly 0x1: wait between 0 and 250 us 0x2: wait between 250 us and 500 us etc.	0xA

Bit	Mode	Symbol	Description	Reset
3:0	R/W	SETTLE_TIME	Defines the delay between applying CLK_FREQ_TRIM_REG and XTAL16_SETTLED in steps of 250 us. 0x0: XTAL16_SETTLED is set direcly 0x1: wait between 0 and 250 us 0x2: wait between 250 us and 500 us etc.	0x2

Table 139: 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. 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. 0x0: lowest frequency 0x7: default 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. 0x0: minimum 0x3: default 0xF: maximum For each application there is an optimal setting for which the startup behaviour is optimal.	0x5
2:1	R/W	XTAL32K_RBIAS	Setting for the bias resistor of the 32 kHz XTAL oscillator. 0x0: maximum 0x3: minimum Prefered setting will be provided by Dialog.	0x3
0	R/W	XTAL32K_ENABLE	Enables the 32 kHz XTAL oscillator	0x0

Table 140: CLK_16M_REG (0x50000022)

Bit	Mode	Symbol	Description	Reset
9	R/W	XTAL16_NOISE_FILT_ENABLE	Enables noise flter in 16 MHz crystal oscillator	0x0
8	R/W	XTAL16_BIAS_SH_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. 0x0: minimum 0x7: maximum	0x5
4:1	R/W	RC16M_TRIM	Controls the frequency of the RC16M oscillator. 0x0: lowest frequency 0xF: highest frequency	0x0

Bit	Mode	Symbol	Description	Reset
0	R/W	RC16M_ENABLE	Enables the 16 MHz RC oscillator	0x0

Table 141: CLK_RCX20K_REG (0x50000024)

Bit	Mode	Symbol	Description	Reset
12	R/W	RCX20K_SELECT	Selects RCX oscillator. 0 : RC32K oscillator 1: RCX oscillator	0
11	R/W	RCX20K_ENABLE	Enable the RCX oscillator	0
10	R/W	RCX20K_LOWF	Extra low frequency	1
9:8	R/W	RCX20K_BIAS	Bias control	1
7:4	R/W	RCX20K_NTC	Temperature control	11
3:0	R/W	RCX20K_TRIM	Controls the frequency of the RCX oscillator. 0x0: lowest frequency 0x7: default 0xF: highest frequency	0

Table 142: BANDGAP_REG (0x50000028)

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

Table 143: ANA_STATUS_REG (0x5000002A)

Bit	Mode	Symbol	Description	Reset
9	R	BOOST_SELECTED	Indicates that DCDC is in boost mode	0x0
8	-	-		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	-		0x0
2	R	VDCDC_OK	Indicates that VDCDC is above threshold.	0x0

Bit	Mode	Symbol	Description	Reset
1	R	VBAT1V_OK	Indicates that VBAT1V is above threshold.	0x0
0	R	VBAT1V_AVAILABLE	Indicates that VBAT1V is available.	0x0

Table 144: POR_PIN_REG (0x50000040)

Bit	Mode	Symbol	Description	Reset
7	R/W	POR_PIN_POLARITY	0: Active Low 1: Active High Note: This applies only for the GPIO pin. Reset pad is always active High	0x0
6	R/W	-		0x0
5:0	R/W	POR_PIN_SELECT	0: GPIO pin POReset disabled 1: P0_0 2: P0_1 3: P0_2 4: P0_3 5: P0_4 6: P0_5 7: P0_6 8: P0_7 9: P1_0 10: P1_1 11: P1_2 12: P1_3 13: P1_4 14: P1_5 15: P2_0 (only in QFN40 and QFN48) 16: P2_1 (only in QFN40 and QFN48) 17: P2_2 (only in QFN40 and QFN48) 18: P2_3 (only in QFN40 and QFN48) 19: P2_4 (only in QFN40 and QFN48) 20: P2_5 (only in QFN40 and QFN48) 21: P2_6 (only in QFN40 and QFN48) 22: P2_7 (only in QFN40 and QFN48) 23: P2_8 (only in QFN40 and QFN48) 24: P2_9 (only in QFN40 and QFN48) 25: P3_0 (only in QFN40 and QFN48) 26: P3_1 (only in QFN48) 27: P3_2 (only in QFN48) 28: P3_3 (only in QFN48) 29: P3_4 (only in QFN48) 30: P3_5 (only in QFN48) 31: P3_6 (only in QFN48) 32: P3_7 (only in QFN48) Note: Pin PID prefered value is 0.	0x0

Table 145: POR_TIMER_REG (0x50000042)

Bit	Mode	Symbol	Description	Reset
6:0	R/W	POR_TIME	Time for the POReset to happen. Formula: Time = POR_TIME x 4096 x RC32 clock period Default value: ~3 seconds	0x18

Table 146: PCM_DIV_REG (0x50001C40)

Bit	Mode	Symbol	Description	Reset
13	R/W	-		0x0
12	R/W	CLK_PCM_EN	Enable for the internally generated PCM clock The PCM_DIV must be set before or together with CLK_PCM_EN.	0x0
11:0	R/W	PCM_DIV	PCM clock divider	0x0

Table 147: PCM_FDIV_REG (0x50001C42)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	PCM_FDIV	These bits define the fractional division part of the PCM clock. The left most '1' defines the denominator, the number of '1' bits define the numerator. E.g. 0x0110 means 2/9, with a distribution of 1.0001.0000 0xfeee means 13/16, with a distribution of 1111.1110.1110.1110	0x0

Table 148: PDM_DIV_REG (0x50001C44)

Bit	Mode	Symbol	Description	Reset
9	R/W	PDM_MASTER_MODE	Master mode selection 0: slave mode 1: master mode	0x0
8	R/W	CLK_PDM_EN	Enable for the internally generated PDM clock The PDM_DIV must be set before or together with CLK_PDM_EN.	0x0
7:0	R/W	PDM_DIV	PDM clock divider	0x0

Table 149: SRC_DIV_REG (0x50001C46)

Bit	Mode	Symbol	Description	Reset
8	R/W	CLK_SRC_EN	Enable for the internally generated SRC clock The SRC_DIV must be set before or together with CLK_SRC_EN.	0x0
7:0	R/W	SRC_DIV	SRC clock divider	0x0

28.5 DCDC Converter Registers

Table 150: Register map crg580_dcdc_nl01

Address	Register	Description
0x50000080	DCDC_CTRL_REG	DCDC control register
0x50000082	DCDC_CTRL2_REG	DCDC second control register
0x50000084	DCDC_CTRL3_REG	DCDC thirth control register

Table 151: DCDC_CTRL_REG (0x50000080)

Bit	Mode	Symbol	Description	Reset
15:14	R/W	DCDC_TUNE	Tune-bits to compensate for parasitic resistance in the current sense circuit of the DCDC-converter.	0x0
13:12	R/W	DCDC_DRIVE_OSW	Drive level of the switch between SWITCH and VDCDC. 00 = 100% 01 = 66% 10 = 33% 11 = off	0x0
11:10	R/W	DCDC_DRIVE_PSW	Drive level of the switch between SWITCH and VBAT3V. 00 = 100% 01 = 66% 10 = 33% 11 = off	0x0
9:8	R/W	DCDC_DRIVE_NSW	Drive level of the switch between SWITCH and GROUND. 00 = 100% 01 = 66% 10 = 33% 11 = off	0x0
7:5	R/W	DCDC_MODE	Testmodes, keep 000.	0x0
4	R/W	-		0x0
3:1	R/W	DCDC_VBAT1V_LEV	If VBAT1V is below this level, the boost converter will be disabled: 110 = 0.6V 101 = 0.8V 011 = 1.0V 111 = 0V (always OK)	0x6
0	R/W	-		0x0

Table 152: DCDC_CTRL2_REG (0x50000082)

Bit	Mode	Symbol	Description	Reset
15:12	R/W	DCDC_VOLT_LEV	Nominal output voltage of the DCDC-converter. VDCDC = 1.2V + N*25mV	0x8
11:9	R/W	DCDC_VBAT3V_LEV	Nominal VBAT3V output voltage of the boost converter.	0x6

Bit	Mode	Symbol	Description	Reset
			000 ... 011 = 1.8V + N*25mV 100 = 2.4V 101 = 2.5V 110 = 2.62V 111 = 2.76V (Note: MSB is automatically on if the OTP LDO is enabled.)	
8:7	R/W	DCDC_TON	This defines the minimum on-time of the comparators. For buck-mode use 0x2, for boost-mode use 0x1	0x0
6:3	R/W	DCDC_CUR_LIM	Current limit in the switches of the DCDC-converter (approximate values): N x 10mA	0x4
2:0	R/W	DCDC_AUTO_CAL	Control of the automatic calibration of the DCDC-converter. For Buck-mode use 0x1, for Boost-mode use 0x6. Automatic calibration is disabled by setting 0x0	0x0

Table 153: DCDC_CTRL3_REG (0x50000084)

Bit	Mode	Symbol	Description	Reset
4:3	R/W	DCDC_TIMEOUT		0x2
2:1	R/W	DCDC_IDLE_CLK	Clock used as trigger for the idle state to check voltage. (Note: when no 16 MHz oscillator is active, the 32 kHz oscillator will be used as trigger independent of the setting below:) 00 = 16 MHz 01 = 4 MHz 10 = 1 MHz 11 = 250 kHz	0x2
0	R/W	BUCK_ENABLE	Enables the buck converter when the device becomes active and VBAT1V is connected to GND.	0x1

28.6 DMA Controller Registers

Table 154: Register map DMA

Address	Register	Description
0x50003600	DMA0_A_STARTL_REG	Start address Low A of DMA channel 0
0x50003602	DMA0_A_STARTH_REG	Start address High A of DMA channel 0
0x50003604	DMA0_B_STARTL_REG	Start address Low B of DMA channel 0
0x50003606	DMA0_B_STARTH_REG	Start address High B of DMA channel 0
0x50003608	DMA0_INT_REG	DMA receive interrupt register channel 0
0x5000360A	DMA0_LEN_REG	DMA receive length register channel 0
0x5000360C	DMA0_CTRL_REG	Control register for the DMA channel 0
0x5000360E	DMA0_IDX_REG	Index value of DMA channel 0
0x50003610	DMA1_A_STARTL_REG	Start address Low A of DMA channel 1
0x50003612	DMA1_A_STARTH_REG	Start address High A of DMA channel 1
0x50003614	DMA1_B_STARTL_REG	Start address Low B of DMA channel 1
0x50003616	DMA1_B_STARTH_REG	Start address High B of DMA channel 1
0x50003618	DMA1_INT_REG	DMA receive interrupt register channel 1
0x5000361A	DMA1_LEN_REG	DMA receive length register channel 1
0x5000361C	DMA1_CTRL_REG	Control register for the DMA channel 1
0x5000361E	DMA1_IDX_REG	Index value of DMA channel 1
0x50003620	DMA2_A_STARTL_REG	Start address Low A of DMA channel 2
0x50003622	DMA2_A_STARTH_REG	Start address High A of DMA channel 2
0x50003624	DMA2_B_STARTL_REG	Start address Low B of DMA channel 2
0x50003626	DMA2_B_STARTH_REG	Start address High B of DMA channel 2
0x50003628	DMA2_INT_REG	DMA receive interrupt register channel 2
0x5000362A	DMA2_LEN_REG	DMA receive length register channel 2
0x5000362C	DMA2_CTRL_REG	Control register for the DMA channel 2
0x5000362E	DMA2_IDX_REG	Index value of DMA channel 2
0x50003630	DMA3_A_STARTL_REG	Start address Low A of DMA channel 3
0x50003632	DMA3_A_STARTH_REG	Start address High A of DMA channel 3

Address	Register	Description
0x50003634	DMA3_B_STARTL_REG	Start address Low B of DMA channel 3
0x50003636	DMA3_B_STARTH_REG	Start address High B of DMA channel 3
0x50003638	DMA3_INT_REG	DMA receive interrupt register channel 3
0x5000363A	DMA3_LEN_REG	DMA receive length register channel 3
0x5000363C	DMA3_CTRL_REG	Control register for the DMA channel 3
0x5000363E	DMA3_IDX_REG	Index value of DMA channel 3
0x50003680	DMA_REQ_MUX_REG	DMA channel assignments
0x50003682	DMA_INT_STATUS_REG	DMA interrupt status register
0x50003684	DMA_CLEAR_INT_REG	DMA clear interrupt register

Table 155: DMA0_A_STARTL_REG (0x50003600)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA0_A_STARTL	Source start address, lower 16 bits	0x0

Table 156: DMA0_A_STARTH_REG (0x50003602)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA0_A_STARTH	Source start address, upper 16 bits	0x0

Table 157: DMA0_B_STARTL_REG (0x50003604)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA0_B_STARTL	Destination start address, lower 16 bits	0x0

Table 158: DMA0_B_STARTH_REG (0x50003606)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA0_B_STARTH	Destination start address, upper 16 bits	0x0

Table 159: DMA0_INT_REG (0x50003608)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA0_INT	Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt.	0x0

Table 160: DMA0_LEN_REG (0x5000360A)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA0_LEN	DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 161: DMA0_CTRL_REG (0x5000360C)

Bit	Mode	Symbol	Description	Reset
15:14	R	-		0x0
13	R/W	REQ_SENSE	0 = DMA operates with level-sensitive peripheral requests (default) 1 = DMA operates with (positive) edge-sensitive peripheral requests	0x0
12	R/W	DMA_INIT	0 = DMA performs copy A1 to B1, A2 to B2, etc ... 1 = DMA performs copy of A1 to B1, B2, etc ... This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.	0x0
11	R/W	DMA_IDLE	0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority. 1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.	0x0
10:8	R/W	DMA_PRIO	The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific: 000 = lowest priority 111 = highest priority If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus.	0x0

Bit	Mode	Symbol	Description	Reset
7	R/W	CIRCULAR	0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed. 1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.	0x0
6	R/W	AINC	Enable increment of source address. 0 = do not increment (source address stays the same during the transfer) 1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")	0x0
5	R/W	BINC	Enable increment of destination address. 0 = do not increment (destination address stays the same during the transfer) 1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")	0x0
4	R/W	DREQ_MODE	0 = DMA channel starts immediately 1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)	0x0
3	R/W	IRQ_ENABLE	0 = disable interrupt on this channel 1 = enable interrupt on this channel	0x0
2:1	R/W	BW	Bus transfer width: 00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI) 01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI) 10 = 4 Bytes (suggested for Memory-to-Memory transfers) 11 = Reserved	0x0
0	R/W	DMA_ON	0 = DMA channel is off, clocks are disabled 1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.	0x0

Table 162: DMA0_IDX_REG (0x5000360E)

Bit	Mode	Symbol	Description	Reset
15:0	R	DMA0_IDX	This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0. The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW.	0x0

