General Description

The DA14592 is a multi-core wireless microcontroller, combining the latest Arm® Cortex® M33™ application processor with floating-point unit, advanced power management functionality, a cryptographic security engine, analog and digital peripherals, a software configurable protocol engine with a radio that is compliant with Bluetooth® Low Energy 5.2 standard and 256 kB of embedded Flash accompanied by 96 kB of RAM, 16 kB of Cache RAM and 288 kB ROM (containing the Bluetooth LE stack). The embedded Flash or SRAM may be expended externally through QSPI. The DA14592 has been optimized for lowest power consumption, including radio, active power, and in hibernation.

The DA14592 utilizes an Arm® Cortex®-M33 CPU with an eight-region MPU and a single-precision FPU offering up to 96 dMIPS at 64 MHz. This dedicated application processor executes code from embedded Flash or RAM through an 8 kB four-way associative cache controller. Bluetooth® 5.2 or other protocol connectivity is supported by a new software-configurable Bluetooth® Low Energy protocol engine (CMAC) based on an Arm® Cortex®-M0+™ with an ultra-low-power radio transceiver, capable of +6 dBm output power and -97 dBm receive sensitivity offering a total link budget of 103 dB.

A variety of standard and advanced peripherals enable interaction with other system components and the development of advanced user interfaces and feature-rich applications.

Key Features

- Compatible with Bluetooth® 5.2, ETSI EN 300 328 and EN 300 440 Class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan)
- Flexible processing power
  - 32 kHz up to 64 MHz 32-bit Arm Cortex-M33™ with 8 kB, four-way associative cache and FPU
  - A flexible and configurable Bluetooth® LE MAC engine based on Arm Cortex-M0+™ with an 8 kB, four-way associative cache
- Memory
  - 256 kB embedded Flash
  - 96 kB Data SRAM with retention
  - 8 kB Caches with retention for Cortex M33 and M0+ respectively
  - 288 kB ROM (including boot ROM, PKI routines, and Bluetooth® LE stack)
- Power/Clock management
  - Buck DCDC converter
  - Hibernation (Shipping mode) <100 nA
  - Programmable thresholds for brownout detection
  - 32 MHz or 64 MHz system clock using a doubler
  - Fast wake-up from sleep in <15 µs
- Application cryptographic engine with AES-256 and SHA-256
- A Real Time Clock with 10 ms resolution
- Four General purpose, 24-bit up/down timers with PWM capabilities
- Analog and Digital interfaces
  - Up to 32 General Purpose I/Os
  - Eight-channel 10-bit SAR ADC, 2 Msamples/s
  - ΣΔ ADC, 15 bits at 1 ksps, 13 bits at 16 ksps with a Programmable Gain Amplifier (PGA)
  - QSPI PSRAM/Flash interface
  - 2 x UARTs up to 3 Mbps, one UART extended to support ISO7816
  - One SPI+™ and one I2C controller at 100 kHz, 400 kHz, or 3.4 MHz
  - PDM interface with hardware sample rate converter
  - I2S/PCM master/slave interface up to eight channels
  - 3-axis capable Quadrature Decoder
- Radio transceiver
  - Single wire antenna: no RF matching or RX/TX switching required
  - High Performance mode: TX output power -22 to +6 dBm, RX sensitivity -97 dBm
  - Low Power mode: TX output power -23 to 4.5 dBm, RX sensitivity -96 dBm

- Packages
  - WLCSP39, 2.48 x 3.32, 0.42 mm diagonal pitch
  - FCQFN52, 5.1 x 4.3 mm, 0.4 mm pitch

Applications

- Crowd-sourced locationing
- IoT (home, retail, industrial)
- Keyboards
- Gaming mouse
- Gaming/AR/VR controllers
- S-pen
- Cold chain tracking and locationing (proximity, AoA) tags
- Connected Medical: BGM/CGM/insulin pump, HRM/BPM, Inhaler
- Robotics
- Wearables
- Toys
- Metering
- ESL
- High-end Voice RCU
- POS readers
- Activity tracker
- Beacons

Key Advantages

- Lowest power consumption
- Smallest system size
- Lowest system cost
- On-chip memory expandable externally through QSPI

Contents

Figures

Tables

1 Block Diagram

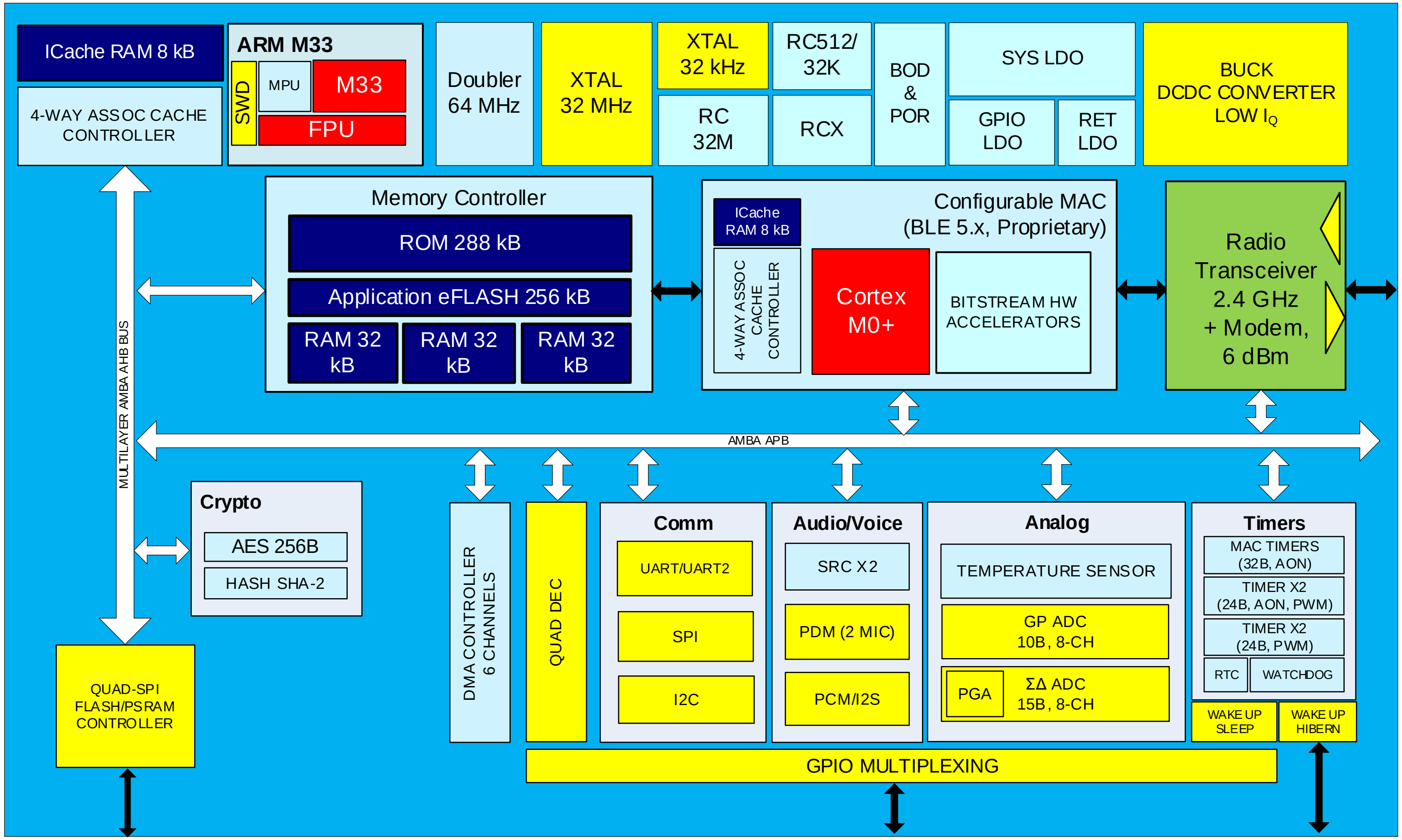

Figure 1. DA14592 block diagram

2 Pinout

The DA14592 comes in two packages:

- A 3.32 mm x 2.48 mm WLCSP with 39 balls, 0.42 mm diagonal pitch.
- A 5.1 mm x 4.3 mm FCQFN with 52 pins, 0.4 mm pitch.

2.1 WLCSP39

This package allows for 19 GPIOs.

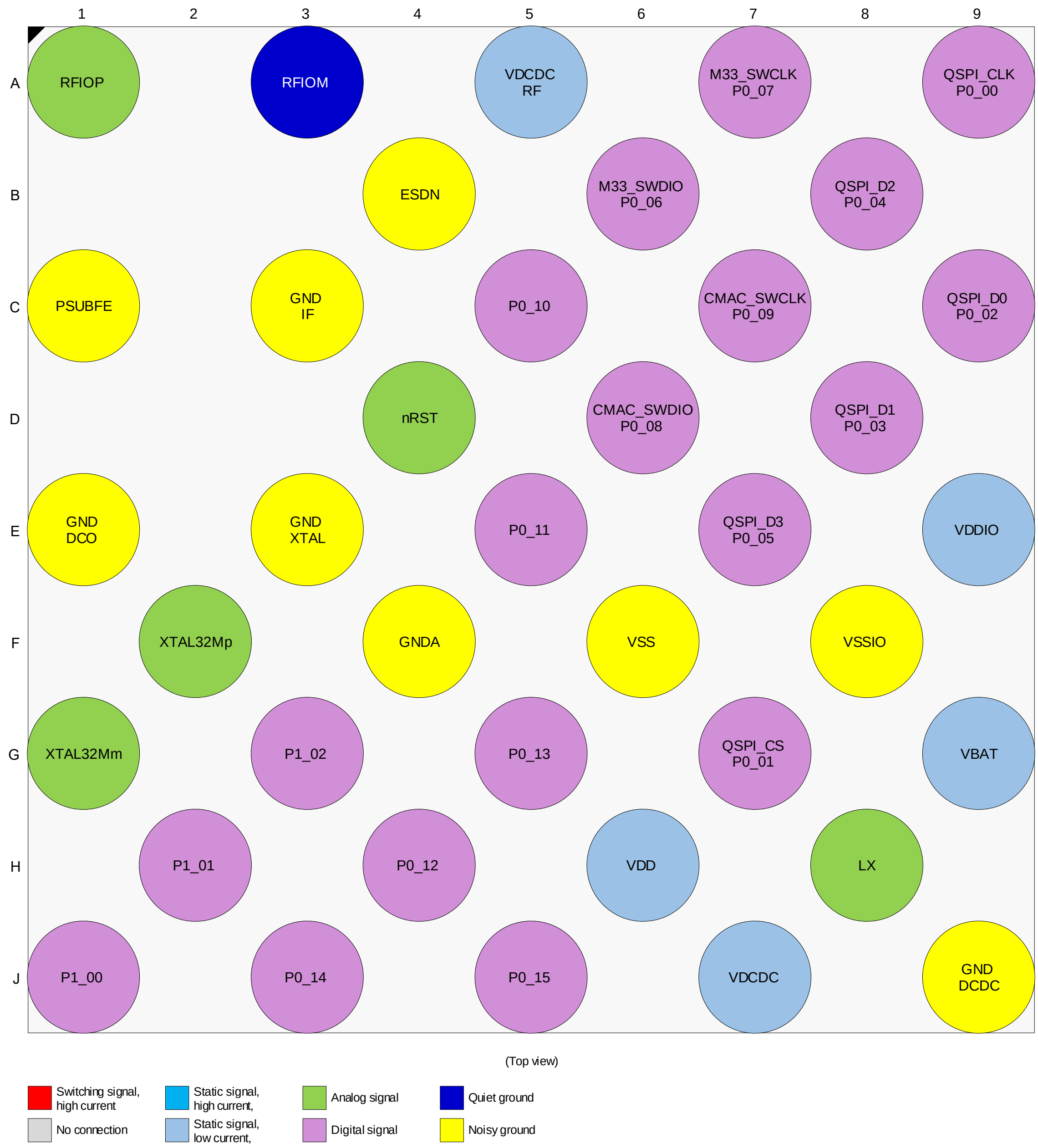

Figure 2. WLCSP39 ball assignment

Table 1: WLCSP39 Ball Description

Ball No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
General Purpose I/Os (fixed pin assignment; additional functions are programmable via Px_xx_MODE_REG)					
A9	P0_00 QSPI_CLK	DIO DO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. QSPI RAM/Flash clock (Note 3)
G7	P0_01 QSPI_CS	DIO DO	4/8/12/16	I-PU	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-up enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. QSPI RAM/Flash chip select (Note 3)
C9	P0_02 QSPI _D0	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 0 (Note 3)
D8	P0_03 QSPI _D1	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 1 (Note 3)
B8	P0_04 QSPI _D2	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 2 (Note 3)
E7	P0_05 QSPI _D3	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 3 (Note 3)
B6	P0_06 M33_SWDIO	DIO DIO	3.5/0.35	I-PU	INPUT/OUTPUT with selectable pull up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. Arm Cortex-M33 Serial Wire Debug data I/O signal.
A7	P0_07 M33_SWCLK	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Arm Cortex-M33 Serial Wire Debug clock signal.
D6	P0_08	DIO	3.5/0.35	I-PU	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-up enabled during and after reset. General-purpose I/O port bit or alternate function

Ball No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
	CMAC_SWDIO DIVN	DIO DO			nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. Arm Cortex-M0+ Serial Wire Debug data I/O signal. OUTPUT. DIVN clock signal output (square wave).
C7	P0_09 CMAC_SWCLK	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Arm Cortex-M0+ Serial Wire Debug clock signal.
C5	P0_10 Timer2.PWM GPADC_3 SDADC_3	DIO DO AI AI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. Timer2/PWM output (PWM) in Sleep mode. INPUT. Analog input of the general-purpose ADC, channel 3. INPUT. Analog input of the SDADC, channel 3.
E5	P0_11 XTAL32M	DIO DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. XTAL32M clock signal output (square wave).
H4	P0_12 LP_CLK Timer.PWM	DIO DO DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Does not contain state retention mechanism during power down. OUTPUT. LP clock signal output (square wave), active during sleep (Note 4). OUTPUT. Timer/PWM output (PWM) in Sleep mode.
G5	P0_13 UART Boot TX	DIO DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. UART Transmit data output during boot.
J3	P0_14 Hibernation Wake up 1	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Wake up from hibernation mode source 1.
J5	P0_15 UART Boot RX	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. UART Receive data input during boot.
J1	P1_00	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism

Ball No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
	PGA_INp GPADC_0 SDADC_0	AI AI AI			during power down. INPUT. Programmable gain amplifier positive input. INPUT. Analog input of the general-purpose ADC, channel 0. INPUT. Analog input of the SDADC, channel 0.
H2	P1_01 PGA_INm GPADC_1 SDADC_1	DIO AI AI AI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Programmable gain amplifier negative input. INPUT. Analog input of the general-purpose ADC, channel 1. INPUT. Analog input of the SDADC, channel 1.
G3	P1_02 GPADC_2 SDADC_2 RC32M	DIO AI AI DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Analog input of the general-purpose ADC, channel 2. INPUT. Analog input of the SDADC, channel 2. OUTPUT. RC32M clock signal output (square wave).
Debug Interface					
B6	M33_SWDIO	DIO		I-PU	INPUT/OUTPUT. Arm Cortex-M33 Serial Wire Debug data I/O signal.
A7	M33_SWCLK	DI		I-PD	INPUT. Arm Cortex-M33 Serial Wire Debug clock signal.
D6	CMAC_SWDIO	DIO		I-PU	INPUT/OUTPUT. Arm Cortex-M0+ Serial Wire Debug data I/O signal.
C7	CMAC_SWCLK	DI		I-PD	INPUT. Arm Cortex-M0+ Serial Wire Debug clock signal.
Clocks					
G1	XTAL32Mm	AO			OUTPUT. Crystal output for the 32 MHz XTAL oscillator.
F2	XTAL32Mp	AI			INPUT. Crystal input for the 32 MHz XTAL oscillator.
QSPI RAM/Flash Interface					
C9	QSPI_D0	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 0.
D8	QSPI_D1	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 1.
B8	QSPI_D2	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 2.
E7	QSPI_D3	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 3.
A9	QSPI_CLK	DO			OUTPUT. QSPI RAM/Flash clock.
G7	QSPI_CS	DO			OUTPUT. QSPI RAM/Flash chip select (active LOW). This output is HIGH in the default and reset states. No pull-up resistor is required.
SPI Bus Interface (mapped on port Px_yy)					
	SPI_DI	DI			INPUT. SPI data input. (Note 1)
	SPI_DO	DO			OUTPUT. SPI data output. (Note 2)
	SPI_CLK	DIO			INPUT/OUTPUT. SPI clock.
	SPI_EN	DI			INPUT. SPI clock enable (chip select).
I2C Bus Interface (mapped on port Px_yy)					

Ball No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
	I2C_SCL	DIO/ DIOD			INPUT/OUTPUT. I2C bus clock with open drain port. Supports bit stretching by a slave in open drain mode.
	I2C_SDA	DIO/ DIOD			INPUT/OUTPUT. I2C bus data with open drain port.
UART Interface (mapped on port Px_yy)					
	UART_RX	DI			INPUT. UART receive data.
	UART_TX	DO			OUTPUT. UART transmit data.
	UART2_RX	DI			INPUT. UART 2 receive data.
	UART2_TX	DO			OUTPUT. UART 2 transmit data.
	UART2_CTS	DI			INPUT. UART 2 clear to send.
	UART2_RTS	DO			OUTPUT. UART 2 request to send.
ISO7816 Bus Interface (mapped on port Px_yy)					
	ISO7816_CLK	DO			OUTPUT. Smart card (ISO7816) clock signal
	ISO7816_DATA	DIO			INPUT/OUTPUT. Smart card (ISO7816) I/O data signal.
	ISO7816_RST	DO			OUTPUT. Smart card (ISO7816) reset signal.
	ISO7816_CI	DI			INPUT. Smart card (ISO7816) inserted signal.
PCM Interface (mapped on port Px_yy)					
	PCM_DI	DI			INPUT. PCM input data.
	PCM_DO	DO			OUTPUT.PCM output data.
	PCM_FSC	DIO			INPUT/OUTPUT. PCM Frame synchronization.
	PCM_CLK	DIO			INPUT/OUTPUT. PCM Clock
PDM Interface (mapped on port Px_yy)					
	PDM_DATA	DIO			INPUT/OUTPUT. PDM data.
	PDM_CLK	DO			OUTPUT. PDM clock output.
Quadrature Decoder Interface (mapped on port Px_yy)					
	QD_CHA_X	DI			INPUT. Channel A for the X axis.
	QD_CHB_X	DI			INPUT. Channel B for the X axis.
	QD_CHA_Y	DI			INPUT. Channel A for the Y axis.
	QD_CHB_Y	DI			INPUT. Channel B for the Y axis.
	QD_CHA_Z	DI			INPUT. Channel A for the Z axis.
	QD_CHB_Z	DI			INPUT. Channel B for the Z axis.
Analog Interface					
J1	SDADC_0	AI			INPUT. Analog input of the SDADC, channel 0.
H2	SDADC_1	AI			INPUT. Analog input of the SDADC, channel 1.
G3	SDADC_2	AI			INPUT. Analog input of the SDADC, channel 2.
C5	SDADC_3	AI			INPUT. Analog input of the SDADC, channel 3.
J1	GPADC_0	AI			INPUT. Analog input of the general-purpose ADC, channel 0.
H2	GPADC_1	AI			INPUT. Analog input of the general-purpose ADC, channel 1.

Ball No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
G3	GPADC_2	AI			INPUT. Analog input of the general-purpose ADC, channel 2.
C5	GPADC_3	AI			INPUT. Analog input of the general-purpose ADC, channel 3.
J1	PGA_INp	AI			INPUT. Analog input of the programmable gain amplifier
H2	PGA_INm	AI			INPUT. Analog input of the programmable gain amplifier
Radio Interface					
A1	RFIOP	AIO			RF input/output. Impedance 50 Ω.
A3	RFIOM	AIO			RF Ground
	Miscellaneous				
D4	nRST	AI			INPUT. Reset signal (active LOW).
H8	LX	AIO			INPUT/OUTPUT. Connection for the external DC-DC converter inductor.
Power Supply					
H6	VDD	AO			OUTPUT. From 0.9 to 1.2 V power rail. 2.2 µF decoupling capacitor required.
E9	VDDIO	AIO			OUTPUT. 1.8 V power rail. 4.7 µF decoupling capacitor required.
A5	VDCDC_RF	AI			INPUT. Radio supply voltage. Connect to VDCDC externally. 4.7 µF decoupling capacitor required.
J7	VDCDC	AO			OUTPUT. From 1.1 V to 1.4 V power rail. 4.7 µF decoupling capacitor required.
G9	VBAT	AI			INPUT. Battery connection
F4	GNDA	-			Analog Ground.
F8	VSSIO	-			Digital Ground
F6	VSS	-			Digital Ground
C3	GNDIF	-			Radio Ground
E1	GND_DCO	-			Radio Ground
E3	GND_XTAL	-			32 MHz Crystal Oscillator Ground
C1	PSUBFE	-			Radio Ground
B4	ESDN	-			Radio Ground
A3	RFIOM	-			Radio Ground

Note 1 Data input only. MOSI in SPI slave mode, MISO in SPI master mode.

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

Note 3 Not available for all family variances.

Note 4 PMU_CTRL_REG[LP_CLK_OUTPUT_EN].

Note 5 RDS = Reduced Driving Strength functionality (P0/1_WEAK_CTRL_REG).

2.2 FCQFN52

This package allows for 32 GPIOs.

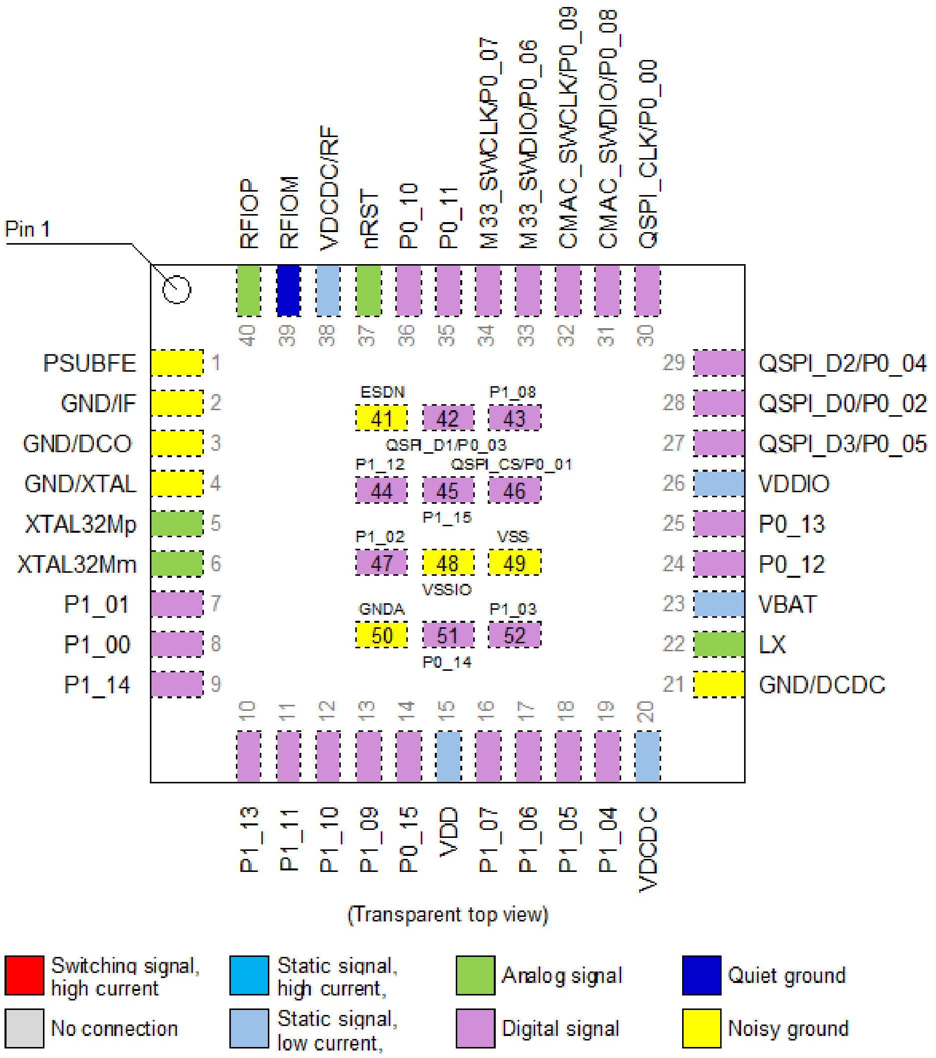

Figure 3. FCQFN52 pin assignment

Table 2: FCQFN52 Pin Description

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
General Purpose I/Os (fixed pin assignment; additional functions are programmable via Px_xx_MODE_REG)					
30	P0_00 QSPI_CLK	DIO DO DO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. QSPI RAM/Flash clock (Note 3)
46	P0_01 QSPI_CS	DIO DO	4/8/12/16	I-PU	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-up enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. QSPI RAM/Flash chip select (Note 3)
28	P0_02 QSPI _D0	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 0 (Note 3)
42	P0_03 QSPI _D1	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 1 (Note 3)
29	P0_04 QSPI _D2	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 2 (Note 3)
27	P0_05 QSPI _D3	DIO DIO	4/8/12/16	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. QSPI RAM/Flash Data line 3 (Note 3)
33	P0_06 M33_SWDIO	DIO DIO	3.5/0.35	I-PU	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. Arm Cortex-M33 Serial Wire Debug data I/O signal.
34	P0_07 M33_SWCLK	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Arm Cortex-M33 Serial Wire Debug clock signal.
31	P0_08	DIO	3.5/0.35	I-PU	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-up enabled during and after reset. General-purpose I/O port bit or

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
	 CMAC_SWDIO DIVN	 DIO DO			alternate function nodes. Contains state retention mechanism during power down. INPUT/OUTPUT. Arm Cortex-M0+ Serial Wire Debug data I/O signal. OUTPUT. DIVN clock signal output (square wave).
32	P0_09 CMAC_SWCLK	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Arm Cortex-M0+ Serial Wire Debug clock signal.
36	P0_10 Timer2.PWM GPADC_3 SDADC_3	DIO DO AI AI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. Timer2/PWM output (PWM) in Sleep mode. INPUT. Analog input of the general-purpose ADC, channel 3. INPUT. Analog input of the SDADC, channel 3.
35	P0_11 XTAL32M	DIO DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. XTAL32M clock signal output (square wave).
24	P0_12 LP_CLK Timer.PWM	DIO DO DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Does not contain state retention mechanism during power down. OUTPUT. LP clock signal output (square wave), active during sleep (Note 4). OUTPUT. Timer/PWM output (PWM) in Sleep mode.
25	P0_13 UART Boot TX	DIO DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. UART Transmit data output during boot.
51	P0_14 Hibernation Wake up 1	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Wake up from hibernation mode source 1.
14	P0_15 UART Boot RX	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. UART Receive data input during boot.
8	P1_00	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
	  PGA_INp GPADC_0 SDADC_0	 AI AI AI			alternate function nodes. Contains state retention mechanism during power down. INPUT. Programmable gain amplifier positive input. INPUT. Analog input of the general-purpose ADC, channel 0. INPUT. Analog input of the SDADC, channel 0.
7	P1_01 PGA_INm GPADC_1 SDADC_1	DIO AI AI AI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Programmable gain amplifier negative input. INPUT. Analog input of the general-purpose ADC, channel 1. INPUT. Analog input of the SDADC, channel 1.
47	P1_02 GPADC_2 SDADC_2 RC32M	DIO AI AI DO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Analog input of the general-purpose ADC, channel 2. INPUT. Analog input of the SDADC, channel 2. OUTPUT. RC32M clock signal output (square wave).
52	P1_03	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.
19	P1_04 Hibernation Wake Up 2	DIO DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Wake up from hibernation mode source 2.
18	P1_05 GPADC_4 SDADC_4 SDADC_REFp SDADC_INT_REF	DIO AI AI AI AO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Analog input of the general-purpose ADC, channel 4. INPUT. Analog input of the SDADC, channel 4. INPUT. Analog input of the external SDADC positive voltage reference OUTPUT. Analog output of the internal SDADC voltage reference.
17	P1_06 GPADC_5 SDADC_5 SDADC_REFn	DIO AI AI AI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Analog input of the general-purpose ADC, channel 5. INPUT. Analog input of the SDADC, channel 5. INPUT. Analog input of the external SDADC negative reference.

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
16	P1_07	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.
43	P1_08	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.
13	P1_09 GPADC_6 SDADC_6	DIO AI AI	3.5/0.35	I-PD	INPUT / OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Analog input of the general-purpose ADC, channel 6. INPUT. Analog input of the SDADC, channel 6.
12	P1_10	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.
11	P1_11 GPADC_7 SDADC_7	DIO AI AI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Analog input of the general-purpose ADC, channel 7. INPUT. Analog input of the SDADC, channel 7.
44	P1_12	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.
10	P1_13 XTAL32km	DIO AO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. OUTPUT. Analog output of the XTAL32k crystal oscillator.
9	P1_14 XTAL32kp	DIO AI DI	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down. INPUT. Analog input of the XTAL32k crystal oscillator. INPUT. Digital input for an external clock (square wave).
45	P1_15	DIO	3.5/0.35	I-PD	INPUT/OUTPUT with selectable pull-up/down resistor and open drain functionality. Pull-down enabled during and after reset. General-purpose I/O port bit or alternate function nodes. Contains state retention mechanism during power down.

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
Debug Interface					
33	M33_SWDIO	DIO		I-PU	INPUT/OUTPUT. Arm Cortex-M33 Serial Wire Debug data I/O signal.
34	M33_SWCLK	DI		I-PD	INPUT. Arm Cortex-M33 Serial Wire Debug clock signal.
31	CMAC_SWDIO	DIO		I-PU	INPUT/OUTPUT. Arm Cortex-M0+ Serial Wire Debug data I/O signal.
32	CMAC_SWCLK	DI		I-PD	INPUT. Arm Cortex-M0+ Serial Wire Debug clock signal.
Clocks					
10	XTAL32km	AO			OUTPUT. Analog output of the 32.768 kHz XTAL oscillator.
9	XTAL32kp	AI DI			INPUT. Analog input of the 32.768 kHz XTAL crystal oscillator. INPUT. Digital input for an external clock (square wave).
6	XTAL32Mm	AO			OUTPUT. Crystal output for the 32 MHz XTAL oscillator.
5	XTAL32Mp	AI			INPUT. Crystal input for the 32 MHz XTAL oscillator.
QSPI RAM/Flash Interface					
28	QSPI_D0	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 0.
42	QSPI_D1	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 1.
29	QSPI_D2	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 2.
27	QSPI_D3	DIO			INPUT/OUTPUT. QSPI RAM/Flash I/O data 3.
30	QSPI_CLK	DO			OUTPUT. QSPI RAM/Flash clock.
46	QSPI_CS	DO			OUTPUT. QSPI RAM/Flash chip select (active LOW). This output is HIGH in the default and reset states. No pull-up resistor is required.
SPI Bus Interface (mapped on port Px_yy)					
	SPI_DI	DI			INPUT. SPI data input. (Note 1)
	SPI_DO	DO			OUTPUT. SPI data output. (Note 2)
	SPI_CLK	DIO			INPUT/OUTPUT. SPI clock.
	SPI_EN	DI			INPUT. SPI clock enable (chip select).
I2C Bus Interface (mapped on port Px_yy)					
	I2C_SCL	DIO/ DIOD			INPUT/OUTPUT. I2C bus clock with open drain port. Supports bit stretching by a slave in open drain mode.
	I2C_SDA	DIO/ DIOD			INPUT/OUTPUT. I2C bus data with open drain port.
UART Interface (mapped on port Px_yy)					
	UART_RX	DI			INPUT. UART receive data.
	UART_TX	DO			OUTPUT. UART transmit data.
	UART2_RX	DI			INPUT. UART 2 receive data.
	UART2_TX	DO			OUTPUT. UART 2 transmit data.

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
	UART2_CTS	DI			INPUT. UART 2 clear to send.
	UART2_RTS	DO			OUTPUT. UART 2 request to send.
ISO7816 Bus Interface (mapped on port Px_yy)					
	ISO7816_CLK	DO			OUTPUT. Smart card (ISO7816) clock signal
	ISO7816_DATA	DIO			INPUT/OUTPUT. Smart card (ISO7816) I/O data signal.
	ISO7816_RST	DO			OUTPUT. Smart card (ISO7816) reset signal.
	ISO7816_CI	DI			INPUT. Smart card (ISO7816) inserted signal.
PCM Interface (mapped on port Px_yy)					
	PCM_DI	DI			INPUT. PCM input data.
	PCM_DO	DO			OUTPUT.PCM output data.
	PCM_FSC	DIO			INPUT/OUTPUT. PCM Frame synchronization.
	PCM_CLK	DIO			INPUT/OUTPUT. PCM Clock
PDM Interface (mapped on port Px_yy)					
	PDM_DATA	DIO			INPUT/OUTPUT. PDM data.
	PDM_CLK	DO			OUTPUT. PDM clock output.
Quadrature Decoder Interface (mapped on port Px_yy)					
	QD_CHA_X	DI			INPUT. Channel A for the X axis.
	QD_CHB_X	DI			INPUT. Channel B for the X axis.
	QD_CHA_Y	DI			INPUT. Channel A for the Y axis.
	QD_CHB_Y	DI			INPUT. Channel B for the Y axis.
	QD_CHA_Z	DI			INPUT. Channel A for the Z axis.
	QD_CHB_Z	DI			INPUT. Channel B for the Z axis.
Analog Interface					
8	SDADC_0	AI			INPUT. Analog input of the SDADC, channel 0.
7	SDADC_1	AI			INPUT. Analog input of the SDADC, channel 1.
47	SDADC_2	AI			INPUT. Analog input of the SDADC, channel 2.
36	SDADC_3	AI			INPUT. Analog input of the SDADC, channel 3.
18	SDADC_4	AI			INPUT. Analog input of the SDADC, channel 4.
17	SDADC_5	AI			INPUT. Analog input of the SDADC, channel 5.
13	SDADC_6	AI			INPUT. Analog input of the SDADC, channel 6.
11	SDADC_7	AI			INPUT. Analog input of the SDADC, channel 7.
8	GPADC_0	AI			INPUT. Analog input of the general-purpose ADC, channel 0.
7	GPADC_1	AI			INPUT. Analog input of the general-purpose ADC, channel 1.
47	GPADC_2	AI			INPUT. Analog input of the general-purpose ADC, channel 2.
36	GPADC_3	AI			INPUT. Analog input of the general-purpose ADC, channel 3.

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
18	GPADC_4	AI			INPUT. Analog input of the general-purpose ADC, channel 4.
17	GPADC_5	AI			INPUT. Analog input of the general-purpose ADC, channel 5.
13	GPADC_6	AI			INPUT. Analog input of the general-purpose ADC, channel 6.
11	GPADC_7	AI			INPUT. Analog input of the general-purpose ADC, channel 7.
8	PGA_INp	AI			INPUT. Analog input of the programmable gain amplifier
7	PGA_INm	AI			INPUT. Analog input of the programmable gain amplifier
17	SDADC_REFm	AI			INPUT. Analog input of the external SDADC negative voltage reference
18	SDADC_REFp	AI			INPUT. Analog input of the external SDADC positive voltage reference
18	SDADC_INT_REF	AO			OUTPUT. Analog output of the internal SDADC voltage reference.
Radio Interface					
40	RFIOP	AIO			RF input/output. Impedance 50 Ω.
39	RFIOM	AIO			RF Ground.
Miscellaneous					
37	nRST	AI			INPUT. Reset signal (active LOW).
22	LX	AIO			INPUT/OUTPUT. Connection for the external DC-DC converter inductor.
Power Supply					
15	VDD	AO			OUTPUT. From 0.9 to 1.2 V power rail. 2.2 µF decoupling capacitor required.
26	VDDIO	AIO			OUTPUT. 1.8 V power rail. 4.7 µF decoupling capacitor required.
38	VDCDC_RF	AI			INPUT. Radio supply voltage. Connect to VDCDC externally. 4.7 µF decoupling capacitor required.
20	VDCDC	AO			OUTPUT. From 1.1 V to 1.4 V power rail. 4.7 µF decoupling capacitor required.
23	VBAT	AI			INPUT. Battery connection.
50	GNDA	-			Analog Ground.
26	VSSIO	-			Digital Ground.
49	VSS	-			Digital Ground.
2	GNDIF	-			Radio Ground.
3	GND_DCO	-			Radio Ground.
4	GND_XTAL	-			32 MHz Crystal Oscillator Ground.
1	PSUBFE	-			Radio Ground.
41	ESDN	-			Radio Ground.