Table 163: DMA1_A_STARTL_REG (0x50003610)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA1_A_STARTL	Source start address, lower 16 bits	0x0

Table 164: DMA1_A_STARTH_REG (0x50003612)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA1_A_STARTH	Source start address, upper 16 bits	0x0

Table 165: DMA1_B_STARTL_REG (0x50003614)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA1_B_STARTL	Destination start address, lower 16 bits	0x0

Table 166: DMA1_B_STARTH_REG (0x50003616)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA1_B_STARTH	Destination start address, upper 16 bits	0x0

Table 167: DMA1_INT_REG (0x50003618)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA1_INT	Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt.	0x0

Table 168: DMA1_LEN_REG (0x5000361A)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA1_LEN	DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 169: DMA1_CTRL_REG (0x5000361C)

Bit	Mode	Symbol	Description	Reset
15:14	R	-		0x0

Bit	Mode	Symbol	Description	Reset
13	R/W	REQ_SENSE	0 = DMA operates with level-sensitive peripheral requests (default) 1 = DMA operates with (positive) edge-sensitive peripheral requests	0x0
12	R/W	DMA_INIT	0 = DMA performs copy A1 to B1, A2 to B2, etc ... 1 = DMA performs copy of A1 to B1, B2, etc ... This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.	0x0
11	R/W	DMA_IDLE	0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority. 1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.	0x0
10:8	R/W	DMA_PRIO	The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific: 000 = lowest priority 111 = highest priority If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus.	0x0
7	R/W	CIRCULAR	0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed. 1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.	0x0
6	R/W	AINC	Enable increment of source address. 0 = do not increment (source address stays the same during the transfer) 1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")	0x0
5	R/W	BINC	Enable increment of destination address. 0 = do not increment (destination address stays the same during the transfer) 1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")	0x0
4	R/W	DREQ_MODE	0 = DMA channel starts immediately 1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)	0x0
3	R/W	IRQ_ENABLE	0 = disable interrupt on this channel 1 = enable interrupt on this channel	0x0

Bit	Mode	Symbol	Description	Reset
2:1	R/W	BW	Bus transfer width: 00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI) 01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI) 10 = 4 Bytes (suggested for Memory-to-Memory transfers) 11 = Reserved	0x0
0	R/W	DMA_ON	0 = DMA channel is off, clocks are disabled 1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.	0x0

Table 170: DMA1_IDX_REG (0x5000361E)

Bit	Mode	Symbol	Description	Reset
15:0	R	DMA1_IDX	This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0. The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW.	0x0

Table 171: DMA2_A_STARTL_REG (0x50003620)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA2_A_STARTL	Source start address, lower 16 bits	0x0

Table 172: DMA2_A_STARTH_REG (0x50003622)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA2_A_STARTH	Source start address, upper 16 bits	0x0

Table 173: DMA2_B_STARTL_REG (0x50003624)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA2_B_STARTL	Destination start address, lower 16 bits	0x0

Table 174: DMA2_B_STARTH_REG (0x50003626)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA2_B_STARTH	Destination start address, upper 16 bits	0x0

Table 175: DMA2_INT_REG (0x50003628)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA2_INT	Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt.	0x0

Table 176: DMA2_LEN_REG (0x5000362A)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA2_LEN	DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 177: DMA2_CTRL_REG (0x5000362C)

Bit	Mode	Symbol	Description	Reset
15:14	R	-		0x0
13	R/W	REQ_SENSE	0 = DMA operates with level-sensitive peripheral requests (default) 1 = DMA operates with (positive) edge-sensitive peripheral requests	0x0
12	R/W	DMA_INIT	0 = DMA performs copy A1 to B1, A2 to B2, etc ... 1 = DMA performs copy of A1 to B1, B2, etc ... This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.	0x0
11	R/W	DMA_IDLE	0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority. 1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.	0x0

Bit	Mode	Symbol	Description	Reset
10:8	R/W	DMA_PRIO	The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific: 000 = lowest priority 111 = highest priority If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus.	0x0
7	R/W	CIRCULAR	0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed. 1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.	0x0
6	R/W	AINC	Enable increment of destination address. 0 = do not increment (destination address stays the same during the transfer) 1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")	0x0
5	R/W	BINC	Enable increment of destination address 0 = do not increment 1 = increment according value of BW	0x0
4	R/W	DREQ_MODE	0 = DMA channel starts immediately 1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)	0x0
3	R/W	IRQ_ENABLE	0 = disable interrupt on this channel 1 = enable interrupt on this channel	0x0
2:1	R/W	BW	Bus transfer width: 00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI) 01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI) 10 = 4 Bytes (suggested for Memory-to-Memory transfers) 11 = Reserved	0x0
0	R/W	DMA_ON	0 = DMA channel is off, clocks are disabled 1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.	0x0

Table 178: DMA2_IDX_REG (0x5000362E)

Bit	Mode	Symbol	Description	Reset
15:0	R	DMA2_IDX	This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0. The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW.	0x0

Table 179: DMA3_A_STARTL_REG (0x50003630)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA3_A_STARTL	Source start address, lower 16 bits	0x0

Table 180: DMA3_A_STARTH_REG (0x50003632)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA3_A_STARTH	Source start address, upper 16 bits	0x0

Table 181: DMA3_B_STARTL_REG (0x50003634)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA3_B_STARTL	Destination start address, lower 16 bits	0x0

Table 182: DMA3_B_STARTH_REG (0x50003636)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA3_B_STARTH	Destination start address, upper 16 bits	0x0

Table 183: DMA3_INT_REG (0x50003638)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA3_INT	Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if DMAx_INT_REG is equal to DMAx_IDX_REG and before DMAx_IDX_REG is incremented. The bit-field IRQ_ENABLE of DMAx_CTRL_REG must be set to '1' to let the controller generate the interrupt.	0x0

Table 184: DMA3_LEN_REG (0x5000363A)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	DMA3_LEN	DMA channel's transfer length. DMAx_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 185: DMA3_CTRL_REG (0x5000363C)

Bit	Mode	Symbol	Description	Reset
15:14	R	-		0x0
13	R/W	REQ_SENSE	0 = DMA operates with level-sensitive peripheral requests (default) 1 = DMA operates with (positive) edge-sensitive peripheral requests	0x0
12	R/W	DMA_INIT	0 = DMA performs copy A1 to B1, A2 to B2, etc ... 1 = DMA performs copy of A1 to B1, B2, etc ... This feature is useful for memory initialization to any value. Thus, BINC must be set to '1', while AINC is don't care, as only one fetch from A is done. This process cannot be interrupted by other DMA channels. It is also noted that DMA_INIT should not be used when DREQ_MODE='1'.	0x0
11	R/W	DMA_IDLE	0 = Blocking mode, the DMA performs a fast back-to-back copy, disabling bus access for any bus master with lower priority. 1 = Interrupting mode, the DMA inserts a wait cycle after each store allowing the CPU to steal cycles or cache to perform a burst read. If DREQ_MODE='1', DMA_IDLE is don't care.	0x0
10:8	R/W	DMA_PRIO	The priority level determines which DMA channel will be granted access for transferring data, in case more than one channels are active and request the bus at the same time. The greater the value, the higher the priority. In specific: 000 = lowest priority 111 = highest priority If different channels with equal priority level values request the bus at the same time, an inherent priority mechanism is applied. According to this mechanism, if, for example, both the DMA0 and DMA1 channels have the same priority level, then DMA0 will first be granted access to the bus.	0x0
7	R/W	CIRCULAR	0 = Normal mode. The DMA channel stops after having completed the transfer of length determined by DMAx_LEN_REG. DMA_ON automatically deasserts when the transfer is completed. 1 = Circular mode (applicable only if DREQ_MODE = '1'). In this mode, DMA_ON never deasserts, as the DMA channel automatically resets DMAx_IDX_REG and starts a new transfer.	0x0
6	R/W	AINC	Enable increment of source address. 0 = do not increment (source address stays the same during the transfer) 1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")	0x0

Bit	Mode	Symbol	Description	Reset
5	R/W	BINC	Enable increment of destination address. 0 = do not increment (destination address stays the same during the transfer) 1 = increment according to the value of BW bit-field (by 1, when BW="00" ; by 2, when BW="01" ; by 4, when BW="10")	0x0
4	R/W	DREQ_MODE	0 = DMA channel starts immediately 1 = DMA channel must be triggered by peripheral DMA request (see also the description of DMA_REQ_MUX_REG)	0x0
3	R/W	IRQ_ENABLE	0 = disable interrupt on this channel 1 = enable interrupt on this channel	0x0
2:1	R/W	BW	Bus transfer width: 00 = 1 Byte (suggested for peripherals like UART and 8-bit SPI) 01 = 2 Bytes (suggested for peripherals like I2C and 16-bit SPI) 10 = 4 Bytes (suggested for Memory-to-Memory transfers) 11 = Reserved	0x0
0	R/W	DMA_ON	0 = DMA channel is off, clocks are disabled 1 = DMA channel is enabled. This bit will be automatically cleared after the completion of a transfer, if circular mode is not enabled. In circular mode, this bit stays set.	0x0

Table 186: DMA3_IDX_REG (0x5000363E)

Bit	Mode	Symbol	Description	Reset
15:0	R	DMA3_IDX	This (read-only) register determines the data items currently fetched by the DMA channel, during an on-going transfer. When the transfer is completed, the register is automatically reset to 0. The DMA channel uses this register to form the source/destination address of the next DMA cycle, considering also AINC/BINC and BW.	0x0

Table 187: DMA_REQ_MUX_REG (0x50003680)

Bit	Mode	Symbol	Description	Reset
15:12	R/W	-		0xF
11:8	R/W	-		0xF
7:4	R/W	DMA23_SEL	Select which combination of peripherals are mapped on the DMA channels. The peripherals are mapped as pairs on two channels. Here, the first DMA request is mapped on channel 2 and the second on channel 3. See DMA01_SEL for the peripherals' mapping.	0xF

Bit	Mode	Symbol	Description	Reset
3:0	R/W	DMA01_SEL	Select which combination of peripherals are mapped on the DMA channels. The peripherals are mapped as pairs on two channels. Here, the first DMA request is mapped on channel 0 and the second on channel 1. 0x0: SPI_rx / SPI_tx 0x1: Reserved 0x2: UART_rx / UART_tx 0x3: UART2_rx / UART2_tx 0x4: I2C_rx / I2C_tx 0x5: Reserved 0x6: Reserved 0x7: Reserved 0x8: PCM_rx / PCM_tx 0x9: SRC_rx / SRC_tx (for all the supported conversions) 0xA: Reserved 0xB: Reserved 0xC: Reserved 0xD: Reserved 0xE: Reserved 0xF: None Note: If any of the two available peripheral selector fields (DMA01_SEL, DMA23_SEL) have the same value, the lesser significant selector has higher priority and will control the dma acknowledge. Hence, if DMA01_SEL = DMA23_SEL, the channels 0 and 1 will generate the DMA acknowledge signals for the selected peripheral. Consequently, it is suggested to assign the intended peripheral value to a unique selector field.	0xF

Table 188: DMA_INT_STATUS_REG (0x50003682)

Bit	Mode	Symbol	Description	Reset
15:8	R	-		0x0
7	R	-		0x0
6	R	-		0x0
5	R	-		0x0
4	R	-		0x0
3	R	DMA_IRQ_CH3	0: IRQ on channel 3 is not set 1: IRQ on channel 3 is set	0x0
2	R	DMA_IRQ_CH2	0: IRQ on channel 2 is not set 1: IRQ on channel 2 is set	0x0
1	R	DMA_IRQ_CH1	0: IRQ on channel 1 is not set 1: IRQ on channel 1 is set	0x0
0	R	DMA_IRQ_CH0	0: IRQ on channel 0 is not set 1: IRQ on channel 0 is set	0x0

Table 189: DMA_CLEAR_INT_REG (0x50003684)

Bit	Mode	Symbol	Description	Reset
15:8	R	-		0x0
7	R	-		0x0
6	R	-		0x0
5	R	-		0x0
4	R	-		0x0
3	R0/W	DMA_RST_IRQ_CH3	Writing a 1 will reset the status bit of DMA_INT_STATUS_REG for channel 3 ; writing a 0 will have no effect	0x0
2	R0/W	DMA_RST_IRQ_CH2	Writing a 1 will reset the status bit of DMA_INT_STATUS_REG for channel 2 ; writing a 0 will have no effect	0x0
1	R0/W	DMA_RST_IRQ_CH1	Writing a 1 will reset the status bit of DMA_INT_STATUS_REG for channel 1 ; writing a 0 will have no effect	0x0
0	R0/W	DMA_RST_IRQ_CH0	Writing a 1 will reset the status bit of DMA_INT_STATUS_REG for channel 0 ; writing a 0 will have no effect	0x0

28.7 General Purpose ADC Registers

Table 190: Register map GPADC

Address	Register	Description
0x50001500	GP_ADC_CTRL_REG	General Purpose ADC Control Register
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

Table 191: 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 dertermine 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 1 = Single ended mode	0x0
10	R/W	GP_ADC_SIGN	0 = Default 1 = Conversion with opposite sign at input and output to cancel out the internal offset of the ADC and low-frequency	0x0

Bit	Mode	Symbol	Description	Reset
9:6	R/W	GP_ADC_SEL	ADC input selection which must be set before the GP_ADC_START bit is enabled. If GP_ADC_SE = 1 (single ended mode): 0000 = P0[0] 0001 = P0[1] 0010 = P0[2] 0011 = P0[3] 0100 = AVS 0101 = VDD_REF 0110 = VDD_RTT (= VDD_REF) 0111 = VBAT3V 1000 = VDCDC 1001 = VBAT1V All other combinations are reserved. If GP_ADC_SE = 0 (differential mode): 0000 = P0[0] vs P0[1] All other combinations are P0[2] vs P0[3].	0x0
5	R/W	GP_ADC_MINT	0 = Disable (mask) GP_ADC_INT. 1 = Enable GP_ADC_INT to ICU.	0x0
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. 1 = Digital clock used.	0x0
2	-	GP_ADC_TEST	Reserved, keep 0.	0x0
1	R/W	GP_ADC_START	0 = ADC conversion ready. 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. 1 = ADC is enabled and sampling of input is started.	0x0

Table 192: GP_ADC_CTRL2_REG (0x50001502)

Bit	Mode	Symbol	Description	Reset
15:4	-	-		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. 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 succesive conversions.	0x0

Table 193: GP_ADC_OFFP_REG (0x50001504)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		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 194: GP_ADC_OFFN_REG (0x50001506)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		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 195: 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 196: GP_ADC_RESULT_REG (0x5000150A)

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

Table 197: GP_ADC_DELAY_REG (0x5000150C)

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

Table 198: 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 µs 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 µs which is the recommended value to wait after enabling the LDO. This is the second step in bringing up the GP ADC.	0x80