Pin No.	Pin Name	Type	Drive (mA) (Note 5)	Reset State	Description
39	RFIOM	-			Radio Ground.

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): VBAT = 3.0 V, T_A = 25 ˚C. All radio measurements are performed with standard RF measurement equipment providing a source/load impedance of 50 Ω.

The specified MIN and MAX capacitor values define the range of the effective capacitance, which may vary over the applied voltage due to voltage derating. Refer to the component manufacturer for the capacitor specifications.

3.1 Absolute maximum ratings

Table 3: Absolute maximum ratings

Parameter	Description	Conditions	Min	Max	Unit
$V_{PIN\_LIM\_DEF}$	Limiting voltage on any pin unless otherwise specified	Default, unless otherwise specified	-0.1	3.6	V
V_{BAT_LIM}	Limiting battery supply voltage	Pin V_{BAT}	0	3.6	V
t_{R_SUP}	Power supply rise time			30	ms
$V_{PIN\_LIM\_3V0}$	Limiting voltage on a pin	3.0 V I/O pins	0	3.45	V
$V_{PIN\_LIM\_1V8}$	Limiting voltage on a pin	1.8 V I/O pins	0	1.98	V
$V_{ESD\_HBM\_FCQFN52}$	Electrostatic discharge voltage (Human Body Model)	FCQFN package		2200	V
$V_{ESD\_HBM\_WLCSP}$	Electrostatic discharge voltage (Human Body Model)	WLCSP package		2200	V
$V_{ESD\_CDM\_FCQFN52}$	Electrostatic discharge voltage (Charged Device Model)	FCQFN package		500	V
$V_{ESD\_CDM\_WLCSP}$	Electrostatic discharge voltage (Charged Device Model)	WLCSP package		500	V
T_{STG}	Storage temperature		-50	150	°C

3.2 Recommended operating conditions

Table 4: Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{BAT}	Battery supply voltage	Pin V_{BAT}	1.7		3.6	V
$V_{PIN\_(default)}$	Voltage on a pin	Voltage between pin and GND	0		min (3.3,VDCDC_RF+0.2)	V
$V_{PIN\_(DCDC\_RF)}$	Voltage on a pin	Supply voltage on VDCDC_RF	0		1.4	V
$V_{PIN\_(1V2)}$	Voltage on a pin	XTAL32Kp, XTAL32Km, XTAL32Mp, XTAL32Mm	0		1.2	V
T_A	Ambient temperature		-40		85	°C

3.3 DC characteristics

Table 5: DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{BAT_IDLE}	Battery supply current	CPU is idle (Wait for Interrupt - WFI); sys_clk = 32 MHz;		481		µA

Parameter	Description	Conditions	Min	Typ	Max	Unit
		pclk = 4 MHz; DC-DC on; FLASH off; peripherals off; V_{BAT} = 3 V.				
$I_{BAT\_RUN\_32MHz\_RAM}$	Average active battery supply current	CPU is executing code from RAM; Cache bypassed; RC32M on; pclk = 4 MHz; DC-DC on; Peripherals off; V_{BAT} = 3 V.		1.52		mA
$I_{BAT\_RUN\_64MHz\_RAM}$	Average active battery supply current	CPU is executing code from RAM; Cache bypassed; XTAL32M on; Doubler on; pclk = 8 MHz; DC-DC on; Peripherals off; VBAT = 3 V.		4.78		mA
$I_{BAT\_RUN\_32MHz\_FLASH}$	Average active battery supply current	CPU is executing code from embedded FLASH; cache bypassed; RC32M on; pclk = 4 MHz; DC-DC on; Peripherals off; V_{BAT} = 3 V.		1.16		mA
$I_{BAT\_RUN\_64MHz\_FLASH}$	Average active battery supply current	CPU is executing code from embedded FLASH; cache bypassed; XTAL32M on; Doubler on; pclk = 8 MHz; DC-DC on; Peripherals off; VBAT = 3 V.		4.52		mA
I_{BAT_HIBERN}	Battery supply current	Hibernation mode; no RAM retained; all clocks off; DC-DC off; V_{BAT} = 3 V.		90		nA
$I_{BAT\_DP\_SLP}$	Battery supply current	Deep Sleep mode; No RAM retained; RCX on; RTC on DC-DC on; V_{BAT} = 3 V. Reset on wake-up		2		µA
$I_{BAT\_EX\_SLP\_32KB\_RET}$	Battery supply current	Extended Sleep mode; Both instruction caches retained. 32 kB (data) RAM retained; RCX on; DC-DC on; RTC on; V_{DD} = 0.75 V; V_{BAT} = 3 V.		2.6		µA
$I_{BAT\_EX\_SLP\_64KB\_RET}$	Battery supply current	Extended Sleep mode; Both instruction caches retained. 64 kB (data) RAM retained; RCX on; DC-DC on; RTC on; V_{DD} = 0.75 V; V_{BAT} = 3 V.		3		µA
$I_{BAT\_EX\_SLP\_96KB\_RET}$	Battery supply current	Extended Sleep mode; Both instruction caches retained. 96 kB (data) RAM retained; RCX on; DC-DC on; RTC on; V_{DD} = 0.75 V; V_{BAT} = 3 V.		3.5		µA
$I_{BAT\_EX\_SLP\_96KB\_RET\_0.9V}$	Battery supply current	Extended Sleep mode; Both instruction caches retained. 96 kB (data) RAM retained; RCX on; DC-DC on; RTC on; V_{DD} = 0.9 V; V_{BAT} = 3 V.		4.8		µA

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{BAT\_Coremark}$	Battery supply current	System CPU is running Coremark from eFlash at 32 MHz speed.		1.1		mA
$I_{BAT\_BLE\_RX\_32M}$	Battery supply current	BLE receive in LP mode; f_{CLK} = 32 MHz; CPU idle; DC-DC on; eFLASH on; V_{BAT} = 3 V. Note 1		2.67		mA
$I_{BAT\_BLE\_TX\_32M\_0dbm}$	Battery supply current	BLE transmit mode in LP mode; TX output power 0 dBm; f_{CLK} = 32 MHz; CPU idle; DC-DC on; eFLASH on; V_{BAT} = 3 V. Note 1		4.33		mA
$I_{BAT\_BLE\_RX\_64M}$	Battery supply current	BLE receive in HP mode; f_{CLK} = 64 MHz; CPU idle; DC-DC on; eFLASH on; V_{BAT} = 3 V. Note 1		4.99		mA
$I_{BAT\_BLE\_TX\_64M\_0dbm}$	Battery supply current	BLE transmit in HP mode; TX output power 0 dBm; f_{CLK} = 64 MHz; CPU idle; DC-DC on; eFLASH on; V_{BAT} = 3 V. Note 1		5.88		mA
$I_{BAT\_BLE\_TX\_64M\_6dbm}$	Battery supply current	BLE transmit in HP mode; TX output power 6 dBm; f_{CLK} = 64 MHz; CPU idle; DC-DC on; eFLASH on; V_{BAT} = 3 V. Note 1		9.56		mA
$I_{BAT\_EFLASH\_PROG}$	Battery supply current during programming the eFlash	V_{BAT} = 3 V, DCDC On, M33 active during programming		2.82		mA
$I_{BAT\_EFLASH\_ERASE}$	Battery supply current during erasing (page or mass) the eFlash	V_{BAT} = 3 V, DCDC On, M33 idle during erase		2.56		mA

Note 1 LP/HP mode: See the "Power Modes and Rails" chapter for details.

3.4 Timing characteristics

Table 6: Timing characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
t_{STA_WAKEUP}	Supply start-up time	Time for normal wake-up from GPIO toggle up to application software execution start running on RC32M. T_A = 25 °C			200	µs
$t_{STA\_FAST\_WAKEUP}$	Supply start-up time	Time for fast wake-up from GPIO toggle up to application software execution start running on RC32M. T_A = 25 °C			10	µs

Parameter	Description	Conditions	Min	Typ	Max	Unit
t_{STA_BOOT}	Power up to booter executed, start-up time	BootROM code execution time without authentication. Max value when in "development mode", min value when in "Normal" mode. T_A = 25 °C Note 1		5	20	ms

Note 1 Development mode: Debugger is enabled and device firmware can be downloaded through UART.

3.5 Thermal characteristics

Table 7: Thermal characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$R_{θJA\_P1\_HQ}$	Thermal resistance (junction to ambient θJA)	WLCSP39 package, 4L JEDEC PCB, 0.24 W Power map, Still air		37.1		°C/W
$R_{θJA\_P2\_HR}$	Thermal resistance (junction to ambient θJA)	FCQFN48 package, 4L JEDEC PCB, 0.24 W Power map, Still air		37.9		°C/W

3.6 Reset characteristics

Table 8: PAD_RST - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{IH}	HIGH level input voltage	Parameters applicable in active/sleep modes; V_{DD} = 0.9 V	0.63			V
V_{IL}	LOW level input voltage	Parameters applicable in active/sleep modes; V_{DD} = 0.9 V			0.27	V
V_{IH_VBAT}	HIGH level input voltage	Parameters applicable in hibernation mode; V_{BAT} = 1.8 V	1.26			V
V_{IL_VBAT}	LOW level input voltage	Parameters applicable in hibernation mode; V_{BAT} = 1.8 V			0.54	V

Table 9: PAD_RST - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{IL_PU}	LOW level input current	V_I=V_{SS} = 0 V, V_{BAT} = 1.8 V	-110		-35	µA

3.7 Brown-out detector characteristics

Table 10: BOD - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$V_{RST\_VDD\_SLEEP}$	Reset voltage		0.67	0.7	0.73	V
$V_{RST\_VDD\_ACTIVE\_1}$	Reset voltage		1.02	1.05	1.08	V
$V_{RST\_VDD\_ACTIVE\_2}$	Reset voltage		0.75	0.78	0.81	V
V_{RST_V18}	Reset voltage		1.52	1.56	1.6	V
$V_{RST\_VDCDC\_ACTIVE}$	Reset voltage		0.96	0.99	1.02	V

Table 11: BOD - Electrical performance

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{HYS}	Hysteresis for rising level		1.25	2.5	3.75	%

3.8 Power on reset at VBAT rail characteristics

Table 12: POR VBAT - DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$V_{HYS\_POR\_VBAT}$	Treshold voltage hysteresis for reset		35		200	mV
$V_{TH\_POR\_VBAT}$	Threshold voltage level to release reset		1100	1500	1700	mV

3.9 General purpose ADC characteristics

Table 13: GP ADC - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
N_{BIT_ADC}	Number of bits (resolution)			10		bit

Table 14: GP ADC - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
E_G	Gain error without software correction (single-ended)	No Chopping, Offset Calibration, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-3		3	%
E_{OFS}	Offset error without software correction (single-ended)	No Chopping, Offset Calibration, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-30		30	LSB
E_{G_COR}	Gain error after software correction (single-ended)	Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-1.5		1.5	%
E_{OFS_COR}	Offset error after software correction (single-ended)	Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-4		8	LSB
E_{G_DIF}	Gain error without software correction (differential)	No Chopping, Offset Calibration, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-3.5		3.5	%
E_{OFS_DIF}	Offset error without software correction (differential)	No Chopping, Offset Calibration, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-20		20	LSB
$E_{G\_DIF\_COR}$	Gain error after software correction (differential)	Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3,	-1.5		1.5	%

Parameter	Description	Conditions	Min	Typ	Max	Unit
		GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7				
$E_{OFS\_DIF\_COR}$	Offset error after software correction (differential)	Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-4		8	LSB
E_{G_ATTNX}	Extra gain error of the attenuator	Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-1		1	%
INL	Integral non-linearity	Single ended, Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-2		2	LSB
DNL	Differential non-linearity	Single ended, Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-2		2	LSB
INL_{DIF}	Integral non-linearity	Differential, Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4, GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7	-2		2	LSB
DNL_{DIF}	Differential non-linearity	Differential, Chopping=1, Offset Calibration, Software Correction, GP_ADC_CTRL2_REG.GP_ADC_STORE_DEL=0x3, GP_ADC_CTRL2_REG.GP_ADC_SMPL_TIME=0x4,	-2		2	LSB

Parameter	Description	Conditions	Min	Typ	Max	Unit
		GP_ADC_CTRL2_REG.GP_ADC_CONV_NRS=0x7				
ENOB	Effective Number of Bits	No averaging, no chopping, Single-Ended: $V_{IN,PP}$ = 800 mV		8		bit
$ENOB_{AVG128}$	Effective Number of Bits	128x averaging, Single-Ended: $V_{IN,PP}$ = 800 mV		11		bit

Table 15: GP ADC - Electrical performance

Parameter	Description	Conditions	Min	Typ	Max	Unit
t_{CONV_ADC}	Conversion time of the ADC				500	ns
t_{ON_LDO}	Turn-on time of the LDO	GP_ADC_CTRL2_REG[GP_ADC_I20U] = 1. Settling of LDO within 10 bit			15	µs

3.10 ΣΔ ADC characteristics

3.10.1 Audio ADC characteristics

Table 16: ADC audio - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_s	Output sample rate	OSR = 62.5x (fixed)		16		kHz
$V_{IN\_FS\_DIF}$	Differential full-scale input voltage (peak-to-peak)			2*VREF/ACL		V
$V_{IN\_FS\_SE}$	Single-Ended full-scale input voltage (peak-to-peak)			VREF/ACL		V
V_{IN_CM}	Common mode input voltage	Internally generated		450		mV

Table 17: ADC audio - Electrical performance

Parameter	Description	Conditions	Min	Typ	Max	Unit
A_{DROOP}	Passband droop	A_{CL} = 0 dB, 20 Hz <= freq <= 7 kHz (decimation filter f_{cutoff} is at 7 kHz)		0.4		dB
A_{STP}	Gain step	f_{signal} = 1 kHz	5.8	6	6.1	dB
$SFDR_{\_DIF}$	Spurious-free dynamic range in Differential mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		71.5		dBFS
$SFDR_{\_SE}$	Spurious-free dynamic range in Single-Ended mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		75		dBFS

Parameter	Description	Conditions	Min	Typ	Max	Unit
$SNR_{\_DIF}$	Signal to noise ratio in Differential mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		73		dB
$SNR_{\_SE}$	Signal to noise ratio in Single-Ended mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		69.5		dB
$THD_{\_DIF}$	Total harmonic distortion in Differential mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		-71		dB
$THD_{\_SE}$	Total harmonic distortion in Single-Ended mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		-73		dB
$SINAD_{\_DIF}$	Signal-to-noise and distortion ratio in Differential mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		68.5		dB
$SINAD_{\_SE}$	Signal-to-noise and distortion ratio in Single-Ended mode	f_{signal} = 1 kHz, A_{CL} = 0 dB, V_{IN} = -2 dBFS		67		dB
V_{IN_OFS}	Input offset voltage		-4	0	4	mV
A_{CL_min}	Closed loop gain	PGA_GAIN = 0x0 (-12 dB) f_{signal} = 1 kHz		-12		dB
A_{CL_max}	Closed loop gain	PGA_GAIN = 0x7 (+30 dB) f_{signal} = 1 kHz		30		dB

Table 18: ADC audio - AC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$PSRR_{\_DIF}$	Power supply rejection ratio	V30, A_{CL} = 0 dB, 20 Hz <= fdist <= 8 kHz	60			dB
$PSRR_{\_SE}$	Power supply rejection ratio	V30, A_{CL} = 0 dB, 20 Hz <= fdist <= 8 kHz	60			dB
Z_I	Input impedance	Differential mode, A_{CL} = 0 dB to +30 dB		20		kΩ

Table 19: ADC audio - External electrical conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
N_{BIT}	Number of bits output word			16		bits

3.10.2 Sensor ADC characteristics

Table 20: ADC sensor - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
N_{BIT}	Number of bits (resolution)	OSR = 1024x		16		bit
V_{REF}	Reference voltage			0.9		V
f_S	Output sample rate	OSR 256x <-> 2048x	488		3906	Hz

Parameter	Description	Conditions	Min	Typ	Max	Unit
$V_{IN\_FS\_DIFF}$	Differential full-scale input voltage (peak-to-peak)			2*VREF		V
$V_{IN\_FS\_SE}$	Single ended full-scale input voltage (peak-to-peak)			VREF		V

Table 21: ADC sensor - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$INL_{\_DIF\_int\_ref}$	Integral non-linearity, LSB wrt 15 bits accuracy Note 1	OSR=1024x, Differential mode, Internal Reference	-2		11.5	LSB
$INL_{\_SE\_int\_ref}$	Integral non-linearity, LSB wrt 15 bits accuracy	OSR=1024x, Single Ended Mode, Internal Reference	-2		12	LSB
$INL_{\_DIF\_ext\_ref}$	Integral non-linearity, LSB wrt 15 bits accuracy	OSR=1024x, Differential Mode, External reference	-1		12	LSB
$INL_{\_SE\_ext\_ref}$	Integral non-linearity, LSB wrt 15 bits accuracy	OSR=1024x, Single Ended Mode, External reference	-4		4	LSB
$DNL_{\_DIF\_int\_ref}$	Differential non-linearity, LSB wrt 15 bits accuracy	OSR=1024x, Differential Mode, Internal reference	-2		2	LSB
$DNL_{\_SE\_int\_ref}$	Differential non-linearity, LSB wrt 15 bits accuracy	OSR=1024x, Single Ended Mode, Internal reference	-2		2	LSB
$DNL_{\_DIF\_ext\_ref}$	Differential non-linearity, LSB wrt 15 bits accuracy	OSR=1024x, Differential Mode, External reference	-1		1	LSB
$DNL_{\_SE\_ext\_ref}$	Differential non-linearity, LSB wrt 15 bits accuracy	OSR=1024x, Single Ended Mode, External reference	-1		1	LSB
$E_{G\_DIF\_ext\_ref}$	Gain error	OSR=1024x, Differential mode, External reference, T = 25	-0.25		0.25	%
$E_{G\_SE\_ext\_ref}$	Gain error	OSR=1024x, Single Ended Mode, External reference, T = 25,	-0.25		0.25	%
$E_{OFS\_DIF\_int\_ref}$	Offset error, LSB wrt 15b accuracy	OSR=1024x, Differential mode, Internal reference, T = 25	-2		2	LSB
$E_{OFS\_SE\_int\_ref}$	Offset error, LSB wrt 15b accuracy	OSR=1024x, Single Ended Mode, Internal reference, T = 25	-2		2	LSB
$E_{OFS\_DIF\_ext\_ref}$	Offset error, LSB wrt 15b accuracy	OSR=1024x, Differential Mode, External reference, T = 25	-1		1	LSB
$E_{OFS\_SE\_ext\_ref}$	Offset error, LSB wrt 15b accuracy	OSR=1024x, Differential mode, External reference, T = 25	-1		1	LSB
I_{SUP_250kHz}	supply current	DCDC off, VBAT = 3 V, f_CLK = 250 kHz		135		µA

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{SUP_500kHz}	supply current	DCDC off, VBAT = 3 V, f_CLK = 500 kHz		150		µA
I_{SUP_1MHz}	supply current	DCDC off, VBAT = 3 V, f_CLK = 1 MHz		175		µA
I_{SUP_2MHz}	supply current	DCDC off, VBAT = 3 V, f_CLK = 2 MHz		220		µA

Note 1 INL is the deviation of a code from a straight line passing through the actual end points of the transfer curve

Table 22: ADC sensor - Electrical performance

Parameter	Description	Conditions	Min	Typ	Max	Unit
$SNR_{DIF\_int\_ref}$	Signal to noise ratio	OSR = 1024x, Differential mode, Internal reference, F_{IN} = 120 Hz, V_{IN} = 0.9*VIN_FS_DIFF	77			dB
$SNR_{SE\_int\_ref}$	Signal to noise ratio	OSR = 1024x, Single Ended mode, Internal reference, F_{IN} = 120 Hz, V_{IN} = 0.9*VIN_FS_SE	70			dB
$SNR_{DIF\_ext\_ref}$	Signal to noise ratio	OSR = 1024x, Differential mode, External reference, F_{IN} = 120 Hz, V_{IN} = 0.9*VIN_FS_DIFF	84			dB
$SNR_{SE\_ext\_ref}$	Signal to noise ratio	OSR = 1024x, Single Ended mode, External reference, F_{IN} = 120 Hz, V_{IN} = 0.9*VIN_FS_SE	82			dB
PSRR	Power supply rejection ratio	At 120 Hz	60			dB

3.11 DCDC converter characteristics

Table 23: DCDC - Recommended Operating Conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
L_{EXT}	External inductor		2	2.2	2.4	µH
C_{OUT}	External capacitor	Effective value at DC output voltage		2		µF
I_{L_ACTIVE}	Load current active mode			15	40	mA
I_{L_SLEEP}	Load current sleep mode			2	300	µA

Table 24: DCDC - DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{ACC_ACTIVE}	Output voltage accuracy	ACTIVE MODE		3		%

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{ACC_SLEEP}	Output voltage accuracy	SLEEP MODE		4		%
$V_{O\_DCDC\_0}$	DC output voltage	DCDC_LEVEL = 0x0		1.1		V
$V_{O\_DCDC\_1}$	DC output voltage	DCDC_LEVEL = 0x1		1.15		V
$V_{O\_DCDC\_2}$	DC output voltage	DCDC_LEVEL = 0x2		1.2		V
$V_{O\_DCDC\_3}$	DC output voltage	DCDC_LEVEL = 0x3		1.25		V
$V_{O\_DCDC\_4}$	DC output voltage	DCDC_LEVEL = 0x4		1.3		V
$V_{O\_DCDC\_5}$	DC output voltage	DCDC_LEVEL = 0x5		1.35		V
$V_{O\_DCDC\_6}$	DC output voltage	DCDC_LEVEL = 0x6		1.4		V
$V_{O\_DCDC\_7}$	DC output voltage	DCDC_LEVEL = 0x7		1.45		V
η_{CONV}	Conversion efficiency	V_{BAT} = 3.0 V V_{OUT} = 1.1 V I_{LOAD} = 20 mA Normal Mode		90		%
V_{RPL_PP}	Ripple voltage	V_{BAT} = 3.0 V V_{OUT} = 1.1 V I_{LOAD} = 1 - 40 mA C_{EXT} = 2 µF	5	15	25	mV
I_{Q_VBAT}	Quiescent current	VBAT = 3.0 V No load	10	18	26	µA

Table 25: DCDC - External electrical conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
ESR_L	Inductor ESR				200	mΩ

3.12 LDOs characteristics

Table 26: LDO_CORE - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{L_LDO}	LDO load current	V_DROP_LDO_CORE >= 200 mV	20		0	mA
C_{L_LDO}	Effective load capacitance		0.47		5	µF
ESR_{CL_LDO}	Equivalent series resistance		1		100	mΩ

Table 27: LDO_CORE - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{Q\_VBAT\_ACT}$	Quiescent current	LDO enabled, Unloaded		10		µA
$I_{Q\_VDCDC\_ACT}$	Quiescent current	LDO enabled, Unloaded		10		µA

Table 28: LDO_CORE_ret - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{L_MAX}	Maximum load current	V_DROP_LDO_CORE_RET >= 200 mV	0.4			mA
C_{L_LDO}	Effective load capacitance		0.47		4.7	µF
ESR_{CL_LDO}	Equivalent series resistance		1		100	mΩ

Table 29: LDO_IO - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{L_LDO}	Load current		0		20	mA
C_{L_LDO}	Effective load capacitance		2.2		100	µF
ESR_{CL_LDO}	Equivalent series resistance		1		100	mΩ

Table 30: LDO_IO - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{LDO}	LDO output voltage	V_{BAT} >= 2.1 V	1.62	1.8	1.98	V

Table 31: LDO_IO_RET - Recommended Operating Conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{L_LDO}	Load current	V_{BAT} > 1.9 V			2	mA
C_{L_LDO}	Effective load capacitance		2.2		100	µF
$ESR_{CL\_LDO\_1v8}$	Equivalent series resistance		1		100	mΩ

Table 32: LDO_LOW - Recommended Operating Conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{L\_LDO\_LOW\_ACTIVE}$	Load current		0.1		40	mA
$I_{L\_LDO\_LOW\_SLEEP}$	Load current		0.1		1000	µA
$C_{L\_LDO\_LOW}$	Effective load capacitance		2.2		10	µF

Parameter	Description	Conditions	Min	Typ	Max	Unit
$ESR_{CL\_LDO\_LOW}$	Equivalent series resistance		1		100	mΩ

3.13 Embedded Flash characteristics

Table 33: eFlash - Absolute Maximum Ratings

Parameter	Description	Conditions	Min	Max	Unit
N_{BIT_WRITE}	Number of times a bit can be programmed before erase			2	-
N_{ENDU}	Endurance in erase cycles per page		10000		cycle
t_{DATA_RET}	Data retention	At 85 °C	10		Years

Table 34: eFlash - Timing Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$t_{ACC\_0.9}$	Access time	VDD = 0.9 V			65	ns
$t_{ACC\_1.2}$	Access time	VDD = 1.2 V			23	ns
t_{PROG}	Time for programming a word	VDD = 1.2 V VBAT = 2.5 V	8		16	μs
t_{ERASE_PAGE}	Page Erase Time (page = 2 kB)	VDD = 1.2 V VBAT = 2.5 V	80		160	ms
t_{ERASE_MASS}	Mass Erase Time	VDD = 1.2 V VBAT = 2.5 V	80		160	ms
t_{SE_SEG}	Suspend erase time segment		1	10		ms
t_{WU_SLP}	Wake-up time from sleep to standby		2.5		3.5	μs

3.14 32 kHz crystal oscillator characteristics

Table 35: XTAL oscillator 32kHz - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_{CLK_EXT}	External clock frequency	At pin XTAL32Kp/P1_14 in GPIO mode	10		100	kHz
ESR	Equivalent series resistance				100	kΩ
C_L	Load capacitance	No external capacitors are required for a 6 pF or 7 pF crystal.	6	7	9	pF
C_0	Shunt capacitance			1	2	pF

Parameter	Description	Conditions	Min	Typ	Max	Unit
Δf_{XTAL}	Crystal frequency tolerance (including aging)	Timing accuracy is dominated by crystal accuracy. A much smaller value is preferred.	-250		250	ppm
P_{DRV_MAX}	Maximum drive power	Note 1	0.1			µW

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

3.15 32 MHz crystal oscillator characteristics

Table 36: XTAL32MHz Oscillator - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
Δf_{XTAL_TRIM}	Crystal frequency trim			2		ppm
V_{CLK_EXT}	External clock voltage	In case of external clock source on XTAL32Mp (XTAL32Mm floating or connected to mid-level 0.6 V)		0.9		V
$P_{DRV(MAX)}$	Maximum drive power	A crystal used that can handle the drive level specified.			100	µW
$\varphi_{N\_EXT\_32M}$	Phase noise	f_C = 50 kHz; in case of external clock source			-130	dBc/Hz
f_{XTAL_32M}	Crystal oscillator frequency			32		MHz
Δf_{XTAL}	Crystal frequency tolerance	After optional trimming; including aging and temperature drift Note 1	-20		20	ppm
Δf_{XTAL_UNT}	Crystal frequency tolerance	Untrimmed; including aging and temperature drift Note 2	-40		40	ppm
$ESR_{\_1pF}$	Equivalent series resistance	C_0 < 1pF			200	Ω
$ESR_{\_3pF}$	Equivalent series resistance	C_0 < 3 pF			80	Ω
$ESR_{\_5pF}$	Equivalent series resistance	C_0 < 5 pF			50	Ω
C_L	Load capacitance	No external capacitors are required	4	6	8	pF

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.

Table 37: XTAL32MHz Oscillator - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
OSF(32M)	Oscillator Safety Factor		3			

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{SUP\_VBAT\_ACTIVE}$	Supply current			5		µA
$I_{SUP\_VDCDC\_ACTIVE}$	Supply current			250		µA

Table 38: XTAL32MHz Oscillator - Timing characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$t_{STA(XTAL)}$(32M)	Crystal oscillator start-up time			0.65		ms

3.16 RCX oscillator characteristics

Table 39: RCX Oscillator - Timing characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$\Delta f_{RC}/\Delta T_{\_1}$	Frequency accuracy per °C	0<T<60			150	ppm/deg
$\Delta f_{RC}/\Delta T_{\_2}$	Frequency accuracy per °C	-40<T<115		100	200	ppm/deg
$\Delta f_{RC}/\Delta V_{\_VBAT}$	Supply voltage dependency (V_{BAT_HIGH})			50		ppm/V
$\Delta f_{RC}/\Delta V_{\_VDD}$	Supply voltage dependency (VDD)			50		ppm/V
f_{RC}	RC oscillator frequency	At default trim setting	12	15	19	kHz

3.17 32/512 kHz RC oscillator characteristics

Table 40: RCLP Oscillator - Timing characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_{RCLP_32}	RCLP oscillator frequency	Trimmed	20	32	50	kHz
f_{RCLP_512}	RCLP oscillator frequency	Trimmed	350	512	650	kHz

3.18 32 MHz RC oscillator characteristics

Table 41: RC32MHz Oscillaor - Timing characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$f_{RC32M\_TRIMMED}$	RC oscillator frequency	RC re-trimmed at every temperature	28.5	30.2	32	MHz

3.19 Clock doubler characteristics

Table 42: Clock doubler - AC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_{CLK_OUT}	Output clock frequency			64		MHz
f_{CLK_IN}	Input clock frequency	Input clock can only be the XTAL32M		32		MHz

3.20 Temperature sensor characteristics

Table 43: Temperature Sensor - Electrical performance

Parameter	Description	Conditions	Min	Typ	Max	Unit
$TC_{\_SENSE}$	Temperature coefficient of the internal temperature sensor, reading via GP_ADC		2.1	2.3	2.5	LSB/°C

3.21 Digital I/O characteristics

Table 44: GPIO - Recommended Operating Conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{DDIO}	Pad supply voltage		1.65		3.6	V
V_{IH}	HIGH level input voltage	V_{DD} = 0.9 V, V_{DDIO} = 1.8 V	0.63			V
V_{IL}	LOW level input voltage	V_{DD} = 0.9 V, V_{DDIO} = 1.8 V			0.27	V

Table 45: GPIO - DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{IH}	HIGH level input current	V_I=VDDIO = 3.0 V, T = 25		0.5		nA
I_{IL}	LOW level input current	V_I=VSSIO = 0 V, VDDIO = 3.0 V, T = 25		0.5		nA
I_{IH_PD}	HIGH level input current	V_I=VDDIO = 1.8 V	35		110	µA
I_{IL_PU}	LOW level input current	V_I=V_{SS} = 0 V, VDDIO = 1.8 V	-110		-35	µA
V_{OH}	HIGH level output voltage	I_O = 3.5 mA, VDDDIO = 1.8 V	1.44			V
V_{OL}	LOW level output voltage	I_O = 3.5 mA, VDDIO = 1.8 V			0.36	V
V_{OH_LOWDRV}	HIGH level output voltage	I_O = 0.35 mA, VDDIO = 1.8 V	1.44			V

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{OL_LOWDRV}	LOW level output voltage	I_O = 0.35 mA, VDDIO = 1.8 V			0.36	V
C_{IN}	Input capacitance			0.75		pF

3.22 Wake-up I/O characteristics

Table 46: GPIO_WAKEUP - Recommended Operating Conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{IH_VBAT}	HIGH level input voltage	V_{BAT} = 1.8 V Note 1	1.26			V
V_{IL_VBAT}	LOW level input voltage	V_{BAT} = 1.8 V			0.54	V

Note 1 Pad characteristics are equal to the standard GPIO pads unless overruled or added in this table

3.23 QSPI characteristics

Table 47: QSPI PAD - Recommended Operating Conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
V_{DDIO}	Pad supply voltage		1.65		3.6	V
V_{IH}	HIGH level input voltage	V_{DDIO} = 1.8 V	1.26			V
V_{IL}	LOW level input voltage	V_{DDIO} = 1.8 V			0.54	V

Table 48: QSPI PAD - DC Characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
I_{IH}	HIGH level input current	V_I=VDDIO = 3.0 V, T = 25		0.5		nA
I_{IL}	LOW level input current	V_I=VSSIO = 0 V, VDDIO = 3.0 V, T = 25		0.5		nA
V_{OH_16mA}	HIGH level output voltage	I_O = 16 mA, VDDIO = 1.8 V	1.44			V
V_{OL_16mA}	LOW level output voltage	I_O = 16 mA, VDDIO = 1.8 V			0.36	V
I_{IH_PD}	HIGH level input current with pull-down	V_I=VDDIO = 1.8 V	25		75	µA
I_{IL_PU}	LOW level input current with pull-up	V_I=VSS, VDDIO = 1.8 V	-75		-25	µA
C_{IN}	Input capacitance			0.87		pF

3.24 Radio characteristics

Table 49: Radio BLE 1M LP - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_{OPER}	Operating frequency		2400		2483.5	MHz
N_{CH}	Number of channels			40		1
f_{CH}	Channel frequency	K = 0 to 39		2402+ K*2		MHz

Table 50: Radio BLE 1M LP - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{BAT\_RF\_RX}$	Battery supply current	Radio receiver and synthesizer active; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		1.2		mA
$I_{BAT\_RF\_TX\_0dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 9; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		2.3		mA
$I_{BAT\_RF\_TX\_-3dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 6; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		1.7		mA
$I_{BAT\_RF\_TX\_-6dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 4; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		1.3		mA
$I_{BAT\_RF\_TX\_-12dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 2; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		0.9		mA
$I_{BAT\_RF\_TX\_-18dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 1; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		0.7		mA

Table 51: Radio BLE 1M LP - AC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
P_{SENS_CLEAN}	Sensitivity level	Dirty Transmitter disabled; DC-DC converter disabled; PER = 30.8%; V_{DCDC_RF} = 1.1 V Note 1		-96		dBm
P_{SENS_EPKT}	Sensitivity level	Extended packet size (255 octets)		-93.5		dBm
P_{SENS}	Sensitivity level	Normal Operating Conditions; DC-DC converter disabled; PER = 30.8% Note 1		-95		dBm
P_{INT_IMD}	Intermodulation distortion interferer power level	Worst-case interferer level @ f_1, f_2 with $2*f_1 - f_2 = f_0$, $\|f_1 - f_2\|$ = n MHz and n = 3, 4, 5; P_{WANTED} = -64 dBm @ f_0; PER = 30.8% Note 2		-28		dBm
CIR_0	Carrier to interferer ratio	n = 0; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		7		dB
CIR_{M1}	Carrier to interferer ratio	n = -1; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-4.5		dB
CIR_{P1}	Carrier to interferer ratio	n = +1; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-3		dB
CIR_{P2}	Carrier to interferer ratio	n = +2; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-29		dB
CIR_{M2}	Carrier to interferer ratio	n = -2 (image frequency); interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-38		dB
CIR_{P3}	Carrier to interferer ratio	n = +3; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-41		dB
CIR_{M3}	Carrier to interferer ratio	n = -3 (image frequency + 1 MHz); interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-47		dB
CIR_{P4}	Carrier to interferer ratio	n = +4; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-48		dB
CIR_{M4}	Carrier to interferer ratio	n = -4; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-51		dB

Parameter	Description	Conditions	Min	Typ	Max	Unit
CIR_5	Carrier to interferer ratio	$\|n\| \geq 5$ (any other BLE channel); interferer @ $f_1 = f_0 + n*1$ MHz Note 2		-51		dB
P_{BL_I}	Blocker power level	30 MHz ≤ f_{BL} ≤ 2000 MHz; P_{WANTED} = -67 dBm Note 2		5		dBm
P_{BL_II}	Blocker power level Note 3	2003 MHz ≤ f_{BL} ≤ 2399 MHz; P_{WANTED} = -67 dBm Note 2		0		dBm
P_{BL_III}	Blocker power level	2484 MHz ≤ f_{BL} ≤ 2997 MHz; P_{WANTED} = -67 dBm Note 2		0		dBm
P_{BL_IV}	Blocker power level	3000 MHz ≤ f_{BL} ≤ 12.75 GHz; P_{WANTED} = -67 dBm Note 2		5		dBm
L_{ACC_RSSI}	Level accuracy	Tolerance at 5% to 95% confidence interval of P_{RF}: when RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		2		dB
L_{RES_RSSI}	Level resolution	Gradient of monotonous range for RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		0.5		dB/LSB
ACP_{2M}	Adjacent channel power level	f_{OFS} = 2 MHz Note 4		-50		dBm
ACP_{3M}	Adjacent channel power level	f_{OFS} ≥ 3 MHz Note 4		-56		dBm
P_{O_15}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 15		4.5		dBm
P_{O_14}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 14		3.5		dBm
P_{O_13}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 13		3		dBm
P_{O_12}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 12		2.5		dBm
P_{O_11}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 11		2		dBm
P_{O_10}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 10		1		dBm
P_{O_09}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 9		0		dBm

Parameter	Description	Conditions	Min	Typ	Max	Unit
P_{O_08}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 8		-1		dBm
P_{O_07}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 7		-2		dBm
P_{O_06}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 6		-3		dBm
P_{O_05}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 5		-4.5		dBm
P_{O_04}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 4		-6.5		dBm
P_{O_03}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 3		-9		dBm
P_{O_02}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 2		-12.5		dBm
P_{O_01}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 1		-18.5		dBm
P_{O_01A1}	Output power level	Power set to -22 dBm		-22		dBm
P_{O_01A2}	Output power level	Power set to -23 dBm		-23		dBm
P_{O_ULP}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 0		-58		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® Core Technical Specification document.