28.8 General Purpose I/O Registers

Table 199: Register map GPIO

Address	Register	Description
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
0x50003070	P01_PADPWR_CTRL_REG	Ports 0 and 1 Output Power Control Register

Address	Register	Description
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

Table 200: P0_DATA_REG (0x50003000)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	P0_DATA	Set P0 output register when written; Returns the value of P0 port when read	0x0

Table 201: P0_SET_DATA_REG (0x50003002)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R0/W	P0_SET	Writing a 1 to P0[y] sets P0[y] to 1. Writing 0 is discarded; Reading returns 0	0x0

Table 202: P0_RESET_DATA_REG (0x50003004)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R0/W	P0_RESET	Writing a 1 to P0[y] sets P0[y] to 0. Writing 0 is discarded; Reading returns 0	0x0

Table 203: P00_MODE_REG (0x50003006)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected In ADC mode, these bits are don't care	0x2
7:5	-	-		0x0
4:0	R/W	PID	Function of port 0 = Port function, PUPD as set above 1 = UART1_RX 2 = UART1_TX 3 = UART2_RX 4 = UART2_TX 5 = SPI_DI 6 = SPI_DO 7 = SPI_CLK 8 = SPI_EN 9 = I2C_SCL 10 = I2C_SDA 11 = UART1_IRDA_RX 12 = UART1_IRDA_TX 13 = UART2_IRDA_RX 14 = UART2_IRDA_TX 15 = ADC (only for P0[3:0]) 16 = PWM0 17 = PWM1 18 = BLE_DIAG (only for P0[7:0] and P1[3:0]) 19 = UART1_CTSN 20 = UART1_RTSN 21 = UART2_CTSN 22 = UART2_RTSN 23 = PWM2 24 = PWM3 25 = PWM4 26 = PCM_DI 27 = PCM_DO 28 = PCM_FSC 29 = PCM_CLK 30 = PDM_DATA 31 = PDM_CLK 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 204: P01_MODE_REG (0x50003008)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected In ADC mode, these bits are don't care	0x2
7:5	-	-		0x0

Bit	Mode	Symbol	Description	Reset
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 205: P02_MODE_REG (0x5000300A)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected In ADC mode, these bits are don't care	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 206: P03_MODE_REG (0x5000300C)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected In ADC mode, these bits are don't care	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 207: P04_MODE_REG (0x5000300E)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 208: P05_MODE_REG (0x50003010)

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

Bit	Mode	Symbol	Description	Reset
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 209: P06_MODE_REG (0x50003012)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 210: P07_MODE_REG (0x50003014)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 211: P1_DATA_REG (0x50003020)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	P1_DATA	Set P1 output register when written; Returns the value of P1 port when read	0x20

Table 212: P1_SET_DATA_REG (0x50003022)

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

Bit	Mode	Symbol	Description	Reset
7:0	R0/W	P1_SET	Writing a 1 to P1[y] sets P1[y] to 1. Writing 0 is discarded; Reading returns 0	0x0

Table 213: P1_RESET_DATA_REG (0x50003024)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R0/W	P1_RESET	Writing a 1 to P1[y] sets P1[y] to 0. Writing 0 is discarded; Reading returns 0	0x0

Table 214: P10_MODE_REG (0x50003026)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 215: P11_MODE_REG (0x50003028)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 216: P12_MODE_REG (0x5000302A)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0

Bit	Mode	Symbol	Description	Reset
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 217: P13_MODE_REG (0x5000302C)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 218: P14_MODE_REG (0x5000302E)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 219: P15_MODE_REG (0x50003030)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected Note that the reset value of this bit-field of P15_MODE_REG is 0x1 (i.e. pulled-up).	0x1
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 220: P2_DATA_REG (0x50003040)

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

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

Table 221: P2_SET_DATA_REG (0x50003042)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:0	R0/W	P2_SET	Writing a 1 to P2[y] sets P2[y] to 1. Writing 0 is discarded; Reading returns 0	0x0

Table 222: P2_RESET_DATA_REG (0x50003044)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:0	R0/W	P2_RESET	Writing a 1 to P2[y] sets P2[y] to 0. Writing 0 is discarded; Reading returns 0	0x0

Table 223: P20_MODE_REG (0x50003046)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 224: P21_MODE_REG (0x50003048)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 225: P22_MODE_REG (0x5000304A)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 226: P23_MODE_REG (0x5000304C)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 227: P24_MODE_REG (0x5000304E)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 228: P25_MODE_REG (0x50003050)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 229: P26_MODE_REG (0x50003052)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 230: P27_MODE_REG (0x50003054)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 231: P28_MODE_REG (0x50003056)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 232: P29_MODE_REG (0x50003058)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	0x2
7:5	-	-		0x0
4:0	R/W	PID	See P00_MODE_REG[PID]	0x0

Table 233: P01_PADPWR_CTRL_REG (0x50003070)

Bit	Mode	Symbol	Description	Reset
15:14	-	-		0x0
13:8	R/W	P1_OUT_CTRL	1 = P1_x port output is powered by the 1V rail 0 = P1_x port output is powered by the 3V rail Bit (8 + x) controls the power of P1[x], x=0, 1,..., 5.	0x0
7:0	R/W	P0_OUT_CTRL	1 = P0_x port output is powered by the 1V rail 0 = P0_x port output is powered by the 3V rail Bit x controls the power of P0[x],x=0, 1,..., 7.	0x0

Table 234: P2_PADPWR_CTRL_REG (0x50003072)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0x0
9:0	R/W	P2_OUT_CTRL	1 = P2_x port output is powered by the 1V rail 0 = P2_x port output is powered by the 3V rail Bit x controls the power of P2[x], x=0, 1,..., 9.	0x0

Table 235: P3_PADPWR_CTRL_REG (0x50003074)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0
7:0	R/W	P3_OUT_CTRL	1 = P3_x port output is powered by the 1 V rail 0 = P3_x port output is powered by the 3 V rail Bit x controls the power of P3[x], x = 0, 1,..., 7.	0

Table 236: P3_DATA_REG (0x50003080)

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

Table 237: P3_SET_DATA_REG (0x50003082)

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

Table 238: P3_RESET_DATA_REG (0x50003084)

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

Table 239: P30_MODE_REG (0x50003086)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

Table 240: P31_MODE_REG (0x50003088)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

Table 241: P32_MODE_REG (0x5000308A)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

Table 242: P33_MODE_REG (0x5000308C)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

Table 243: P34_MODE_REG (0x5000308E)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

Table 244: P35_MODE_REG (0x50003090)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

Table 245: P36_MODE_REG (0x50003092)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

Table 246: P37_MODE_REG (0x50003094)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		0
9:8	R/W	PUPD	00 = Input, no resistors selected 01 = Input, pull-up selected 10 = Input, pull-down selected 11 = Output, no resistors selected	2
7:5	-	-		0
4:0	R/W	PID	See P00_MODE_REG[PID]	0

28.9 General Purpose Registers

Table 247: Register map GPREG

Address	Register	Description
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.
0x5000330A	BLE_FINECNT_SAMP_REG	BLE FINECNT sampled value while in deep sleep state.

Table 248: SET_FREEZE_REG (0x50003300)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R/W	FRZ_DMA	If '1', the DMA is frozen, '0' is discarded.	0x0
3	R/W	FRZ_WDOG	If '1', the watchdog timer is frozen, '0' is discarded. WATCHDOG_CTRL_REG[NMI_RST] must be '0' to allow the freeze function.	0x0
2	R/W	FRZ_BLETIM	If '1', the BLE master clock is frozen, '0' is discarded.	0x0
1	R/W	FRZ_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 249: RESET_FREEZE_REG (0x50003302)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R/W	FRZ_DMA	If '1', the DMA continues, '0' is discarded.	0x0
3	R/W	FRZ_WDOG	If '1', the watchdog timer continues, '0' is discarded.	0x0
2	R/W	FRZ_BLETIM	If '1', the 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 250: DEBUG_REG (0x50003304)

Bit	Mode	Symbol	Description	Reset
15:1	R/W	-		0x0

Bit	Mode	Symbol	Description	Reset
0	R/W	DEBUGS_FREEZE_EN	Default '1', freezing of the on-chip timers is enabled when the Cortex-M0 is halted in DEBUG State. If '0', freezing of the on-chip timers is depending on FREEZE_REG when the Cortex-M0 is halted in DEBUG State except the watchdog timer. The watchdog timer is always frozen when the Cortex-M0 is halted in DEBUG State.	0x1

Table 251: GP_STATUS_REG (0x50003306)

Bit	Mode	Symbol	Description	Reset
15:2	-	-		0x0
1	R/W	-		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 occured.	0x0

Table 252: GP_CONTROL_REG (0x50003308)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R/W	CPU_DMA_BUS_PRIO	Controls the CPU DMA system bus priority: If '0', the CPU has highest priority. If '1', the DMA has highest priority.	0x0
3	-	-		0x0
2	R	BLE_WAKEUP_LP_IRQ	The current value of the BLE_WAKEUP_LP_IRQ interrupt request.	0x0
1	-	-		0x0
0	R/W	BLE_WAKEUP_REQ	If '1', the BLE wakes up. Must be kept high at least for 1 low power clock period. 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 253: BLE_FINECNT_SAMP_REG (0x5000330A)

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

Bit	Mode	Symbol	Description	Reset
9:0	R/W	BLE_FINECNT_SAMP	This register is located at the Always On Power Domain and it holds the automatically sampled value of the BLE FINECNT timer The HW automatically samples the value into this register during the sequence of "BLE Sleep On" and restores automatically the value during the BLE Wake up sequence. The Software may read and modify the value while the BLE is in Sleep state. While the BLE is awake, the value of the register has no meaning, while changing the value by writing another one will have no effect in the operation of the BLE core.	0x0

28.10 I2C Interface Registers

Table 254: Register map I2C

Address	Register	Description
0x50001300	I2C_CON_REG	I2C Control Register
0x50001304	I2C_TAR_REG	I2C Target Address Register
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

Address	Register	Description
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
0x50001388	I2C_DMA_CR_REG	DMA Control Register
0x5000138C	I2C_DMA_TDLR_REG	DMA Transmit Data Level Register
0x50001390	I2C_DMA_RDLR_REG	I2C Receive Data Level 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

Table 255: I2C_CON_REG (0x50001300)

Bit	Mode	Symbol	Description	Reset
15:7	-	-		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. 0=slave is enabled 1=slave is disabled 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 0= disable 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. 0= 7-bit addressing 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. 0= 7-bit addressing 1= 10-bit addressing	0x1

Bit	Mode	Symbol	Description	Reset
2:1	R/W	I2C_SPEED	These bits control at which speed the controller operates. 1= standard mode (100 kbit/s) 2= fast mode (400 kbit/s) Note: The actuall speed depends on the pcb traces capacitance as well as on the values of the external pull-up resistorts. For an exact speed match, trimming might be required, by adjusting the values of I2C_SS_SCL_HCNT_REG, I2C_SS_SCL_LCNT_REG, I2C_FS_SCL_HCNT_REG, I2C_FS_SCL_LCNT_REG registers. The reset values of those registers were calculated with the assumption of 4.3kOhms external pull-up resistors.	0x2
0	R/W	I2C_MASTER_MODE	This bit controls whether the controller master is enabled. 0= master disabled 1= master enabled Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'.	0x1

Table 256: I2C_TAR_REG (0x50001304)

Bit	Mode	Symbol	Description	Reset
15:12	-	-		0x0
11	R/W	SPECIAL	This bit indicates whether software performs a General Call or START BYTE command. 0: ignore bit 10 GC_OR_START and use IC_TAR normally 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. 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. 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. 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 257: I2C_SAR_REG (0x50001308)

Bit	Mode	Symbol	Description	Reset
15:10	-	-		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 258: I2C_DATA_CMD_REG (0x50001310)

Bit	Mode	Symbol	Description	Reset
15:9	-	-		0x0
8	R/W	CMD	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. 1 = Read 0 = Write 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. If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. 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	0x0
7:0	R/W	DAT	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the controller. However, when you read this register, these bits return the value of data received on the controller's interface.	0x0

Table 259: I2C_SS_SCL_HCNT_REG (0x50001314)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	IC_SS_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 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. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. 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.	0x48

Table 260: 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. 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. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set.	0x4F

Table 261: 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. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.	0x8

Table 262: 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. 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 263: I2C_INTR_STAT_REG (0x5000132C)

Bit	Mode	Symbol	Description	Reset
15:12	-	-		0x0
11	R	R_GEN_CALL	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling controller or when the CPU reads bit 0 of the I2C_CLR_GEN_CALL register. The controller stores the received data in the Rx buffer.	0x0
10	R	R_START_DET	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.	0x0
9	R	R_STOP_DET	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.	0x0
8	R	R_ACTIVITY	This bit captures I2C Ctrl activity and stays set until it is cleared. There are four ways to clear it: => Disabling the I2C Ctrl => Reading the IC_CLR_ACTIVITY register => Reading the IC_CLR_INTR register => System reset 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

Bit	Mode	Symbol	Description	Reset
6	R	R_TX_ABRT	This bit indicates if the controller, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a "transmit abort". When this bit is set to 1, the I2C_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. NOTE: The controller flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register I2C_CLR_TX_ABRT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.	0x0
5	R	R_RD_REQ	This bit is set to 1 when the controller is acting as a slave and another I2C master is attempting to read data from the controller. The controller holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the I2C_DATA_CMD register. This bit is set to 0 just after the processor reads the I2C_CLR_RD_REQ register	0x0
4	R	R_TX_EMPTY	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the I2C_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en=0, this bit is set to 0.	0x0
3	R	R_TX_OVER	Set during transmit if the transmit buffer is filled to 32 and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared	0x0
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

Bit	Mode	Symbol	Description	Reset
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 264: I2C_INTR_MASK_REG (0x50001330)

Bit	Mode	Symbol	Description	Reset
15:12	-	-		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 265: I2C_RAW_INTR_STAT_REG (0x50001334)

Bit	Mode	Symbol	Description	Reset
15:12	-	-		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

Bit	Mode	Symbol	Description	Reset
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: => Disabling the I2C Ctrl => Reading the IC_CLR_ACTIVITY register => Reading the IC_CLR_INTR register => System reset 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". When this bit is set to 1, the I2C_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. 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

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 266: I2C_RX_TL_REG (0x50001338)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		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 267: I2C_TX_TL_REG (0x5000133C)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0

Bit	Mode	Symbol	Description	Reset
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 268: I2C_CLR_INTR_REG (0x50001340)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 269: I2C_CLR_RX_UNDER_REG (0x50001344)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 270: I2C_CLR_RX_OVER_REG (0x50001348)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 271: I2C_CLR_TX_OVER_REG (0x5000134C)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 272: I2C_CLR_RD_REQ_REG (0x50001350)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 273: I2C_CLR_TX_ABRT_REG (0x50001354)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 274: I2C_CLR_RX_DONE_REG (0x50001358)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 275: I2C_CLR_ACTIVITY_REG (0x5000135C)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 276: I2C_CLR_STOP_DET_REG (0x50001360)

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

Bit	Mode	Symbol	Description	Reset
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 277: I2C_CLR_START_DET_REG (0x50001364)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 278: I2C_CLR_GEN_CALL_REG (0x50001368)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 279: I2C_ENABLE_REG (0x5000136C)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
0	R/W	CTRL_ENABLE	Controls whether the controller is enabled. 0: Disables the controller (TX and RX FIFOs are held in an erased state) 1: Enables the controller 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: * The TX FIFO and RX FIFO get flushed. * Status bits in the IC_INTR_STAT register are still active until the controller goes into IDLE state. 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. There is a two ic_clk delay when enabling or disabling the controller	0x0

Table 280: I2C_STATUS_REG (0x50001370)

Bit	Mode	Symbol	Description	Reset
15:7	-	-		0x0
6	R	SLV_ACTIVITY	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Slave FSM is in IDLE state so the Slave part of the controller is not Active 1: Slave FSM is not in IDLE state so the Slave part of the controller is Active	0x0
5	R	MST_ACTIVITY	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0: Master FSM is in IDLE state so the Master part of the controller is not Active 1: Master FSM is not in IDLE state so the Master part of the controller is Active	0x0
4	R	RFF	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0: Receive FIFO is not full 1: Receive FIFO is full	0x0
3	R	RFNE	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty. 0: Receive FIFO is empty 1: Receive FIFO is not empty	0x0
2	R	TFE	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0: Transmit FIFO is not empty 1: Transmit FIFO is empty	0x1
1	R	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0: Transmit FIFO is full 1: Transmit FIFO is not full	0x1
0	R	I2C_ACTIVITY	I2C Activity Status.	0x0

Table 281: I2C_TXFLR_REG (0x50001374)

Bit	Mode	Symbol	Description	Reset
15:6	-	-		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 282: I2C_RXFLR_REG (0x50001378)

Bit	Mode	Symbol	Description	Reset
15:6	-	-		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 283: I2C_SDA_HOLD_REG (0x5000137C)

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

Table 284: 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_NORSTRT	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

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_NORSTRT	1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode	0x0
7	R	ABRT_SBYTE_ACKDET	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).	0x0
6	R	ABRT_HS_ACKDET	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).	0x0
5	R	ABRT_GCALL_READ	1: the controller in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).	0x0
4	R	ABRT_GCALL_NOACK	1: the controller in master mode sent a General Call and no slave on the bus acknowledged the General Call.	0x0
3	R	ABRT_TXDATA_NOACK	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).	0x0
2	R	ABRT_10ADDR2_NOACK	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave.	0x0
1	R	ABRT_10ADDR1_NOACK	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.	0x0
0	R	ABRT_7B_ADDR_NOACK	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.	0x0

Table 285: I2C_DMA_CR_REG (0x50001388)

Bit	Mode	Symbol	Description	Reset
1	R/W	TDMAE	Transmit DMA Enable. //This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled	0x0
0	R/W	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled	0x0

Table 286: I2C_DMA_TDLR_REG (0x5000138C)

Bit	Mode	Symbol	Description	Reset
4:0	R/W	DMATDL	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.	0x0

Table 287: I2C_DMA_RDLR_REG (0x50001390)

Bit	Mode	Symbol	Description	Reset
4:0	R/W	DMARDL	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.	0x0

Table 288: I2C_SDA_SETUP_REG (0x50001394)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	SDA_SETUP	SDA Setup. 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. 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 289: I2C_ACK_GENERAL_CALL_REG (0x50001398)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		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 290: I2C_ENABLE_STATUS_REG (0x5000139C)

Bit	Mode	Symbol	Description	Reset
15:3	-	-		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: (a) I2C Ctrl is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, the controller is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in I2C Ctrl (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the the controller has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, the controller is deemed to have been disabled when there is master activity, or when the I2C bus is idle. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.	0x0
0	R	IC_EN	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, the controller is deemed to be in an enabled state. When read as 0, the controller is deemed completely inactive. NOTE: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1).	0x0

Table 291: I2C_IC_FS_SPKLEN_REG (0x500013A0)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		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 1; hardware prevents values less than this being written, and if attempted results in 1 being set.	0x1

28.11 Keyboard Registers

Table 292: Register map KBRD

Address	Register	Description
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

Table 293: GPIO_IRQ0_IN_SEL_REG (0x50001400)

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

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

Table 294: GPIO_IRQ1_IN_SEL_REG (0x50001402)

Bit	Mode	Symbol	Description	Reset
15:6	-	-		0x0
5:0	R/W	KBRD_IRQ1_SEL	see KBRD_IRQ0_SEL	0x0

Table 295: GPIO_IRQ2_IN_SEL_REG (0x50001404)

Bit	Mode	Symbol	Description	Reset
15:6	-	-		0x0
5:0	R/W	KBRD_IRQ2_SEL	see KBRD_IRQ0_SEL	0x0

Table 296: GPIO_IRQ3_IN_SEL_REG (0x50001406)

Bit	Mode	Symbol	Description	Reset
15:6	-	-		0x0
5:0	R/W	KBRD_IRQ3_SEL	see KBRD_IRQ0_SEL	0x0

Table 297: GPIO_IRQ4_IN_SEL_REG (0x50001408)

Bit	Mode	Symbol	Description	Reset
15:6	-	-		0x0
5:0	R/W	KBRD_IRQ4_SEL	see KBRD_IRQ0_SEL	0x0

Table 298: GPIO_DEBOUNCE_REG (0x5000140C)

Bit	Mode	Symbol	Description	Reset
15:14	-	-		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	-	-		0x0
5:0	R/W	DEB_VALUE	Keyboard debounce time if enabled. Generate KEYB_INT after specified time. Debounce time: N*1 ms. N =0..63	0x0

Table 299: GPIO_RESET_IRQ_REG (0x5000140E)

Bit	Mode	Symbol	Description	Reset
15:6	-	-		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

Bit	Mode	Symbol	Description	Reset
0	R0/W	RESET_GPIO0_IRQ	writing a 1 to this bit will reset the GPIO0 IRQ. Reading returns 0.	0x0

Table 300: GPIO_INT_LEVEL_CTRL_REG (0x50001410)

Bit	Mode	Symbol	Description	Reset
15:14	-	-		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 1: wait for key release after interrupt was reset for IRQ0	0x0
7:6	-	-		0x0
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. 1 = selected input will generate GPIO IRQ0 if that input is low.	0x0

Table 301: KBRD_IRQ_IN_SEL0_REG (0x50001412)

Bit	Mode	Symbol	Description	Reset
15	R/W	KBRD_REL	0 = No interrupt on key release 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. 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. Repeat time: N*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

Bit	Mode	Symbol	Description	Reset
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 302: 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
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 303: 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

28.12 OTP Controller Registers

Table 304: Register map OTPC

Address	Register	Description
0x07F40000	OTPC_MODE_REG	Mode register
0x07F40004	OTPC_PCTRL_REG	Bit-programming control register
0x07F40008	OTPC_STAT_REG	Status register
0x07F4000C	OTPC_AHBADR_REG	AHB master start address
0x07F40010	OTPC_CELADR_REG	Macrocell start address
0x07F40014	OTPC_NWORDS_REG	Number of words
0x07F40018	OTPC_FFPRT_REG	Ports access to fifo logic
0x07F4001C	OTPC_FFRD_REG	The data which have taken with the latest read from the OTPC_FFPRT_REG
0x07F40020	OTPC_PWORDL_REG	The 32 lower bits of the 64-bit word that will be programmed, when the MPROG mode is used.
0x07F40024	OTPC_PWORDH_REG	The 32 higher bits of the 64-bit word that will be programmed, when the MPROG mode is used.
0x07F40028	OTPC_TIM1_REG	Various timing parameters of the OTP cell.
0x07F4002C	OTPC_TIM2_REG	Various timing parameters of the OTP cell.
0x07F40034	OTPC_BCSTS_REG	Blank check status register

Table 305: OTPC_MODE_REG (0x07F40000)

Bit	Mode	Symbol	Description	Reset
31:15	-	-		0x0
14:12	R/W	OTPC_MODE_BCHK_THR	The maximum number of OTP words (minus one) that can be repaired during the blank check, when OTPC_MODE_BCHK_ALG=3.	0x0

Bit	Mode	Symbol	Description	Reset
11:10	R/W	OTPC_MODE_BCHK_ALG	Defines that algorithm that will be used for the implementation of the blank check (GUSTOC_MODE_MODE=0x5). Has no effect for the other modes of operation of the controller. 0x0: The OTP memory that has in at least one word with two or more bits that are not blank, is recognized by the controller as faulty (GUSTOC_STAT_TERROR=1). There is no addition of a repair record for the non-blank words. The test terminated immediately with the recognition of the first word with two or more non-zero bits. 0x1: The OTP memory that has even one bit that is not blank, is recognized by the controller as faulty (GUSTOC_STAT_TERROR=1). The test terminated immediately with the recognition of the first word with a non-zero bit. 0x2: The controller checks all the positions of the OTP memory and updates the two counters GUSTOC_BCHK_EQ1_WCNT and GUSTOC_BCHK_BEQ2_WCNT. After the completion of the test the GUSTOC_STAT_TERROR is always 0. 0x3: The controller creates a repair record for each OTP word that is not blank. The maximum number of the words that can be repaired is defined by the GUSTOC_MODE_BCHK_THR. If there are more than non-blank (GUSTOC_MODE_BCHK_THR+1) OTP words, the OTP memory is recognized as faulty (GUSTOC_STAT_TERROR=1)	0x0
9	R/W	OTPC_MODE_RLD_RR_REQ	Write with 1 in order to be requested the reloading of the repair records. The reloading of the repair records will be performed at the next enabling of the OTP cell. That means that first the controller should be configured to the STBY mode and after should be activated any other mode. The hardware will clear this register, when the reloading will be performed. The reloading has meaning only if the repair records have been updated manually (MPROG mode).	0x0
8	R/W	OTPC_MODE_USE_SP_ROWS	Selects the memory area of the OTP cell that will be used. 0: Uses the normal memory area of the OTP cell 1: Uses the spare rows of the OTP cell This selection has meaning only if the mode of the controller is not TDEC and TWR. The controller should be in STBY mode, in order to takes into account this bit. The selection will take effect at the next mode that will be enabled.	0x0
7	R/W	-		0x0

Bit	Mode	Symbol	Description	Reset
6	R/W	OTPC_MODE_ERR_RESP_DIS	When is performed a read from the OTP memory in the MREAD mode, a double error is likely be detected during the retrieving of the data from the OTP. This error condition is always indicated in the status bit OTPC_STAT_REG[OTPC_STAT_RERROR]. However, the OTP controller has also the ability to indicates this error condition, by generating an ERROR response in the AHB bus. The generation of the ERROR response can be avoided with the help of this configuration bit. 0: The OTP controller generates an ERROR response in the AHB bus, when a double error is detected during a reading in MREAD mode. The OTPC_STAT_REG[OTPC_STAT_RERROR] is also updated. The receiving of an ERROR response by the CPU causes a Hard Fault exception in the CPU. 1: Only the OTPC_STAT_REG[OTPC_STAT_RERROR] is updated in a case of such error. The OTP controller will not generate an ERROR response in the AHB bus.	0x0
5	R0/W	OTPC_MODE_FIFO_FLUSH	By writing with 1, removes any content from the fifo. This bit returns automatically to value 0.	0x0
4	R/W	OTPC_MODE_USE_DMA	Selects the use of the dma, when the controller is configured in one of the modes: AREAD or APROG. 0: The dma is not used. The data should be transferred from/to controller through the register OTPC_FFPRT_REG. 1: The dma is used. The data transfers from/to controller are performed automatically, with the help of the internal DMA of the OTP controller. The AHB base address should be configured in register OTPC_AHBADR_REG, before the selection of one of the two modes: AREAD or APROG.	0x0
3	-	-		0x0
2:0	R/W	OTPC_MODE_MODE	Defines the mode of operation of the OTPC controller. The encoding of the modes is as follows: 0x0: STBY mode 0x1: MREAD mode 0x2: MPROG mode 0x3: AREAD mode 0x4: APROG mode 0x5: TBLANK mode 0x6: TDEC mode 0x7: TWR mode	0x0

Table 306: OTPC_PCTRL_REG (0x07F40004)

Bit	Mode	Symbol	Description	Reset
31:16	-	-		0x0

Bit	Mode	Symbol	Description	Reset
15	R0/W	OTPC_PCTRL_PSTART	Write with '1' to trigger the programming of one OTP word, in the case where the MPROG mode is selected. The bit is cleared automatically. The 64-bits that will be programmed into the OTP memory are contained into the two registers OTPC_PWORDx_REG. This bit should be used when a new programming is initiated, but also when the programming must be retried. The OTPC_PCTRL_WADDR defines the OTP position where will be performed the programming.	0x0
14	R/W	OTPC_PCTRL_PRETRY	It distinguishes the first attempt of a programming of an OTP position, from a retry of programming. 0: A new value will be programmed in a blank OTP position. The hardware will try to write all the bits that are equal to '1'. 1: The programming that is applied is not the first attempt, but is a request for reprogramming. Will be processed only the bits that were failed to be programmed during the previous attempt. The hardware knows the bits that were failed during the previous attempt. The registers OTPC_PWORDx_REG should contain the 64 bits of the value that should be programmed, independent of the value of the OTPC_PCTRL_PRETRY bit. Also, the OTPC_PCTRL_WADDR should contain always the required OTP address. A retry of a programming should be requested only if the previous action was the first attempt of programming or a retry of programming. Should not be requested a retry if the first attempt has not been performed.	0x0
13	-	-		0x0
12:0	R/W	OTPC_PCTRL_WADDR	Defines the OTP position where will be programmed the 64-bits that are contained into the registers OTPC_PWORDx_REG. It points to a physical 72 bits OTP word.	0x0

Table 307: OTPC_STAT_REG (0x07F40008)

Bit	Mode	Symbol	Description	Reset
31:30	-	-		0x0
29:16	R	OTPC_STAT_NWORDS	It contains the "live" value of the number of (32 bits) words that remain to be processed by the controller.	0x0
15:12	-	-		0x0
11:8	R	OTPC_STAT_FWORDS	Indicates the number of words which contained in the fifo of the controller.	0x0
7	R	OTPC_STAT_RERROR	Indicates that during a normal reading (MREAD or AREAD) was reported a double error by the SECDED logic. That means that the data are corrupted. 0: The read data are considered as correct. 1: The SECDED logic detects a double error. This bit can be cleared only with a write with '1'.	0x0