Note 3 Frequencies close to the ISM band can show slightly worse performance

Note 4 Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.2.3.

Table 52: Radio BLE 1M HP - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_{OPER}	Operating frequency		2400		2483.5	MHz
N_{CH}	Number of channels			40		1
f_{CH}	Channel frequency	K = 0 to 39		2402+ K*2		MHz

Table 53: Radio BLE 1M HP - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{BAT\_RF\_RX}$	Battery supply current	Radio receiver and synthesizer active; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		2.1		mA

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{BAT\_RF\_TX\_+6dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 15; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		6.2		mA
$I_{BAT\_RF\_TX\_0dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 8; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		3		mA
$I_{BAT\_RF\_TX\_-3dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 6; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		2.5		mA
$I_{BAT\_RF\_TX\_-6dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 4; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		1.9		mA
$I_{BAT\_RF\_TX\_-12dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 2; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		1.3		mA
$I_{BAT\_RF\_TX\_-18dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 1; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		1.1		mA

Table 54: Radio BLE 1M HP - AC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
P_{SENS_CLEAN}	Sensitivity level	Dirty Transmitter disabled; DC-DC converter disabled; PER = 30.8%; V_{DCDC_RF} = 1.4 V Note 1		-97		dBm
P_{SENS_EPKT}	Sensitivity level	Extended packet size (255 octets)		-94.5		dBm
P_{SENS}	Sensitivity level	Normal Operating Conditions; DC-DC converter disabled; PER = 30.8% Note 1		-96		dBm

Parameter	Description	Conditions	Min	Typ	Max	Unit
P_{INT_IMD}	Intermodulation distortion interferer power level	Worst-case interferer level @ f_1, f_2 with $2{*}f_1 - f_2 = f_0$, $\|f_1 - f_2\|$ = n MHz and n = 3, 4, 5; P_{WANTED} = -64 dBm @ f_0; PER = 30.8% Note 2		-27		dBm
CIR_0	Carrier to interferer ratio	n = 0; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		6.5		dB
CIR_{M1}	Carrier to interferer ratio	n = -1; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-5		dB
CIR_{P1}	Carrier to interferer ratio	n = +1; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-3		dB
CIR_{P2}	Carrier to interferer ratio	n = +2; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-30		dB
CIR_{M2}	Carrier to interferer ratio	n = -2 (image frequency); interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-38		dB
CIR_{P3}	Carrier to interferer ratio	n = +3; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-42		dB
CIR_{M3}	Carrier to interferer ratio	n = -3 (image frequency + 1 MHz); interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-48		dB
CIR_{P4}	Carrier to interferer ratio	n = +4; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-49		dB
CIR_{M4}	Carrier to interferer ratio	n = -4; interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-51		dB
CIR_5	Carrier to interferer ratio	|n| ≥ 5 (any other BLE channel); interferer @ $f_1 = f_0$ + n*1 MHz Note 2		-53		dB
P_{BL_I}	Blocker power level	30 MHz ≤ f_{BL} ≤ 2000 MHz; P_{WANTED} = -67 dBm Note 2		5		dBm
P_{BL_II}	Blocker power level Note 3	2003 MHz ≤ f_{BL} ≤ 2399 MHz; P_{WANTED} = -67 dBm Note 2		0		dBm
P_{BL_III}	Blocker power level	2484 MHz ≤ f_{BL} ≤ 2997 MHz; P_{WANTED} = -67 dBm		0		dBm

Parameter	Description	Conditions	Min	Typ	Max	Unit
		Note 2				
P_{BL_IV}	Blocker power level	3000 MHz ≤ f_{BL} ≤ 12.75 GHz; P_{WANTED} = -67 dBm Note 2		5		dBm
L_{ACC_RSSI}	Level accuracy	Tolerance at 5% to 95% confidence interval of P_{RF}: when RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		2		dB
L_{RES_RSSI}	Level resolution	Gradient of monotonous range for RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		0.5		dB/LSB
ACP_{2M}	Adjacent channel power level	f_{OFS} = 2 MHz Note 4		-49		dBm
ACP_{3M}	Adjacent channel power level	f_{OFS} ≥ 3 MHz Note 4		-57		dBm
P_{O_15}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 15		6		dBm
P_{O_14}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 14		5		dBm
P_{O_13}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 13		4.5		dBm
P_{O_12}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 12		3.5		dBm
P_{O_11}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 11		2.5		dBm
P_{O_10}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 10		2		dBm
P_{O_09}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 9		1		dBm
P_{O_08}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 8		0		dBm
P_{O_07}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 7		-1		dBm
P_{O_06}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 6		-2.5		dBm
P_{O_05}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 5		-4		dBm
P_{O_04}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 4		-6		dBm
P_{O_03}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 3		-8.5		dBm

Parameter	Description	Conditions	Min	Typ	Max	Unit
P_{O_02}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 2		-12		dBm
P_{O_01}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 1		-18		dBm
P_{O_01A1}	Output power level	Power set to -21 dBm		-21		dBm
P_{O_01A2}	Output power level	Power set to -22 dBm		-22		dBm
P_{O_ULP}	Output power level	RF_ATTR_REG[PA_POWER_SETTING] = 0		-52		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® Core Technical Specification document.

Note 3 Frequencies close to the ISM band can show slightly worse performance

Note 4 Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.2.3.

Table 55: Radio BLE 2M LP - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_{OPER}	Operating frequency		2400		2483.5	MHz
N_{CH}	Number of channels			40		1
f_{CH}	Channel frequency	K = 0 to 39		2402+K*2		MHz

Table 56: Radio BLE 2M LP - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{BAT\_RF\_RX}$	Battery supply current	Radio receiver and synthesizer active; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		1.3		mA
$I_{BAT\_RF\_TX\_0dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 9; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		2.3		mA
$I_{BAT\_RF\_TX\_-3dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 6; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		1.7		mA
$I_{BAT\_RF\_TX\_-6dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 4; ideal DC-DC		1.3		mA

Parameter	Description	Conditions	Min	Typ	Max	Unit
		converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C				
$I_{BAT\_RF\_TX\_-12dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 2; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		0.9		mA
$I_{BAT\_RF\_TX\_-18dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 1; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.1 V; T_A = 25 °C		0.7		mA

Table 57: Radio BLE 2M LP - AC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
P_{SENS_CLEAN}	Sensitivity level	Dirty Transmitter disabled; DC-DC converter disabled; PER = 30.8%; V_{DCDC_RF} = 1.1 V Note 1		-93		dBm
P_{SENS_EPKT}	Sensitivity level	Extended packet size (255 octets)		-91		dBm
P_{SENS}	Sensitivity level	Normal Operating Conditions; DC-DC converter disabled; PER = 30.8% Note 1		-93		dBm
P_{INT_IMD}	Intermodulation distortion interferer power level	Worst-case interferer level @ f_1, f_2 with $2*f_1 - f_2 = f_0$, $\|f_1 - f_2\|$ = n x 2 MHz and n = 3, 4, 5; P_{WANTED} = -64 dBm @ f_0; PER = 30.8% Note 2		-28		dBm
CIR_0	Carrier to interferer ratio	n = 0; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		7		dB
CIR_{P1}	Carrier to interferer ratio	n = +1; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		-4		dB
CIR_{M1}	Carrier to interferer ratio	n = -1; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		-4		dB
CIR_{P2}	Carrier to interferer ratio	n = +2; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		-29		dB

Parameter	Description	Conditions	Min	Typ	Max	Unit
CIR_{M2}	Carrier to interferer ratio	n = -2 (image frequency); interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-36		dB
CIR_{P3}	Carrier to interferer ratio	n = +3; interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-41		dB
CIR_{M3}	Carrier to interferer ratio	n = -3 (image frequency + 2 MHz); interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-47		dB
CIR_{M4}	Carrier to interferer ratio	n = -4; interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-46		dB
CIR_{P4}	Carrier to interferer ratio	n = +4; interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-41		dB
CIR_5	Carrier to interferer ratio	$\|n\| \geq 5$ (any other BLE channel); interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-52		dB
P_{BL_I}	Blocker power level	30 MHz ≤ f_{BL} ≤ 2000 MHz; P_{WANTED} = -67 dBm Note 2		5		dBm
P_{BL_II}	Blocker power level Note 3	2003 MHz ≤ f_{BL} ≤ 2399 MHz; P_{WANTED} = -67 dBm Note 2		0		dBm
P_{BL_III}	Blocker power level	2484 MHz ≤ f_{BL} ≤ 2997 MHz; P_{WANTED} = -67 dBm Note 2		0		dBm
P_{BL_IV}	Blocker power level	3000 MHz ≤ f_{BL} ≤ 12.75 GHz; P_{WANTED} = -67 dBm Note 2		5		dBm
L_{ACC_RSSI}	Level accuracy	Tolerance at 5% to 95% confidence interval of P_{RF}: when RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		2		dB
L_{RES_RSSI}	Level resolution	Gradient of monotonous range for RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		0.5		dB/LSB
ACP_{4M}	Adjacent channel power level	f_{OFS} = 4 MHz Note 4		-54		dBm
ACP_{5M}	Adjacent channel power level	f_{OFS} = 5 MHz Note 4		-60		dBm

Parameter	Description	Conditions	Min	Typ	Max	Unit
ACP_{6M}	Adjacent channel power level	f_{OFS} ≥ 6 MHz Note 4		-58		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® Core Technical Specification document.

Note 3 Frequencies close to the ISM band can show slightly worse performance.

Note 4 Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.2.3.

Table 58: Radio BLE 2M HP - Recommended operating conditions

Parameter	Description	Conditions	Min	Typ	Max	Unit
f_{OPER}	Operating frequency		2400		2483.5	MHz
N_{CH}	Number of channels			40		1
f_{CH}	Channel frequency	K = 0 to 39		2402+ K*2		MHz

Table 59: Radio BLE 2M HP - DC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{BAT\_RF\_RX}$	Battery supply current	Radio receiver and synthesizer active; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		2.2		mA
$I_{BAT\_RF\_TX\_+6dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 15; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		6.2		mA
$I_{BAT\_RF\_TX\_0dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 8; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		3		mA
$I_{BAT\_RF\_TX\_-3dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 6; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		2.5		mA
$I_{BAT\_RF\_TX\_-6dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 4; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		1.9		mA

Parameter	Description	Conditions	Min	Typ	Max	Unit
$I_{BAT\_RF\_TX\_-12dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 2; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		1.3		mA
$I_{BAT\_RF\_TX\_-18dBm}$	Battery supply current	Radio transmitter and synthesizer active; power setting = 1; ideal DC-DC converter with V_{BAT} = 3 V and V_{DCDC_RF} = 1.4 V; T_A = 25 °C		1.1		mA

Table 60: Radio BLE 2M HP - AC characteristics

Parameter	Description	Conditions	Min	Typ	Max	Unit
P_{SENS_CLEAN}	Sensitivity level	Dirty Transmitter disabled; DC-DC converter disabled; PER = 30.8%; V_{DCDC_RF} = 1.4 V Note 1		-94		dBm
P_{SENS_EPKT}	Sensitivity level	Extended packet size (255 octets)		-91.5		dBm
P_{SENS}	Sensitivity level	Normal Operating Conditions; DC-DC converter disabled; PER = 30.8% Note 1		-93.5		dBm
P_{INT_IMD}	Intermodulation distortion interferer power level	Worst-case interferer level @ f_1, f_2 with $2*f_1 - f_2 = f_0$, $\|f_1 - f_2\|$ = n x 2 MHz and n = 3, 4, 5; P_{WANTED} = -64 dBm @ f_0; PER = 30.8% Note 2		-27		dBm
CIR_0	Carrier to interferer ratio	n = 0; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		6.5		dB
CIR_{P1}	Carrier to interferer ratio	n = +1; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		-4.5		dB
CIR_{M1}	Carrier to interferer ratio	n = -1; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		-4		dB
CIR_{P2}	Carrier to interferer ratio	n = +2; interferer @ $f_1 = f_0$ + n*2 MHz Note 2		-30		dB
CIR_{M2}	Carrier to interferer ratio	n = -2 (image frequency); interferer @ $f_1 = f_0$ + n*2 MHz Note 2		-37		dB

Parameter	Description	Conditions	Min	Typ	Max	Unit
CIR_{P3}	Carrier to interferer ratio	n = +3; interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-42		dB
CIR_{M3}	Carrier to interferer ratio	n = -3 (image frequency + 2 MHz); interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-49		dB
CIR_{M4}	Carrier to interferer ratio	n = -4; interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-48		dB
CIR_{P4}	Carrier to interferer ratio	n = +4; interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-42		dB
CIR_5	Carrier to interferer ratio	$\|n\| \geq 5$ (any other BLE channel); interferer @ $f_1 = f_0 + n*2$ MHz Note 2		-54		dB
P_{BL_I}	Blocker power level	30 MHz ≤ f_{BL} ≤ 2000 MHz; P_{WANTED} = -67 dBm Note 2		5		dBm
P_{BL_II}	Blocker power level Note 3	2003 MHz ≤ f_{BL} ≤ 2399 MHz; P_{WANTED} = -67 dBm Note 2		0		dBm
P_{BL_III}	Blocker power level	2484 MHz ≤ f_{BL} ≤ 2997 MHz; P_{WANTED} = -67 dBm Note 2		0		dBm
P_{BL_IV}	Blocker power level	3000 MHz ≤ f_{BL} ≤ 12.75 GHz; P_{WANTED} = -67 dBm Note 2		5		dBm
L_{ACC_RSSI}	Level accuracy	Tolerance at 5% to 95% confidence interval of P_{RF}: when RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		2		dB
L_{RES_RSSI}	Level resolution	Gradient of monotonous range for RXRSSI[7:0] = *X*, 50 < *X* < 175; burst mode, 1500 packets		0.5		dB/LSB
ACP_{4M}	Adjacent channel power level	f_{OFS} = 4 MHz Note 4		-53		dBm
ACP_{5M}	Adjacent channel power level	f_{OFS} = 5 MHz Note 4		-59		dBm
ACP_{6M}	Adjacent channel power level	f_{OFS} ≥ 6 MHz Note 4		-59		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® Core Technical Specification document.

Note 3 Frequencies close to the ISM band can show slightly worse performance.

Note 4 Measured according to Bluetooth® Low Energy Test Specification RF-PHY.TS/4.0.1, section 6.2.3.

4 System Overview

4.1 Internal blocks

The DA14592 contains the following blocks:

Arm Cortex™ M33 CPU. This processor provides 1.5 dMIPS/MHz (96 dMIPS when operating at a maximum clock speed of 64 MHz) and is used for the application but also for implementing the upper layers of the Bluetooth® Low Energy protocol (Host). This CPU has a powerful cache controller with four-way associativity, 8 bytes cache line size, and 8 kB of RAM. The CPU executes code from embedded Flash through the cache controller, but it can also bypass it and execute code from embedded Flash directly or RAM.

Configurable MAC (CMAC). This is based on the Arm Cortex M0+ CPU and hardware accelerators implementing all timing critical tasks of the Bluetooth® Low Energy Controller. Moreover, it is capable of supporting proprietary, ISM band protocols (for example, ANT+).

ROM. This is a 288 kB ROM containing the booter code, the Bluetooth® Low Energy stack code, and various routines for implementing authentication of the eFlash image (using Elliptic Curves).

Data RAM. 96 kB Data RAM (DataRAM) which is shared among all masters of the system. It is used for storing code and data of CMAC and application data (Cortex M33). It comprises three RAM cells of 32 kB each, all with content retaining as well as complete power switch-off capability.

Embedded Flash Controller. This controller implements the interface to the 256-kB embedded Flash (eFlash) cell which contains the application image. It serves both Cortex CPUs for code execution as well as data. It implements all timing related tasks and read/write/erase/suspend/resume commands towards the eFlash cell.

QSPI Controller. A Quad SPI controller interfacing to external PSRAM/Flash devices used to extend the embedded RAM or to store data in non-volatile while executing code.

Cryptography Controller. It consists of an AES 256-bit block and a HASH controller implementing SHA-2. These hardware accelerators can be used by the Cortex M33 serving any application security requirements.

UART, UART2. Asynchronous serial interfaces. UART2 implements hardware flow control and is amended with ISO7816 functionality for connecting to a secure element while UART has no flow control. Both UARTs are equipped with a FIFO of 16 bytes depth and are supported by the general-purpose DMA engine. Both UART and UART2 can reach up to 3 Mbps.

SPI. This is the serial peripheral interface (SPI) with master/slave capability, and it has separate FIFOs for RX and TX of two 16-bit words each.

I2C. These are Master/Slave I2C interfaces used for sensors and/or host MCU communication. Each controller includes 32 locations deep FIFO (8-bits RX, 10-bits TX). They can both achieve up to 2.9 Mbps with a 32 MHz system clock.

Audio Controller. This block enables audio streaming by means of a Pulse Density Modulation (PDM), two Sample Rate Converters (SRC) that enable simultaneous inbound and outbound streams and a Pulse Code Modulation (PCM) interface. It can support up to two digital microphones or two digital loudspeakers using the PDM interface or connect an external CODEC at the PCM/I2S interface.

General Purpose (GP) ADC. This is a 10-bit analog to digital converter with eight external input channels and oversampling capability which increases the effective number of bits (ENOB) to 11.

Application (SD) ADC. This is a 15-bit analog to digital converter with eight external input channels and a sampling rate of 1 Ksps up to 2 Msps. It comes with a programmable gain amplifier and supports two modes of operation, namely, the audio mode and the sensor mode.

Radio Transceiver. This block implements the digital and analog PHY of the Bluetooth® Low Energy protocol at 2.4 GHz.

General Purpose Timers. Four general purpose timers of 24-bit width each are available for the user. They provide a number of features like PWM generation, two capture channels that save a snapshot of the timer, up/down counting with free-running mode, selectable clock source, and one-shot pulse generation with configurable width.

Real Time Clock. This is a hardware controller that supports the complete time of day clock: 12/24 hours, minutes, seconds, milliseconds, and hundredths of milliseconds. It comprises a configurable alarm function and can be programmed to generate an interrupt on any event like a rollover of month, day, hour, minute, second, or hundredths of milliseconds.

Watchdog Timers. The system comprises two watchdog timers, 13-bit wide each. One for CMAC software monitoring (CMAC Wdog) and another for the System CPU (System Wdog). The System watchdog is constantly counting down and automatically starts right after POWERUP, it is powered by the always-on domain and generates an NMI and a hardware reset when 0 and -16 are reached respectively. Its maximum counting time is 84 seconds or 3 minutes depending on the clock used (RC32K or RCX). The CMAC Watchdog resides in the Radio power domain and generates an interrupt to the CMAC CPU when 0 is reached. It also generates a hardware reset if -16 is reached. Both watchdogs are automatically frozen when either of the two CPUs is in debug mode.

Wake-up Controllers. There are two different wake-up controllers: one for capturing external events that can be used as a wake-up trigger from extended sleep modes at any GPIO and another one that can capture events on two specific GPIOs only to wake up from Hibernation. While both support programmable polarity, only the first one comprises a single debouncing structure for generating a wake-up interrupt upon a button press while the latter guarantees minimum power consumption.

DMA Engine. This is a general-purpose DMA engine with six channels that can be multiplexed to support data transfers between memory resources but also between memory and peripherals in single or burst modes (where applicable). It is also used when secure features are enabled to perform key transfers from protected eFlash memory spaces to registers without the CPU having access.

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.

4.2 Digital power domains

The DA14592 supports a number of digital power domains that can be turned ON and OFF independently from one another.

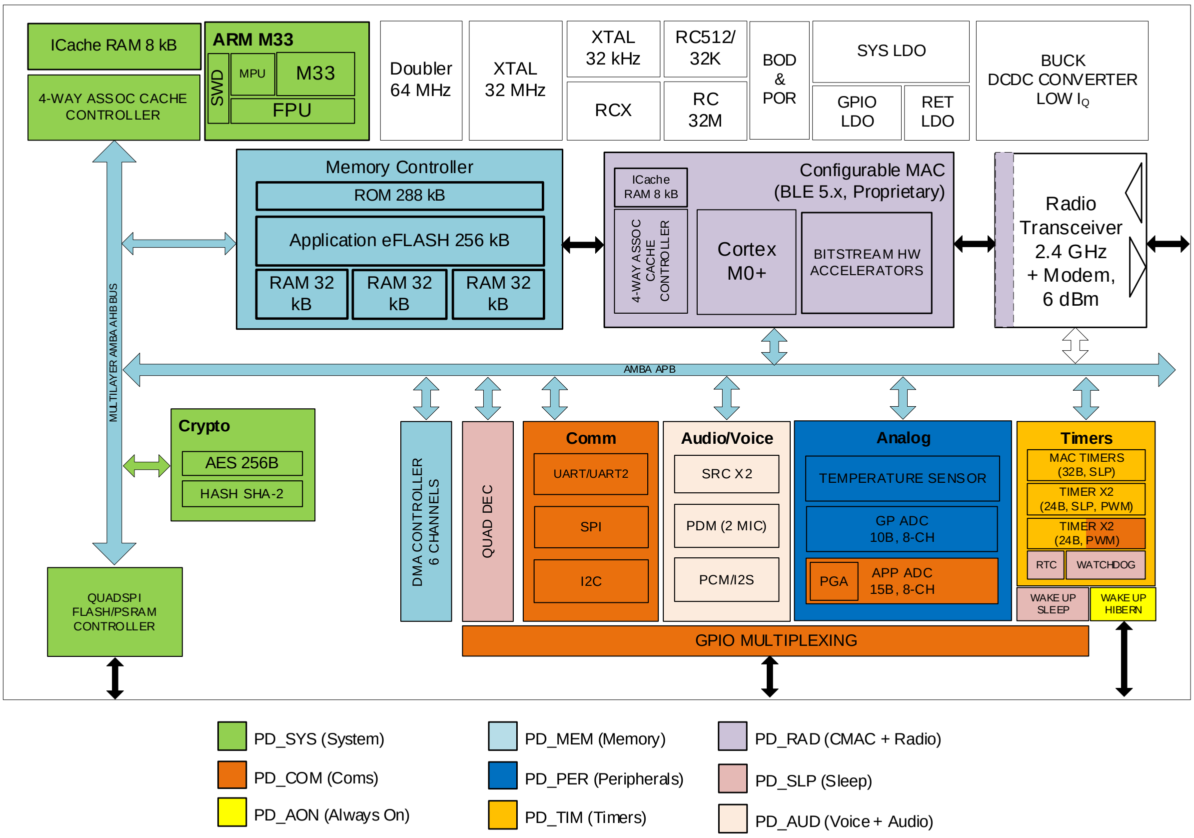

Figure 4. Digital power domains and blocks mapping

Table 61 shows the description and domain names used throughout this document.

Table 61: Power Domains Description

Domain Name	Description
PD_SYS	System Power Domain. It comprises the Arm M33 CPU, the QSPI controller, and the Crypto block.
PD_MEM	Memory Power Domain. It comprises the RAM, ROM, and eFlash controllers, the DCDC digital FSM, and the DMA engine. It also contains the complete bus fabric.
PD_RAD	Radio Power Domain. It comprises the CMAC and the digital PHY of the Radio.
PD_COM	Communications Power Domain. It comprises all serial interfaces, SD ADC, and one of the Timers. It also contains the GPIO multiplexing.
PD_PER	Peripherals Power Domain. It comprises the Temperature Sensor and the GP ADC.
PD_SLP	Sleep Power Domain. It comprises the RTC, the Watchdog, and the respective Wake-up controller.
PD_AON	Always-On Power Domain. It comprises the wake-up from the hibernation controller.
PD_TIM	Timer Power Domain. It comprises the MAC timer, three general-purpose Timers, and the XTAL32M digital state machine.
PD_AUD	Audio power domain. It comprises the PCM, PDM, and SCRs blocks.

4.3 Hardware FSM (POWERUP, WAKEUP, GOTO SLEEP)

4.3.1 Hardware FSM

Figure 5 shows the hardware FSM responsible for the POWERUP, WAKEUP, and GOTO SLEEP processes of the system.

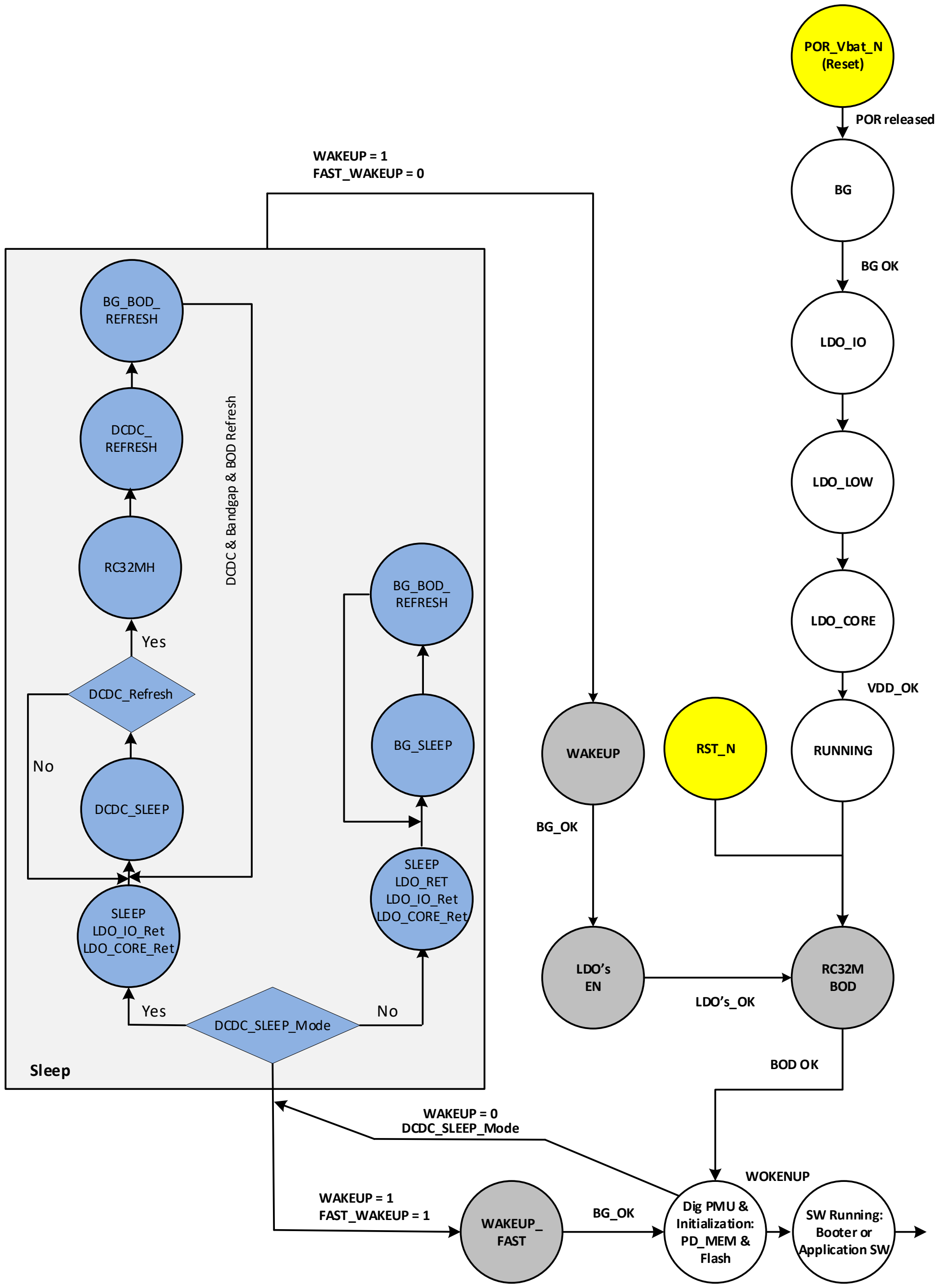

Figure 5. POWERUP, WAKEUP, GOTO SLEEP hardware FSM

The WAKEUP signal indicates that the hardware FSM has finished, and the digital power domains are initialized.

4.3.2 Power-up

After POR_Vbat is released, the FSM enters the state BG where the following actions are taken:

- The COLD_BOOT bit is asserted indicating that the system recovers from a POR (previous state has to be POR_Vbat_NOK).
- Bandgap is enabled.

This state does not exceed 10 µs. When the BG reference voltage is settled to 0.75 V, the BG OK signal is generated, and the FSM continues with the next state LDO_LOW LDO_IO:

- LDO_IO is enabled depending on the programmed register bit in POWER_CTRL_REG.
- LDO_LOW is enabled depending on the programmed register bit in the POWER_CTRL_REG register.

Those LDOs are enabled one after the other. The LDO_LOW is only enabled when the VDDIO is settled and generates an LDO_OK signal.

- The LDO_CORE is enabled after the aforementioned LDOs are settled and their outputs are stable. The LDO_CORE sets VDD to 0.9 V.

The RCLP clock then switches to 512 kHz and the RC32M and BOD comparators are enabled. When the BOD comparator levels are OK, the WOKENUP signal is asserted (which can be mapped on a GPIO or monitored at a register bit). This is the time when the FSM has reached the end.

The WOKENUP signal triggers the digital FSM by starting the Digital Power Control. First, there is a timeout counter to be able to configure the test mode. Next, the required power domains are powered up. After this, the software starts running.

The timing diagram in Figure 6 summarizes the aforementioned description.

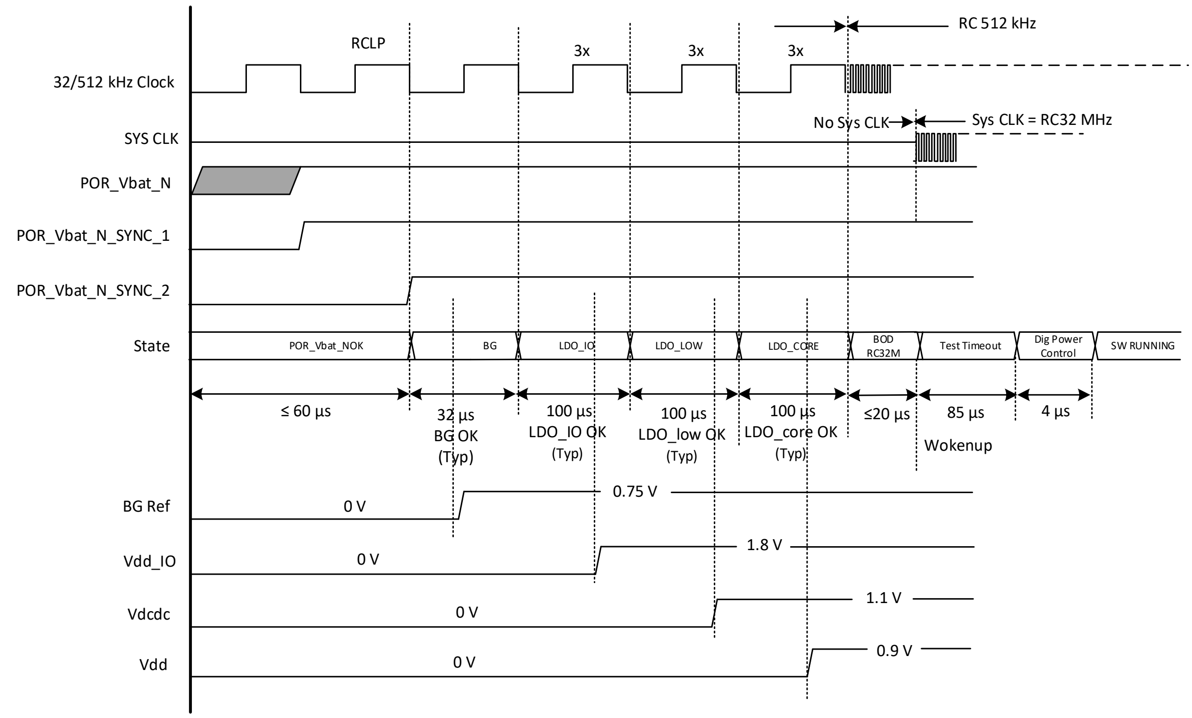

Figure 6. POWERUP timing diagram

The total time required for the cold boot powerup is about 0.5 ms. This is a typical number and depends largely on the capacitor values at the supply rails on the PCB.

4.3.3 Wake-up options

The system supports two different wake-up modes. You can configure which one to select depending on the application requirements and constraints. Table 62 summarizes the wake-up mode characteristics.

Table 62: Wake-Up Modes Overview

Mode	Latency	Description	Constraints
Normal wake-up	~200 µs	All LDOs of the Power Management are powered sequentially. Software starts running after all LDO OK signals are evaluated by the hardware FSM	None
Fast wake-up	10 µs	DCDC is ON. Wake-up detection is performed using the RC512kHz clock and FSM is running with the RC32MHz clock. Software starts running within 10 µs from the wake-up trigger assuming all voltage outputs are stable.	Low load (< 0.5 mA) should be applied for the first 100 µs. Sleep core voltage (VDD) must be 0.9 V.

4.3.3.1 Normal wake-up

The slow WAKEUP is following the states of the cold boot. It, however, deasserts the COLD_BOOT flag. Figure 7 shows the timing diagram of the slow WAKEUP.