Bit	Mode	Symbol	Description	Reset
6	R	OTPC_STAT_ARDY	Should be used to monitor the progress of the AREAD and APROG modes. 0: One of the APROG or AREAD mode is selected. The controller is busy. 1: The controller is not in an active AREAD or APROG mode.	0x1
5	R	OTPC_STAT_TERROR	Indicates the result of a test sequence. Should be checked after the end of a TBLANK, TDEC and TWR mode (OTPC_STAT_TRDY = 1). 0: The test sequence ends with no error. 1: The test sequence has failed.	0x0
4	R	OTPC_STAT_TRDY	Indicates the state of a test mode. Should be used to monitor the progress of the TBLANK, TDEC and TWR modes. 0: The controller is busy. One of the test modes is in progress. 1: There is no active test mode.	0x1
3	R	OTPC_STAT_PZERO	Indicates that the programming sequence has been avoided during a programming request, due to that the word that should be programmed is equal to zero. 0: At least one bit has been programmed into the OTP. 1: The programming has not been performed. All the bits of the word that should be programmed are equal to zero. When the controller is in MPROG mode, this bit can be checked after the end of the programming process (OTPC_STAT_PRDY = 1). During APROG mode, the value of this field it is normal to changing periodically. After the end of the APROG mode (OTPC_STAT_ARDY = 1), this field indicates that one or more of words that have been processed are equal to zero.	0x0
2	R	OTPC_STAT_PERR_COR	Indicates that a correctable error has been occurred during the word programming process. 0: There is no correctable error in the word-programming process. 1: The process of word - programming reported a correctable error. The correctable error occurs when exactly one bit in an OTP position cannot take the required value. This is not a critical failure in the programming process. The data can still be retrieved correctly by the OTP memory, due to that the error correcting algorithm can repair the corrupted bit. When the controller is in MPROG mode, this bit can be checked after the end of the programming process (OTPC_STAT_PRDY = 1). During APROG mode, the value of this field it is normal to changing periodically. After the end of the APROG mode (OTPC_STAT_ARDY = 1), this field indicates that one or more words had a correctable error.	0x0

Bit	Mode	Symbol	Description	Reset
1	R	OTPC_STAT_PERR_UNC	Indicates that an uncorrectable error has been occurred during the word programming process. 0: There is no uncorrectable error in the word-programming process. 1: The process of word-programming failed due to an uncorrectable error. An uncorrectable error is considered when two or more of the bits in an OTP position cannot take the required values. This is a critical failure in the programming process, which means that the data cannot corrected by the single error correcting algorithm. When the controller is in MPROG mode, this bit should be checked after the end of the programming process (OTPC_STAT_PRDY = 1). During APROG mode, the value of this field it is normal to changing periodically. After the end of the APROG mode (OTPC_STAT_ARDY = 1), this field indicates if the programming was failed or ended successfully.	0x0
0	R	OTPC_STAT_PRDY	Indicates the state of a bit-programming process. 0: The controller is busy. A bit-programming is in progress 1: The logic which performs bit-programming is idle. When the controller is in MPROG mode, this bit should be used to monitor the progress of a programming request. During APROG mode, the value of this field it is normal to changing periodically.	0x1

Table 308: OTPC_AHBADR_REG (0x07F4000C)

Bit	Mode	Symbol	Description	Reset
31:2	R/W	OTPC_AHBADR	It is the AHB address used by the AHB master interface of the controller (the bits [31:2]). The bits [1:0] of the address are considered always as equal to zero. The value of the register remains unchanged, by the internal logic of the controller.	0x1FF0000
1:0	R	-		0x0

Table 309: OTPC_CELADR_REG (0x07F40010)

Bit	Mode	Symbol	Description	Reset
31:30	-	-		0x0
29:16	R	OTPC_CELADR_LV	This is a readonly field that contains the "live" value of the OTP cell address as it is used by the hardware of the OTPC controller during the AREAD and the APROG modes. The value of the register is updated only while the OTPC is in AREAD or the APROG mode.	0x0
15:14	-	-		0x0

Bit	Mode	Symbol	Description	Reset
13:0	R/W	OTPC_CELADR	It represents an OTP address, where the OTP word width should be considered equal to 32-bits. The physical word width of the OTP memory is 72 bits. The 8-bits of them are used for the implementation of an error correcting code and are not available for the application. The remaining 64 bits of the physical word are available for the application. The OTPC_CELADDR can distinguish the upper 32 bits from the lower 32 bits of the available for the application bits of the OTP word. When OTPC_CELADDR[0] = 1 the address refers to the upper 32 bits of the physical OTP address OTPC_CELADDR[14:1]. The register is used during the modes: AREAD and APROG. The value of the register remains unchanged, by the internal logic of the controller.	0x0

Table 310: OTPC_NWORDS_REG (0x07F40014)

Bit	Mode	Symbol	Description	Reset
31:14	-	-		0x0
13:0	R/W	OTPC_NWORDS	The number of words (minus one) for reading /programming during the AREAD/APROG mode. The width of the word should be considered equal to 32-bits. The value of the register remains unchanged, by the internal logic of the controller. During mirroring, this register reflects the current ammount of copied data.	0x0

Table 311: OTPC_FFPRT_REG (0x07F40018)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	OTPC_FFPRT	Provides access to the fifo through an access port. Write to this register with the corresponding data, when the APROG mode is selected and the dma is disabled. Read from this register the corresponding data, when the AREAD mode is selected and the dma is disabled. The software should check the OTPCC_STAT_FWORDS register for the availability of data/space, before accessing the fifo.	0x0

Table 312: OTPC_FFRD_REG (0x07F4001C)

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

Table 313: OTPC_PWORDL_REG (0x07F40020)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	OTPC_PWORDL	Contains the lower 32 bits that can be programmed with the help of the OTPC_PCTRL_REG, while the controller is in MPROG mode.	0x0

Table 314: OTPC_PWORDH_REG (0x07F40024)

Bit	Mode	Symbol	Description	Reset
31:0	R/W	OTPC_PWORDH	Contains the upper 32 bits that can be programmed with the help of the OTPC_PCTRL_REG, while the controller is in MPROG mode.	0x0

Table 315: OTPC_TIM1_REG (0x07F40028)

Bit	Mode	Symbol	Description	Reset
31	R/W	OTPC_TIM1_CC_T_25NS	The number of hclk_c clock periods (minus one) that give a time interval at least higher than 25 ns.	0x0
30:27	R/W	OTPC_TIM1_CC_T_200NS	The number of hclk_c clock periods (minus one) that give a time interval at least higher than 200 ns.	0x3
26:22	R/W	OTPC_TIM1_CC_T_500NS	The number of hclk_c clock periods (minus one) that give a time interval at least higher than 500 ns	0x8
21:16	R/W	OTPC_TIM1_CC_T_1US	The number of hclk_c clock periods (minus one) that give a time interval at least higher than 1 us.	0x10
15:8	R/W	OTPC_TIM1_CC_T_PW	The number of hclk_c clock periods (minus one) that give a time interval that is - at least higher than 4.8 us - and lower than 5.2 us It is preferred the programmed value to give a time interval equal to 5 us. It defines the duration of the programming pulse for every bit that written in the OTP cell.	0x4F
7:0	R/W	OTPC_TIM1_CC_T_CADX	The number of hclk_c clock periods (minus one) that give a time interval at least higher than 2 us. It is used as a wait time each time where the OTP cell is enabled.	0x20

Table 316: OTPC_TIM2_REG (0x07F4002C)

Bit	Mode	Symbol	Description	Reset
31:24	-	-		0x0

Bit	Mode	Symbol	Description	Reset
23	R/W	OTPC_TIM2_RDENL_PROT	This bit has meaning only when the OTPC_TIM1_CC_T_25NS = 1, otherwise has no functionality. 0: The minimum number of clock cycles for which the signal read_enable of the OTP memory stays inactive is one clock cycle. This is also applicable if OTPC_TIM1_CC_T_25NS = 0. 1: The minimum number of clock cycles for which the signal read_enable of the OTP memory stays inactive is two clock cycles. The controller adds one extra wait state in the AHB access , if it is required, in order to achieves this constraint. This setting is applicable only if OTPC_TIM1_CC_T_25NS = 1.	0x0
22:16	R/W	OTPC_TIM2_CC_T_BCHK	The number of hclk_c clock periods (minus one) that give a time interval between 100 ns and 200 ns. This time interval is used for the reading of the contents of the OTP cell during the TBLANK mode.	0x1
15:10	-	-		0x0
9:0	R/W	OTPC_TIM2_CC_STBY_THR	This register controls a power saving feature, which is applicable only in MREAD mode. The controller monitors the accesses in the OTP cell. If there is no access for more than OTPC_TIM2_CC_STBY_THR hclk_c clock cycles, the OTP cell goes to the standby while the controller itself remains in the MREAD mode. The OTP cell will be enabled again when will be applied a new read request. The enabling of the OTP cell has a cost of 2 us (OTPC_TIM1_CC_T_CADX hclk_c clock cycles). When OTPC_TIM2_CC_STBY_THR = 0 the power saving feature is disabled and the OTP cell remains active while the controller is in MREAD mode.	0x0

Table 317: OTPC_BCSTS_REG (0x07F40034)

Bit	Mode	Symbol	Description	Reset
31:30	-	-		0x0
29:16	R	OTPC_BCHK_BEQ2_WCNT	This is a counter that counts the OTP words that have two or more non-blank bits. It is updated only after the execution of the blank check test (OTPC_MODE_MODE=0x5).	0x0
15:14	-	-		0x0
13:0	R	OTPC_BCHK_EQ1_WCNT	This is a counter that counts the OTP words that have exactly one non-blank bit. It is updated only after the execution of the blank check test (OTPC_MODE_MODE=0x5).	0x0

28.13 Quadrature Decoder Registers

Table 318: Register map QDEC

Address	Register	Description
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
0x5000020A	QDEC_ZCNT_REG	Z_counter

Table 319: QDEC_CTRL_REG (0x50000200)

Bit	Mode	Symbol	Description	Reset
15:10	R/W	-		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 occured.	0x0
1	R/W	QD_IRQ_CLR	Writing 1 to this bit clears the interrupt. This bit is autocleared	0x0
0	R/W	QD_IRQ_MASK	0: interrupt is masked 1: interrupt is enabled	0x0

Table 320: 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 321: 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 322: 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 323: QDEC_CTRL2_REG (0x50000208)

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

Table 324: 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

28.14 SPI Interface Registers

Table 325: Register map SPI

Address	Register	Description
0x50001200	SPI_CTRL_REG	SPI control register 0
0x50001202	SPI_RX_TX_REG0	SPI RX/TX register0
0x50001204	SPI_RX_TX_REG1	SPI RX/TX register1
0x50001206	SPI_CLEAR_INT_REG	SPI clear interrupt register
0x50001208	SPI_CTRL_REG1	SPI control register 1

Table 326: 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. 1 = SPI_EN pin enabled in slave mode.	0x0
14	R/W	SPI_MINT	0 = Disable SPI_INT_BIT to the Interrupt Controller 1 = Enable SPI_INT_BIT to the Interrupt Controller	0x0
13	R	SPI_INT_BIT	0 = RX Register or FIFO is empty. 1 = SPI interrupt. Data has been transmitted and receivedMust 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. 1 = TX-FIFO is full, data can not be written.	0x0
10	R/W	SPI_FORCE_DO	0 = normal operation 1 = Force SPIDO output level to value of SPI_DO.	0x0
9	R/W	SPI_RST	0 = normal operation 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 01 = 16 bit mode, only SPI_RX_TX_REG0 used 10 = 32 bits mode, SPI_RX_TX_REG0 & SPI_RX_TX_REG1 used 11 = 9 bits mode. Only valid in master mode.	0x0
6	R/W	SPI_SMN	Master/slave mode 0 = Master, 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) 01 = (XTAL) / (CLK_PER_REG *4) 10 = (XTAL) / (CLK_PER_REG *2) 11 = (XTAL) / (CLK_PER_REG *14)	0x0
2	R/W	SPI_POL	Select SPI_CLK polarity. 0 = SPI_CLK is initially low. 1 = SPI_CLK is initially high.	0x0

Bit	Mode	Symbol	Description	Reset
1	R/W	SPI_PHA	Select SPI_CLK phase. See functional timing diagrams in SPI chapter	0x0
0	R/W	SPI_ON	0 = SPI Module switched off (power saving). Everything is reset except SPI_CTRL_REG0 and SPI_CTRL_REG1. When this bit is cleared the SPI will remain active in master mode until the shift register and holding register are both empty. 1 = SPI Module switched on. Should only be set after all control bits have their desired values. So two writes are needed!	0x0

Table 327: 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) Read: SPI_RX_REG0 input register 0 (RX-FIFO) In 8 or 9 bits mode bits 15 to 8 are not used, they contain old data.	0x0

Table 328: 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) Read: SPI_RX_REG1 input register 1 (MSB's of RX-FIFO) In 8 or 9 or 16 bits mode bits this register is not used.	0x0

Table 329: 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] Reading returns 0.	0x0

Table 330: SPI_CTRL_REG1 (0x50001208)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R/W	SPI_9BIT_VAL	Determines the value of the first bit in 9 bits SPI mode.	0x0

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. 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. 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). 1: RX-FIFO used (Read Only Mode) TX-FIFO single depth, no flow control 2: TX-FIFO used (Write Only Mode), RX-FIFO single depth, no flow control 3: No FIFOs used (backwards compatible mode)	0x3

28.15 Timer and Triple PWM Registers

Table 331: Register map Timer+3PWM

Address	Register	Description
0x50003400	TIMER0_CTRL_REG	Timer0 control register
0x50003402	TIMER0_ON_REG	Timer0 on control register
0x50003404	TIMER0_RELOAD_M_REG	16 bits reload value for Timer0
0x50003406	TIMER0_RELOAD_N_REG	16 bits reload value for Timer0
0x50003408	PWM2_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 332: TIMER0_CTRL_REG (0x50003400)

Bit	Mode	Symbol	Description	Reset
15:4	-	-		0x0
3	R/W	PWM_MODE	0 = PWM signals are '1' during high time. 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. 0 = Timer0 uses selected clock frequency divided by 10. 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. 0 = Timer0 uses 32 kHz (slow) clock frequency.	0x0
0	R/W	TIM0_CTRL	0 = Timer0 is off and in reset state. 1 = Timer0 is running.	0x0

Table 333: TIMER0_ON_REG (0x50003402)

Bit	Mode	Symbol	Description	Reset
15:0	R0/W	TIM0_ON	Timer0 On reload value: If read the actual counter value ON_CNTer is returned	0x0

Table 334: TIMER0_RELOAD_M_REG (0x50003404)

Bit	Mode	Symbol	Description	Reset
15:0	R0/W	TIM0_M	Timer0 'high' reload valueIf read the actual counter value T0_CNTer is returned	0x0

Table 335: TIMER0_RELOAD_N_REG (0x50003406)

Bit	Mode	Symbol	Description	Reset
15:0	R0/W	TIM0_N	Timer0 'low' reload value: If read the actual counter value T0_CNTer is returned	0x0

Table 336: PWM2_DUTY_CYCLE (0x50003408)

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

Table 337: PWM3_DUTY_CYCLE (0x5000340A)

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

Table 338: PWM4_DUTY_CYCLE (0x5000340C)

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

Table 339: TRIPLE_PWM_FREQUENCY (0x5000340E)

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

Table 340: 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_ENABLE	'1' = PWM 2 3 4 is enabled	0x0

28.16 UART Interface Registers

Table 341: Register map UART

Address	Register	Description
0x50001000	UART_RBR_THR_DLL_REG	Receive Buffer Register/Transmit Holding Register/Divisor Latch Low
0x50001004	UART_IER_DLH_REG	Interrupt Enable Register/Divisor Latch High
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
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
0x50001070	UART_FAR_REG	FIFO Access Register
0x5000107C	UART_USR_REG	UART Status Register

Address	Register	Description
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
0x500010A8	UART_DMASA_REG	DMA Software Acknowledge
0x500010C0	UART_DLF_REG	Divisor Latch Fraction Register
0x500010F4	UART_CPR_REG	Component Parameter Register
0x500010F8	UART_UCV_REG	Component Version
0x500010FC	UART_CTR_REG	Component Type Register
0x50001100	UART2_RBR_THR_DLL_REG	Receive Buffer Register/Transmit Holding Register/Divisor Latch Low
0x50001104	UART2_IER_DLH_REG	Interrupt Enable Register/Divisor Latch High
0x50001108	UART2_IIR_FCR_REG	Interrupt Identification Register/FIFO Control Register
0x5000110C	UART2_LCR_REG	Line Control Register
0x50001110	UART2_MCR_REG	Modem Control Register
0x50001114	UART2_LSR_REG	Line Status Register
0x50001118	UART2_MSR_REG	Modem Status Register
0x5000111C	UART2_SCR_REG	Scratchpad Register
0x50001130	UART2_SRBR_STHR0_REG	Shadow Receive/Transmit Buffer Register
0x50001134	UART2_SRBR_STHR1_REG	Shadow Receive/Transmit Buffer Register
0x50001138	UART2_SRBR_STHR2_REG	Shadow Receive/Transmit Buffer Register
0x5000113C	UART2_SRBR_STHR3_REG	Shadow Receive/Transmit Buffer Register
0x50001140	UART2_SRBR_STHR4_REG	Shadow Receive/Transmit Buffer Register
0x50001144	UART2_SRBR_STHR5_REG	Shadow Receive/Transmit Buffer Register
0x50001148	UART2_SRBR_STHR6_REG	Shadow Receive/Transmit Buffer Register
0x5000114C	UART2_SRBR_STHR7_REG	Shadow Receive/Transmit Buffer Register
0x50001150	UART2_SRBR_STHR8_REG	Shadow Receive/Transmit Buffer Register

Address	Register	Description
0x50001154	UART2_SRBR_STHR9_REG	Shadow Receive/Transmit Buffer Register
0x50001158	UART2_SRBR_STHR10_REG	Shadow Receive/Transmit Buffer Register
0x5000115C	UART2_SRBR_STHR11_REG	Shadow Receive/Transmit Buffer Register
0x50001160	UART2_SRBR_STHR12_REG	Shadow Receive/Transmit Buffer Register
0x50001164	UART2_SRBR_STHR13_REG	Shadow Receive/Transmit Buffer Register
0x50001168	UART2_SRBR_STHR14_REG	Shadow Receive/Transmit Buffer Register
0x5000116C	UART2_SRBR_STHR15_REG	Shadow Receive/Transmit Buffer Register
0x50001170	UART2_FAR_REG	FIFO Access Register
0x5000117C	UART2_USR_REG	UART Status Register
0x50001180	UART2_TFL_REG	Transmit FIFO Level
0x50001184	UART2_RFL_REG	Receive FIFO Level
0x50001188	UART2_SRR_REG	Software Reset Register.
0x5000118C	UART2_SRTS_REG	Shadow Request to Send
0x50001190	UART2_SBCR_REG	Shadow Break Control Register
0x50001194	UART2_SDMAM_REG	Shadow DMA Mode
0x50001198	UART2_SFE_REG	Shadow FIFO Enable
0x5000119C	UART2_SRT_REG	Shadow RCVR Trigger
0x500011A0	UART2_STET_REG	Shadow TX Empty Trigger
0x500011A4	UART2_HTX_REG	Halt TX
0x500011A8	UART2_DMASA_REG	DMA Software Acknowledge
0x500011C0	UART2_DLF_REG	Divisor Latch Fraction Register
0x500011F4	UART2_CPR_REG	Component Parameter Register
0x500011F8	UART2_UCV_REG	Component Version
0x500011FC	UART2_CTR_REG	Component Type Register

Table 342: UART_RBR_THR_DLL_REG (0x50001000)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	RBR_THR_DLL	**Receive Buffer Register: (RBR).** 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: (THR)** 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): (DLL)** 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: baud rate = (serial clock freq) / (16 * divisor) 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 Divisor Latch is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data. For the Divisor Latch (High) bits, see register UART_IER_DLH_REG.	0x0

Table 343: UART_IER_DLH_REG (0x50001004)

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

Bit	Mode	Symbol	Description	Reset
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): DLH7**, Bit 7 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	0x0
6:4	R/W	dlh6_4	**Divisor Latch (High): DLH6 to DLH4**, Bits 6 to 4 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set, otherwise, this field is reserved. See register UART_RBR_THR_DLL_REG.	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): DLH3**, Bit 3 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	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): DLH2**, Bit 2 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	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): DLH1**, Bit 1 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	0x0

Bit	Mode	Symbol	Description	Reset
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): DLH0**, Bit 0 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	0x0

Table 344: UART_IIR_FCR_REG (0x50001008)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	IIR_FCR	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:0001 = no interrupt pending. 0010 = THR empty. 0100 = received data available. 0110 = receiver line status. 0111 = busy detect. 1100 = character timeout. FIFO Control Register Bits[7:6], RCVR Trigger (or RT):. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode it is used to determine when the rts_n signal will be de-asserted. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than full Bits[5:4], TX Empty Trigger (or TET): This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. It also determines when the dma_tx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full Bit[3], DMA Mode (or DMAM): This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. 0 = mode 0 1 = mode 1 Bit[2], XMIT FIFO Reset (or XFIFOR): This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[1], RCVR FIFO Reset (or RFIFOR): This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[0], FIFO Enable (or FIFOE): This enables/disables the transmit (XMIT) and receive (RCVR) FIFO's. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFO's will be reset.	0x0

Table 345: UART_LCR_REG (0x5000100C)

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

Bit	Mode	Symbol	Description	Reset
7	R/W	UART_DLAB	Divisor Latch Access Bit. 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. This bit must be cleared after initial baud rate setup in order to access other registers.	0x0
6	R/W	UART_BC	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.	0x0
5	-	-		0x0
4	R/W	UART_EPS	Even Parity Select. Writeable only when UART is not busy (USR[0] is zero). This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.	0x0
3	R/W	UART_PEN	Parity Enable. Writeable only when UART is not busy (USR[0] is zero) This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled	0x0
2	R/W	UART_STOP	Number of stop bits. 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. 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. 0 = 1 stop bit 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. 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: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	0x0

Table 346: UART_MCR_REG (0x50001010)

Bit	Mode	Symbol	Description	Reset
15:7	-	-		0x0
6	R/W	UART_SIRE	SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol". 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled	0x0
5	R/W	UART_AFCE	Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in "Auto Flow Control". 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled	0x0
4	R/W	UART_LB	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. 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. If operating in infrared mode (SIR_MODE active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.	0x0
3	R/W	UART_OUT2	OUT2. 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: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input.	0x0
2	R/W	UART_OUT1	OUT1. 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: 0 = out1_n de-asserted (logic 1) 1 = out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input.	0x0

Bit	Mode	Symbol	Description	Reset
1	R/W	UART_RTS	Request to Send. 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. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.	0x0
0	R/W	-		0x0

Table 347: UART_LSR_REG (0x50001014)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7	R	UART_RFE	Receiver FIFO Error bit. 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. 0 = no error in RX FIFO 1 = error in RX FIFO 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. 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. 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. 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

Bit	Mode	Symbol	Description	Reset
4	R	UART_BI	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. 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. 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. 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. 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
3	R	UART_FE	Framing Error bit. 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. 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. 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]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit.	0x0
2	R	UART_PE	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. 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. 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]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit.	0x0

Bit	Mode	Symbol	Description	Reset
1	R	UART_OE	Overrun error bit. 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. 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. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit.	0x0
0	R	UART_DR	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.	0x0

Table 348: UART_MSR_REG (0x50001018)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R	UART_CTS	Clear to Send. 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. 0 = cts_n input is de-asserted (logic 1) 1 = cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).	0x0
3:1	-	-		0x0
0	R	UART_DCTS	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. 0 = no change on cts_n since last read of MSR 1 = change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). 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 349: UART_SCR_REG (0x5000101C)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	UART_SCRATCH_PAD	This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART Ctrl.	0x0

Table 350: UART_SRBR_STHR0_REG (0x50001030)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 351: UART_SRBR_STHR1_REG (0x50001034)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 352: UART_SRBR_STHR2_REG (0x50001038)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 353: UART_SRBR_STHR3_REG (0x5000103C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 354: UART_SRBR_STHR4_REG (0x50001040)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 355: UART_SRBR_STHR5_REG (0x50001044)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 356: UART_SRBR_STHR6_REG (0x50001048)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 357: UART_SRBR_STHR7_REG (0x5000104C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 358: UART_SRBR_STHR8_REG (0x50001050)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 359: UART_SRBR_STHR9_REG (0x50001054)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 360: UART_SRBR_STHR10_REG (0x50001058)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 361: UART_SRBR_STHR11_REG (0x5000105C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 362: UART_SRBR_STHR12_REG (0x50001060)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 363: UART_SRBR_STHR13_REG (0x50001064)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 364: UART_SRBR_STHR14_REG (0x50001068)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 365: UART_SRBR_STHR15_REG (0x5000106C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 366: UART_FAR_REG (0x50001070)

Bit	Mode	Symbol	Description	Reset
0	R	UART_FAR	Description: Writes will have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFO's are implemented and enabled. When FIFO's are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFO's are treated as empty.	0x0

Table 367: UART_USR_REG (0x5000107C)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R	UART_RFF	Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.	0x0
3	R	UART_RFNE	Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.	0x0
2	R	UART_TFE	Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.	0x1
1	R	UART_TFNF	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.	0x1
0	R	UART_BUSY	UART Busy. This indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive. 0 - DW_apb_uart is idle or inactive 1 - DW_apb_uart is busy (actively transferring data) Note that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in the THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled) the assertion of this bit will also be delayed by several cycles of the slower clock.	0x0

Table 368: UART_TFL_REG (0x50001080)

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

Table 369: UART_RFL_REG (0x50001084)

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

Table 370: UART_SRR_REG (0x50001088)

Bit	Mode	Symbol	Description	Reset
15:3	-	-		0x0
2	W	UART_XFR	XMIT FIFO Reset. 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. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.	0x0
0	W	UART_UR	UART Reset. This asynchronously resets the UART Ctrl and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.	0x0

Table 371: UART_SRTS_REG (0x5000108C)

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

Bit	Mode	Symbol	Description	Reset
0	R/W	UART_SHADOW_REQUEST_TO_SEND	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART Ctrl is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input.	0x0

Table 372: UART_SBCR_REG (0x50001090)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
0	R/W	UART_SHADOW_BREAK_CONTROL	Shadow Break Control Bit. 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. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.	0x0

Table 373: UART_SDMAM_REG (0x50001094)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
0	R/W	UART_SHADOW_DMA_MODE	Shadow DMA Mode. 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. 0 = mode 0 1 = mode 1	0x0

Table 374: UART_SFE_REG (0x50001098)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
0	R/W	UART_SHADOW_FIFO_ENABLE	Shadow FIFO Enable. 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 375: UART_SRT_REG (0x5000109C)

Bit	Mode	Symbol	Description	Reset
15:2	-	-		0x0
1:0	R/W	UART_SHADOW_RCVR_TRIGGER	Shadow RCVR Trigger. 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. 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: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full	0x0

Table 376: UART_STET_REG (0x500010A0)

Bit	Mode	Symbol	Description	Reset
15:2	-	-		0x0

Bit	Mode	Symbol	Description	Reset
1:0	R/W	UART_SHADOW_TX_EMPTY_TRIGGER	Shadow TX Empty Trigger. 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. 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: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full	0x0

Table 377: UART_HTX_REG (0x500010A4)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
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. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.	0x0

Table 378: UART_DMASA_REG (0x500010A8)

Bit	Mode	Symbol	Description	Reset
0	W	DMASA	This register is use to perform DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This will cause the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing' and it is not necessary to clear this bit.	0x0

Table 379: UART_DLF_REG (0x500010C0)

Bit	Mode	Symbol	Description	Reset
3:0	R/W	UART_DLF	The fractional value is added to integer value set by DLH, DLL. Fractional value is equal UART_DLF/16	0x0

Table 380: UART_CPR_REG (0x500010F4)

Bit	Mode	Symbol	Description	Reset
15:0	R	CPR	Component Parameter Register	0x3D71

Table 381: UART_UCV_REG (0x500010F8)

Bit	Mode	Symbol	Description	Reset
15:0	R	UCV	Component Version	0x3331352A

Table 382: UART_CTR_REG (0x500010FC)

Bit	Mode	Symbol	Description	Reset
15:0	R	CTR	Component Type Register	0x44570110

Table 383: UART2_RBR_THR_DLL_REG (0x50001100)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	RBR_THR_DLL	**Receive Buffer Register: (RBR).** 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: (THR)** 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): (DLL)** 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: baud rate = (serial clock freq) / (16 * divisor) 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 Divisor Latch is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data. For the Divisor Latch (High) bits, see register UART_IER_DLH_REG.	0x0

Table 384: UART2_IER_DLH_REG (0x50001104)

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

Bit	Mode	Symbol	Description	Reset
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): DLH7**, Bit 7 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	0x0
6:4	R/W	dlh6_4	**Divisor Latch (High): DLH6 to DLH4**, Bits 6 to 4 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set, otherwise, this field is reserved. See register UART_RBR_THR_DLL_REG.	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): DLH3**, Bit 3 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	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): DLH2**, Bit 2 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	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): DLH1**, Bit 1 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	0x0