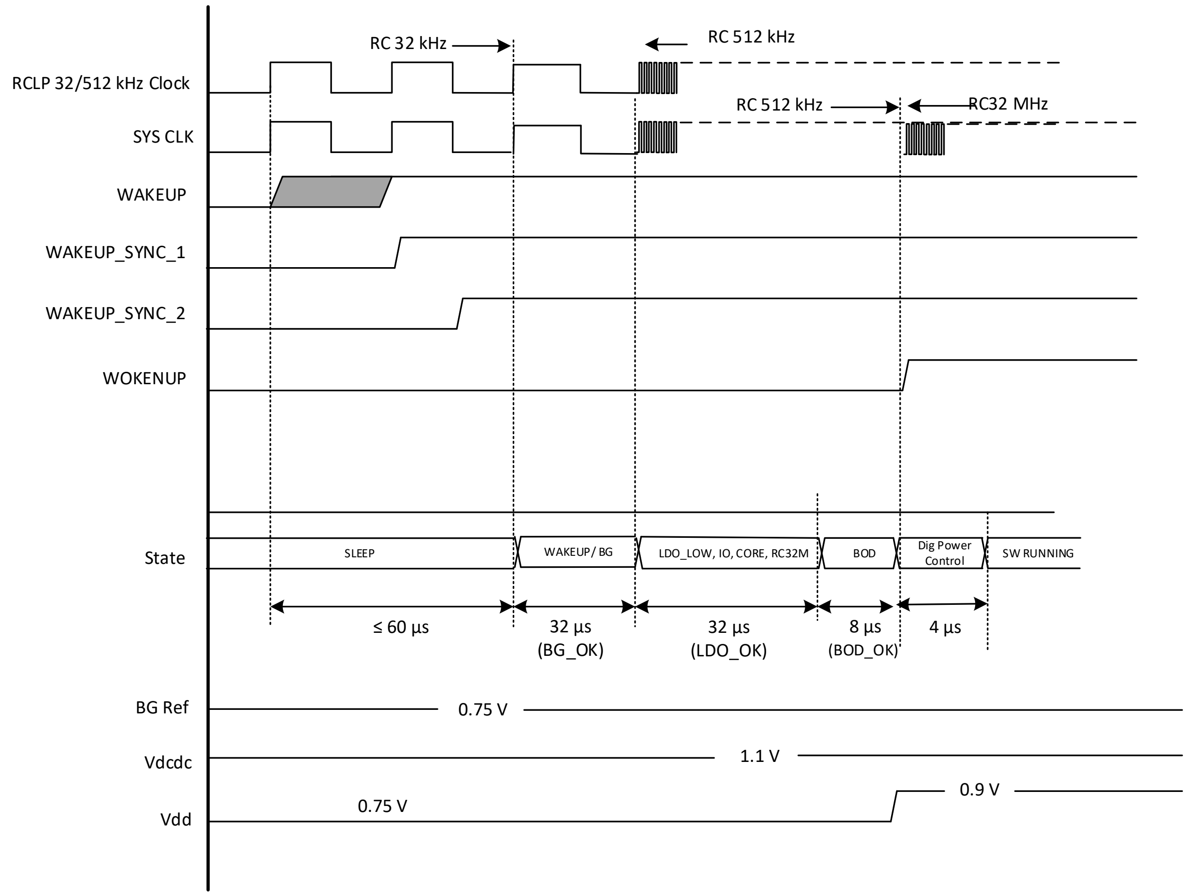

Figure 7. Normal wake-up timing

The maximum latency of the normal wake-up is in the range of 200 µs.

4.3.3.2 Fast wake-up

Enabling fast wake-up mode reduces the wake-up time. In this mode, the DCDC is running during sleep and the VDD is set at 0.9 V. The RC32/512k clock is active running at 512 kHz and clocking the wake-up logic. The major difference on the latency is achieved if the Bandgap is kept activated during sleep, thus it requires no settling time when waking up. If this is the case, the Analog FSM state WAKEUP_FAST only takes 2 µs, else 12 µs, until the Bandgap is settled and ready.

Figure 8 shows the timing diagram.

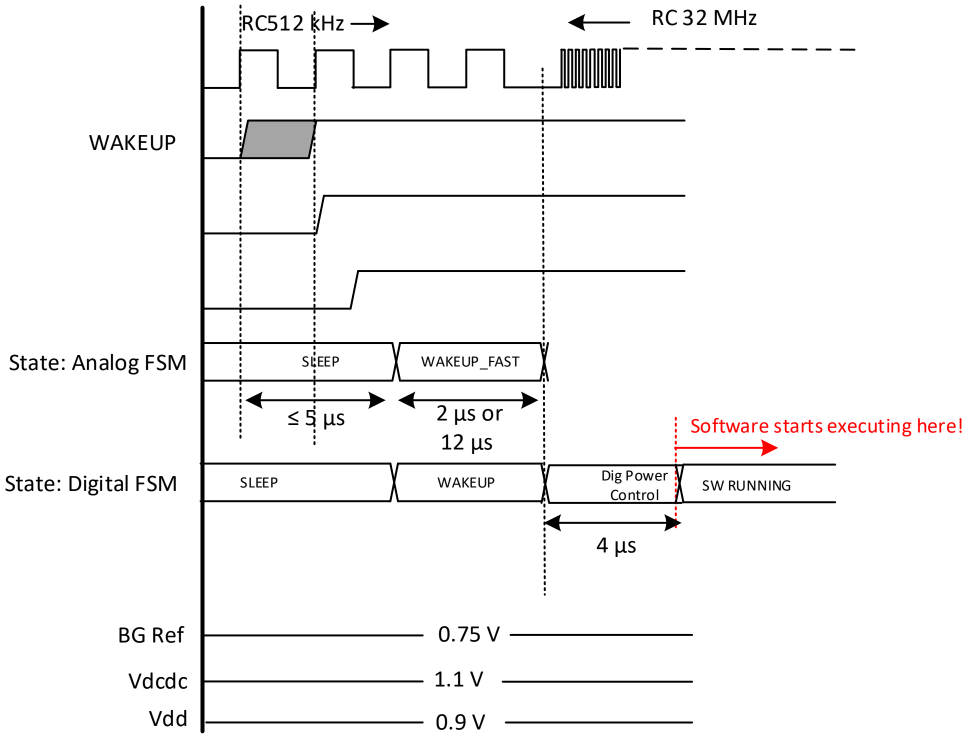

Figure 8. Fast wake-up timing

Latency of this mode does not exceed 10 µs considering Bandgap is enabled during sleep.

To enable the Fast Wake-up mode, the PMU_SLEEP_REG[FAST_WAKEUP] bit should be set followed by a PDC entry with the peripheral trigger ID set to Fast Wakeup (PID = 13).

Note
When fast wake-up is required, VDD should be set to 1.2 V (in active mode) before entering sleep mode. As soon as the device wakes up, VDD can be switched back to 0.9 V.

4.3.4 Go-to-sleep

While sleeping, the VDCDC rail can be supplied either by the DCDC or by the LDO_LOW_RET. You can select between two go-to-sleep sequences:

- DCDC Go-To-Sleep where VDCDC is supplied by the DCDC during sleep.
- BG Go-To-Sleep where VDCDC is supplied by the LDO_LOW_RET during sleep.

Figure 9 shows the timing sequence of BG Go-To-Sleep. When the system enters this sleep state two operations should be configured by software to happen periodically:

- Refresh the reference of the LDO_CORE_RET which provides the sleep voltage by refreshing the bandgap in a programmable interval between 2 ms and 8 s. The refresh interval can be set by PMU_SLEEP_REG[BG_REFRESH_INTERVAL] bit field (1 LSB = 64 x 32 kHz clock cycles).
- Check on the voltage levels by allowing the BOD to sense each voltage rail while the bandgap is enabled.

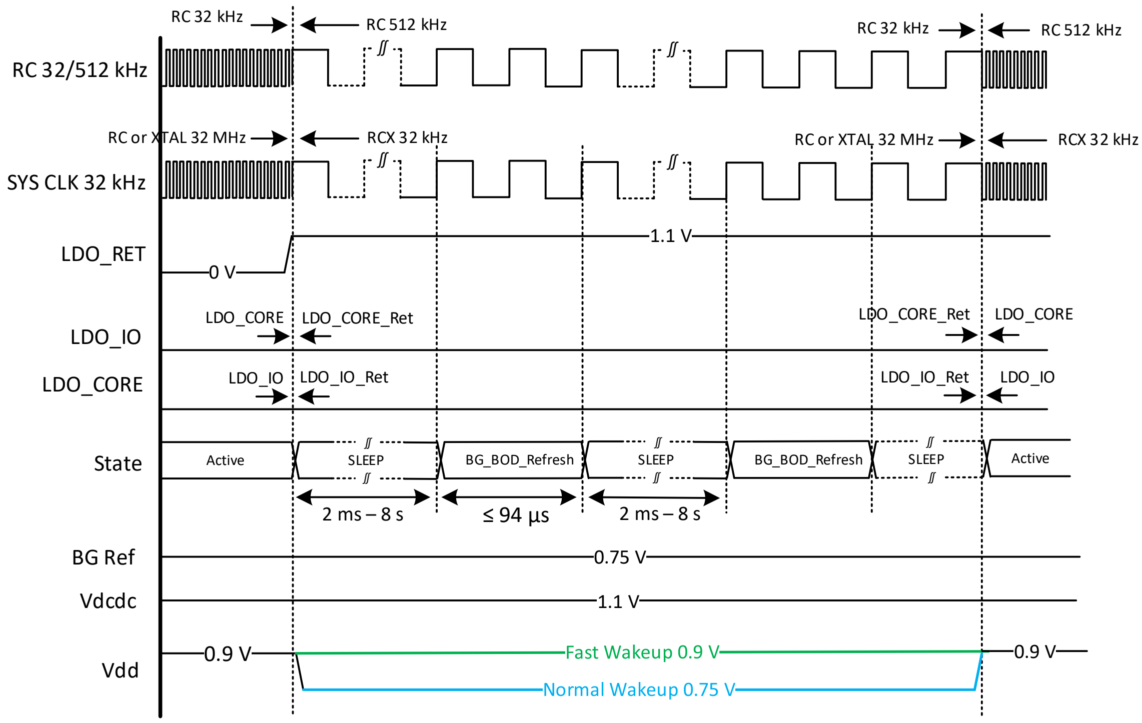

Figure 9. BG Go-To-Sleep

Figure 8 shows the timing sequence of DCDC Go-To-Sleep.

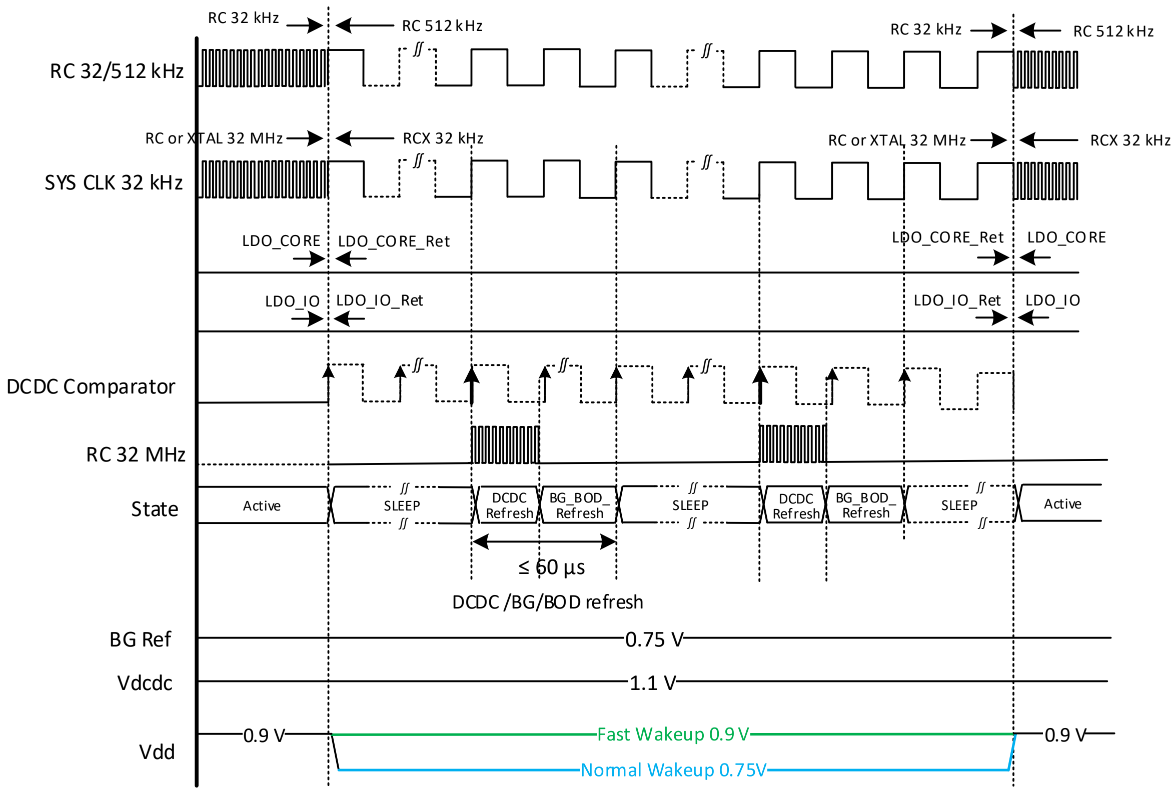

Figure 10. DCDC Go-to-Sleep

DCDC is running during sleep as well. A voltage comparator is used to activate the DCDC for a refresh when the voltage at the VDCDC rail drops below a threshold. During the DCDC refresh, the Bandgap is refreshed, and the BOD is activated to sense the voltage rails.

4.4 Power modes and rails

There are four main power modes in the device. These modes are not hardwired into any state machine, they are there to ease the SDK development in the sense of the offered capabilities for customers.

- **Hibernation mode**. This is supposed to be the "shipping mode". No RAM is retained, no clocks are running (so no RTC), all domains are off, except for PD_AON, and the system can only be woken up by POR or a hibernation wake-up trigger that can only happen on dedicated GPIOs.
- **Deep Sleep mode.** This is supposed to be the "shipping mode + RTC". No RAM is retained, RCX is running (so RTC is on), all domains, except for PD_AON and PD_SLP, are off, and the system can only be woken up by POR, RTC alarm, watchdog reset, or a GPIO trigger (reset on wake-up).
- **Extended Sleep mode.** This is supposed to be the "Bluetooth connection sleep mode". There can be programmable RAM retained, RCX is running (so RTC can be on), all domains are off, except for PD_AON PD_SLP and PD_TIM, the system can only be woken up by POR, RTC alarm (if RCX is used), MAC timer, other Timer, or a GPIO trigger. There is a sleep variance based on the VDD voltage level while sleeping as shown in Table 63.
- **Active.** The system is up and running with a number of power domains enabled depending on the use case requirements. There are two variances in this mode: the Low-Power (LP) and the High-Performance (HP) mainly related to the Radio TX output performance as shown in Table 63.

Table 63: Power Modes, Rails Voltage Levels

Power Mode	VDCDC	VDD	System Clock Frequency	eFlash Mode	Radio Mode	Comment
Active	1.1	0.9 V	32 MHz	Read	OFF LP	Default startup/Low Power mode.
	1.4 V	0.9 V 1.2 V	32 MHz 64 MHz	Read/Program/ Erase	OFF HP	High-Performance mode
Ext. Sleep	1.1 V	0.75 V	RCX	OFF	OFF	Normal Wake-up
		0.9 V				Fast Wake-up
Deep Sleep	1.1 V	0.75 V	RCX	OFF	OFF	Wake-up in the reset state
Hibernation	OFF	OFF	None	OFF	OFF	Wake-up in the reset state only from designated IOs

For a better understanding of the V_{DD} and V_{DCDC} rails, see the Power section.

4.5 Embedded flash layout

The embedded Flash (eFlash) is used for executing code through the cache controllers of the Cortex M33 or the Cortex M0+ but also without them. Moreover, it is also used for storing chip-related trimming values security keys, and product headers (primary and backup).

Table 64 shows the eFlash that consists of several different sectors.

Table 64: Embedded Flash Layout

Sector	Description	Address Range	Remark
IP	Manufacturing trim values	0x00040000-0x000407FF	This is a separate sector, called "Information Page" which holds the chip's trim values. This sector is write- and erase-protected.
1	User Configuration Script and security key pointers	0x00000000-0x000007FF	This sector contains the customer's configuration and the pointers to keys for application cryptography or secure boot.
2	User Application Expanded Encryption Keys	0x00000800-0x00000BFF	This sector contains the actual security keys.
	Signature Keys	0x00000C00-0x00000FFF	
3	Product Header	0x00001000-0x00001800	This sector contains the product header.
4	Product Header Backup copy	0x00001800-0x00002000	This sector contains the product header backup used in case of OTA failing.
5-12	Customer Application Area	0x00002000-0x00006000	These eight sectors can be used for various software modules such as a secondary bootloader, various binary libraries, and so on. It can be observed as the start of the firmware image.
Var.	Firmware Image X	0x00006000-	The actual application firmware image with variable length, always with a start address sector aligned.

Note 1 Every sector is 2 kB long.

4.6 Configuration script

The Configuration Script (CS) is used for programming registers with values that are defined during production testing, storing a trim value for the application software, or configuring a particular system feature during boot time. It comprises ten commands. The CS begins with the Start of CS command followed by a set of commands that refer to certain things like register configurations, trim values, and so on. The CS is ended with the Stop of CS command. Empty entries between commands are not allowed. If an empty entry is found, then all the following commands are discarded.

The booter executes the script to prepare and initialize the system before the CPU starts running application code from eFlash.

Table 65 explains the format of the commands in the script.

Table 65: Configuration Script Commands

#	Command Type	Description
1	Start Command	One 32-bit word containing 0xA5A5A5A5 to signal a valid CS is in place.
2	Register Configuration	• One 32-bit word containing an address of an existing register. • One 32-bit word containing the data value of the register. These are always in pairs with the address sitting in even memory addresses.
3	Trim Value	• One 32-bit word which is equal to 0x**9**000YYXX indicates that the next word is a value stored during production testing. More specifically: ○ **9**: indicates that the following word(s) are not to be stored in registers but are used by the SDK software. ○ YY: indicates that YY amount of words follow.

#	Command Type	Description
		○ XX: is an increasing value and can be used for indexing by the software application. If YY > 1 then this number is not increased for the words that belong to the same value. ● One or more 32-bit words which represent the value.
4	Booter Value	One 32-bit word which is equal to 0x6XXXXXXX indicating this is a value pointing to the Flash product header in flash at address 0xXXXXXXX
5	Development Mode Disable	One 32-bit word which is equal to 0x70000000, disabling the development mode. Development Mode is enabled by default at the initialization phase of the booter.
6	UART STX Timeout	One 32-bit word which is equal to 0x8YXXXXXX. ● The XXXXXX is used to program the selected STX timeout in multiples of 100 µs. So, 0x80000028 is 40x100 µs = 4 ms. ● The Y is used to select a baud rate setting other than the default (115K2) ○ 0x1: 4K8 ○ 0x2: 9K6 ○ 0x3: 14K4 ○ 0x4: 19K2 ○ 0x5: 28K8 ○ 0x6: 38K4 ○ 0x7: 57K6 ○ 0x8: 115K2 ○ 0x9: 230K4 ○ 0xA: 500K ○ 0xB: 1M ○ 0xC- 0xF: reserved (defaults to 115K2).
7	XTAL32M Settling Value	● One 32-bit word which is equal to **0xAYXXXXXX**. The XXXXXX is used to program the selected XTAL32M settling time in multiples of 100 µs. So, 0xA0000028 is 40x100 µs = 4 ms. The Y is used to instruct the booter to overrule the XTAL32M settling and: ○ Y = 1: Ignore XTAL32M settling time and continue. ○ Y = 2: Ignore XTAL32M settling time and continue booting using RC32M.
8	Stop Command	One 32-bit word containing 0x00000000 designating that execution should be terminated since the configuration script has reached the end and is locked for further entries with one exception, being the minimum version keyword for the rollback prevention feature.
9	Minimum Firmware Version	One 32-bit word containing the **0xBXXXXXXX,** where XXXXXXX is used to store the minimum firmware version that the booter should accept.
10	Set-Once Bits Configuration	One 32-bit word containing the **0xC000XXXX** where the XXXX is used to program the set-once (sticky) bits for hardware configuration and security features. ● 0: PROT_CONFIG_SCRIPT ● 1: PROT_APP_KEY ● 2: PROT_VALID_KEY ● 3: PROT_USER_APP_CODE ● 4: Enable DCDC at boot ● 5: FORCE_M33_DEBUGGER_OFF ● 6: FORCE_CMAC_DEBUGGER_OFF ● 7: SECURE_BOOT ● Rest: Reserved.

4.7 Product header layout

The DA14592 is capable of booting firmware images either from eFlash or external QSPI Flash devices. Depending on the user's preference, the Product Header should be placed accordingly.

4.7.1 eFlash product header layout

The eFlash product header layout is presented in Figure 11. The optional fields are marked with orange color while mandatory fields are marked with blue color.

The user needs to make sure that:

- A firmware image size must not exceed the selected CACHE_EFLASH_REG[EFLASH_REGION_SIZE]. The default value is set to 128 kB but it can be overruled using the Configuration Script.
- All firmware images must start at a CACHE_EFLASH_REG[EFLASH_REGION_SIZE] aligned address.
- A padding of 0xFF is used between the end of the Image header and the start of the Image itself. The image is always 1 kB aligned.

In case the image header contains a Device Administration section, the signature will be calculated over the Signed firmware version field, the Device Administration section, the padding, and the image. If the Device Administration section is not included, the signature will be calculated over the Signed firmware version field, the padding, and image.

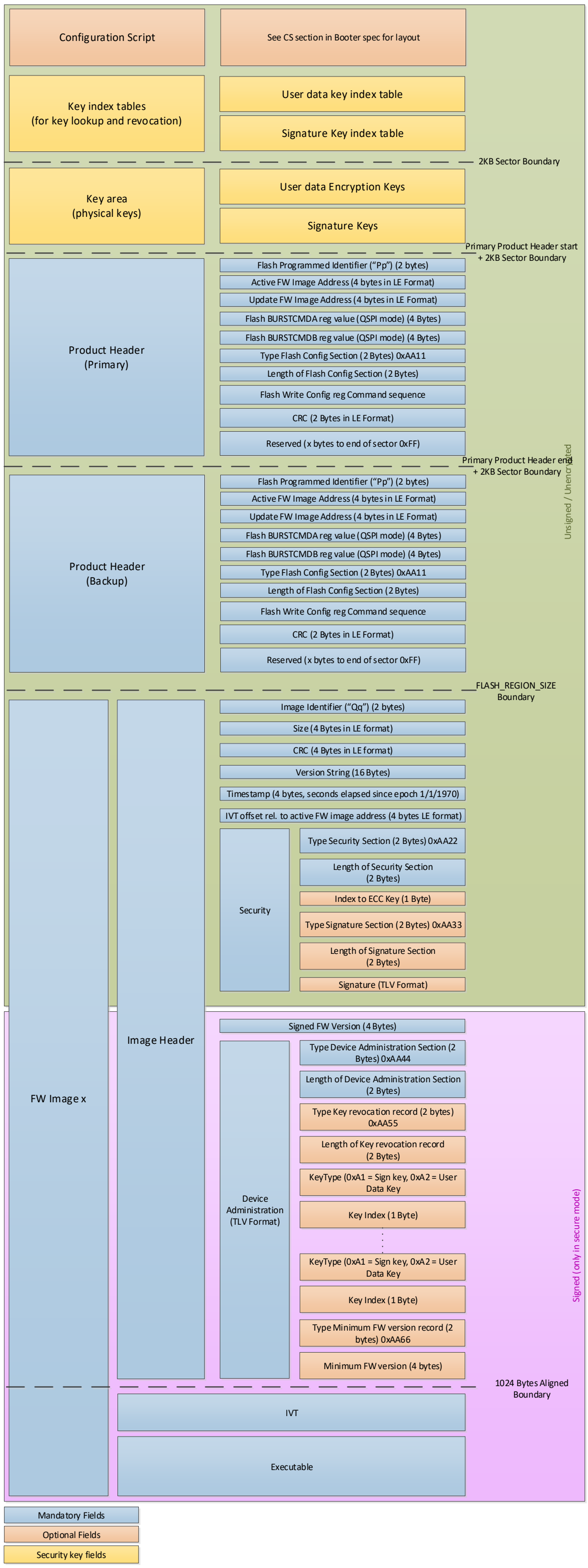

Figure 11. eFlash product header layout

4.7.2 QSPI Flash product header layout

Figure 12 shows the QSPI Flash product header layout. The optional fields are marked with orange color while mandatory fields are marked with blue color.

The user needs to make sure that:

- A firmware image size must not exceed the selected CACHE_FLASH_REG[FLASH_REGION_SIZE]. The default value is set to 0.5 MB but can be overruled using the Configuration Script.
- All firmware images must start at a CACHE_FLASH_REG[FLASH_REGION_SIZE] aligned address.
- The QSPI Flash size used must be at least 0.5 MB (4 Mbit). The minimum FLASH_REGION_SIZE allowed is 0.25 MB and at least two regions are needed (product header and one firmware image).
- A padding of 0xFF is used between the end of the Image header and the start of the Image itself. The image is always 1 kB aligned.

In case the image header contains a Device Administration section, the signature will be calculated over the Signed firmware version field, the Device Administration section, the padding, and the image. If the Device Administration section is not included, the signature will be calculated over the Signed firmware version field, the padding, and the image.

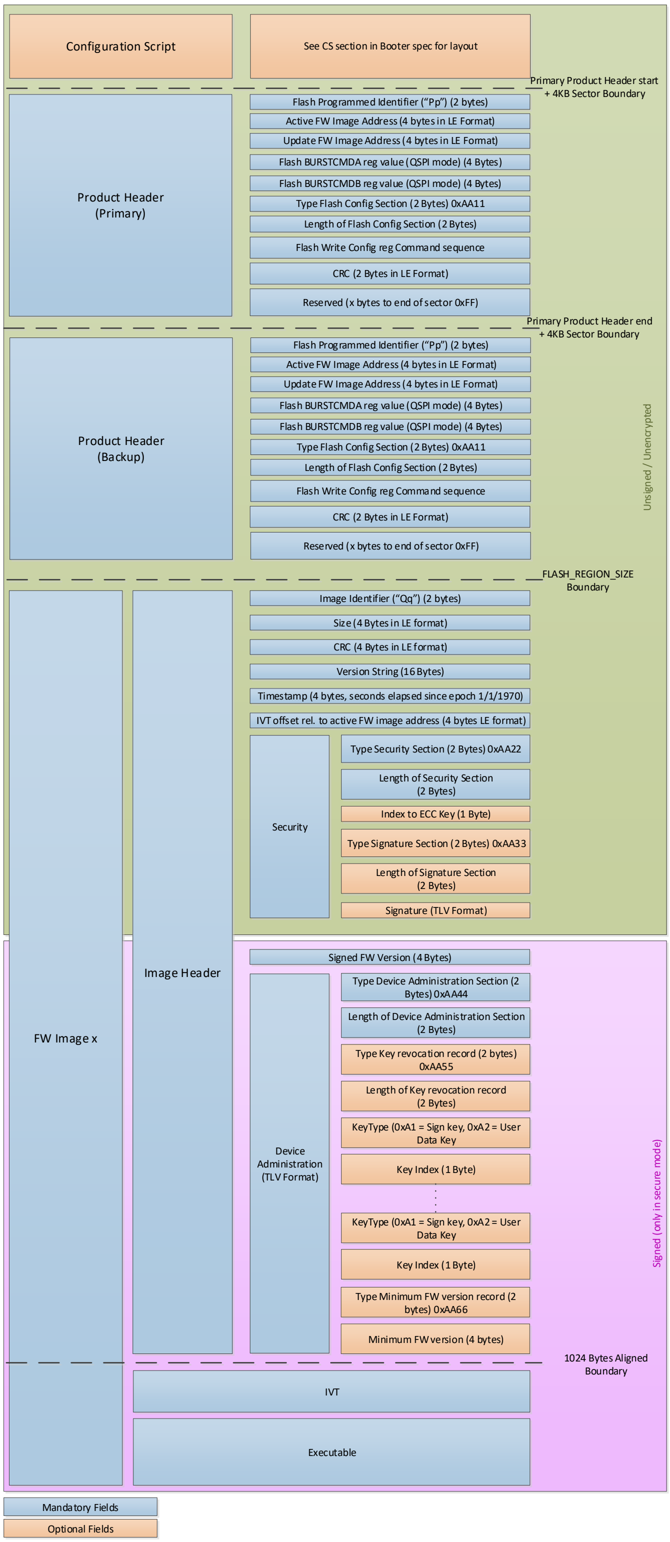

Figure 12. QSPI Flash product header layout

4.8 Booting

The booter is always executed when a POR, a Hardware Reset, or the RESET_ON_WAKEUP feature is configured. Different booting flavors are supported:

- Boot from eFlash cached (secure/non-secure), Configuration Script in eFlash
- Boot from QSPI cached (non-secure), Configuration Script in QSPI

- Boot from QSPI cached (secure), Configuration Script in eFlash
- Boot from UART (non-secure), no eFlash/QSPI Flash image available.

The booter also detects available software updates and applies them according to the eFlash/QSPI header.

The Boot flow is divided into five separate phases:

1. Initialization.
2. Run configuration script.
3. Retrieve the application code.
4. Device administration.
5. Load image.

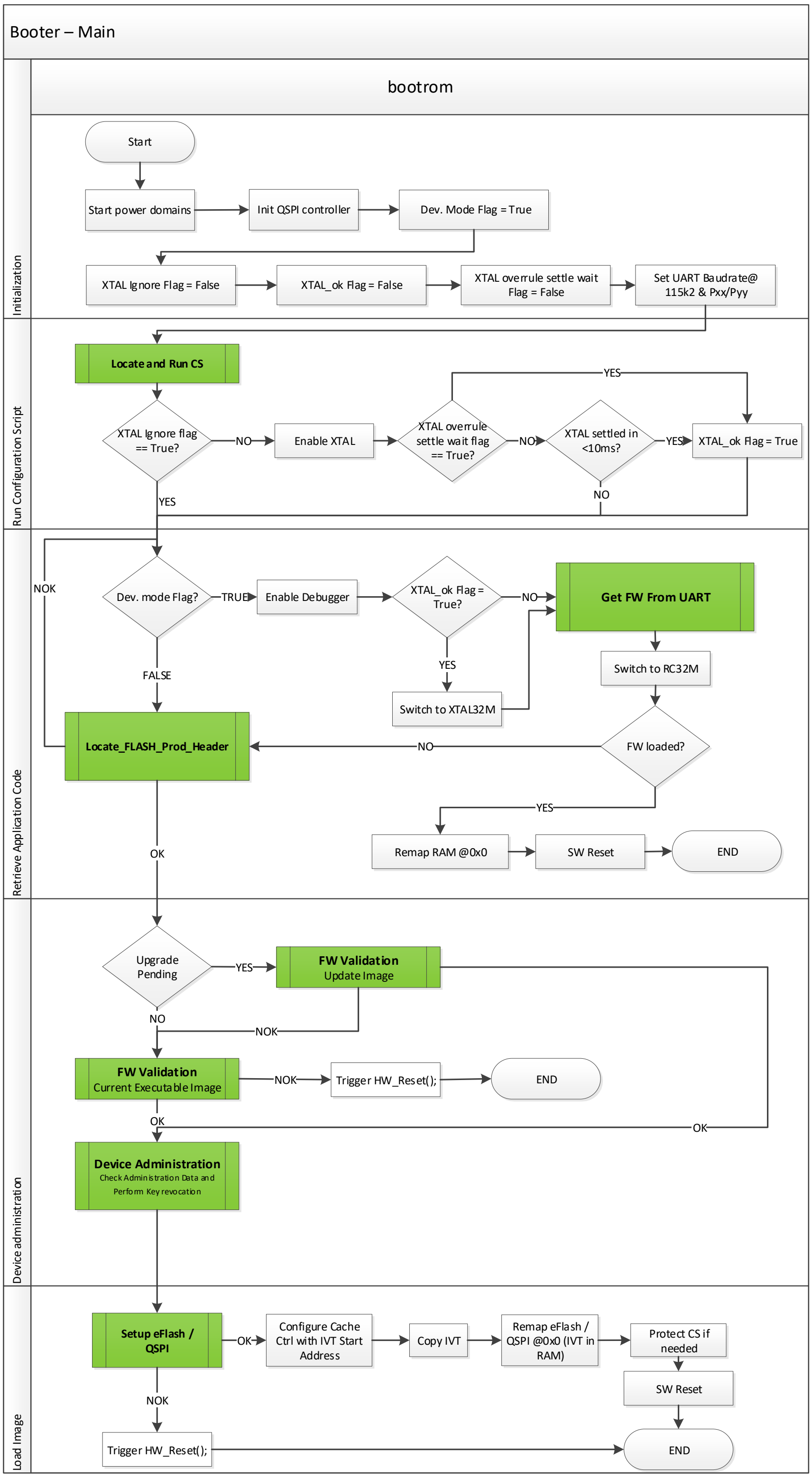

Figure 13. BootROM flowchart

4.8.1 Initialization phase

The initialization phase takes care of enabling the Power domains. Then it initializes the QSPI interface and Clocks. Following this, it enables Development mode. This is done so that it can be disabled by the CS in the next phase if desired. Development mode can be used for having the debugger on and updating the device firmware using UART.

Next, it sets all XTAL-related flags to "False" and finally, it initializes the UART (but not enables it yet) with the default settings, which can still be overwritten by the configuration script.

4.8.2 Configuration script phase

This phase tries to locate and execute the Configuration Script (CS) from eFlash or QSPI Flash. The booter tries to locate the CS in the eFlash customer area at address 0. If no CS is found, it looks at the start of QSPI Flash after performing the generic reset QSPI function. This function performs a generic wake-up and reset that should be able to reset and wake up the majority of QSPI Flash devices.

After the CS is processed, the "XTAL Ignore" flag is checked. If it is not set, the XTAL32M is enabled and the booter checks if the settle and trim-ready bits are set within 10 ms. If not, the XTAL_ok Flag is kept at False, if it is settled correctly it is set to True. This flag later is used to detect if it is safe to switch to XTAL32M, or to continue using the default sys clock (RC32M).

The Booter uses a default 10 ms timeout between enabling the XTAL32M and checking the settling bits. This time can be overruled by the CS. Also, checking the settle bits can be overruled completely by the same keyword in the CS and the booter just assumes the XTAL is OK and settled.

4.8.3 Retrieve application code phase

During this phase, the booter scans to check if the development mode flag is disabled or not. If it is still in development mode, then it enables the Debug interface and tries to boot from UART. If booting from UART is successful, then it issues a software reset after having remapped address zero to RAM. If not, then it tries to locate a valid Flash header (first in eFlash then in QSPI Flash), trying to boot from Flash. If not in development mode, the booter continues to the next phase after having identified a valid product header in any of the embedded or QSPI Flashes.

4.8.4 Device administration phase

The device administration phase is used to check for pending updates and validate the Flash images. It also processes corresponding image headers and revoke keys, if needed. The booter checks if the "Active FW Image address" and the "Upgrade Image address" fields of the Flash product header are the same. If not, an upgrade image is available. The firmware validation is executed if the secure boot bit has already been set by the Configuration Script (CS).

Validation is done by identifying the public key, checking if the key is already revoked, and if not, proceeding with invoking the Ed25519 verification algorithm. This happens with the doubler being enabled and the system CPU running at 64 MHz.

NOTE
The booter does not manipulate the product headers in case of a firmware update. This should be taken care of by the application level which is also responsible for updating both product headers (primary and backup).

4.8.5 Load image phase

In this final phase, the actual firmware image is loaded. This is done by setting up the eFlash/QSPI and cache controllers and executing the QSPI Loader if the application is running from QSPI, which is located in the Product header of Flash. The cache controller is then configured to point to the interrupt vector table (copied in RAM) and address zero is remapped to eFlash/QSPI Flash. Any error condition during this or previous phases results in a hardware reset.

4.9 Memory map

Table 66 shows the mapping of the system's internal resources. Note that *_C defines a Controller (registers) while *_M defines Memory (RAM) space. All resources are 32-bit aligned.