Bit	Mode	Symbol	Description	Reset
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): DLH0**, Bit 0 of the upper part of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. See register UART_RBR_THR_DLL_REG.	0x0

Table 385: UART2_IIR_FCR_REG (0x50001108)

Bit	Mode	Symbol	Description	Reset
15:0	R/W	IIR_FCR	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:0001 = no interrupt pending. 0010 = THR empty. 0100 = received data available. 0110 = receiver line status. 0111 = busy detect. 1100 = character timeout. FIFO Control Register Bits[7:6], RCVR Trigger (or RT):. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode it is used to determine when the rts_n signal will be de-asserted. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than full Bits[5:4], TX Empty Trigger (or TET): This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. It also determines when the dma_tx_req_n signal will be asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full Bit[3], DMA Mode (or DMAM): This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals. 0 = mode 0 1 = mode 1 Bit[2], XMIT FIFO Reset (or XFIFOR): This resets the control portion of the transmit FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[1], RCVR FIFO Reset (or RFIFOR): This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing' and it is not necessary to clear this bit. Bit[0], FIFO Enable (or FIFOE): This enables/disables the transmit (XMIT) and receive (RCVR) FIFO's. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFO's will be reset.	0x0

Table 386: UART2_LCR_REG (0x5000110C)

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

Bit	Mode	Symbol	Description	Reset
7	R/W	UART_DLAB	Divisor Latch Access Bit. 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. This bit must be cleared after initial baud rate setup in order to access other registers.	0x0
6	R/W	UART_BC	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.	0x0
5	-	-		0x0
4	R/W	UART_EPS	Even Parity Select. Writeable only when UART is not busy (USR[0] is zero). This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.	0x0
3	R/W	UART_PEN	Parity Enable. Writeable only when UART is not busy (USR[0] is zero) This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled	0x0
2	R/W	UART_STOP	Number of stop bits. 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. 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. 0 = 1 stop bit 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. 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: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits	0x0

Table 387: UART2_MCR_REG (0x50001110)

Bit	Mode	Symbol	Description	Reset
15:7	-	-		0x0
6	R/W	UART_SIRE	SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol". 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled	0x0
5	R/W	UART_AFCE	Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in "Auto Flow Control". 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled	0x0
4	R/W	UART_LB	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. 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. If operating in infrared mode (SIR_MODE active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.	0x0
3	R/W	UART_OUT2	OUT2. 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: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input.	0x0
2	R/W	UART_OUT1	OUT1. 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: 0 = out1_n de-asserted (logic 1) 1 = out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input.	0x0

Bit	Mode	Symbol	Description	Reset
1	R/W	UART_RTS	Request to Send. 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. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.	0x0
0	R/W	-		0x0

Table 388: UART2_LSR_REG (0x50001114)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7	R	UART_RFE	Receiver FIFO Error bit. 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. 0 = no error in RX FIFO 1 = error in RX FIFO 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. 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. 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. 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

Bit	Mode	Symbol	Description	Reset
4	R	UART_BI	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. 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. 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. 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. 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
3	R	UART_FE	Framing Error bit. 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. 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. 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]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit.	0x0
2	R	UART_PE	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. 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. 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]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit.	0x0

Bit	Mode	Symbol	Description	Reset
1	R	UART_OE	Overrun error bit. 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. 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. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit.	0x0
0	R	UART_DR	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.	0x0

Table 389: UART2_MSR_REG (0x50001118)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R	UART_CTS	Clear to Send. 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. 0 = cts_n input is de-asserted (logic 1) 1 = cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).	0x0
3:1	-	-		0x0
0	R	UART_DCTS	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. 0 = no change on cts_n since last read of MSR 1 = change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). 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 390: UART2_SCR_REG (0x5000111C)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	UART_SCRATCH_PAD	This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART Ctrl.	0x0

Table 391: UART2_SRBR_STHR0_REG (0x50001130)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 392: UART2_SRBR_STHR1_REG (0x50001134)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 393: UART2_SRBR_STHR2_REG (0x50001138)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 394: UART2_SRBR_STHR3_REG (0x5000113C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 395: UART2_SRBR_STHR4_REG (0x50001140)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 396: UART2_SRBR_STHR5_REG (0x50001144)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 397: UART2_SRBR_STHR6_REG (0x50001148)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 398: UART2_SRBR_STHR7_REG (0x5000114C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 399: UART2_SRBR_STHR8_REG (0x50001150)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 400: UART2_SRBR_STHR9_REG (0x50001154)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 401: UART2_SRBR_STHR10_REG (0x50001158)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 402: UART2_SRBR_STHR11_REG (0x5000115C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 403: UART2_SRBR_STHR12_REG (0x50001160)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 404: UART2_SRBR_STHR13_REG (0x50001164)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 405: UART2_SRBR_STHR14_REG (0x50001168)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 406: UART2_SRBR_STHR15_REG (0x5000116C)

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

Bit	Mode	Symbol	Description	Reset
7:0	R/W	SRBR_STHRx	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. 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.	0x0

Table 407: UART2_FAR_REG (0x50001170)

Bit	Mode	Symbol	Description	Reset
0	R	UART_FAR	Description: Writes will have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFO's are implemented and enabled. When FIFO's are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFO's are treated as empty.	0x0

Table 408: UART2_USR_REG (0x5000117C)

Bit	Mode	Symbol	Description	Reset
15:5	-	-		0x0
4	R	UART_RFF	Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.	0x0
3	R	UART_RFNE	Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.	0x0
2	R	UART_TFE	Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.	0x1
1	R	UART_TFNF	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.	0x1
0	R	UART_BUSY	UART Busy. This indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive. 0 - DW_apb_uart is idle or inactive 1 - DW_apb_uart is busy (actively transferring data) Note that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in the THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled) the assertion of this bit will also be delayed by several cycles of the slower clock.	0x0

Table 409: UART2_TFL_REG (0x50001180)

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

Table 410: UART2_RFL_REG (0x50001184)

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

Table 411: UART2_SRR_REG (0x50001188)

Bit	Mode	Symbol	Description	Reset
15:3	-	-		0x0
2	W	UART_XFR	XMIT FIFO Reset. 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. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.	0x0
0	W	UART_UR	UART Reset. This asynchronously resets the UART Ctrl and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.	0x0

Table 412: UART2_SRTS_REG (0x5000118C)

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

Bit	Mode	Symbol	Description	Reset
0	R/W	UART_SHADOW_REQUEST_TO_SEND	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART Ctrl is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input.	0x0

Table 413: UART2_SBCR_REG (0x50001190)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
0	R/W	UART_SHADOW_BREAK_CONTROL	Shadow Break Control Bit. 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. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.	0x0

Table 414: UART2_SDMAM_REG (0x50001194)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
0	R/W	UART_SHADOW_DMA_MODE	Shadow DMA Mode. 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. 0 = mode 0 1 = mode 1	0x0

Table 415: UART2_SFE_REG (0x50001198)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
0	R/W	UART_SHADOW_FIFO_ENABLE	Shadow FIFO Enable. 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 416: UART2_SRT_REG (0x5000119C)

Bit	Mode	Symbol	Description	Reset
15:2	-	-		0x0
1:0	R/W	UART_SHADOW_RCVR_TRIGGER	Shadow RCVR Trigger. 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. 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: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full	0x0

Table 417: UART2_STET_REG (0x500011A0)

Bit	Mode	Symbol	Description	Reset
15:2	-	-		0x0

Bit	Mode	Symbol	Description	Reset
1:0	R/W	UART_SHADOW_TX_EMPTY_TRIGGER	Shadow TX Empty Trigger. 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. 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: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO ¼ full 11 = FIFO ½ full	0x0

Table 418: UART2_HTX_REG (0x500011A4)

Bit	Mode	Symbol	Description	Reset
15:1	-	-		0x0
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. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.	0x0

Table 419: UART2_DMASA_REG (0x500011A8)

Bit	Mode	Symbol	Description	Reset
0	W	DMASA	This register is use to perform DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This will cause the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing' and it is not necessary to clear this bit.	0x0

Table 420: UART2_DLF_REG (0x500011C0)

Bit	Mode	Symbol	Description	Reset
3:0	R/W	UART_DLF	The fractional value is added to integer value set by DLH, DLL. Fractional value is equal UART_DLF/16	0x0

Table 421: UART2_CPR_REG (0x500011F4)

Bit	Mode	Symbol	Description	Reset
15:0	R	CPR	Component Parameter Register	0x3D71

Table 422: UART2_UCV_REG (0x500011F8)

Bit	Mode	Symbol	Description	Reset
15:0	R	UCV	Component Version	0x3331352A

Table 423: UART2_CTR_REG (0x500011FC)

Bit	Mode	Symbol	Description	Reset
15:0	R	CTR	Component Type Register	0x44570110

28.17 Chip Version Registers

Table 424: Register map Version

Address	Register	Description
0x50003200	CHIP_ID1_REG	Chip identification register 1.
0x50003201	CHIP_ID2_REG	Chip identification register 2.
0x50003202	CHIP_ID3_REG	Chip identification register 3.
0x50003203	CHIP_SWC_REG	Software compatibility register.
0x50003204	CHIP_REVISION_REG	Chip revision register.

Table 425: CHIP_ID1_REG (0x50003200)

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

Table 426: CHIP_ID2_REG (0x50003201)

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

Table 427: CHIP_ID3_REG (0x50003202)

Bit	Mode	Symbol	Description	Reset
7:0	R	CHIP_ID3	Third character of device type "585" in ASCII.	0x35

Table 428: CHIP_SWC_REG (0x50003203)

Bit	Mode	Symbol	Description	Reset
7:4	R	-		0x0
3:0	R	CHIP_SWC	**S**oft**Wa**re **C**ompatibility code. Integer (default = 0) which is incremented if a silicon change has impact on the CPU Firmware. Can be used by software developers to write silicon revision dependent code.	0x1

Table 429: CHIP_REVISION_REG (0x50003204)

Bit	Mode	Symbol	Description	Reset
7:0	R	REVISION_ID	Chip version, corresponds with type number in ASCII. 0x41 = 'A', 0x42 = 'B'	0x41

28.18 Wake-Up Registers

Table 430: Register map WKUP

Address	Register	Description
0x50000100	WKUP_CTRL_REG	Control register for the wakeup counter
0x50000102	WKUP_COMPARE_REG	Number of events before wakeup interrupt
0x50000104	WKUP_RESET_IRQ_REG	Reset wakeup interrupt
0x50000106	WKUP_COUNTER_REG	Actual number of events of the wakeup counter
0x50000108	WKUP_RESET_CNTR_REG	Reset the event counter
0x5000010A	WKUP_SELECT_P0_REG	Select which inputs from P0 port can trigger wkup counter
0x5000010C	WKUP_SELECT_P1_REG	Select which inputs from P1 port can trigger wkup counter
0x5000010E	WKUP_SELECT_P2_REG	Select which inputs from P2 port can trigger wkup counter
0x50000110	WKUP_SELECT_P3_REG	Select which inputs from P3 port can trigger wkup counter
0x50000112	WKUP_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

Table 431: WKUP_CTRL_REG (0x50000100)

Bit	Mode	Symbol	Description	Reset
15:14	-	-		0x0
7	R/W	WKUP_ENABLE_IRQ	0: no interrupt will be enabled 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 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). 0x0: no debouncing 0x1 to 0x3F: 1 ms to 63 ms debounce time	0x0

Table 432: WKUP_COMPARE_REG (0x50000102)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		0x0
7:0	R/W	COMPARE	The number of events that have to be counted before the wakeup interrupt will be given	0x0

Table 433: 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 434: WKUP_COUNTER_REG (0x50000106)

Bit	Mode	Symbol	Description	Reset
15:8	-	-		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 435: 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 436: 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 1: input P0x is enabled for wakeup event counter	0x0

Table 437: 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 1: input P1x is enabled for wakeup event counter	0x0

Table 438: 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 1: input P2x is enabled for wakeup event counter	0x0

Table 439: 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 1: input P3x is enabled for wakeup event counter	0x0

Table 440: 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 1: enabled input P0x will increment the event counter if that input goes low	0x0

Table 441: 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 1: enabled input P1x will increment the event counter if that input goes low	0x0

Table 442: 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 1: enabled input P2x will increment the event counter if that input goes low	0x0

Table 443: 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 1: enabled input P3x will increment the event counter if that input goes low	0x0

28.19 Watchdog Registers

Table 444: Register map WDOG

Address	Register	Description
0x50003100	WATCHDOG_REG	Watchdog timer register.
0x50003102	WATCHDOG_CTRL_REG	Watchdog control register.

Table 445: WATCHDOG_REG (0x50003100)

Bit	Mode	Symbol	Description	Reset
15:9	R0/W	WDOG_WEN	0000.000 = Write enable for Watchdog timer else Write disable. This filter prevents unintentional presetting the watchdog with a SW run-away.	0x0
8	R/W	WDOG_VAL_NEG	0 = Watchdog timer value is positive. 1 = Watchdog timer value is negative.	0x0

7:0	R/W	WDOG_VAL	Write: Watchdog timer reload value. Note that all bits 15-9 must be 0 to reload this register.
			Read: Actual Watchdog timer value. Decremented by 1 every 10.24 msec. Bit 8 indicates a negative counter value. 2, 1, 0, $1FF_{16}$, $1FE_{16}$ etc. An NMI or WDOG (SYS) reset is generated under the following conditions:
			If WATCHDOG_CTRL_REG[NMI_RST] = 0 then
			If WDOG_VAL = 0 -> NMI (Non Maskable Interrupt)
			if WDOG_VAL = $1F0_{16}$ -> WDOG reset -> reload FF_{16}
			If WATCHDOG_CTRL_REG[NMI_RST] = 1 then
			if WDOG_VAL <= 0 -> WDOG reset -> reload FF_{16}
				0xFF

Table 446: WATCHDOG_CTRL_REG (0x50003102)

Bit	Mode	Symbol	Description	Reset
15:2	-	-		0x0
1	R/W	-		0x0

Bit	Mode	Symbol	Description	Reset
0	R/W	NMI_RST	0 = Watchdog timer generates NMI at value 0, and WDOG (SYS) reset at <=-16. Timer can be frozen /resumed using SET_FREEZE_REG[FRZ_WDOG]/ RESET_FREEZE_REG[FRZ_WDOG]. 1 = Watchdog timer generates a WDOG (SYS) reset at value 0 and can not be frozen by Software. Note that this bit can only be set to 1 by SW and only be reset with a WDOG (SYS) reset or SW reset. The watchdog is always frozen when the Cortex-M0 is halted in DEBUG State.	0x0

29 Ordering Information

Table 447: Ordering Information (Samples)

Part Number	Package	Size (mm)	Shipment Form	Pack Quantity
DA14585-00000AT2	QFN40	5 x 5 x 0.9	Reel	100/1000
DA14585-00000VV2	WLCSP34	2.40 x 2.66 x 0.39	Reel	100/1000
DA14585	QFN48	6 x 6 x 0.9	Contact Dialog sales for availability.	