Table 66: Memory Map

	Start Address	End Address	Size (kB)	PD	AMBA	Comments
Remapped Devices	0	800000	8192	PD_SYS	AHB	Remap IVT into SYSRAM
SYSRAM (code)	800000	818000	96	PD_MEM	AHB	Remapped from 0x20000000
CACHE_SYS_RAM	818000	81A000	8	PD_SYS	AHB	Cortex-M33 C-Bus access
Reserved	81A000	900000	920			
ROM	900000	948000	288	PD_MEM	AHB	Physical address. Remappable at 0x0
Reserved	948000	A00000	736			
eFlash(code)	A00000	A40800	258	PD_MEM	AHB	Physical address. Remappable at 0x0.
Reserved	A40800	16000000	349950			
QSPIC_M	16000000	18000000	32768	PD_SYS	AHB	Physical address. 8 Mbytes Remappable at 0x0
QSPIC_C	18000000	1A000000	32768	PD_SYS	AHB	
Reserved	1A000000	20000000	98304			
SYSRAM (data)	20000000	20018000	96	PD_MEM	AHB	Physical address of SYSRAM. Remappable at 0x0
CACHE_SYS_RAM	20018000	2001A000	8	PD_SYS	AHB	Cortex-M33 S-Bus access
Reserved	2001A000	30020000	262168			
AHB_DMA_B	30020000	30020400	1	PD_MEM	AHB	
Reserved	30020400	30040000	127			
AES_HASH_C	30040000	30050000	64	PD_SYS	AHB	
Reserved	30050000	31000000	16064			
eFlash(data)	31000000	31040800	258	PD_MEM	AHB	No read buffering is performed in this address range
Reserved	31040800	32000000	16126			
QSPIC_M	32000000	34000000	32768	PD_SYS	AHB	Same physical address as 0x16000000
QSPIC_C	34000000	36000000	32768	PD_SYS	AHB	Same physical address as 0x18000000
Reserved	36000000	36008000	32			
CACHE_SYS_C	36008000	36008400	1	PD_SYS	AHB	
Reserved	36008400	40000000	163807			
CMAC	40000000	40003000	12	PD_RAD	AHB	
RFCU	40003000	40003400	1	PD_RAD	AHB	
DEMOD	40003400	40003800	1	PD_RAD	AHB	
ADPLL	40003800	40010000	50	PD_RAD	AHB	

	Start Address	End Address	Size (kB)	PD	AMBA	Comments
Reserved	40010000	40108000	992			
CACHE_CMAC _RAM	40108000	4010A000	8	PD_RAD	AHB	Accessible by M33 when CMAC cache is disable
Reserved	4010A000	40200000	984			
PATCH_C	40200000	40300000	1024	PD_RAD	AHB	
Reserved	40300000	50000000	259072			
CRG	50000000	50000100	0,25	PD_SLP	APB32	
WKUP	50000100	50000200	0,25	PD_SLP	APB32	
PDC	50000200	50000300	0,25	PD_SLP	APB32	
DCDC	50000300	50000400	0,25	PD_SLP	APB32	
RTC	50000400	50000500	0,25	PD_SLP	APB32	
QUADDEC	50000500	50000700	0,5	PD_SLP	APB32	
WDOG	50000700	50000800	0,25	PD_SLP	APB32	
Reserved	50000800	50010000	62			
CRG_XTAL	50010000	50010300	0,75	PD_TIM	APB32	
TIMER	50010300	50010400	0,25	PD_TIM	APB32	
TIMER2	50010400	50010500	0,25	PD_TIM	APB32	
TIMER3	50010500	50010600	0,25	PD_TIM	APB32	
CMAC_TIM	50010600	50010700	0,25	PD_TIM	APB32	
Reserved	50010700	50020000	62,25			
UART	50020000	50020100	0,25	PD_COM	APB32	
UAR2	50020100	50020200	0,25	PD_COM	APB32	
SPI	50020200	50020300	0,25	PD_COM	APB32	
I2C	50020300	50020400	0,25	PD_COM	APB32	
SDADC	50020400	50020500	0,25	PD_COM	APB32	
CRG_COM	50020500	50020600	0,25	PD_COM	APB32	
GPIOMUX	50020600	50020800	0,5	PD_COM	APB32	
Reserved	50020800	50020A00	0,5			
TIMER4	50020A00	50020B00	0,25	PD_COM	APB32	
Reserved	50020B00	50030000	61,25			
CRG_AUD	50030000	50030100	0,25	PD_AUD	APB32	
SRC1/PDM	50030100	50030200	0,25	PD_AUD	APB32	
SRC2/PDM	50030200	50030300	0,25	PD_AUD	APB32	
PCM	50030300	50030400	0,25	PD_AUD	APB32	
Reserved	50030400	50040900	65,25			
GPADC	50040900	50040B00	0,5	PD_PER	APB32	
ANAMISC	50040B00	50040C00	0,25	PD_PER	APB32	
CRG_PER	50040C00	50040E00	0,5	PD_PER	APB32	
Reserved	50040E00	50050200	61			

	Start Address	End Address	Size (kB)	PD	AMBA	Comments
VERSION	50050200	50050300	0,25	PD_SYS	APB32	
GPREG	50050300	50050500	0,5	PD_SYS	APB32	
CRG_SYS	50050500	50050600	0,25	PD_SYS	APB32	
RFMON	50050600	50050700	0,25	PD_SYS	APB32	
Reserved	50050700	50060000	62,25			
MEMCTRL_C	50060000	50060100	0,25	PD_MEM	APB32	
FCU	50060100	50060200	0,25	PD_MEM	APB32	
DMA_C	50060200	50060400	0,5	PD_MEM	APB32	
AMBA	50060400	50060500	0,25	PD_MEM	APB32	
Reserved	50060500	E0000000	235891 0,75			
ARM Internal Bus	E0000000	FFFFFFFF	524287 ,999	PD_SYS		

Note 1 Access to QSPI Flash memory from peripherals (for example, DMA) is done through 0x36000000 memory space.

4.10 Busy status register

The DA14592 has two processing units that might request access to one or more of the system's peripheral controllers. The application software can map certain resources to one of the processing units. But there are times, for example, when CMAC needs to use the General Purpose ADC to read the die temperature and trigger a radio calibration, while the same ADC is required by the application software running a State of Charge algorithm by checking the Battery Voltage on a regular basis.

To avoid race conditions and provide both processing units with a robust way of identifying who the owner of the peripheral is, a hardware mutex is implemented. This is a 32-bit register (BUSY_STAT_REG) that can be set/reset by using write-only registers (BUSY_SET_REG) and (BUSY_RESET_REG). This prevents race conditions because two processing units are writing at the same time. Two bits are reserved per resource so that the value represents the owner of the resource:

- 0x0: resource is available for use
- 0x1: reserved
- 0x2: resource is busy, controlled by the Arm M33
- 0x3: resource is busy, controlled by the CMAC.

Such a mutex register, which decides resources that require sharing, can be defined by application software. Table 67 shows a possible configuration of such a register.

Table 67: Busy Status Register Indicative Assignment

31-28	26-27	24-25	22-23	20-21	18-19	16-17	14-15	12-13	10-11	8-9	6-7	4-5	2-3	0-1	Bit
RESERVED	Timer4	Timer3	Timer2	Timer	PDM	PCM	SRC2	SRC	SDADC	GPADC	I2C	SPI	UART2	UART	

Notice that the register is in the PD_MEM that is automatically activated if one of the three masters is alive. However, if the system is in sleep, then this register is not retained.

4.11 Remapping

Table 68: Remapping Options

Remap Field in the SYS_CTRL_REG – 3 bits	
0x0	Remap ROM to address 0
0x1	Remap eFlash to address 0
0x2	Remap QSPI Flash/RAM cached area to address 0
0x3	Remap SysRAM1 to address 0
0x4	Remap SysRAM3 to address 0
0x5	Remap System Cache RAM to address 0
0x6	Remap QSPI Flash/RAM area to address 0 (uncached)
0x7	RESERVED

In the case of eFlash (which is expected to be the normal use case), a register defining the actual base address, where the image resides in the eFlash, needs to be created. This register should be retained and serve the cache controller addresses requests upon cache misses.

4.12 Security features

The DA14592 supports a number of security features that can be configured. This is done using the configuration script to program the following set-once (sticky) register bits (Note that these bits can only be reset by hardware or POReset).

Table 69: Security Configuration Options

Bit Field	Description
FORCE_M33_DEBUGGER_OFF	This bit permanently disables the M33 debugger.
FORCE_CMAC_DEBUGGER_OFF	This bit permanently disables the CMAC debugger.
PROT_USER_APP_CODE	This bit permanently disables write/erase operations on sectors 5-12 (Customer Application Area) in the embedded Flash.
PROT_APP_KEY	This bit permanently disables read/erase operations on sector 2, address range 0x00000800-0x00000BFF (User Application Expanded Encryption Keys) in the embedded Flash.
PROT_VALID_KEY	This bit permanently disables read/erase operations on sector 2, address range 0x00000C00-0x00000FFF (Signature Keys) in the embedded Flash.
PROT_CONFIG_SCRIPT	This bit permanently disables write/erase operations on sector 1 (User Configuration Script and security keys pointers) in the embedded Flash.
SECURE_BOOT	This bit enables authentication of the image in the embedded Flash while the system is booting.

4.12.1 Secure keys manipulation

This feature allows for programming up to eight different 256-bit symmetric keys for the user application cases and up to eight different 256-bit ECC keys for the authentication of the embedded Flash image. A revocation mechanism is supported through the booter allowing for changing the current key of any of the two operations while the product is in the field.

4.12.2 Secure boot

This feature is enabled by programming the SECURE_BOOT_REG[SECURE_BOOT] bit in the User Configuration Script and security keys pointers sector.

This forces authentication of the eFlash image before booting is finished. The booter code starts the Doubler, switches the system clock to 64 MHz, and then executes the Ed25519 verification algorithm. If the generated signature and the signature stored in the eFlash match, authentication is successful and booting is continued. If not, a hardware reset is issued.

The secure boot feature is not supported when booting from QSPI Flash is selected.

4.12.3 Secure access

Permanently disabling the SWI interface prevents unwanted access to the DA14592 CPUs. This is done by programming the SECURE_BOOT_REG[FORCE_M33_DEBUGGER_OFF] and SECURE_BOOT_REG[FORCE_CMAC_DEBUGGER_OFF] in the configuration script. This disconnects the SWI signals from the respective CPU's SWI controller.

Except for the sticky bit, the debugger has its own enable bit, namely, SYS_CTRL_REG[DEBUGGER_ENABLE] which is by default disabled. This bit is enabled during booting at the "Retrieve Application Code" phase.

If nothing is programmed in the configuration script, JTAG is enabled a few microseconds after POWERUP. If the sticky bit is programmed, JTAG is permanently disabled.

4.12.4 Validation

Every device can be uniquely identified using the Position, Package, and Time Stamp information put in the eFlash Information Page during manufacturing. This is a 64-bit word, which contains information about the position of the die, the wafer number, the package, and the time stamp of the production testing that is compared to the Tester ID and site.

4.12.5 Cryptography operations

The DA14592 is equipped with hardware acceleration for supporting all modern cryptography operations. More specifically, it consists of:

- A 256-bit capable AES encryption/decryption and key expansion engine that implements ECB/CBC/CTR modes covering all symmetric key application needs.
- A complete HASH block supporting SHA-2.

5 Power

5.1 Introduction

The DA14592 has a complete integrated Power Management Unit (PMU). This includes a buck DCDC converter, several LDOs for the different power rails of the system, and brownout detection. Figure 14 shows the system diagram of the analog Power Management Unit (PMU).

Features

- Synchronous Single Inductance Single Output Buck PCM DCDC converter with programmable output 1.1 V to 1.4 V.
- Capability of using the DCDC converter both in active as well as in Sleep modes.
- An LDO that can be used to instead of the DCDC converter in active and Sleep mode.
- Active and retention LDO for the digital core with 20 mA/2 mA driving capability.
- Active and retention LDO for the I/O-s with bypass functionality.
- Separate LDOs for the radio operation supporting High-Performance and Low Power modes.
- Brownout detection on all internal voltage rails.

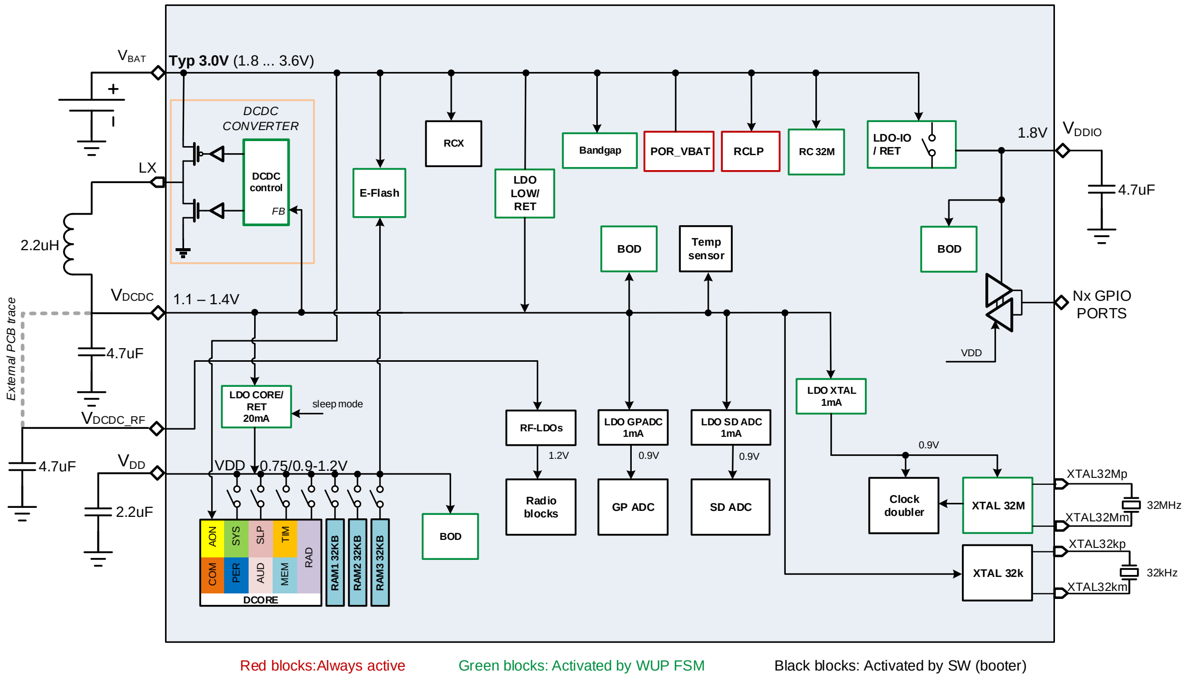

Figure 14. Power management unit architecture

5.2 Architecture

There are certain parts of the PMU that are powered directly from VBAT, namely the LDO_LOW, the LDO_IO, the Bandgap, the POR and RCLP and the AON power domain that contains special low leakage standard cells implementing the cold boot and wake-up from hibernation.

The V_{DCDC}, V_{DDIO}, and V_{DD} rails all have an external decoupling capacitor. All these rails have the possibility to use a BOD. At cold boot, the LDO_LOW is used to power the V_{DCDC} rail. The DCDC can be enabled by the software.

The digital core is further divided into power domains as explained in the "System Overview" section. These domains are controlled by Software and in some cases the Power Domains Controller (PDC).

Note
V_{DDIO} rail must be powered on during the wake up from Sleep mode. The valid register settings for a successful wakeup are: ● LDO_IO mode: ○ POWER_CTRL_REG[LDO_IO_ENABLE] = 1 ○ POWER_CTRL_REG[LDO_IO_RET_ENABLE_SLEEP] = 1 ● LDO_IO_BYPASS mode: ○ POWER_CTRL_REG[LDO_IO_BYPASS_ACTIVE] = 1 ○ POWER_CTRL_REG[LDO_IO_BYPASS_SLEEP] = 1

5.2.1 Buck DCDC converter

A Buck DCDC converter is used to supply the V_{DCDC} power rail that is basically providing power to the digital core, the radio, the embedded Flash, the ADCs, and the clocking sources. The block diagram is shown in Figure 15.

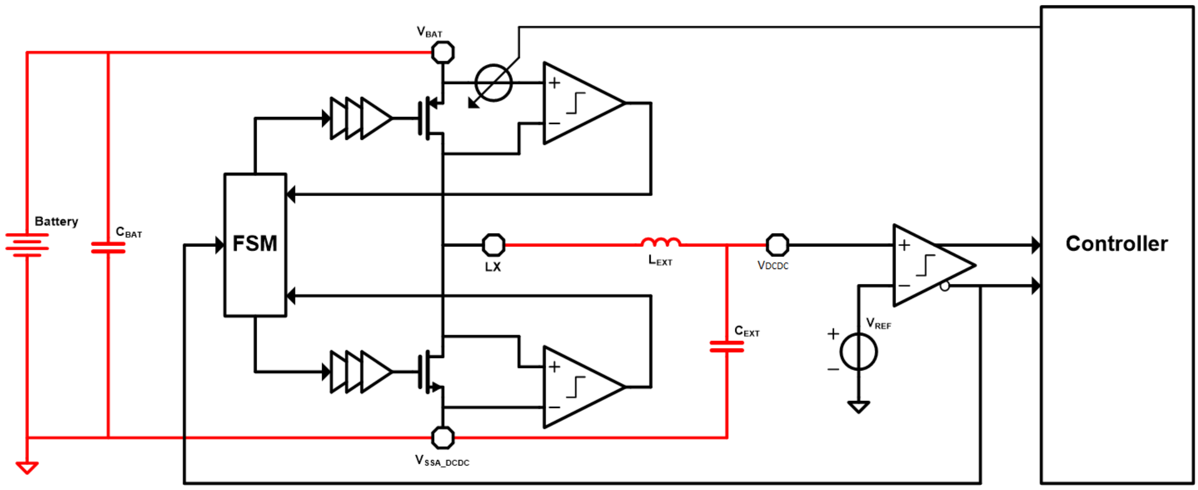

Figure 15. DCDC converter block diagram

The DCDC converter can be enabled/disabled by DCDC_CTRL_REG[DCDC_ENABLE]. The LDO_LOW is disabled/enabled automatically. The FSM is clocked by the RC32M clock. The level can be programmed by POWER_LEVEL_REG[VDCDC_LEVEL].

To enable the DCDC converter in Sleep mode, the POWER_CTRL_REG[DCDC_ENABLE_SLEEP] should be used. When this bit is set and the system goes to sleep, the DCDC converter is enabled instead of the retention LDOs. When in Sleep mode, the V_{DCDC} rail is monitored by the same comparator as in Active mode but the RCLP clock is used instead of the RC32M clock. When undervoltage is detected, the RC32M is enabled together with the DCDC-FSM and the V_{DCDC} is charged. If the voltage on the rail is OK, the RC32M and DCDC-FSM are turned off again. This results in duty cycled operation of the DCDC converter, where the duty cycle depends on the load current.

The expected DCDC efficiency versus load current for various battery voltages is shown in Figure 16. This is an indication of the possible efficiency of the DCDC converter. The quality of the external components can affect the overall DCDC efficiency.

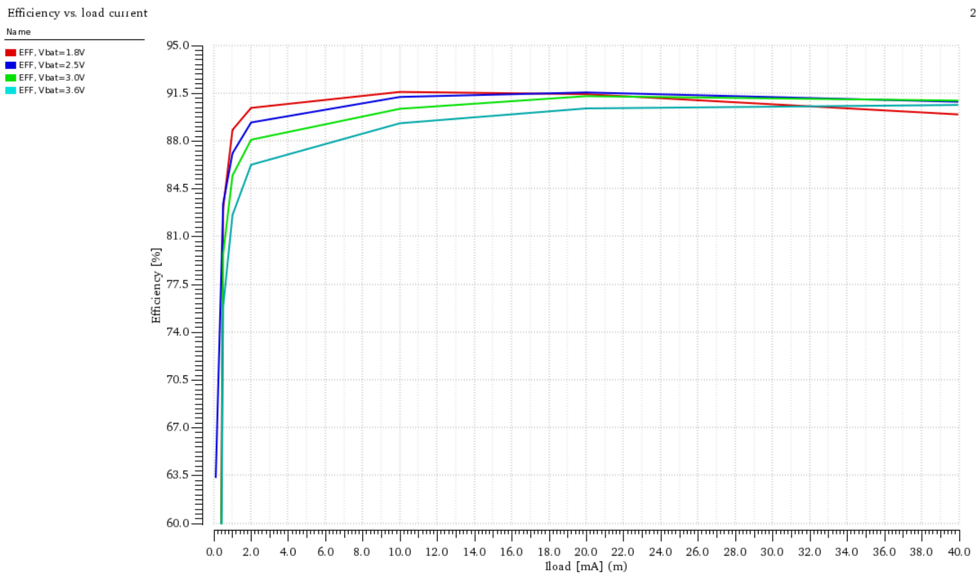

Figure 16. DCDC converter efficiency (active)

5.2.2 LDOs

Several LDOs are used to provide a stable power supply to the rails and the building blocks:

- LDO_IO provides a stable 1.8 V to the V_{DDIO} rail (which supplies the GPIOs) in Active mode. The LDO is enabled by the PMU FSM at cold boot. The LDO can be configured with the POWER_CTRL_REG. The LDO_IO_BYPASS_ACTIVE setting provides the option to connect the V_{DDIO} rail directly to V_{BAT}. When LDO_IO is set to 1.8 V output, the total current sourced from V_{DDIO} rail (including GPIO pins) must not exceed LDO_IO I_{L_LDO} max value.
- LDO_IO_RET provides a stable 1.8 V to the V_{DDIO} rail in Sleep mode. The LDO is enabled by the PMU FSM when the chip enters Sleep mode. The LDO can be configured with the POWER_CTRL_REG. The LDO_IO_BYPASS_SLEEP setting provides the option to connect the V_{DDIO} rail directly to V_{BAT}. When LDO_IO_RET is set to 1.8 V output, the total current sourced from V_{DDIO} rail (including GPIO pins) must not exceed LDO_IO_RET I_{L_LDO} max value.
- LDO_LOW supplies the V_{DCDC} rail when the DCDC converter is not enabled. The LDO is enabled by the PMU FSM at cold boot and can be used in Active mode as well as in Sleep mode (with different characteristics). When the DCDC converter is enabled, the LDO is disabled by the FSM. The output voltage is programmable between 1.1 V and 1.45 V in steps of 50 mV by POWER_LEVEL_REG[VDCDC_LEVEL]. (The same setting is used for the DCDC converter).
- LDO_CORE supplies the V_{DD} rail in Active mode. The LDO is enabled by the PMU FSM at cold boot. The output voltage can be programmed between 0.9 V and 1.25 V in steps of 50 mV by POWER_LEVEL_REG[VDD_LEVEL_ACTIVE].
- LDO_CORE_RET supplies the V_{DD} rail in Sleep mode. The LDO is enabled by the PMU FSM when the chip enters Sleep mode. The output voltage can be set to 0.75 V or 0.9 V by POWER_LEVEL_REG[VDD_LEVEL_SLEEP].
- LDOs for the analog blocks like the GP_ADC, SD_ADC and XTAL32M are typically enabled when the block is enabled and do provide a 0.9 V voltage internally for the block.
- LDOs for the radio blocks provide an internal voltage and are powered on only just before an RF transmission or reception.

The LDO_IO_RET, LDO_CORE_RET, and LDO_LOW (in Sleep mode) circuits operate in a sample and hold manner. They contain a reference voltage capacitance which is used to regulate the output voltage. However, due to leakage, this internal reference capacitor is discharged. To keep a stable voltage reference, an automatic mechanism is built in hardware to start the Bandgap, sample the voltage reference in the LDOs, and shut it down again. This periodic operation is programmable in terms of timing with use of the BG_REFRESH_INTERVAL, which counts sleep clock ticks.

5.2.3 POR circuit

The POR_VBAT is cleared when the battery voltage crosses V_{TH} for three RCLP clock periods. This is the trigger for the AON FSM to start a cold boot. When the battery voltage drops below the V_{TH}-V_{HYS}, the POR is issued again. Supply filtering has been added to prevent POR triggering on fast supply spikes.

6 Power Domain Controller

6.1 Introduction

The Power Domain Controller (PDC) is responsible for taking action after waking up or before going to sleep regarding the activation/de-activation of the seven digital power domains of the system. It follows the hardware FSM for start-up/wake-up and it precedes the go-to-sleep hardware FSM described in Section 4.3. It is a digital hardware state machine that defines the following parameters after consulting a programmable look-up table (LUT):

- Which power domain to activate after a wake-up trigger.
- If XTAL32M needs to be started after a wake-up trigger.
- If the system is allowed to go to hibernation (trigger the hardware FSM to disable DCDC, Bandgap, and LDOs).

The PDC can be triggered by any GPIO, general purpose timers, RTC, the MAC timer, SWD presence, or even by a Software trigger issued by one of the three masters of the system (Cortex-M33, CMAC, or FCU).

Features

- 12 places of 13-bit words Look-Up Table (LUT) for defining what the system should do upon any wake-up trigger
- Triggers from IOs, internal timers/controllers, or software
- Triggers from internal diagnostic signals
- Monitors if a power domain is actively used by any master before shutting it down
- System wake-up can be overridden by accessing the PMU_CTRL_REG.

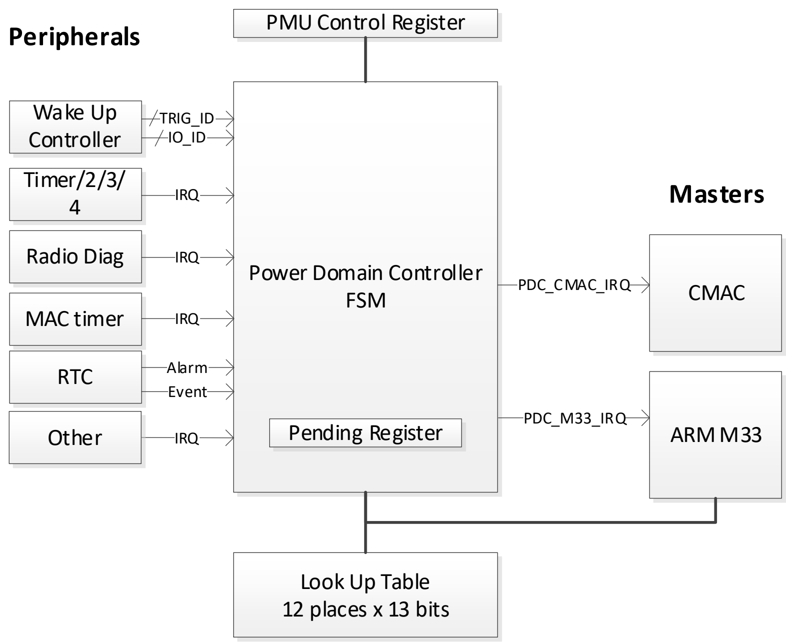

Figure 17. Power domain controller block diagram

6.2 Architecture

6.2.1 Look-up table

The Look-Up Table (LUT) is initialized at cold boot by the Cortex-M33. It is instructing the PDC which digital power domains to activate based on the triggering source.

Table 70 shows the format of the LUT.

Table 70: PDC LUT Format

	Bit	Function			
Triggers	0	if 0x00 then P0 GPIO	if 0x01 then P1 GPIO	if 0x02 or 0x3 then Peripheral	
	1				
	2	Pin_ID	Pin_ID	Periph_ID	
	3	Pin_ID	Pin_ID	Periph_ID	
	4	Pin_ID	Pin_ID	Periph_ID	
	5	Pin_ID	Pin_ID	Periph_ID	
	6	Pin_ID	Pin_ID		
Action	7	Enable XTAL32M			
	8	Enable PD_TMR			
	9	Enable PD_PER			
	10	Enable PD_COM (Needed for using the GPIOs)			
	11	Wake up Master_ID			
	12	Wake up Master_ID			

A look up table of 13 bits width describes what happens on each trigger. The depth of the LUT is 12 places, the system supports up to 12 different configurations given a specific trigger for waking up, but this could be changed dynamically by application software if needed.

There are three different trigger types that are evaluated when a trigger comes according to the value bits 0 and 1 of each LUT entry:

- If Bits[1:0]=0x0 then it is a GPIO toggle from P0. Bits [6:2] contain the pin number that has triggered.
- If Bits[1:0]=0x1 then it is a GPIO toggle from P1. Bits [6:2] contain the pin number that has triggered.
- If Bits[1:0]=0x2 or 0x3 then it is a trigger from some peripheral. Bits [5:2] define the peripheral according to Table 71.

Table 71: Peripheral Trigger Encoding

ID	Peripheral
0	Timer
1	Timer2
2	Timer3 || Quadrature Decoder
3	Timer4
4	RTC Alarm/Rollover
5	RTC Timer
6	MAC Timer
7	Reserved
8	XTAL32MRDY_IRQ
9	RFDIAG_IRQ
10	Debounced IO || JTAG present || CMAC2SYS_IRQ

ID	Peripheral
11	Reserved
12	FCU_IRQ
13	Fast wake-up
14	Reserved
15	Software Trigger Only

Bits[12:7] explain what needs to be done upon a trigger from the triggering sources as explained so far. More specifically, a triggering source might request to:

- Enable the XTAL32M. This bit is set if clock precision is required by the application. Note that the PDC state machine only enables the XTAL32MHz block but switching the system clock to the XTAL32M is not done. This must be done by software. Since multiple masters might want to switch to XTAL32M, switching to this clock can be done by any master but switching back should not be allowed. Turning off the XTAL32M is done automatically when the system enters sleep (for example, the PDC allows the hardware FSM to turn off all LDOs and Bandgap).
- Enable PD_TMR or PD_PER. These digital power domains do not contain a master. PD_MEM always is enabled since all masters are using this power domain.
- Enable PD_COM. To use GPIOs, the PD_COM must be activated. Some exceptions apply, for example, Timers' outputs during sleep, and so forth.
- Issue a wake-up IRQ/Signal to any other master. Hence a master can wake up another master with the use of a LUT entry. The Master ID is encoded as in Table 72.

Table 72: Master Trigger Encoding

ID	Master
0x0	Reserved
0x1	Cortex-M33
0x2	CMAC
0x3	Reserved

6.2.2 Operation

The PDC re-evaluates all available information every time there is a trigger input. It does the same every time there is feedback from one of the two masters (M33 or CMAC) indicating that they have finished what they were triggered to do. The latter is implemented using signaling between the masters and the PDC, namely:

- A deep sleep signal coming from Cortex-M33 or CMAC which is asserted when the Cortex-M33 SCR bit is set and the WFI command executed.

Every time a trigger occurs, the PDC follows the action plan as programmed in the LUT.

The PDC, upon triggering, asserts the respective bit number in the PENDING register that keeps one bit per LUT entry (for example, if LUT entry #2 is triggered, PENDING[2] is also asserted). The master that has been woken up, acknowledges the PENDING bit and after finishing its tasks, it notifies the PDC that any request for power domains activation is not valid anymore. The PDC cross-checks the complete LUT to decide if there is any other active entry that still uses any of the power domains requested. If not, these power domains are shut down. If other masters still use one of the domains, then it does not allow this domain to be powered down.

The PENDING register should be acknowledged as soon as possible and well before issuing a SLP/WFI command.

If there is no active LUT entry where one of the two masters is still up and running, then the PDC allows the system to go to deep sleep (that is, invoke the hardware FSM and turn off bandgap and LDOs). Any power domain is allowed to turn off, depending on the value of the respective field in

PMU_CTRL_REG. Hence, if for example, PMU_CTRL_REG[RADIO_SLEEP] = 1, then given the fact that no LUT entry is active, the power domain is turned off.

6.3 Programming

To program and use the Power Domain controller:

1. Add a PDC entry by writing the PDC_CTRLx_REG:
   a. Select Trigger (TRIG_SELECT):
      i. 0: Trigger is a GPIO on Port 0 (selectable through Wake-Up Controller).
      ii. 1: Trigger is a GPIO on Port 1 (selectable through Wake-Up Controller).
      iii. 2: Trigger is a Peripheral IRQ (see register description).
      iv. 3: Trigger is another Master (see register description).
   b. Select Trigger ID (TRIG_ID):
      Valid when TRIG_SELECT = 0x0, 0x1 or 0x2 (see register description for options).
   c. Enable extra options if needed (EN_COM, EN_PER, EN_TMR, EN_XTAL).
   d. Select the master to wake up (PDC_MASTER):
      i. 0: PDC entry is disabled.
      ii. 1: Wake up Arm Cortex-M33.
      iii. 2: Wake up CMAC.
2. Check if a PDC entry has been triggered (PDC_PENDING_REG).
3. If needed, trigger a PDC by software (PDC_SET_PENDING_REG).
4. Clear any pending requests and/or IRQs (PDC_ACKNOWLEDGE_REG).

7 Brown-Out Detector

7.1 Introduction

The brown-out detector (BOD) is a voltage monitoring circuit that triggers a hardware reset if LDOs or supply voltages go below a certain threshold.

The BOD comprises a set of analog comparators that are always active in either ACTIVE, SLEEP, or DEEP_SLEEP mode. The outputs of the comparators can selectively be masked or disabled.

Features

- Independent voltage monitoring of three power rails (VBAT, VDCDC, VDD)
- Programmable BOD level for VDD rail
- Periodic detection mechanism, available during active and sleep periods.

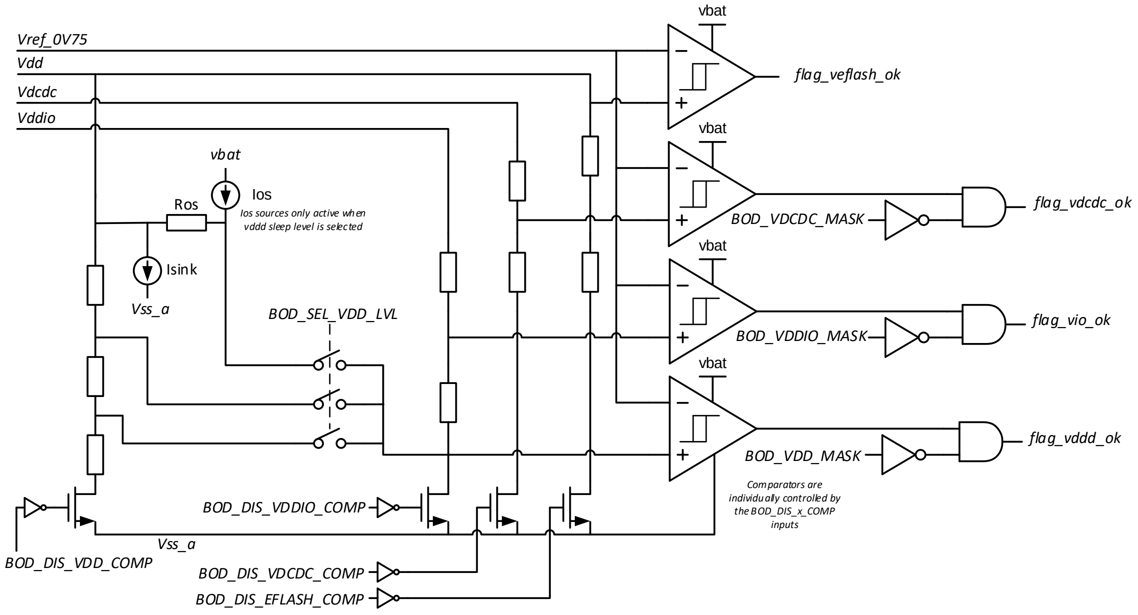

Figure 18. Brown-out detector block diagram

7.2 Architecture

The Brown-out Detector block is an analog block that utilizes four comparators that monitor during active or sleep mode the three power rails of the PMU. Table 73 shows the voltage level of each comparator.

Table 73: Brown-out Detectors and Default Levels

Comparator	Default POR Level (V)
VDDIO	1.65
VDCDC	0.99
VDD (Active)	0.78
VDD (Sleep)	0.7

For the VDD rail the voltage level threshold is programmable (BOD_CTRL_REG[BOD_SEL_VDD_LVL]) and can vary from 0.7 V to 1.05 V. The load in sleep mode is specific and not affected by the user application. It is recommended that the BOD is disabled for the VDD rail during the sleep mode.

Due to the eFlash internal mechanism when a power loss happens during programming or erasing the eFlash, the hardware reset signal coming from the BOD_VDD is gated by the FCU. This signal is an input to the FCU block indicating that a power loss is imminent. When this signal is asserted the FCU discharges the internal high voltage rails of the eFlash using an internal damage relief mechanism. This operation lasts for at least 5 μs. As soon as that time elapses, an FCU signal releases the hardware reset signal coming from the BOD_VDD.

7.3 Programming

To configure the Brown-Out Detector:

1. Configure the BOD voltage level for the VDD power rail (BOD_CTRL_REG[BOD_SEL_VDD_LVL]).
2. Enable/disable the BOD monitoring for the chosen power rails (BOD_CTRL_REG).

8 Reset

8.1 Introduction

The DA14592 has a nRST pad which is active low. It contains an RC filter for spikes suppression with 400 kΩ resistor and a 2.8 pF capacitor. It also includes a 25 kΩ pull-up resistor. This pad should be driven externally using FET or a single button connected to Ground. The typical latency of the nRST pad is about 2 µs.

Additionally, a configurable Power-on Reset circuitry is included to allow for a programmable time delayed POR functionality from a configurable reset source. By default, this circuitry is connected to the nRST pin, but software can remap it additionally to any GPIO.

A software reset is available also. This is done by either programming a specific register or triggering it through the debugger interface.

Features

- RC spike filter on nRST to suppress external spikes (400 kΩ, 2.8 pF).
- Three different reset lines (software, hardware, and POR).
- Reset cause is latched in a specific register.
- Configurable POR circuitry.

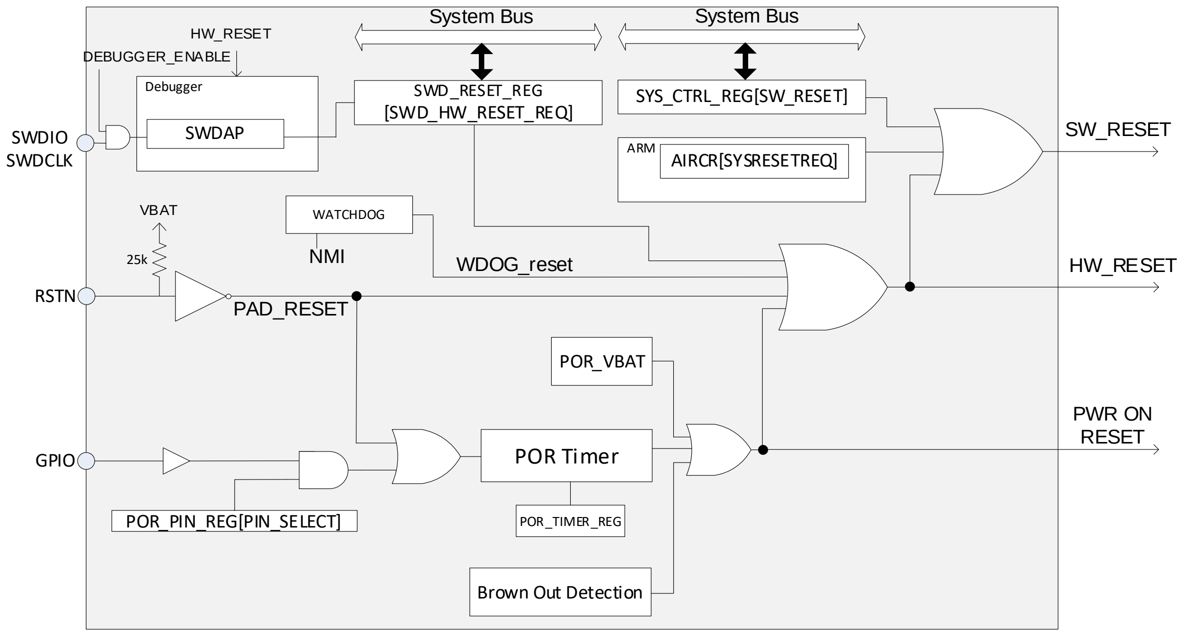

Figure 19. Reset block diagram

There are three main reset signals in DA14592:

1. The PWR ON reset, triggered by a GPIO set as POR source with selectable polarity and/or the nRST pad, after a programmable time delay.
2. The hardware reset, triggered by the RST pad when it becomes active for a short period of time (less than the programmable delay for POR).
3. The software reset, triggered by writing the SYS_CTRL_REG[SW_RESET] bit or Arm's AIRCR[SYSRESETREQ] register.

8.2 Architecture

The Power-on Reset (POR) signal is generated as follows:

- Internally; it will release the system's flip flops as soon as the VBAT voltage crosses the minimum threshold value.
- Brown-Out Detection senses the various internal voltage levels to be higher than the programmed thresholds.
- Externally by a Power-on Reset source (nRST pad or GPIO).

The hardware reset can be automatically triggered at system wake-up from the Extended or Deep Sleep mode by programming bit PMU_CTRL_REG[RESET_ON_WAKEUP]. The PWR ON reset and the hardware reset runs the cold start-up sequence and the BootROM code is executed.

The software 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 is remapped.

Certain registers are reset by Power-on Reset only. Table 74 lists these registers.

Table 74: Reset Signals and Registers

Reset Source	Registers
POR Reset	RST_CTRL_REG SYS_STAT_REG[POWER_IS_UP] CLK_XTAL32K_REG CLK_RTCDIV_REG BANDGAP_REG RESET_STAT_REG[PORESET_STAT] POR_PIN_REG POR_TIMER_REG RTC_CONTROL_REG RTC_KEEP_RTC_REG RTC_EVENT_CTRL_REG RTC_PDC_EVENT_PERIOD_REG
HW Reset	CACHE_FLASH_REG CACHE_EFLASH_REG CLK_AMBA_REG CLK_RADIO_REG CLK_CTRL_REG PMU_CTRL_REG CLK_RCLP_REG CLK_RC32M_REG CLK_RCX_REG P0_PAD_LATCH_REG P0_SET_PAD_LATCH_REG P0_RESET_PAD_LATCH_REG P1_PAD_LATCH_REG P1_SET_PAD_LATCH_REG P1_RESET_PAD_LATCH_REG BIAS_VREF_SEL_REG RAM_PWR_CTRL_REG RESET_STAT_REG[HWRESET_STAT] RESET_STAT_REG[SWD_HWRESET_STAT] SECURE_BOOT_REG BOD_CTRL_REG POWER_CTRL_REG POWER_LEVEL_REG

Reset Source	Registers
	HIBERN_CTRL_REG PMU_SLEEP_REG DCDC_CTRL_REG PDC_CTRL0_REG PDC_CTRL1_REG PDC_CTRL2_REG PDC_CTRL3_REG PDC_CTRL4_REG PDC_CTRL5_REG PDC_CTRL6_REG PDC_CTRL7_REG PDC_CTRL8_REG PDC_CTRL9_REG PDC_CTRL10_REG PDC_CTRL11_REG PDC_ACKNOWLEDGE_REG PDC_PENDING_REG PDC_PENDING_CM33_REG PDC_PENDING_CMAC_REG PDC_SET_PENDING_REG PDC_CONFIG_REG XTAL32M_START_REG XTAL32M_SETTLE_REG XTAL32M_TRIM_REG XTAL32M_CAP_MEAS_REG XTAL32M_FSM_REG XTAL32M_CTRL_REG XTAL32M_IRQ_CTRL_REG XTAL32M_STAT0_REG XTAL32M_IRQ_STAT_REG CLKDBLR_CTRL1_REG CLKDBLR_CTRL2_REG FLASH_PTWK_SP_REG QDEC_CTRL_REG QDEC_XCNT_REG QDEC_YCNT_REG QDEC_CLOCKDIV_REG QDEC_CTRL2_REG QDEC_ZCNT_REG QDEC_EVENT_CNT_REG QSPIC_CTRLBUS_REG QSPIC_CTRLMODE_REG QSPIC_RECVDATA_REG QSPIC_BURSTCMDA_REG QSPIC_BURSTCMDB_REG QSPIC_STATUS_REG QSPIC_WRITEDATA_REG QSPIC_READDATA_REG QSPIC_DUMMYDATA_REG QSPIC_ERASECTRL_REG

Reset Source	Registers
	QSPIC_ERASECMDA_REG QSPIC_ERASECMDB_REG QSPIC_BURSTBRK_REG QSPIC_STATUSCMD_REG QSPIC_CHCKERASE_REG QSPIC_AWRITECMD_REG QSPIC_MEMBLEN_REG
Watchdog Reset	RESET_STAT_REG[CMAC_WDOGRESET_STAT] RESET_STAT_REG[WDOGRESET_STAT]
SW Reset	The rest of the Register File

8.2.1 Power-on reset from pin

The Power-on Reset function can be triggered at timer expiration. It is available at two sources:

- Reset Pad (nRST): 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 POR_TIMER_REG configures the time needed for the POReset signal to be active. The register field POR_TIME is a 7-bits field (maximum value is 0x7F), which holds a multiplication factor for counting RCLP clock periods. This is explained in the following formula:

Total time for POR = POR_TIME x 4096 x RCLP clock period,

where RCLP clock period equals to 31.25 or 1.95 μs at 25 °C.

When RCLP runs at 32 kHz, the maximum time for issuing a POR is ~16.2 seconds at 25 °C, while the default value is ~3 seconds (POR_TIME=0x18).

The RCLP clock is temperature dependent and as such drift over the temperature should be expected.

The Power-on Reset timer is clocked by the RCLP clock. If the application disables the RCLP, then hardware takes care of enabling the RCLP clock when the POR source (Reset pad or GPIO) is asserted. If POR is generated from the Reset pad, RCLP operates with the default (reset) trim value. If a GPIO is used as a POR source, the RCLP clock is trimmed. The timing deviation between both cases is expected to be minor.

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.

Figure 20 shows the operation of the Power-on Reset for both Reset pad and GPIO.

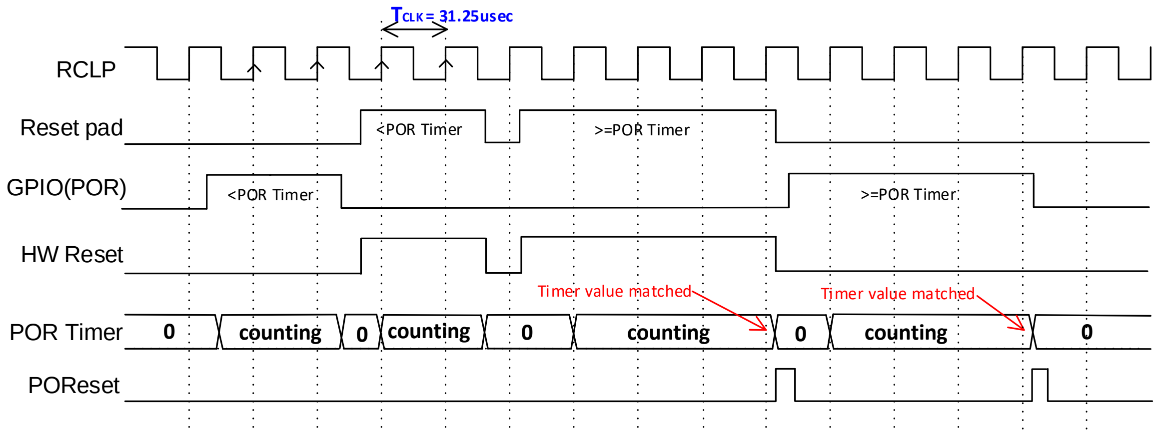

Figure 20. Power-on reset timing diagram

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 resets to 0. If a second source is asserted while the first is already asserted and the first is released after that point, POR occurs; 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, hardware, software) hence 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.

8.3 Programming

To configure the POR from GPIO functionality:

1. Select the GPIO to be set as POR source (POR_PIN_REG[POR_PIN_SELECT]).
2. Set up the input polarity of the GPIO that causes POR (POR_PIN_REG[POR_PIN_POLARITY]).
3. Configure the time for the POR to happen (POR_TIMER_REG[POR_TIME]). The default time is ~3 seconds.

NOTE
To set up the time that the Reset pin produces a POR, only POR_TIMER_REG has to be set.

9 Arm Cortex-M33

9.1 Introduction

The Cortex-M33 processor is a low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications. The processor is based on the Armv8-M architecture and is primarily for use in environments where security is an important consideration. The interfaces that the processor supports include:

- Code AHB (C-AHB) interface
- System AHB (S-AHB) interface
- External PPB (EPPB) APB interface
- Debug AHB (D-AHB) interface.

Arm Cortex-M33 does not retain its status when going to any sleep modes that switch off its power domain. So, the CPU always wakes up in reset. Restoration of the CPU state is done by software.

Features

- Supports the Armv8-M Main Extension. The processor has optional support for one or more of the following extensions:
  - The Floating-point Extension.
  - The Digital Signal Processing (DSP) Extension.
  - The Debug Extension.
- An in-order issue pipeline.
- Thumb-2 technology.
- Configurable to perform data accesses as either big or little endian.
- Nested Vectored Interrupt Controller (NVIC).
- Floating Point Unit (FPU) supporting single-precision arithmetic:
  - Combined multiply-add instructions for increased precision (Fused MAC).
  - Hardware support for conversion, addition, subtraction, multiplication with optional accumulate, division, and square root.
  - Hardware support for denormals and all IEEE Standard 754-2008 rounding modes.
  - 32 32-bit single-precision registers or 16 64-bit double-precision registers.
  - Lazy floating-point context save.
- Support for exception-continuable instructions.
- Micro Trace Buffer (MTB).

9.2 Architecture

9.2.1 Interrupts

This section lists all 35 interrupt lines, except the NMI interrupt, and describes their source and functionality. Table 75 shows the overview of the interrupts.

Table 75: Interrupt List

#	Name	Type	Polarity	Description
0	Reserved	-	-	-
1	DMA_IRQ	Pulse	Active High	General Purpose DMA interrupt line
2	CMAC2SYS_IRQ	Level	Active High	CMAC and mailbox interrupt line
3	UART_IRQ	Level	Active High	UART interrupt line

#	Name	Type	Polarity	Description
4	UART2_IRQ	Level	Active High	UART2 interrupt line
5	I2C_IRQ	Level	Active High	I2C interrupt line
6	SPI_IRQ	Level	Active High	SPI interrupt line
7	FCU_IRQ	Level	Active High	eFlash interrupt line
8	PCM_IRQ	Pulse	Active High	PCM interrupt line
9	SRC_IN_IRQ	Level/Pulse	Active High	SRC input interrupt line
10	SRC_OUT_IRQ	Level/Pulse	Active High	SRC output interrupt line
11	SRC2_IN_IRQ	Level/Pulse	Active High	SRC2 input interrupt line
12	SRC2_OUT_IRQ	Level/Pulse	Active High	SRC2 output interrupt line
13	Reserved			
14	TIMER_IRQ	Level	Active High	TIMER interrupt line
15	TIMER2_IRQ	Level	Active High	TIMER2 interrupt line
16	RTC_IRQ	Level	Active High	RTC interrupt line
17	KEY_WKUP_GPIO_IRQ	Level	Active High	Debounced button press interrupt. This interrupt is first driven to the PDC and then directed to the required masters to be waken up/notified
18	PDC_M33_IRQ	Level	Active High	This is an interrupt coming from the PDC indicating that the M33 needs to be waken up due to a GPIO/Peripheral/other master request.
19	MRM_IRQ	Pulse	Active High	Cache miss rate monitor interrupt
20	Reserved			
21	QUADDEC_IRQ	Level	Active High	Quadrature Decoder interrupt line
22	Reserved			
23	XTAL32M_RDY_IRQ	Pulse	Active High	Indicates that XTAL32M oscillator is trimmed and settled and can provide a reliable 32 MHz clock
24	CLK_CALIBRATION_IRQ	Level	Active High	Indicates that the calibration of the selected clock has been performed successfully
25	GPADC_IRQ	Level	Active High	SAR analog-digital converter interrupt
26	SDADC_IRQ	Level	Active High	SD analog-digital converter interrupt
27	CRYPTO_IRQ	Level	Active High	Crypto interrupt. Sources: AES or HASH function interrupt
28	CAPTIMER_IRQ	Pulse	Active High	GPIO triggered Timer1 Capture interrupt
29	RFDIAG_IRQ	Pulse	Active High	Baseband or Radio Diagnostics Interrupt. Required for signaling Radio or Baseband internal events. 2 signals per Radio and 2 per BB
30	TIMER3_IRQ	Level	Active High	TIMER3 interrupt line
31	TIMER4_IRQ	Level	Active High	TIMER4 interrupt line
32	RTC_EVENT	Level	Active High	RTC event interrupt line
33	GPIO_P0_IRQ	Level	Active High	GPIO port 0 toggle interrupt line
34	GPIO_P1_IRQ	Level	Active High	GPIO port 1 toggle interrupt line

9.2.2 Reference

The register descriptions for the Nested Vectored Interrupt Controller (NVIC), the System Control Block (SCB), and the System Timer (SysTick) of the Arm Cortex-M33 can be found in the following documents, available on the Arm website:

Devices generic user guide:

https://developer.arm.com/documentation/100235/0100/

Technical reference manual:

https://developer.arm.com/docs/100230/0004

10 Cache Controller

10.1 Introduction

The cache controller is used to accelerate the system performance of the Arm Cortex-M33 executing from eFlash. The cache controller dynamically loads program and data code into the cache RAM and Arm Cortex-M33 executes from there, resulting in reduced accesses on the eFlash for more power efficient operation.

The cache controller is controlled via the CACHE_*_REGs. The cache administration is kept in the TAG memory region, which is part of the cache RAM. The cache RAM is initialized when the cache controller is enabled.

For debugging, the cache RAM memory can be monitored on the AHB-SYS bus (see memory map). Furthermore, the cache RAM can be used for static code and data storage when the cache controller is disabled.

Features

- Four-way set associative cache implementation.
- 8 kB cache size and 8 B cache line size.
- Support up to 32 MB of cacheable range.
- Built-in cache memory invalidation/initialization when cache is enabled.
- Least Recently Used (LRU) replacement strategy.
- Cache RAM monitoring.
- Instruction and Data caching upon read access, no write path to cache available.
- Bypass mode.
- Configurable critical-word first functionality for cache misses.
- Cache internal latency:
  - Zero wait cycle for cache hits on Most Recently Used (MRU) words.
  - One wait cycle for cache hits on LRU words.
  - 1-wait cycle for cache misses with critical-word first functionality enabled.
  - Max four-wait cycles for cache misses without critical-word first functionality enabled.
  - Zero wait cycle in transparent bypass mode.

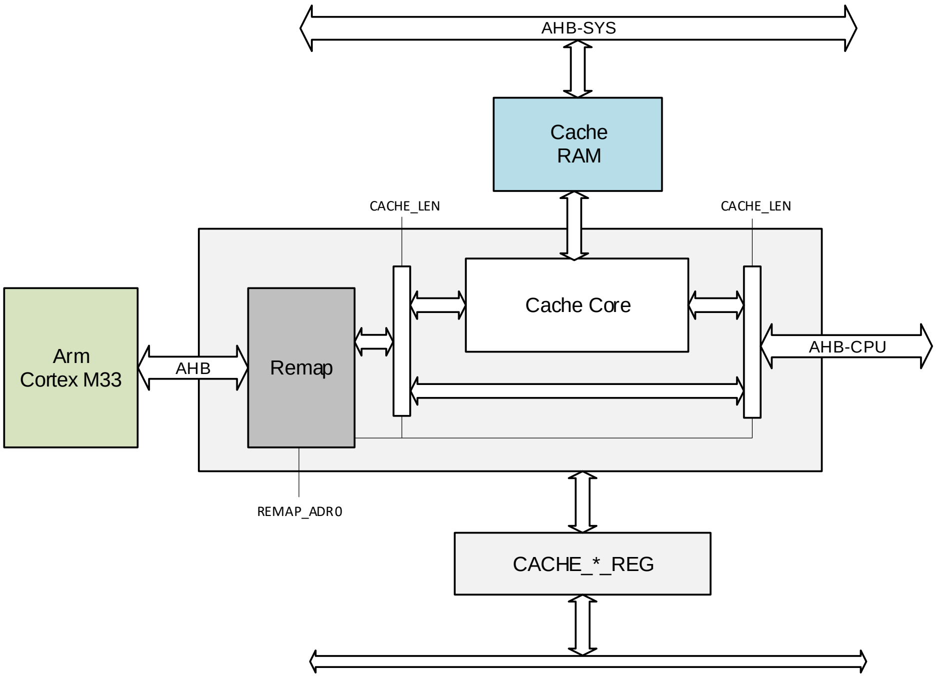

Figure 21. Cache controller block diagram

10.2 Architecture

10.2.1 Cacheable range

The cache controller caches address range 0-0x1FFFFFFF (32 MB). If REMAP_ADR0 = 0x1 or 0x2, all addresses from 0 to CACHE_EF_LEN are cached, else the cache controller automatically asserts the bypass mode. The bypass mode can be forced for all addresses by setting CACHERAM_MUX = 0 or the CACHE_EF_LEN = 0.

10.2.2 Cache replacement strategy

The cache controller stores usage statistics for each cache line to be able to indicate the Least Recently Used (LRU) cache line among the four ways. In case of a cache miss, the LRU cache line is replaced with a new cache line, which is fetched from eFlash. The data fetch can be configured to be critical word first, in which case the requested word from the Arm Cortex M33 is the first word that is requested by the eFlash or regular data fetch in which the first word in the cache line is requested by the eFlash. This functionality can be configured by the CACHE_CWF_DISABLE bit in CACHE_CTRL2_REG.

10.2.3 Cache reset

The cache controller comprises two reset signals connected:

- The HW_RESET: When this reset is activated, all cache logic and registers are reset to their default values, while all data in the cache memories are cleared and all CACHE_*_REG are set to their reset values. It takes around 300 AHB cycles to clear the cache memories. If a fetch request occurs during this reset period, the request is the reset and wait-states is inserted.
- The SW_RESET: Upon a SW_RESET the cache state machine and cache memories are reset, but the CACHE_*_REG are not affected and remains as programmed.

Furthermore, the cache controller can be disabled or enabled with the use of SYS_CTRL_REG[CACHERAM_MUX] configuration bit. Every time the cache controller is enabled it performs a cache RAM initialization process to clear all the TAG information and cached data from the cache RAM. If the cache is disabled, the cache RAM can be used by the system as a regular RAM.

10.2.4 Miss rate monitor

The system includes a cache miss rate monitor circuitry that gives real time information on the number of cache misses within a certain amount of time. When a programmable threshold is reached, an interrupt is sent to the CPU to act. This block only operates while the system is in active mode. The main features are:

- Up to 10 ms active time interval counter.
- Registered amount of cache misses.
- Registered amount of cache hits.
- Programmable threshold of cache misses that generates an interrupt.
- The CACHE_MRM_HITS_REG contains the amount of cache hits. The CACHE_MRM_MISSES_REG contains the number of misses counted within the time interval programmed at the CACHE_MRM_TINT_REG in CPU clock cycles.

10.3 Programming

10.3.1 Cache controller programming

To configure the Cache Controller:

1. Disable the Cache by clearing the CACHE_CTRL2_REG[CACHE_EF_LEN] bit field.
2. Enable the cache (CACHE_CTRL2_REG[CACHE_LEN] != 0).
   The size of the cacheable eFlash memory can be specified in CACHE_CTRL2_REG[CACHE_EF_LEN] when CACHE_CTRL2_REG[CACHE_EF_LEN] != 1. If CACHE_CTRL2_REG [CACHE_EF_LEN] = 1, then the memory space is specified by CACHE_EFLASH_REG [EFLASH_REGION_OFFSET].

10.3.2 Miss rate monitor programming

To configure the Miss Rate Monitor:

1. Freeze all counters by clearing the CACHE_MRM_CTRL_REG[MRM_START] bit.
2. Set up the thresholds that produce interrupts:
   a. Cache Misses threshold: CACHE_MRM_MISSES_THRES_REG[MRM_MISSES_THRES].
   b. Cache Hits threshold: CACHE_MRM_HITS_THRES_REG[MRM_HITS_THRES].
   c. Time passed threshold: CACHE_MRM_TINT_REG[MRM_TINT].
3. Unmask all interrupts by setting the CACHE_MRM_CTRL_REG[MRM_IRQ_MASK] bit.
4. Enable the chosen interrupts:
   a. CACHE_MRM_CTRL_REG[MRM_IRQ_HITS_THRES_STATUS]: The number of cache hits reached the programmed threshold.
   b. CACHE_MRM_CTRL_REG[MRM_IRQ_MISSES_THRES_STATUS]: The number of cache misses reached the programmed threshold.
   c. CACHE_MRM_CTRL_REG[MRM_IRQ_TINT_STATUS]: The time interval counter reached the end.
5. Enable all counters by setting the CACHE_MRM_CTRL_REG[MRM_START] bit.
6. Read the results (misses and/or hits) in the CACHE_MRM_MISSES_REG and CACHE_MRM_HITS_REG registers.

11 Bus

11.1 Introduction

The DA14592 is equipped with a multi-layer AMBA bus that enables parallel data paths among different masters and slaves.

The bus matrix comprises four main busses:

1. AHB-CPUC bus where the System Cache is master. This is the primary bus for executing code from. Memory map range is 0x00000000- 0x1FFFFFFF (No address decoding above 0x20000000).
2. AHB-CPUS bus where the Arm M33 is master. This is the bus for the system. Memory map range is 0x20000000 – 0xDFFFFFFF and above 0xE0100000 (No address decoding below 0x20000000).
3. AHB-DMA bus where the RFMON, the Generic DMA, and the Crypto block (AES/HASH) can be masters.
4. AHB-CMAC bus where the CMAC is the only master.

There are several slaves, sitting behind interconnection multiplexers (ICMs) that allow access from the AHB busses; namely:

- QSPI RAM/Flash memory controller
- 32-bit APB peripheral registers
- The RAM controller
- The ROM controller
- The eFlash (FCU) controller
- The AHB register file contains registers for the QSPI RAM/Flash controller, CMAC, the Crypto, and so on.

Features

- Enables parallelization of data transfers from:
  - Peripherals to memory (GP DMA)
  - Cortex-M33 data read/writes from RAM
  - Cortex-M33 executing code from eFlash
  - Cortex-M33 executing code from QSPI RAM/Flash.
- Supports parallel accessing of GP DMA, MACCPU, and SYSCPU on (different) memory segments without collisions.

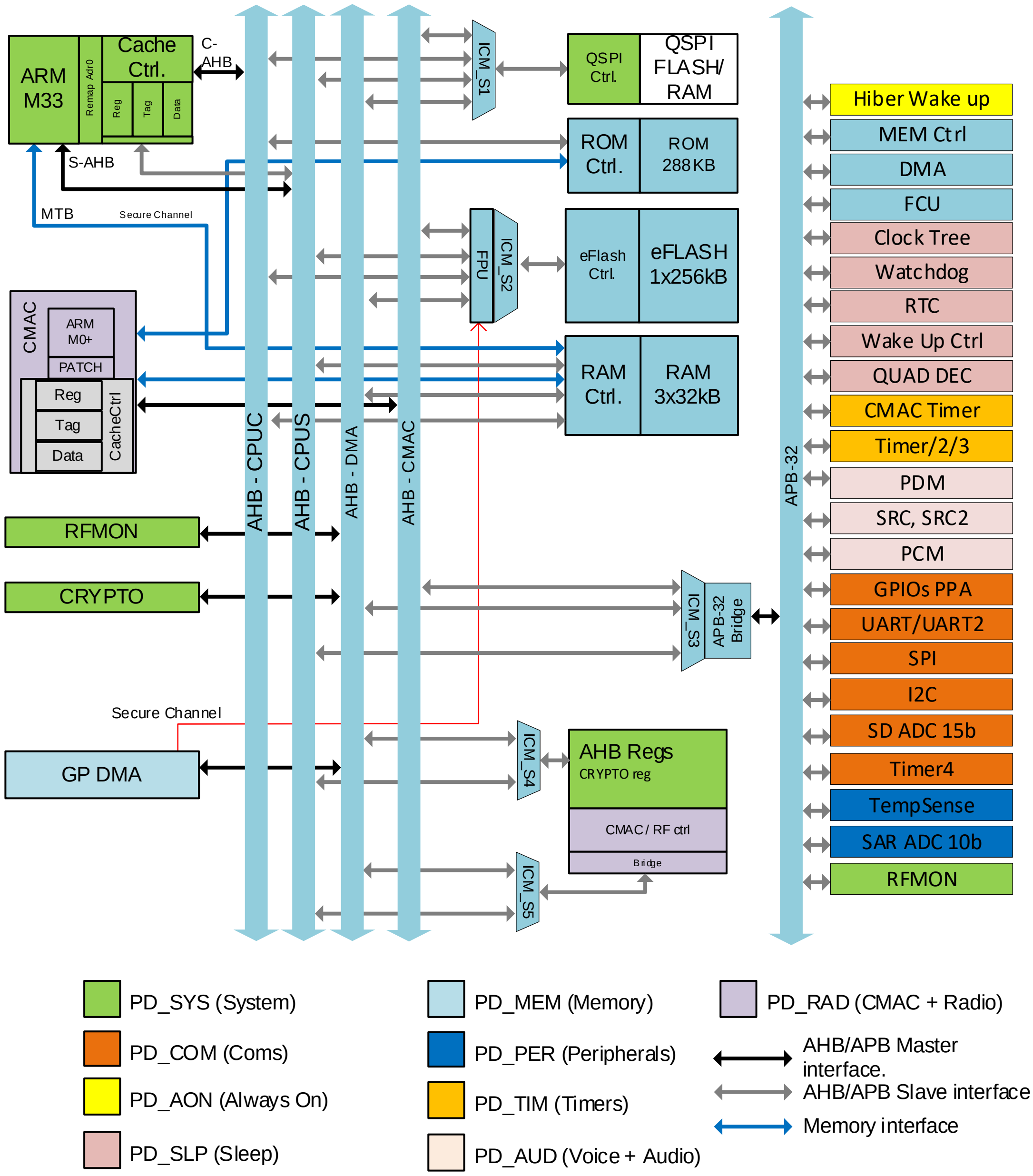

Figure 22. Bus architecture

11.2 Architecture

Figure 22 shows the architecture of the AMBA bus matrix. There are three potential masters for the APB32, namely: CMAC, Cortex-M33, and the DMA engines. A multiplexer is required there to allow access to a single master on the APB32 multiplexing.

There is one protection unit – the eFlash Controller Register Unit (FPU) – that forms part of the system's security perimeter. FPU protects the eFlash region where private keys are stored; keys that the CPU cannot read. Access to this space is only allowed when a special signal (secure channel) is activated by the GP DMA. Furthermore, the Crypto Controller registers that keep the AES key are write only registers and cannot be written by CPU.

Additionally, FPU blocks any access to specific regions of the eFlash based on the configuration of the SECURE_BOOT_REG. Rest of accesses to other register files or eFlash regions are completely transparent and are not gated by FPU.

The default priorities of the four buses are listed in the following tables.

Table 76: AHB-DMA Master Fixed Priorities

Priority	Master
1 (highest)	RFMON
2	GPDMA
3	CRYPTO
4 (lowest)	Reserved

Table 77: ICM1 Programmable Priorities

Priority	AHB Bus (Default)	AHB Bus (Programmable)
1 (highest)	AHB-CMAC	AHB-DMA
2	AHB-CPUS	AHB-CMAC
3	AHB-CPUC	AHB-CPUS
4 (lowest)	AHB-DMA	AHB-CPUC

Table 78: ICM2 Priorities

Priority	AHB Bus
1 (highest)	AHB-CMAC
2	AHB-CPUS
3	AHB-CPUC
4 (lowest)	AHB-DMA
Note: Bursts can brake	

Table 79: ICM3 Priorities

Priority	AHB Bus
1 (highest)	AHB-CMAC
2	AHB-DMA
3	Reserved
4 (lowest)	AHB-CPUS

Table 80: ICM4 Priorities

Priority	AHB Bus
1 (highest)	AHB-DMA
2 (lowest)	AHB-CPUS

Table 81: ICM5 Priorities

Priority	AHB Bus
1 (highest)	AHB-DMA
2 (lowest)	AHB-CPUS

12 Configurable MAC

12.1 Introduction

The Configurable Medium Access Controller (CMAC) is a Flexible MAC hardware block based on Arm Cortex-M0+ that can be programmed to support multiple protocols.

The CMAC executes code from the system RAM and ROM with zero wait states. It also has access to all APB peripherals of the system.

Features

- Implements Bluetooth® LE 5.2 controller stack, including HCI.
- Optional Bluetooth® LE 5.2 Features supported:
  - 2 Mbps
  - Channel selection algorithm #2
  - High Duty Cycle Non-Connectable Advertising
  - Advertising Extensions (Note 1)
  - Periodic Advertising (Note 1)
  - AoA/AoD (Note 1)
- Autonomous operation for Advertising or keep-alive connections.
- Autonomous execution and sleep cycles.
- Rapid wake-up and go-to-sleep operation.
- Size and base address of RAM for code execution is configurable.
- Accelerators in hardware:
  - Link Layer and framing Timers
  - AES-128-bit crypto engine
  - Configurable Whitening engine compliant with Bluetooth® LE standard
  - Configurable Whitening engine compliant with ANT+ standard
  - Configurable CRC engine, up to 32-bit order primitive polynomials
  - 32 bits wide Correlator
  - AoA/AoD support in hardware (Note 1).

Note 1 Supported in DA14592-02, which also supports Bluetooth 5.3 core features.

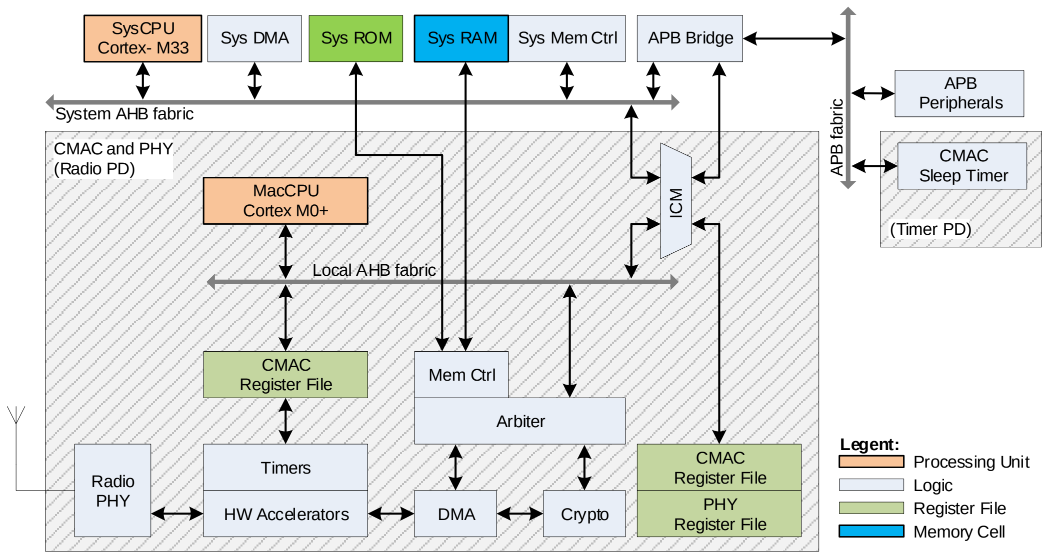

Figure 23. Configurable MAC block diagram

12.2 Architecture

The CMAC block is an autonomous system that can execute Bluetooth® LE and other protocols, if there is enough hardware accelerators and Firmware throughput.

A dedicated Cortex-M0+ executes code through the local memory controller that reuses the system memory RAM for storing code and data. A hardware bit stream controller performs all real time functions by using hardware accelerators like whitening, CRC, and crypto engines. A Link Layer Timer schedules protocol operations and radio transactions. The radio transactions are programmed to the CMAC registers. The hardware executes them automatically in the scheduled moment, activating the proper accelerators.

Radio Register File and CMAC Sleep Timer Register Files are accessible by both CPUs but typically only CMAC CPU accesses them.

12.2.1 Dataflow

CMAC can implement the Bluetooth® LE Data Link Layer, providing an HCI interface towards the system processor Cortex-M33.

The communication of CMAC processor Cortex-M0+ with the system processor Cortex-M33 is performed through IRQ signals and the common system memory RAM. A mailbox mechanism can be used to exchange command and data structures between the two CPUs.

12.2.2 Diagnostics

Table 82 shows available CMAC diagnostics signals.

Table 82: CMAC Diagnostic Signals

Diagnostics	Signal	Description
CMAC_DIAG_0	TX EN	Data transmit enable signal
CMAC_DIAG_1	RX EN	Data receive enable signal
CMAC_DIAG_2	DATA EN	TX/RX Data Enable pulse
CMAC_DIAG_3	DATA COMB	TX and RX data bits (combined on the same line)
CMAC_DIAG_4	Reserved	-
CMAC_DIAG_5	Reserved	-
CMAC_DIAG_6	Reserved	-
CMAC_DIAG_7	CORR COMB	Indicates that the correlator is active
CMAC_DIAG_8	CRC	RX CRC LFSR is zero to indicate correctly received packet
CMAC_DIAG_9 to 15	Reserved	-

Figure 24 shows the functionality of the diagnostic signals.