Table 448: Ordering Information (Production)

Part Number	Package	Size (mm)	Shipment Form	Pack Quantity
DA14585-00000AT2	QFN40	5 x 5 x 0.9	Reel	5000
DA14585-00000VV2	WLCSP34	2.40 x 2.66 x 0.39	Reel	7500
DA14585	QFN48	6 x 6 x 0.9	Contact Dialog sales for availability.	

Part Number Legend:

DA14585-RRXXXYYZ

RR: chip revision number

XXX: variant (000: No Flash)

YY: package code (AT: QFN40, A3: QFN48, VV: WLCSP34)

Z: packing method (1: Tray, 2: Reel, A: Mini-Reel)

30 Package information

30.1 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

30.2 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.

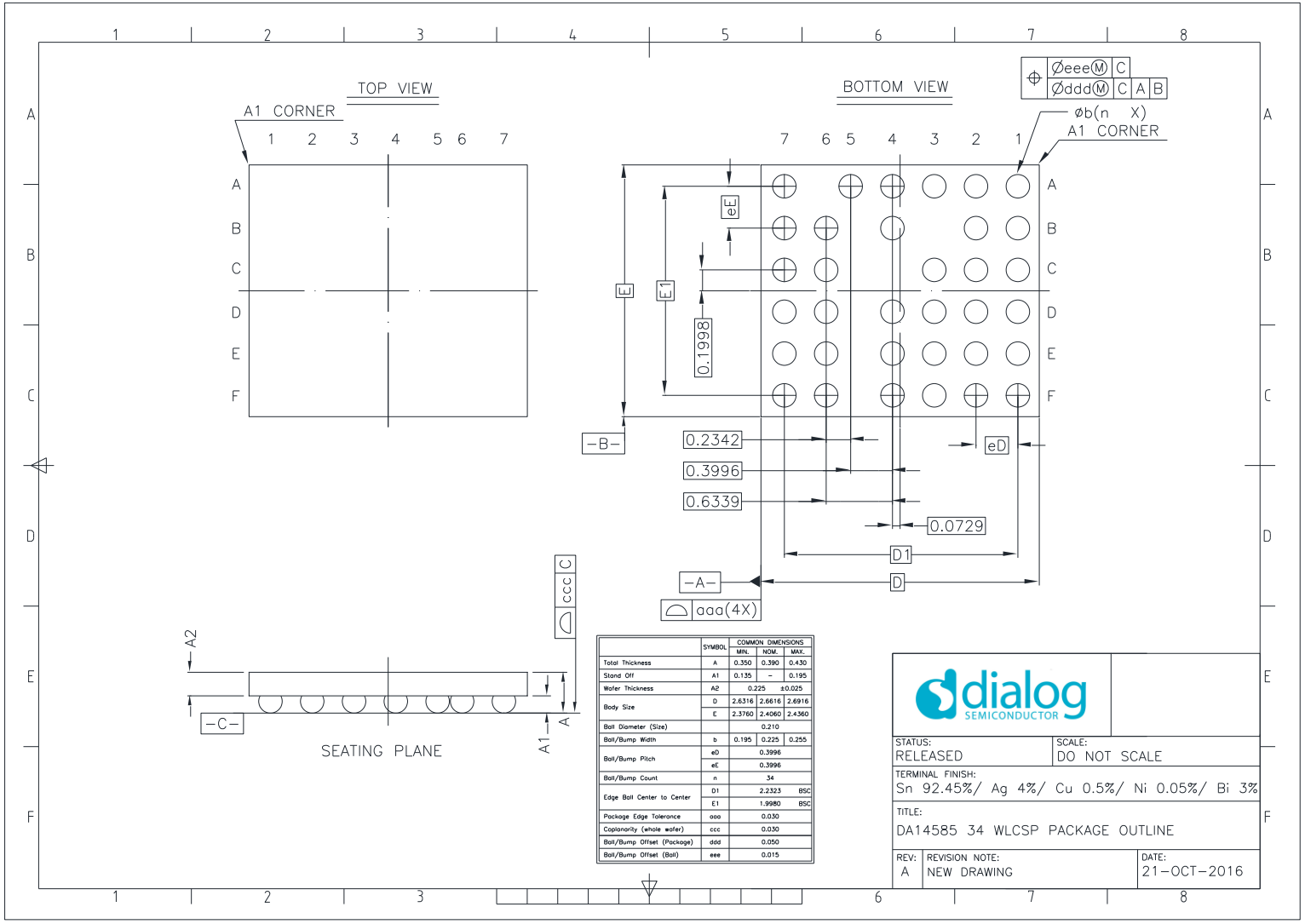
Removal will cause damage to the solder balls and therefore a removed sample cannot be reused.

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

30.3 Soldering information

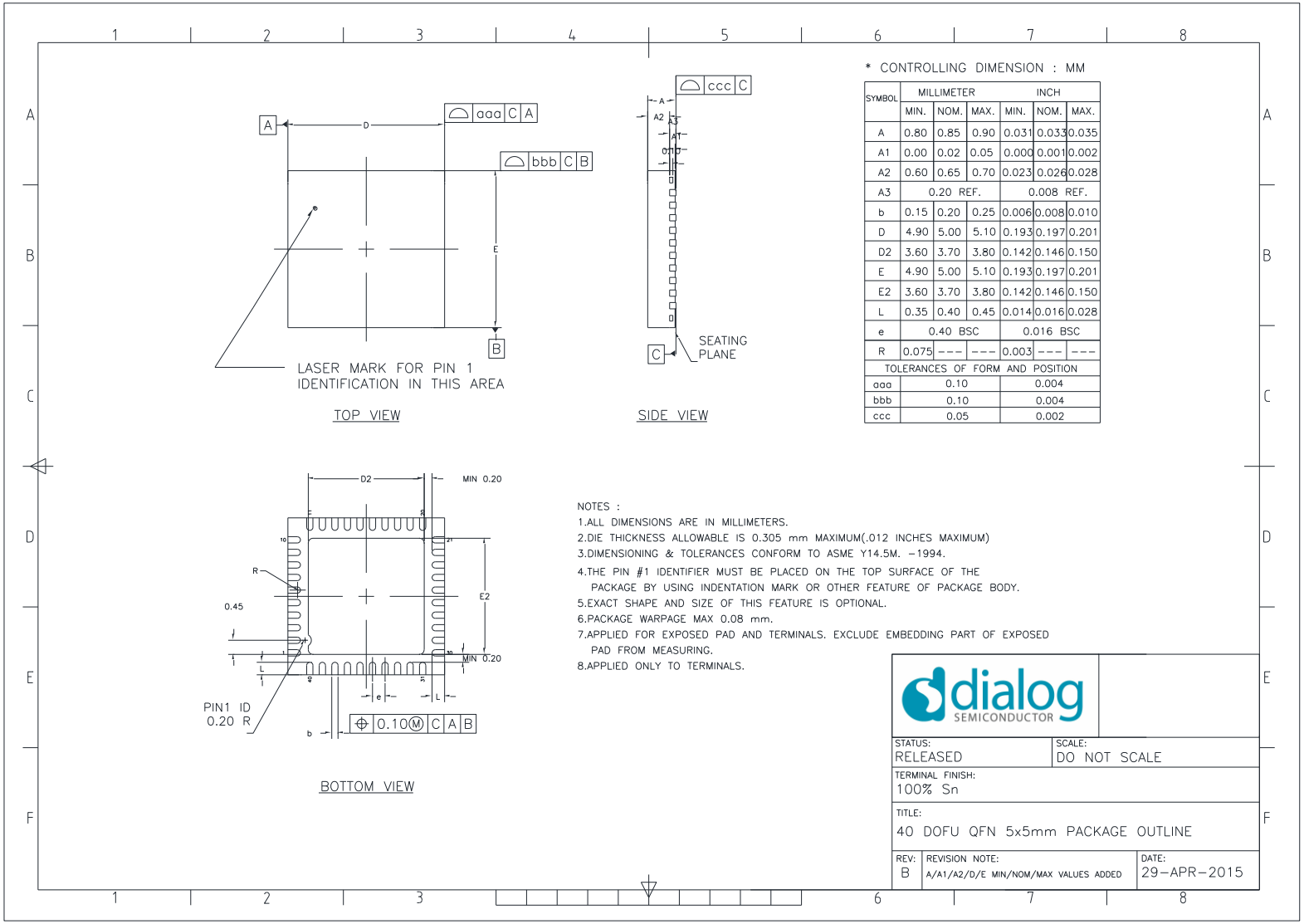
Refer to the JEDEC standard J-STD-020 for relevant soldering information. This document can be downloaded from http://www.jedec.org.

30.4 Package outlines



	SYMBOL	COMMON DIMENSIONS MIN.	NOM.	MAX.
Total Thickness	A	0.350	0.390	0.430
Stand Off	A1	0.135	–	0.195
Wafer Thickness	A2	0.225	±0.025	
Body Size	D	2.6316	2.6616	2.6916
	E	2.3760	2.4060	2.4360
Ball Diameter (Size)		0.210		
Ball/Bump Width	b	0.195	0.225	0.255
Ball/Bump Pitch	eD	0.3996		
	eE	0.3996		
Ball/Bump Count	n	34		
Edge Ball Center to Center	D1	2.2323	BSC	
	E1	1.9980	BSC	
Package Edge Tolerance	aaa	0.030		
Coplanarity (whole wafer)	ccc	0.030		
Ball/Bump Offset (Package)	ddd	0.050		
Ball/Bump Offset (Ball)	eee	0.015		

Figure 85: WLCSP34 Package Outline Drawing



SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.80	0.85	0.90	0.031	0.033	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A2	0.60	0.65	0.70	0.023	0.026	0.028
A3	0.20 REF.			0.008 REF.		
b	0.15	0.20	0.25	0.006	0.008	0.010
D	4.90	5.00	5.10	0.193	0.197	0.201
D2	3.60	3.70	3.80	0.142	0.146	0.150
E	4.90	5.00	5.10	0.193	0.197	0.201
E2	3.60	3.70	3.80	0.142	0.146	0.150
L	0.35	0.40	0.45	0.014	0.016	0.028
e	0.40 BSC			0.016 BSC		
R	0.075	---	---	0.003	---	---
TOLERANCES OF FORM AND POSITION						
aaa	0.10			0.004		
bbb	0.10			0.004		
ccc	0.05			0.002		

Figure 86: QFN40 Package Outline Drawing

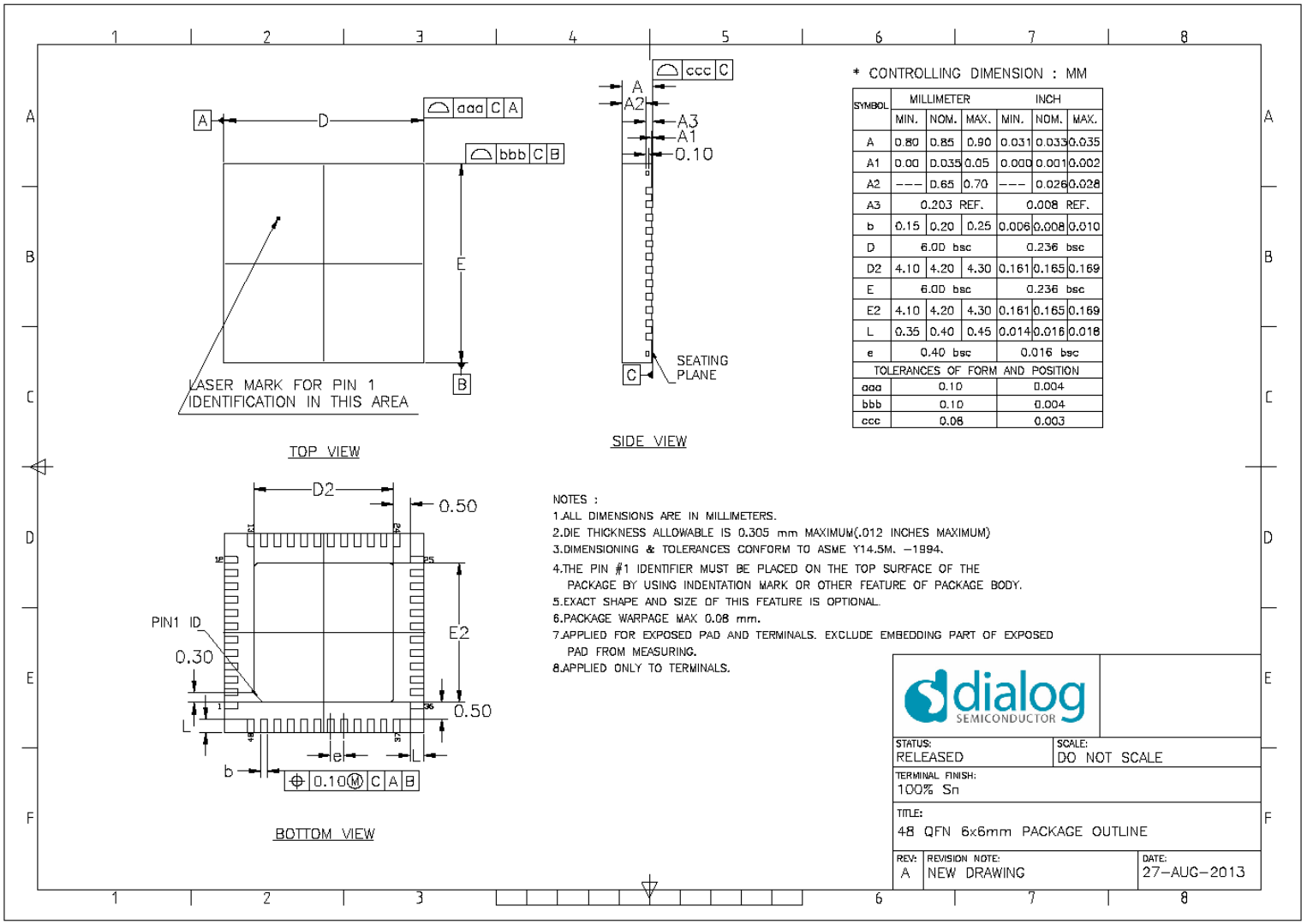


Figure 87: QFN48 Package Outline Drawing (Contact Dialog sales for availability)

Status Definitions

Revision	Datasheet Status	Product Status	Definition
1.<n>	Target	Development	This datasheet contains the design specifications for product development. Specifications may be changed 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 www.dialog-semiconductor.com.
4.<n>	Obsolete	Archived	This datasheet contains the specifications for discontinued products. The information is provided for reference only.

Disclaimer

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

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

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