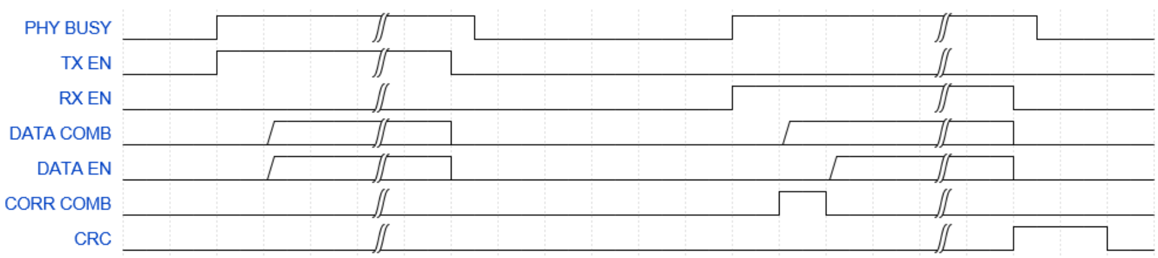

Figure 24. CMAC diagnostics timing diagram

13 Memory Controller

13.1 Introduction

The memory controller allows access to the 96 kB RAM pool by the system's masters; these include the: SysCPU, MACCPU and the General-Purpose DMA controller. It comprises three AHB ports (GPDMA, CPU-C and CPU-S) and two Memory ports (MACCPU and Micro Trace Buffer (MTB)).

The memory controller implements an intelligent addressing scheme so that RAM is not fragmented by various data or code allocations. At the same time, it allows parallel access of multiple data streams on different RAM cells, transparent to the application software. So, it allows the System CPU to store/read application variables, while the MAC CPU executes code and at the same time, without conflict and arbitration that would result in wait states.

Definition of the different segments allocated in memory is fully programmable. In cases where two or more segments are sharing the same RAM cell, hardware arbitration resolves conflicts by issuing wait states to the masters having the least priority.

Features

- Supports flexible memory allocation per master to avoid fragmentation.
- Parallelizes data flows from/to various masters on the system.
- Allows for programmable priority scheme per RAM cell.
- Generates wait cycles to AHB masters.

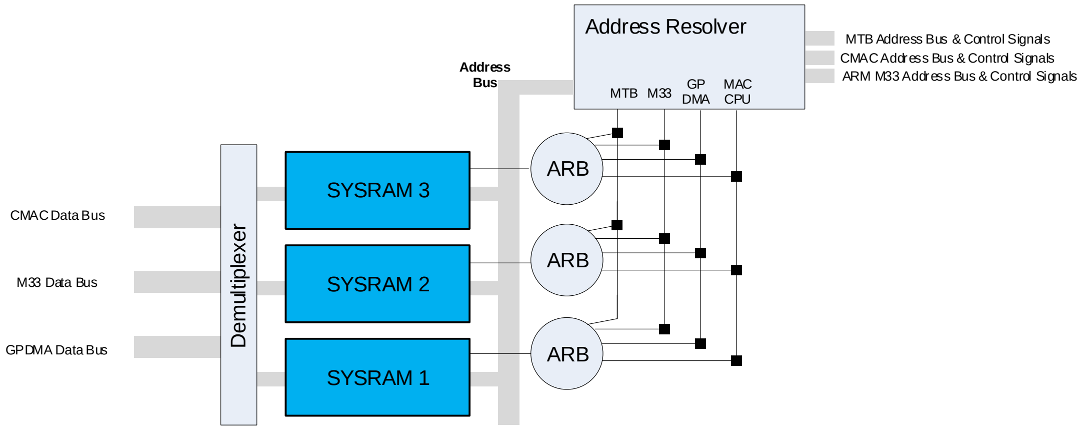

Figure 25. Internal architecture of the memory controller

13.2 Architecture

Figure 25 shows the internal architecture of the memory controller.

Where two or more masters need access to the same RAM cell but different addresses (each on its own segment), then arbitration is used. If wait cycles (for the least prioritized) are not acceptable, a reconfiguration of the segment boundaries, at the cost of some RAM fragmentation, should be considered.

All masters support a "Ready" like signal functionality, so they can be stalled for a specific amount of clock cycles. This makes the arbitration signals for the AHB interfaces (SysCPU, GP DMA, and so on) HREADY, while the memory interfaces (MTB) also support a respective input (MREADY) and can proceed in reading/writing the RAM cell if this signal is high.

The CMAC does not support any MREADY functionality and always has the highest priority.

Table 83 shows the basic metrics of each master accessing the memory controller.

Table 83: Memory Controller Masters Access Metrics

Master	Frequency Range	Stall Mechanism	Access Rate	Wait States Tolerance
Cortex-M33	32 MHz – 64 MHz	AHB HREADY	Non-burst accesses for data and/or code	Should keep that < 4 AHB clocks to avoid performance degradation
CMAC	32 MHz – 64 MHz	None	Non-burst accesses for data and/or code	0
MTB	32 MHz – 64 MHz	None	Non-burst accesses for data and/or code	0
GP DMA	32 MHz – 64 MHz	AHB HREADY	8-beat	< 6 AHB clocks

Two different arbitration schemes are provided:

1. Static priority of masters. The MEM_PRIO_REG defines three fields, one for each AHB layer. The priority level of each channel can be defined by programming these fields. Arbitration is always done in favor of the highest priority master.
2. Round Robing priority of masters. To avoid stalling a master for too long, another register (MEM_STALL_REG) can be programmed with the maximum amount of clock cycles that a channel might be stalled. As soon as this number is reached, this channel automatically retrieves the highest priority.

Programming the MEM_STALL_REG with values other than 0 enables the second scheme.

13.3 Programming

Programmability is required for the range (Start and Stop address) of segments that need to be defined in the memory depending on the type and size of the application. The following list contains the least number of segments needed for proper operation of the system.

Table 84: Memory Segments Description

Segment Name	Description	Access	Start/Stop Address Registers
CMAC stack	Controller stack code and temporary variables	Cortex-M0+ (read, write) Cortex-M33 (write) DMA (write)	CMI_CODE_BASE_REG (Note 1) CMI_DATA_BASE_REG CMI_END_REG

Segment Name	Description	Access	Start/Stop Address Registers
Mailbox	Implements the structure which is used for exchanging commands and data between the 2 CPUs	Cortex-M0+ (read, write) Cortex-M33 (read, write)	
Cortex-M33 Code	Special code segment for SysCPU. Hardware Patching code can be placed here as well	Cortex-M33 (read, write)	
Cortex-M33 Data	Application data space	Cortex-M33 (read, write) GP DMA (read, write)	
System	IVT and others	0x00000	0x000800

Note 1 This value should also be programmed in the Cortex-M33 MPU region 1 to ensure the Application does not corrupt CMAC code/data.

14 Embedded Flash Controller

14.1 Introduction

The DA14592 comprises an Embedded Flash Controller (FCU) that is responsible for accessing the Embedded Flash Memory (eFlash) allowing Read, Write and Erase accesses. The FCU resides in the power domain PD_MEM.

The Embedded Flash (eFlash) memory is a 2 Mb on-chip Flash memory with a minimum endurance of 10000 program and erase cycles for a single bit and minimum data retention of ten years at 85 °C.

Features

- Supports flexible memory allocation per master to avoid fragmentation.
- Parallelizes data flows from/to various masters on the system.
- Generates wait cycles to AHB masters.

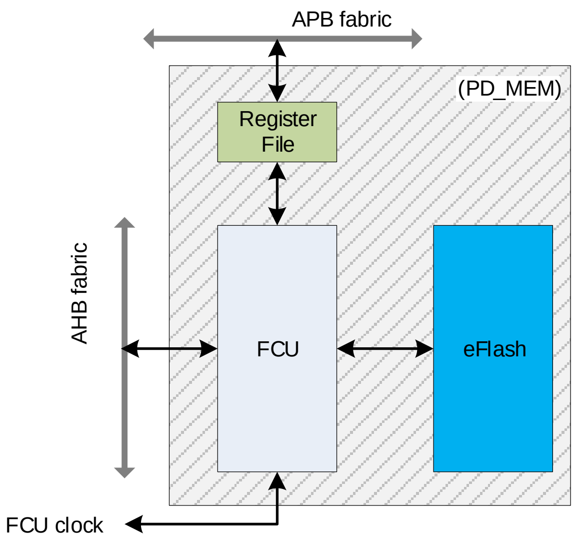

Figure 26. Embedded flash controller block diagram

14.2 Architecture

As shown in Figure 27, eFlash comprises of:

- One block (main block) of 32768 x 64-bits words, divided in 128 x 2 kB (2048) pages.
- The Info Page (1 page = 2048 bytes), mapped at the end of the main block.

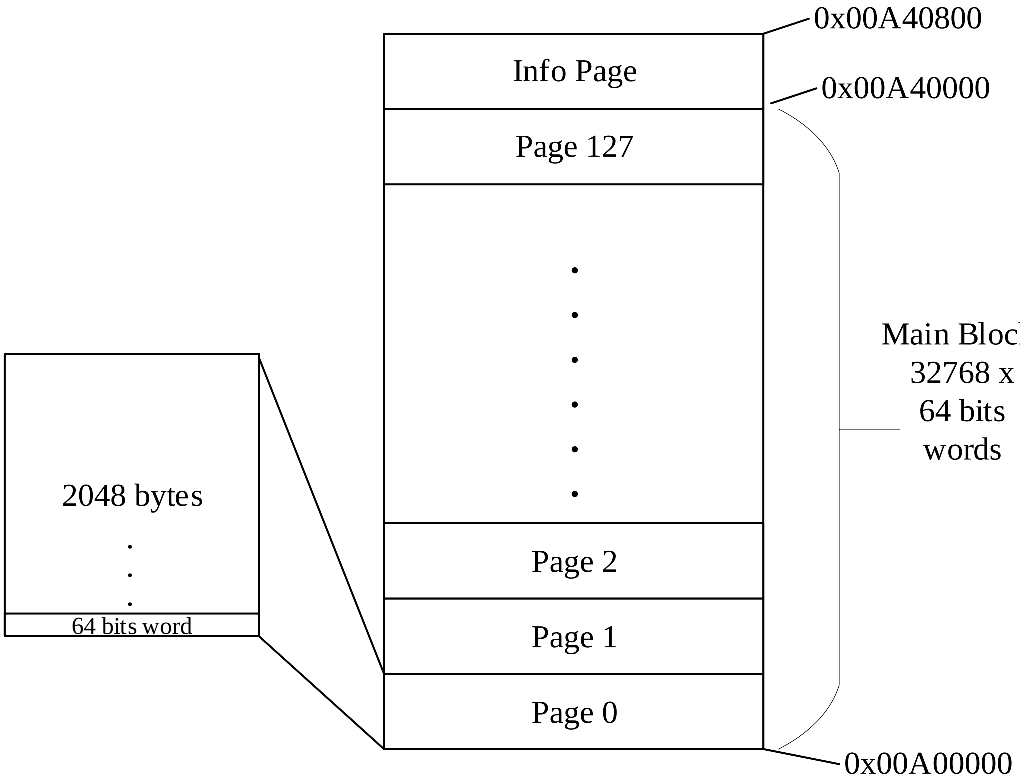

Figure 27. eFlash organization

The FCU is part of the power domain PD_MEM and has two power supply inputs supplied by VBAT rail and VDD rail. The minimum supported voltage level in VBAT is 1.8 V and in VDD 0.9 V. For write and erase mode, eFlash requires minimum VDD of 1.2 V.

14.2.1 Reading from eFlash

To enter the read mode the FLASH_CTRL_REG[PROG_SEL] and FLASH_CTRL_REG[PROG_MODE] bits must be cleared.

When the digital core (VDD) is running at 0.9 V, the worst case for eFlash access time is Tacc = 65 ns and when running at 1.2 V, the worst access time is Tacc = 23 ns. The reduced operational frequency of eFlash, in comparison to the M33/M0+ frequency (hclk), results in increased delay on read accesses (instruction fetching). For example, when eFlash operates at 16 MHz and M33/M0+ at 32 MHz, each read access costs two clock cycles (one wait cycle). The caches on M33 and M0+ reduce the number of read accesses from the CPUs to eFlash (only cache misses result in eFlash accesses). The number of eFlash wait cycles can be set in the FLASH_CTRL_REG[WAIT_CYCLES] bit field. Table 85 shows some use cases and the required eFlash wait cycles.

Table 85: Use Cases and eFlash Wait Cycles

hclk	VDD	eFlash Wait Cycles
64 MHz	1.2 V	1
32 MHz	0.9 V	2 (Note 1)
32 MHz	1.2 V	0
16 MHz	0.9 V	1
16 MHz	1.2 V	0
<16 MHz		0

Note 1 Default conditions on power up.

14.2.1.1 Instruction access buffering

eFlash supports 64-bit word width to reduce the delay caused by the wait cycles, while the instructions are 32-bit wide. A buffering mechanism (IAB) in the FCU is introduced to result in reduced delays on each cache-miss (assuming a high probability on sequential instruction reads from the processors – high code locality). The 32 MSBs or 32 LSBs of each eFlash word (only for instruction accesses) is stored in a register, depending on the requested address, for the next processor instruction request. If the next request is the other half of the same 64-bit eFlash word, then the FCU does not access eFlash but provides the requested word from the register (buffer) at the next clock cycle. This results in reduced delays on instruction reading for both CPUs and lower power consumption for the system. The IAB functionality is programmable by software for each of the processors.

For non-instruction accesses, the FCU accesses eFlash for each read request and selects the 32 MSBs or 32 LSBs of the 64-bit eFlash word based on the requested address. The rest 32 MSBs or 32 LSBs (second half) are discarded.

14.2.2 Writing to eFlash

To enter the write mode, the FLASH_CTRL_REG[PROG_SEL] bit must be set and FLASH_CTRL_REG[PROG_MODE] = 0x01.

The 64-bits wide eFlash can be programmed in groups of 32-bits words, either 32 MSBs or 32 LSBs. When programming the 32 MSBs, the FCU sets the 32 LSBs to 0xFFFFFFFF. That does not affect the eFlash content. When programming the 32 LSBs, the FCU sets the 32 MSBs to 0xFFFFFFFF.

Every time a write access is completed, the FCU generates an IRQ signal that can be cleared by writing to the FLASH_CTRL_REG[IRQ_CLEAR].

A successful data write is only possible when the address written is previously erased.

14.2.3 Erasing the eFlash

To enter the erase mode, the FLASH_CTRL_REG[PROG_SEL] bit must be set and FLASH_CTRL_REG[PROG_MODE] = 0x02 (Page Erase) or 0x03 (Mass Erase).

Erasing eFlash can be started after writing a dummy data to any address that is part of the desired page or block to be erased. If page erase is selected in PROG_MODE, the page containing the address is erased. In case of a mass erase, if the dummy word address belongs to the information block, both information and main block are mass erased. If the dummy word address belongs to the main block, only the main block is mass erased.

Every time an erase access is completed, the FCU generates an IRQ signal that can be cleared by writing to the FLASH_CTRL_REG[IRQ_CLEAR].

14.2.3.1 Suspend erase/resume

The FCU supports automatic suspend on page erase command, when a read access occurs on a different page than the one it is being erased. This functionality is enabled by ERASE_SUSPEND_EN bit field of FLASH_CTRL_REG.

The FLASH_PTERASE_SEG_REG holds the minimum erase time segment (register value is in clock cycles of the FCU clock with reset value 1000 – 1 ms. FCU clock has 1 μs period). When a page erase command is issued, the FCU initializes the (internal) erase segment counter with the FLASH_PTERASE_SEG_REG value and the (internal) total erase counter with the FLASH_PTERASE_REG value.

Note
The FCU must be allowed at least 1 period (1 µs) from the start of the erase (FLASH_CTRL_REG[PROG_ERS] = 1) until it can be suspended for a read access (FLASH_CTRL_REG[PROG_ERS] = 1 and FLASH_CTRL_REG[PROG_RMIN] = 1). Before that, all reads do not suspend the FCU erase.

Two modes of suspend erase/resume are supported. Those can be selected through ERASE_SUSPEND_MODE bit field of FLASH_CTRL_REG:

- Mode 0: When an erase command is issued, the FCU starts counting down with the erase segment counter. As soon as a read command is issued on a different page, the FCU suspends the erase operation. If the erase segment counter reaches the value of 0 before the suspend, the FLASH_PTERASE_SEG_REG value is subtracted from the total erase counter and the erase segment counter is reinitialized to the FLASH_PTERASE_SEG_REG value. Then it continues to count down. If a suspend operation (read command while erasing another page) occurs before the erase segment counter's value becomes zero, the current erase time segment (value of FLASH_PTERASE_SEG_REG) is not subtracted from the total erase counter and the erase segment counter is not reinitialized with the FLASH_PTERASE_SEG_REG value before the resume of the erase operation. A page is assumed to be erased if the value of the total erase counter reaches 0.
  - If the FLASH_PTERASE_SEG_REG value is equal to the FLASH_PTERASE_REG value, the erase process starts over after a read operation is issued.
  - If the FLASH_PTERASE_SEG_REG value is 1, the erase process resumes from the exact point (same total erase counter value) it was suspended.
- Mode 1: The same principles with Mode 0 stand, thus, if a read operation is issued (while erasing another page), the FCU waits (stalling the read) until the segment erase counter reaches the value 0 and then it suspends the erase operation.

If the erase is automatically suspended, it is required to be manually resumed by writing 1 to the ERASE_RESUME register field of FLASH_CTRL_REG.
FLASH_CTRL_REG[ERASE_SUSPEND_STAT] is used to check if the erase (is suspended and) needs to be resumed. Software is responsible to keep track of the suspend erase/resume procedure, by checking the ERASE_SUSPEND_STAT.

The FLASH_RTERASE_TOT_CNT_REG holds the total erase time that has passed with respect to the FLASH_PTERASE_REG.

The FLASH_RTERASE_SEG_CNT_REG holds the erase segment time that has passed with respect to the FLASH_PTERASE_SEG_REG.

All the read accesses in the page that an erase command is running result in AHB error, without suspending the erase process. When the erase suspend functionality is disabled by the software all the read accesses during a page erase result in an AHB error.

14.2.4 Standby/Sleep mode

The FCU comprises a state machine that sets eFlash in Sleep or Standby mode. In Sleep mode, the eFlash block dissipates lower leakage current. When the FCU is powered on, the state machine is in the SLEEP state and eFlash is in Sleep mode.

The state machine automatically switches from Sleep to Standby mode when an AHB read access is detected. The read access is delayed for the duration eFlash needs to wake-up. The wake-up time can be programmed in the FLASH_PTWK_SP_REG and has a default value of 3 (counted in µs). The FLASH_CTRL_REG[SLEEP_MODE] shows the sleep status of eFlash. If this bit is set, eFlash block is in Sleep mode, while if it is cleared, it is in Standby mode.

To perform write or erase, eFlash needs to be in Standby mode. To switch eFlash from Standby to Sleep mode, the FLASH_CTRL_REG[SLEEP_MODE] bit must be set.

The FCU can be safely powered off after the state machine is in the SLEEP state.

Before entering the Sleep mode, FLASH_CTRL_REG[PROG_MODE] bit must be cleared, otherwise a bus error occurs when a read access tries to wake up the eFlash.

14.2.5 Special considerations

The value of the FLASH_CTRL_REG should not be changed by the software before any FCU operation is finished. The software should check that the status FLASH_CTRL_REG bits PROG_ERS, PROG_WRS, PROG_RMIN, and SLEEP are all 0 before changing the values of the registers PROG_MODE and PROG_SEL. If the above requirements are not followed, the behavior of the FCU is unpredictable.

Moreover, the software must access the status registers at least once after changing any of the values of the programming registers. This is necessary to allow the time for the APB request to change the registers' values before an AHB request to reach the FCU occurs. In any other case, an error response would be generated.

For power saving reasons, it is recommended that the FCU is set to Sleep mode before the Arm Cortex-M33 enters the Sleep mode (call of WFI()) in case the PDC gates the shutting down of the PD_MEM power domain due to CMAC activity.

15 Clock Generation

15.1.1 Clock tree

Figure 28 shows the generation of the system's clocks in detail.

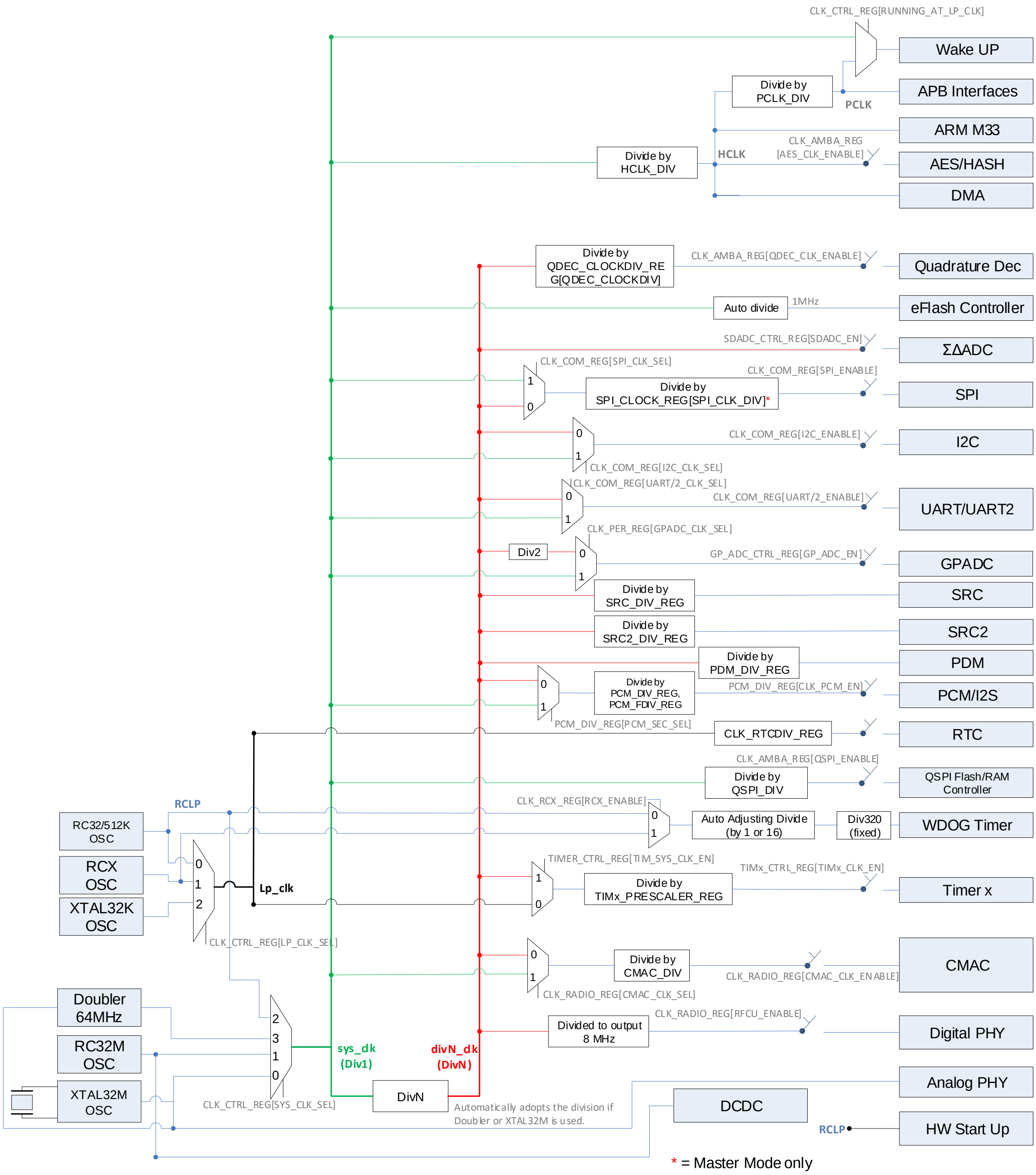

Figure 28. Clock tree diagram

The diagram shows the possible clock sources and all different divisions and multiplexing paths towards the generation of each block's clock. Also, the required registers that must be programmed, are also labelled on the same diagram. There are some main clock lines which are of interest:

- lp_clk (black bold line): this is the low power clock used for the sleep modes and can only be the RCX, RC32/512K(RCLP) or XTAL32K.
- sys_clk (green line): this is the system clock, – used for the AMBA clock (hclk) – which runs the CPU, memories, and the bus. The source of this clock can be the XTAL32M, the RC32M, the Doubler, the RCLP, or even an externally supplied digital clock.
- divn_clk (red line): this is a clock which automatically adjusts the division factor on the sys_clk to always generate 32 MHz. This enables the dynamic activation of the Doubler to provide more processing power at the CPU, without affecting the operation of blocks designed for 32 MHz.

15.2 Crystal oscillators

The Digital Controlled Xtal Oscillators (DXCO) are designed for low power consumption and high stability. There are two such crystal oscillators in the system, one at 32 MHz (XTAL32M) 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 32 MHz oscillator can be trimmed.

Figure 29 shows the principal schematic of the two oscillators. No external components to the DA14592 are required other than the crystal itself. If the crystal has a case connection, it is advised to connect the case to ground.

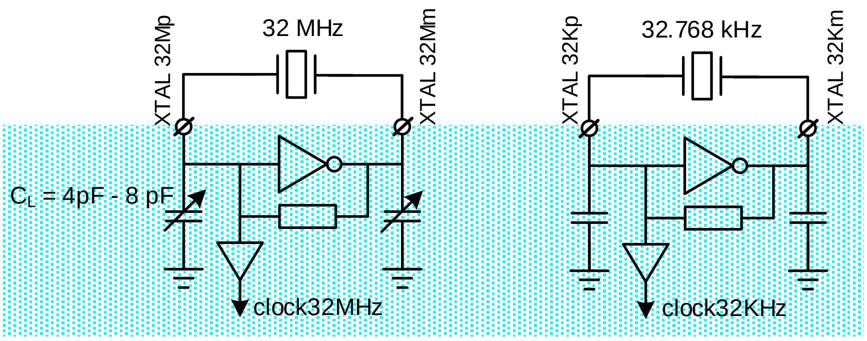

Figure 29. Crystal oscillator circuits

Note
The option to connect an XTAL32kHz crystal oscillator is only available in the FCQFN52 variant of the device.

15.2.1 Frequency control (32 MHZ crystal)

The register XTAL32M_TRIM_REG controls the trimming of the 32 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 the variable capacitor banks varies from the minimum to the maximum value in steps based on the bits selected on XTAL32M_TRIM_REG[XTAL32M_TRIM]. With XTAL32M_TRIM_REG[XTAL32M_TRIM] = 0x00, the minimum capacitance and thus the maximum frequency are selected. With XTAL32M_TRIM_REG [XTAL32M_TRIM] = 0xFF, the maximum capacitance and thus the minimum frequency are selected.

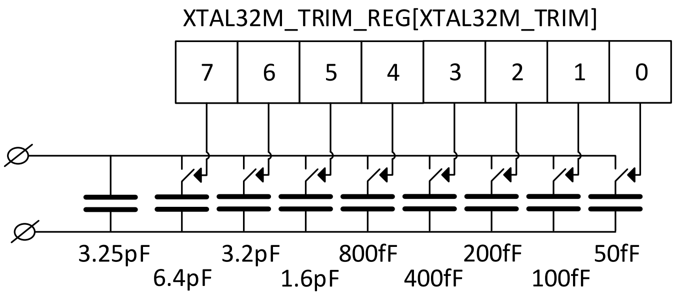

Figure 30. XTAL32MHz oscillator frequency trimming

15.3 RC oscillators

There are three RC oscillators in DA14592:

- One providing 32 MHz (RC32M)
- One providing 32 kHz and 512 kHz (RC32K/512K)
- One providing a frequency of 15 kHz (RCX).

The RC32M is powered by V_{BAT} which is available during Active or Sleep mode. The output clock is slower than 32 MHz if untrimmed and it is used to clock the CPU and the digital part of the chip during power-up or wake-up, while the XTAL32M oscillator is settling.

The simple RCLP oscillator (RC32K/512K) operates on V_{BAT} and provides 32 kHz or 512 kHz. The main usage of the RC32K/512K oscillator is for internal clocking during power-up or start-up. It clocks the hardware FSM that brings up the power management system of the chip. In the power-up or start-up sequence, the clock dynamically changes from 32 kHz to 512 kHz to speed up the sequence.

The enhanced RC oscillator (RCX) provides a stable 15 kHz frequency and operates on V_{BAT} that is available during Active or Sleep mode. The RCX oscillator is the main low power clock (lp_clk) and replaces the 32.768 kHz crystal, since it has a precision of < 500 ppm, while its output frequency needs to be recalibrated over temperature.

15.3.1 Frequency calibration

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

The measurement procedure is as follows:

1. REF_CNT_VAL = N (the higher N, the more accurate and longer the calibration is).
2. CLK_REF_SEL_REG[REF_CLK_SEL] = 0 (RC32/512K) or
   CLK_REF_SEL_REG[REF_CLK_SEL] = 1 (RC32M) or
   CLK_REF_SEL_REG[REF_CLK_SEL] = 2 (XTAL32K) or
   CLK_REF_SEL_REG[REF_CLK_SEL] = 3 (RCX).
3. Start the calibration: CLK_REF_SEL_REG[REF_CAL_START] = 1.
4. Wait until CLK_REF_SEL_REG[REF_CAL_START] = 0.
5. Read CLK_REF_VAL_REG = M (32-bits value).
6. Frequency = (N/M) * 32 MHz.

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

15.4 Doubler

The low power Doubler multiplies (doubles) the XTAL32M clock to produce a 64 MHz clock with very high precision.

Changing the system's clock into the Doubler output can be done dynamically without affecting the operation of the device. Its main purpose is to provide more processing power to the CPU for computational hungry applications.

The doubler is powered by the LDO_XTAL and the output duty cycle is 45% to 55% of the input clock over temperature.

When doubler is enabled, VDD should be set to 1.2 V.

16 Quad SPI RAM/Flash Controller

16.1 Introduction

The Quad SPI Controller (QSPIC) provides a low pin count interface to serial QSPI Flash or PSRAM memory devices. The QSPIC supports the standard Serial Peripheral Interface (SPI) and a high performance Dual/Quad SPI Interface. The QSPI RAM feature provides a low-cost RAM extension for infrequently used data.

The QSPIC automatically generates all the control signals for the QSPI bus needed to access data from the serial quad memory. The controller has a vendor independent register file that provides a rich set of control fields for supporting a wide range of Flash and RAM devices.

The QSPIC provides memory mapped Execute-In-Place (XIP) as well as transparent memory mapped RAM access.

Features

- SPI modes:
  - Single: Data transfer through two unidirectional pins
  - Dual: Data transfer through two bidirectional pins
  - Quad: Data transfer through four bidirectional pins.
- Auto mode: up to 32 MB memory mapped Read/Write Data access with 3-byte and 4-byte addressing modes.
- Manual mode: Direct register access using the QSPIC register file.
- QSPI clock up-to 64 MHz. Clock modes 0 and 3. Master mode only.
- Vendor independent Instruction Sequencer.
- In Auto mode the Flash control signals are fully programmable.
- Use of a special read instruction in the case of a specific (programmable) wrapping burst access.
- Erase suspend/resume to Support for Code and Data storage.

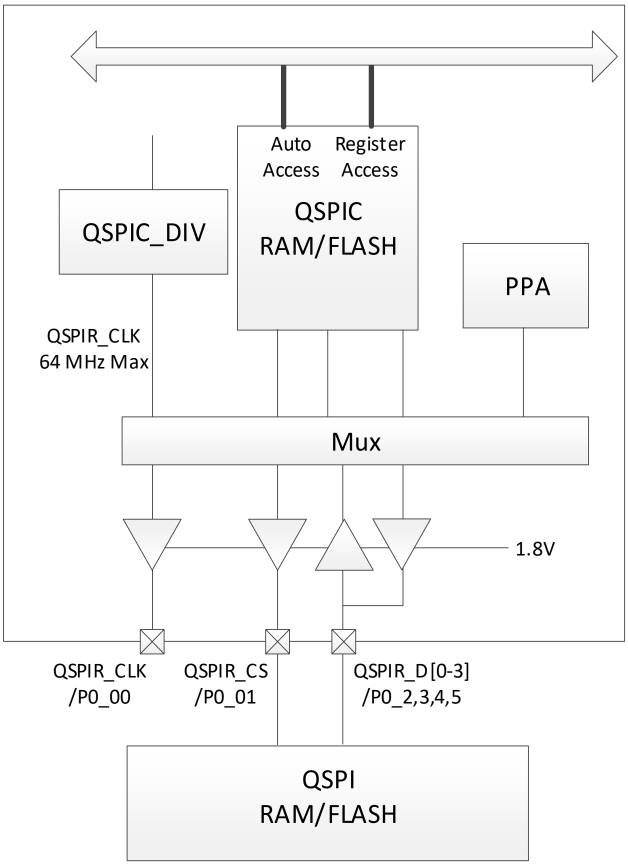

Figure 31. Quad SPI RAM/Flash controller

16.2 Architecture

16.2.1 Interface

- QSPI_SCK: output serial clock
- QSPI_CS: Active Low output Chip select
- QSPI_IO0:
  - DO (output) in Single SPI mode
  - IO0 (bidirectional) in Dual/Quad SPI mode
- QSPI_IO1:
  - DI (input) in Standard SPI mode
  - IO1 (bidirectional) in Dual/Quad SPI mode
- QSPI_IO2:
  - General purpose (output) (for example, WPn Write Protect) in Standard SPI mode
  - IO2 (bidirectional) in Quad SPI mode
- QSPI_IO3:
  - General purpose (output) (for example, HOLDn) in Single SPI mode
  - IO3 (bidirectional) at Quad SPI mode.

16.2.2 SPI modes

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

- Single: Data transfer through two unidirectional pins. The QSPIC supports communication to any single/dual or Quad SPI Flash memory. In contradiction to the Standard SPI interface, the supported Single SPI interface does not support the bus modes 1 and 2, does not support full-duplex communications and does not support any SPI slave mode.
- Dual: Data transfer through two bidirectional pins.
- Quad: Data transfer through four bidirectional pins.

16.2.3 Access modes

You can access a serial memory (Flash or RAM) connected to the QSPIC in one of the modes:

- Auto mode
- Manual mode.

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

In Auto mode, 3-byte and 4-byte addressing modes are supported. With QSPIC_USE_32BA = 0, up to 16 MB serial memory (3-byte addressing) can be accessed. If QSPIC_USE_32BA = 1, the 4-byte addressing is enabled for accessing up to 32 MB serial memory.

Auto mode Flash access

In auto mode (QSPIC_AUTO_MD = 1), the read access to a serial Flash memory is performed in a fully transparent way through the SPI bus. A read access to the memory space, where the external memory is mapped, is translated by the controller to the respective SPI bus command sequence, which is needed for retrieving of the requested data from the serial Flash memory.

When the auto mode is disabled (QSPIC_AUTO_MD = 0), any access (read or write) to the mapped memory space is ignored by the controller.

Only read accesses are supported when the connected external device is a Flash memory (QSPIC_SRAM_EN = 0). A write access causes a hard fault at the CPU.

The read access can be single access or incremental burst or wrapping burst access. The wrapping burst is supported even when the controlled serial Flash does not support any special instruction for wrapping burst. A special read instruction can be used in the case of a specific (programmable) wrapping burst access. When a serial Flash supports a special instruction for wrapping burst access, this feature saves access time (less wait states). For maximizing the utilization of the bus and minimizing the number of wait states, it is recommended to use burst accesses. However, non-sequential random accesses are still supported at cost of more wait states.

Auto mode RAM access

If connected to a serial RAM device, QSPIC can provide both read and write functionality.

The configuration register must be programmed to enable the RAM functionality (QSPIC_SRAM_EN = 1). As in the case where the external device is a Flash, the auto mode must also be enabled (QSPIC_AUTO_MD = 1). In the case where the auto mode is disabled (QSPIC_AUTO_MD = 0), any access (read or write) in the memory space, where the external device has been mapped, is ignored by the QSPI controller.

The read access in the memory space of the external serial RAM memory, is done in a fully transparent way through the QSPI bus. The capability of the controller to handle the various types of read accesses, is the same as in the case of the Flash device. Single access, incremental burst or wrapping burst are all supported.

A write access to the memory space where the external memory is mapped, does not cause a hard fault to the CPU. On the contrary, a write access is interpreted by the QSPIC in the respective QSPI bus protocol and write data is stored in the external RAM device.

The controller is capable of handling write accesses of all kinds of burst: single access, incremental burst, or wrapping burst access. The throughput that can be achieved varies depending on the burst length, the word width, the protocol of the external memory device, and the frequency of the QSPI clock.

Burst access ensures the highest throughput. The non-sequential random accesses are supported at cost of more wait states. The maximum throughput that can be achieved depends on the burst length.

Manual mode

In manual mode, the external serial memory is controlled by a register file. All instructions that are supported by the serial memory can be programmed by using the register file. Moreover, the mode of interface (SPI, Dual SPI, Quad SPI) and the mode of operation (Auto or Manual mode) can be configured via this register file. The register file supports the following data sizes for reading and writing accesses: 8-bits, 16-bits, and 32 bits.

16.2.4 Endianness

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

16.2.5 Erase suspend/resume

A QSPI Flash memory can be used for data storage, combining the EEPROM functionality and Program storage in one single device.

For this purpose, QSPI ERASE/SUSPEND ERASE/RESUME are automatically executed as shown in Figure 32.

To store data in QSPI Flash memory, the sector designated for storage must be erased first.

The ERASE/SUSPEND ERASE/RESUME process is only meaningful if the external device is a serial Flash memory (QSPIC_SRAM_EN = 0).

Erase procedure

1. The controller is in Auto mode and the read requests are served. The Erase procedure is initiated by setting QSPIC_ERASE_EN to 1. The address of the sector, which is erased, is defined by QSPIC_ERS_ADDR. When an Erase procedure is requested, the controller jumps to state 2.
2. The read requests are still served. As soon as the read requests stop (also possible due to late bus master change, for example, DMA) and there is no new read request for a number of AHB clock cycles equal to QSPIC_ERSRES_HLD, the QSPIC_WEN_INST and the QSPIC_ERS_INST instructions are automatically sent to QSPI Flash. The QSPIC_RESSUS_DLY counter is started. After this, the controller jumps to state 3.
3. The erasing is in progress in the QSPI Flash memory. The QSPI controller waits for one of the following:
    - A status check request. This request can be forced by writing QSPIC_CHCKERASE_REG. This makes the QSPIC controller read the Flash memory status and checks the end of *erasing*. Reading the status is delayed by QSPIC_RESSTS_DLY cycles, or by QSPIC_RESSUS_DLY cycles. The QSPIC_RESSTS_DLY delay is based on the SPI bus clock, while the QSPIC_RESSUS_DLY delay is based on a 222 kHz clock. The selection between the two delays is configured with the help of the QSPIC_STSDLY_SEL bit. After erasing, the QSPI controller returns to the normal operation (state 1) and sets QSPIC_ERASE_EN = 0, otherwise it remains in state 3.
    - A read data request on the AHB bus. The QSPI controller reads the status of the Flash memory and checks the end of erasing. Status reading is delayed again by QSPIC_RESSTS_DLY cycles, or QSPIC_RESSUS_DLY cycles. The QSPIC_STSDLY_SEL bit does the selection between two delays. At the end of erasing, the controller returns to

normal operation (state 1) and sets QSPIC_ERASE_EN to 0. The read request is served in state 1. If erasing has not ended, the controller proceeds to state 4.

4. The QSPIC_SUS_INST is sent as soon as the QSPIC_RESSUS_DLY counter is 0. The controller enters state 5.
5. The controller reads the status register until the Flash device is ready (erasing is suspended). When the Flash device is ready, the controller enters state 6.
6. Erasing process in Flash is suspended and the controller can read the Flash. The requested data is retrieved from the Flash device. If reading on the AHB stops (for example, Cache hit) and there is no new read request for a number of AHB clock cycles equal to QSPIC_ERSRES_HLD, the controller enters state 7.
7. The QSPIC_RES_INST instruction is applied and state 3 is entered. Also, the QSPIC_RESSUS_DLY counter is started. The QSPIC_RES_INST makes sure the erase procedure is continued (erase resume) in the flash device.

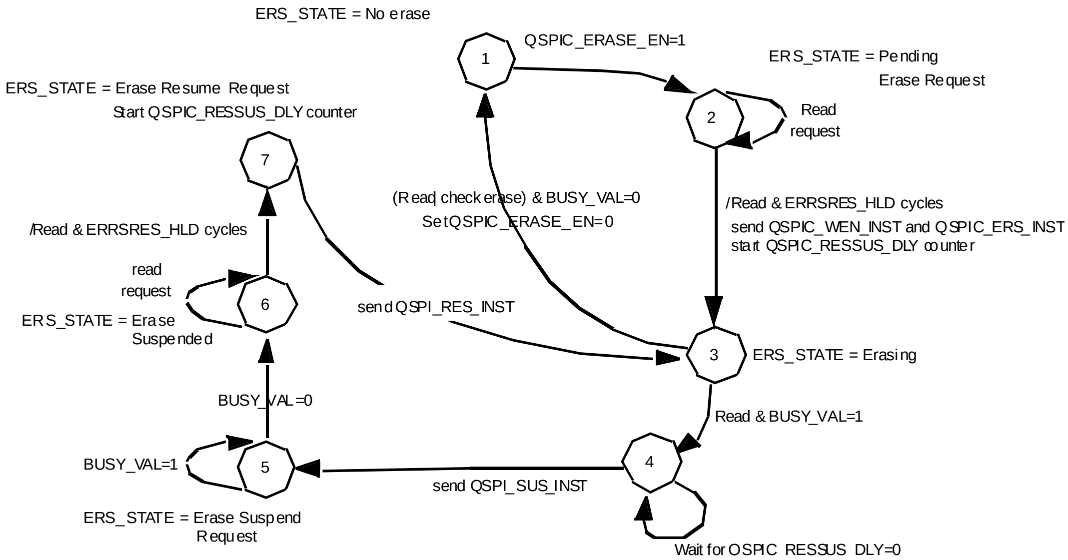

Figure 32. Erase suspend/resume in auto mode

NOTE
QSPI_RESSTS_DLY is counted with the QSPI_CLK, so before changing the QSPI_CLK, make sure that QSPI_RESSTS_DLY is set large enough to meet the timing parameter requirements

QSPI Flash programming procedure

Sectors are programmed in manual mode by polling the status bit in QSPI Flash. During programming, the CPU MUST execute code from shared or non-shared RAM. Also, interrupts must be disabled while executing the write command to the Flash.

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

16.2.6 Low power considerations

To reduce the power dissipation in QSPI Flash, the QSPI_CLK must always be the highest possible system clock to keep the burst access to Flash as short as possible. The CPU must run as slow as possible for minimum power.

For lowest power with slow CPU (for example, 2 MHz) and high QSPI_CLK (for example, 32 MHz) bit QSPIC_CTRLMODE_REG[QSPIC_FORCENSEQ_EN] must be set to 1. This enables split burst mode, reducing the power dissipation during active burst only, while disabling the Flash when the burst is done compared to high efficiency burst. These two modes are explained in Figure 33 and Figure 34.

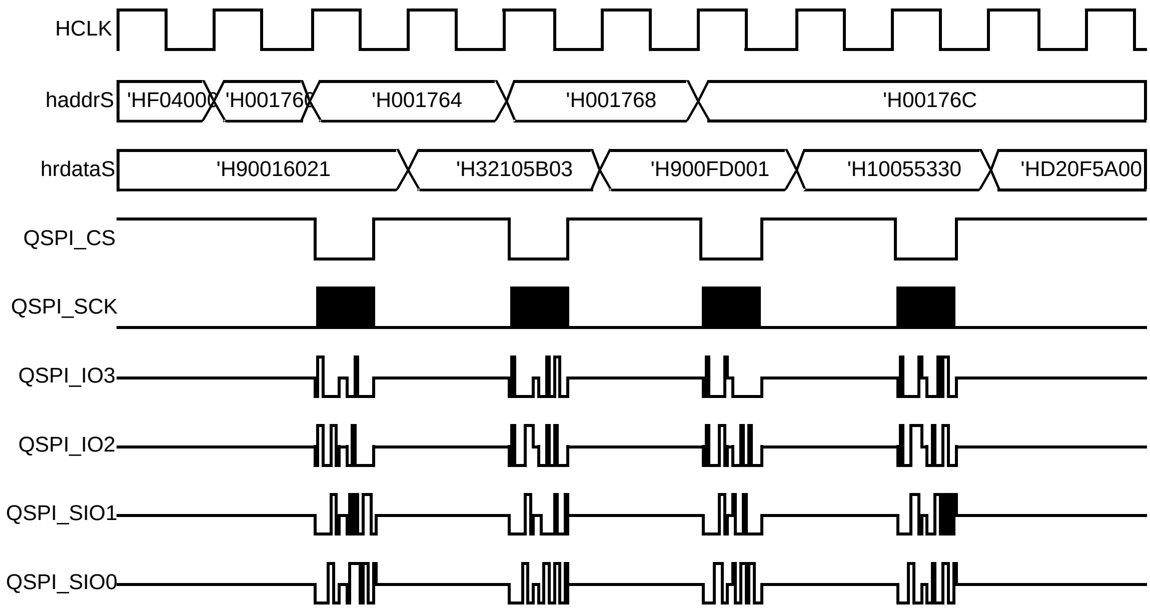

Figure 33. QSPI split burst timing for low power (QSPI_FORENSEQ_EN = 1)

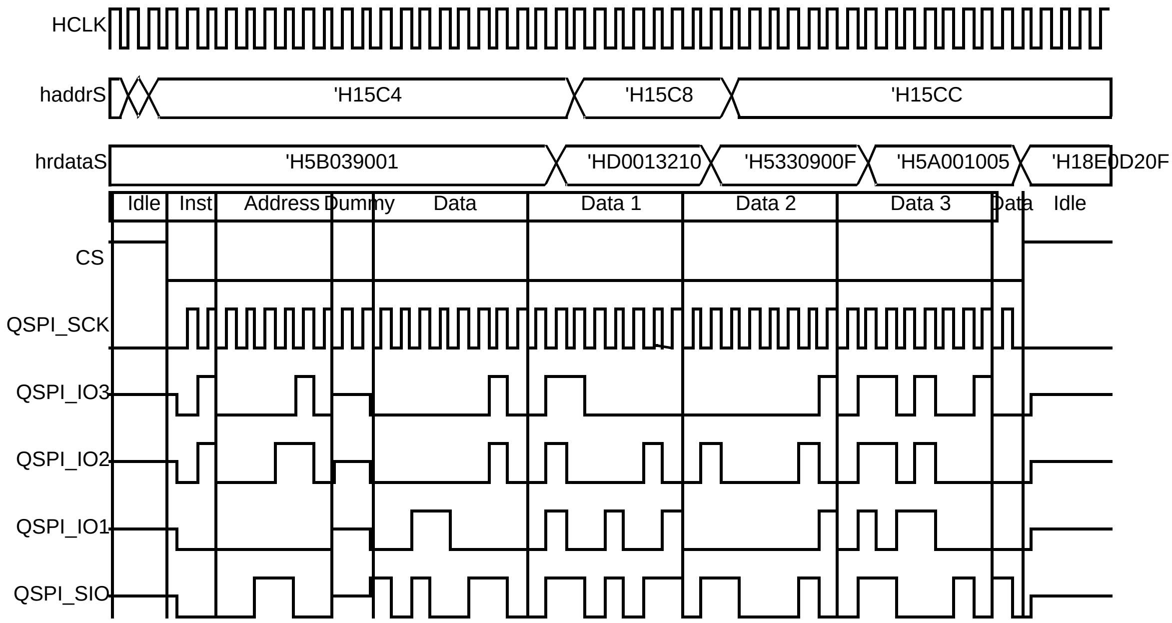

Figure 34. QSPI burst timing for high performance (QSPI_FORENSEQ_EN = 0)

If QSPI_FORCENSE_EN = 0, the QSPIC reads data in a burst address1/extra/dummy/data1, data2, dataN, keeping the QSPI_CS low during the complete burst. If set to 1, the burst is split into non-sequential accesses address1/extra/dummy/data1, address2/extra/dummy/data2, and so on making the QSPI_CS high between the accesses.

16.3 Programming

16.3.1 Auto mode

Chip selection

In auto mode, QSPI executes from address 0. See Arm chapter remap function.

Notice that certain PSRAM RAM (for example, APmemory APS3204J) connected to the QSPI interface have a minimum time t_{CEM} that the #CE may stay low. Software must make sure that the QSPI CLK is not going too slow during a burst, otherwise RAM data might get lost.

Read burst in auto mode

In the case of a read from the external device, in Auto mode, the QSPI controller generates a sequence of control signals. This is analyzed in the following phases: instruction phase, address phase, extra byte phase, dummy clocks phase, and read data phase. These phases can be programmed through the QSPIC_BURSTCMDA_REG and QSPIC_BURSTCMDB_REG registers.

The QSPIC_INST bits are used to set the selected instruction for the cases of incremental burst or single read access. If the QSPIC_WRAP_MD bit is equal to 1, the QSPIC_INST_WB bits can be used to set the used instruction for a wrapping burst read access. The length and size is described by the QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE bits respectively. In all other cases, QSPIC_INST is the selected instruction.

If instruction is to be transmitted only during the first access after the selection of Auto mode, QSPIC_INST_MD must be equal to 1.

To enable the extra byte phase, set 1 to the QSPIC_EXT_BYTE_EN register. The transmitted byte during the extra byte phase is specified from the QSPIC_EXT_BYTE register. To disable (hi-z) the output pads during the transmission of bits [3:0] of extra byte, write 1 to the QSPIC_EXT_HF_DS register.

The number of dummy bytes during the dummy clocks phase is specified at QSPIC_DMY_NUM.

The SPI BUS mode during each phase can be configured as follows:

- QSPIC_INST_TX_MD for the instruction phase
- QSPIC_ADR_TX_MD for the address phase
- QSPIC_EXT_TX_MD for the extra byte phase
- QSPIC_DMY_TX_MD for the dummy byte phase
- QSPIC_DAT_RX_MD for the read data phase.

If the Quad SPI mode is selected in any of the above phases, write 0 to QSPIC_IO3_OEN and QSPIC_IO2_OEN.

If the serial Flash ьemory must be prepared for reading with the use of any instruction except the read instruction, then the Manual mode must be used for the programming of the above instructions.

The final step to enable the use of Auto mode of operation is to set QSPIC_AUTO_MD equal to 1.

Write bursts in auto mode

In the case where the connected memory is a serial PSRAM, the controller can serve requests for write accesses. This is implemented, as in the case of the read burst, when the Auto mode of operation is active (QSPIC_AUTO_MD = 1). Additionally, the external device must be declared to the QSPI controller as a serial RAM (QSPIC_SRAM_EN = 1).

Under these conditions the QSPI controller generates a sequence of control signals in SPI BUS, for each request for write burst access towards the external device. This sequence of control signals is analyzed in the following phases: instruction phase, address phase, extra byte phase, and write data phase. These phases can be programmed through the QSPIC_AWRITECMD_REG register.

The QSPIC_WR_INST bits are used to define the write instruction. This instruction is used for all the cases of bursts: single access, incremental burst, or wrapping burst. The controller handles them accordingly to implement all the lengths of the bursts.

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

- QSPIC_WR_INST_TX_MD for the instruction phase.
- QSPIC_WR_ADR_TX_MD for the address phase.
- QSPIC_WR_DAT_TX_MD for the read data phase.

If the serial RAM has to be configured with a special command sequence prior to the write instruction, manual mode must be used.

16.3.2 Manual mode

To enable Manual mode, QSPIC_AUTO_MD must be equal to zero. Manual operation of the bus signal is done through QSPIC_CTRLBUS_REG:

- Start/End of an access can be controlled using the QSPIC_EN_CS and QSPIC_DIS_CS bits respectively.
- SPI mode configured with the QSPIC_SET_SINGLE, QSPIC_SET_DUAL, and QSPIC_SET_QUAD bits.

Writing to the QSPIC_WRITEDATA register generates a data transfer from the QSPIC to the SPI bus. A read access at QSPIC_READDATA register generates a data transfer from the SPI bus. Writing to QSPIC_DUMMYDATA register generates a number of clock pulses to the SPI bus. During this activity in the SPI bus, the QSPI_IO data pads are in *hi-z* state.

When access to the SPI bus through QSPIC_WRITEDATA, QSPIC_READDATA, and QSPIC_DUMMYDATA is very slow, the delay on the internal AHB bus is very high. In this case, set the QSPIC_HRDY_MD register equal to 1. With this, the *hready* signal of the SB slave interface is always equal to 1, when accessing the WriteData, ReadData, and DummyData registers. All masters can access the AHB bus without waiting for transmission completion on SPI Bus. A read of the QSPIC_BUSY register must be done to check the end of the activity at the SPI bus, before any more accesses are triggered. In this case, the QSPIC_RECVDATA register contains the received data at the end of a read access.

16.3.3 Clock selection

The SPI clock mode is set with the QSPIC_CLK_MD bit. The supported modes for the generated SPI clock are:

- 0 = Mode 0. The QSPI_SCK is low, when the bus is idle (QSPI_CS is high).
- 1= Mode 3. The QSPI_SCK is high, when the bus is idle (QSPI_CS is high).

The QSPI_CLK frequency has a programmable divider CLK_AMBA_REG[QSPIC_DIV] which divides the system clock by 1, 2, 4, 8.

16.3.4 Received data

The standard method to sample the received data is by using the positive edge of the QSPI_SCK. However, when the output delay of the serial Flash is high, timing issues at the read path are very likely. For this reason, the QSPIC can be programmed to sample the received data with the negative edge of the QSPI_SCK. This is specified with the QSPIC_RXD_NEG register.

Furthermore, the receive data can be pipelined by setting QSPI_RPIPE_EN = 1 and the sample clock can be delayed using QSPI_PCLK_MD. See the timing chapter for detailed QSPI Timing information.

16.3.5 Delay line configuration

When VDD = 0.9 V (POWER_LEVEL_REG[VDD_LEVEL_ACTIVE] = 0x0) and QSPI_CLK = 32 MHz, the QSPIC_CTRLMODE_REG[QSPIC_PCLK_MD] bit field should be equal to 2. On the contrary, if VDD = 1.2 V, then the QSPIC_CTRLMODE_REG[QSPIC_PCLK_MD] bit field should be equal to 7.

17 DMA Controller

17.1 Introduction

The DMA controller has six Direct Memory Access (DMA) channels for fast data transfers to and from SPI, UART/2, I2C, SRC, PCM, GP_ADC, SD_ADC, QSPI Flash and eFlash to and from any on-chip RAM. The DMA controller off-loads the Arm interrupt rate, if an interrupt is generated after a programmable amount of transfers. A number of peripheral requests are multiplexed on the six available channels, to increase utilization of the DMA. Figure 29 shows the block diagram of the DMA controller.

Features

- Six channels with optional peripheral trigger.
- Full 32-bit source and destination pointers.
- Flexible interrupt generation.
- Programmable transfer length.
- Flexible peripheral request per channel.
- Option to initialize memory.
- Programmable Edge-Sensitive request support.
- Programmable support of AHB burst reads/writes, supporting both Memory-to-Memory and Memory-to-Peripheral transfers.
  - Burst lengths supported are 8-beat (INCR8) and 4-beat (INCR4).
- Programmable bus error detection support and IRQ generation upon detection.
- FREEZE support also in on-going Memory-to-Memory transfers.
- Support of "secure transfer" mode by a dedicated, conditionally secure DMA channel (DMA5).

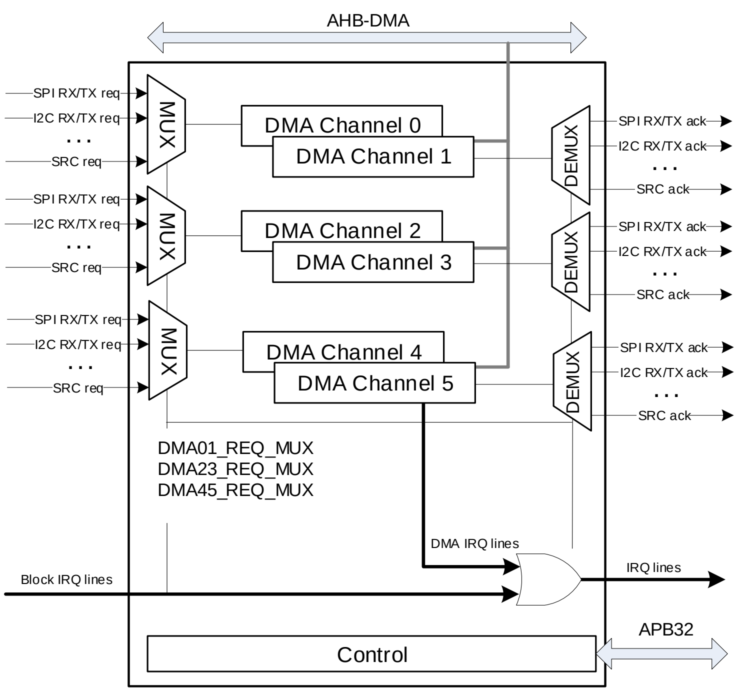

Figure 35. DMA controller block diagram

17.2 Architecture

17.2.1 DMA peripherals

By default, DMA assumes memory-to-memory transactions. Each DMA channel can also be connected with the hand-shaking signals or other request signals of the corresponding peripherals as shown in Table 86.

Table 86: DMA Served Peripherals

Block	Direction(s)	Supported Access Rate
SPI	RX and TX	Single
QSPI Flash	Read	Burst
eFlash	Read and Write	Burst (Read) – Single (Read/Write)
UART	RX and TX	Single/Burst
UART2	RX and TX	Single/Burst
I2C	RX and TX	Single/Burst
RAM	Read and Write	Burst
GPADC	Read	Single
SDADC	Read	Single
SRC1	Left and Right	Single/Burst (Left or Right)
SRC2	Left and Right	Single/Burst (Left or Right)
PCM	Left and Right	Single

17.2.2 Input/output multiplexer

The multiplexing of peripheral requests is controlled by DMA_REQ_MUX_REG. So, if DMA_REQ_MUX_REG[DMAxy_SEL] is set to a certain (non-reserved) value, the TX/RX request from the corresponding peripheral is routed to DMA channels x (TX request) and y (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 is given priority (see also the register's description).

17.2.3 DMA channel operation

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

If DREQ_MODE is 1, a DMA channel can be triggered by a hardware request coming from a selected peripheral. All DMA channels support either level (default) or edge-sensitive requests via the bit-field REQ_SENSE of DMAx_CTRL_REG (x = 0, 1, 2, 3). If this bit-field is set (recommended for Memory-to-UART/UART2 and Memory-to-I2C transfers), the channel detects a positive edge on the request signal of the selected peripheral to start up a new transfer cycle. The edge-sensitive requests can be used globally, if desired, for all the peripherals interfacing with the DMA.

When 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 eight, 16, or 32 bits wide. The address increment is realized with an internal 16-bit counter DMAx_IDX_REG, which is set to 0 when the DMA transfer starts and is compared with the DMAx_LEN_REG after each transfer. The register value is multiplied by the values of the automatic increment of source address (AINC), the automatic increment of destination address (BINC), and bus transfer width (BW) before it is added to DMAx_A_START_REG and DMAx_B_START_REG. AINC or BINC must be 0 for register access.

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 DREQ_MODE is 0 or if DREQ_MODE is set to 1 and CIRCULAR bit is not set.

If a hand shaking is attached to the specific DMA channel at the end of a DMA cycle, the channel is blocked for as long as the peripheral is not ready for the next transaction.

If the 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 always stops, regardless of the state of CIRCULAR.

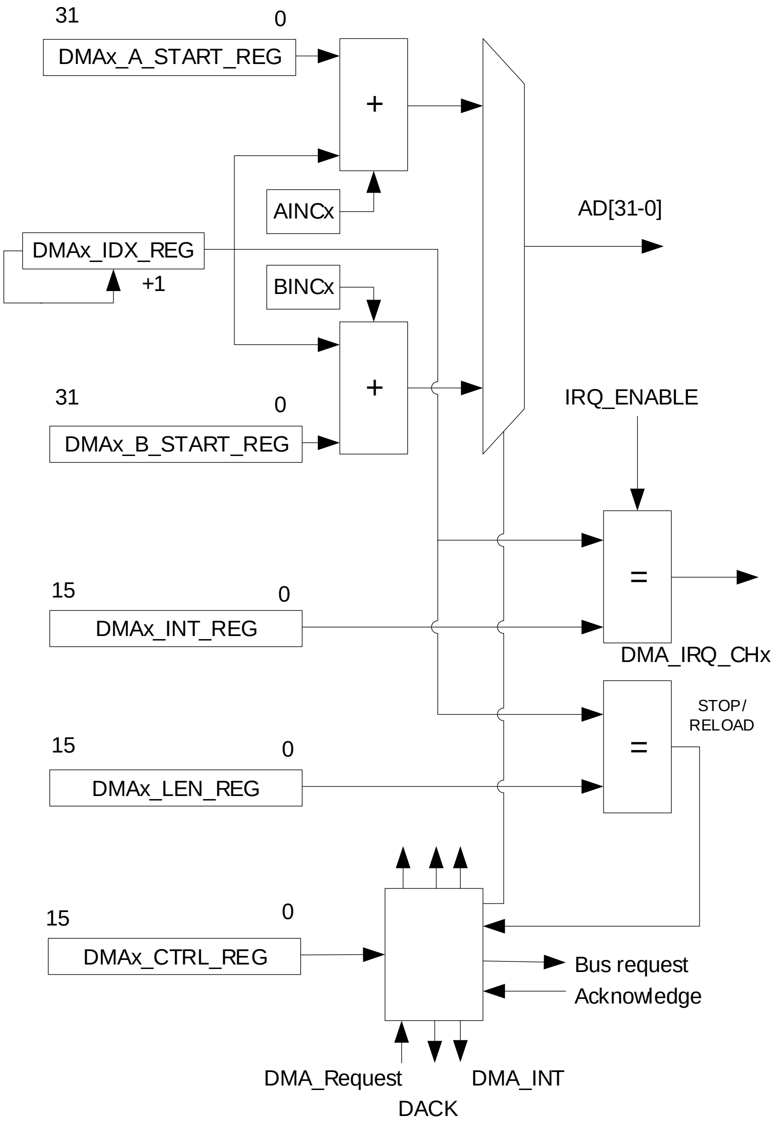

Figure 36. DMA channel diagram

Each DMA channel can generate an interrupt if the index counter DMAx_IDX_REG reaches the value of the channel's interrupt transfer length register, DMAx_INT_REG. After the transfer and before DMAx_IDX_REG is incremented, the interrupt is generated.

For example, if DMA_x_INT_REG = 0 and DMA_x_LEN_REG = 0, there is one transfer and an interrupt.

17.2.4 DMA arbitration

The priority level of a DMA channel can be set with bits DMA_PRIO[2:0]. These bits determine which DMA channel is 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 channel 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 is not changed during the transfer. However, it is not guaranteed that the DMA transfer is not interrupted by other channels of higher priority when these request access to the bus at the same time.

17.2.5 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[FRZ_DMA].

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

It is noted that the on-going Memory-to-Memory transfers (DREQ_MODE = 0) can also be interrupted (freeze).

17.2.6 Secure DMA channel

If the security configuration option (PROT_APP_KEY) is enabled in the configuration script, then DMA channel #5 becomes a secure channel. This channel is then only used to move keys from the Symmetric Key Area to the AES block for encryption/decryption without the CPU being able to intervene.

17.3 Programming

To configure the DMA Controller, do the following procedures.

17.3.1 Memory to memory transfer

1. Set the length of data to be transferred (DMAx_LEN_REG).
2. Set the source address (DMAx_A_START_REG).
3. Set the destination address (DMAx_B_START_REG).
4. Configure the number of transfers until an interrupt is generated (DMAx_INT_REG).
5. Enable the IRQ of the used DMA channel if needed (DMA_INT_MASK_REG).
6. Configure transfer options:
   a. DMAx_CTRL_REG[AINC]: Automatic increment of source address.
   b. DMAx_CTRL_REG[BINC]: Automatic increment of destination address.
   c. DMAx_CTRL_REG[BW]: Bus transfer width.
   d. DMAx_CTRL_REG[BURST_MODE]: Enable DMA read/write bursts, if needed.
7. Start the DMA transfer by setting the DMAx_CTRL_REG[DMA_ON] bit.
8. Wait until transfer is finished (DMAx_CTRL_REG[DMA_ON] = 0) or the corresponding interrupt.
9. Clear the IRQ if it was enabled (DMA_INT_STATUS_REG).

17.3.2 Peripheral to memory transfer

1. Set the length of data to be transferred (DMAx_LEN_REG).
2. Set the source address (DMAx_A_START_REG) equal to the peripheral RX register (for example, I2C_DATA_CMD_REG).
3. Set the destination address (DMAx_B_START_REG).
4. Configure the number of transfers until an interrupt is generated (DMAx_INT_REG).
5. Map the peripheral to the selected channels pair (DMA_REQ_MUX_REG[DMAxy_SEL]).
6. Configure transfer options:
   a. DMAx_CTRL_REG[AINC]: Disable automatic increment of source address.
   b. DMAx_CTRL_REG[BINC]: Automatic increment of destination address.
   c. DMAx_CTRL_REG[BW]: Bus transfer width.
   d. DMAx_CTRL_REG[DREQ_MODE]: Enable triggering by peripheral DMA request.
   e. DMAx_CTRL_REG[DMA_PRIO]: Set channel priority.
   f. DMAx_CTRL_REG[BURST_MODE]: Enable DMA read/write bursts, if needed.
   Note that SPI does not support burst transfers.
7. Enable the IRQ of the used DMA channel (DMA_INT_MASK_REG).
8. Start the DMA transfer by setting the DMAx_CTRL_REG[DMA_ON] bit.
9. Enable peripheral’s DMA request (for example, I2C_DMA_CR_REG[TDMAE]).

18 Crypto Engine

18.1 Introduction

The Crypto engine aims to accelerate the algorithm calculations that are needed to implement the RFC4835. It implements AES in ECB, CBC, and CTR modes. It also comprises HASH functions (SHA224/256). It supports AES128, AES192, AES 256 as well as HMAC-SHA-256 authentication protocol.

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

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

Features

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

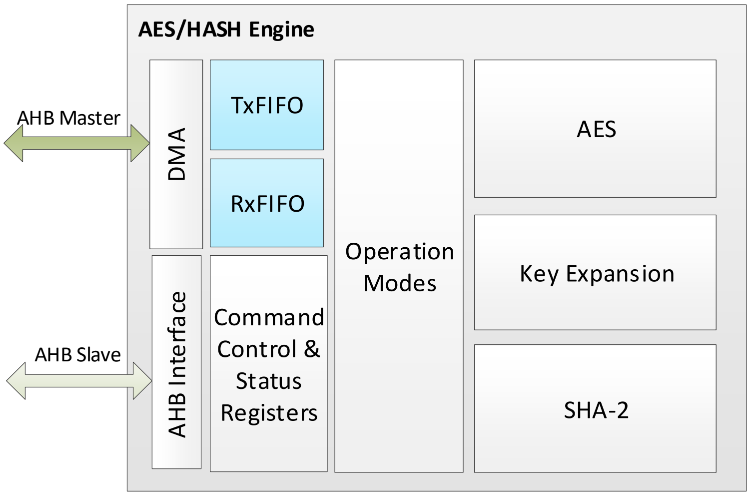

Figure 37. AES/HASH architecture

18.2 Architecture

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

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

The "Ctrl FSM" block checks the FIFO’s and DMA status continuously and decides for: the amount of data traffic, plus which of the FIFO’s are used. It also decides the "switching off" of the AES/HASH after transferring all results to the memory. The "HASH" block contains all the logic required for the realization of the hash algorithms calculations, as well as circuitry for the padding of data. It also contains glue logic for the transfer of the results to the "Modes" block.

18.2.1 AES

This part of the architecture implements the AES algorithm described in the AES-FIPS PUB 197. The capability it offers is the encryption and decryption of 128 bits data blocks by using 128-, 192-, or 256-bits encryption key.

18.2.2 Operation modes

The "Modes" block uses the AES to implement the following modes of operations:

- ECB (Electronic Code Book)
- CBC (Cipher Block Chaining
- CTR (Counter).

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

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

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

18.2.3 HASH

Padding at the input data is automatically applied as required by the hash algorithms. The purpose is to ensure that the message is a multiple of 512. Padding is done as follows. After the last data byte, one extra byte of value 0x80 is added. Next, a number of bytes (0x00) is added, so that the overall size of the data block (including the extra bytes) mod 512 is 448. Following that, a 64 bits big-endian number is attached, which represents the size of the data block, in bits (without the padding). While in this process, TX/RX FIFOs are switched into 8-bytes mode.

The block packetizes the algorithm result (up to 224 or 256 bits) into blocks of 64 bits, so that they can be shifted to the TX FIFO. The SHA-224/256: FIPS PUB180-4 hash algorithms is supported.

18.3 Programming

18.3.1 AES engine programming

To program the AES Engine:

1. Enable the clock by setting the CLK_AMBA_REG[AES_CLK_ENABLE] bit.
2. Select the AES mode (CRYPTO_CTRL_REG[CRYPTO_HASH_SEL] = 0).
3. Define the mode of operation of the AES algorithm (CRYPTO_CTRL_REG[CRYPTO_ALG_MD]). For the CBC/CTR mode of operation the corresponding IV/CTR block should be defined in the CRYPTO_MREGx_REG registers.
4. Select the AES algorithm (CRYPTO_CTRL_REG[CRYPTO_ALG]).
5. Select the size of AES Key (CRYPTO_CTRL_REG[CRYPTO_AES_KEY_SZ]).
6. Use AES keys expansion if needed (CRYPTO_CTRL_REG[CRYPTO_AES_KEXP]). If the key expansion is enabled, define the (basic) key in the local CRYPTO_KEYS_START memory.
7. Set up data fetching address (CRYPTO_FETCH_ADDR_REG).
8. Set up data destination address (CRYPTO_DEST_ADDR_REG).
9. Set data length (CRYPTO_LEN_REG).
10. Select to Encrypt or Decrypt (CRYPTO_CTRL_REG[CRYPTO_ENCDEC]).
11. Enable the process by setting the CRYPTO_START_REG[CRYPTO_START] bit.

12. Wait for the process to finish (CRYPTO_STATUS_REG[CRYPTO_INACTIVE] = 1).

18.3.2 HASH engine programming

To program the HASH Engine:

1. Enable the clock by setting the CLK_AMBA_REG[AES_CLK_ENABLE] bit.
2. Select the HASH mode (CRYPTO_CTRL_REG[CRYPTO_HASH_SEL] = 1).
3. Set the number of bytes which are saved at the memory by the DMA (CRYPTO_CTRL_REG[CRYPTO_HASH_OUT_LEN], see register description).
4. Define the mode of operation of the HASH algorithm (CRYPTO_CTRL_REG[CRYPTO_ALG_MD], see register description).
5. Select the HASH algorithm (CRYPTO_CTRL_REG[CRYPTO_ALG], see register description).
6. Set up data fetching address (CRYPTO_FETCH_ADDR_REG).
7. Set up data destination address (CRYPTO_DEST_ADDR_REG).
8. Set data length (CRYPTO_LEN_REG).
9. Enable the process by setting the CRYPTO_START_REG[CRYPTO_START] bit.
10. Wait for the process to finish (CRYPTO_STATUS_REG[CRYPTO_INACTIVE] = 1).

For both cases, an interrupt can be enabled upon the completion of the processing, by setting 1 to CRYPTO_IRQ_EN (the interrupt should be enabled also in CPU).

The clock of the Crypto Block should be disabled after the completion of each operation, by clearing the CLK_AMBA_REG[AES_CLK_ENABLE] bit, if there is no other operation to be performed.

19 Sleep Wake-Up Controller

19.1 Introduction

The wake-up controller can be programmed to wake up DA14592 from the Extended/Deep Sleep (clocked) mode. It consists of two parallel circuits that can be programmed individually, one with a debouncing counter usually used for buttons and another that indicates which GPIO has toggled.

Figure 38 shows the block diagram of the wake-up function.

Features

- Monitors any GPIO state change.
- Implements debouncing time from 0 up to 63 ms.
- Latches the status of the monitored lines.
- Generates three interrupts to Arm Cortex-M33.
- Generates signals towards PDC indicating which GPIO has toggled.
- Wakes up the system from Deep or Extended Sleep.

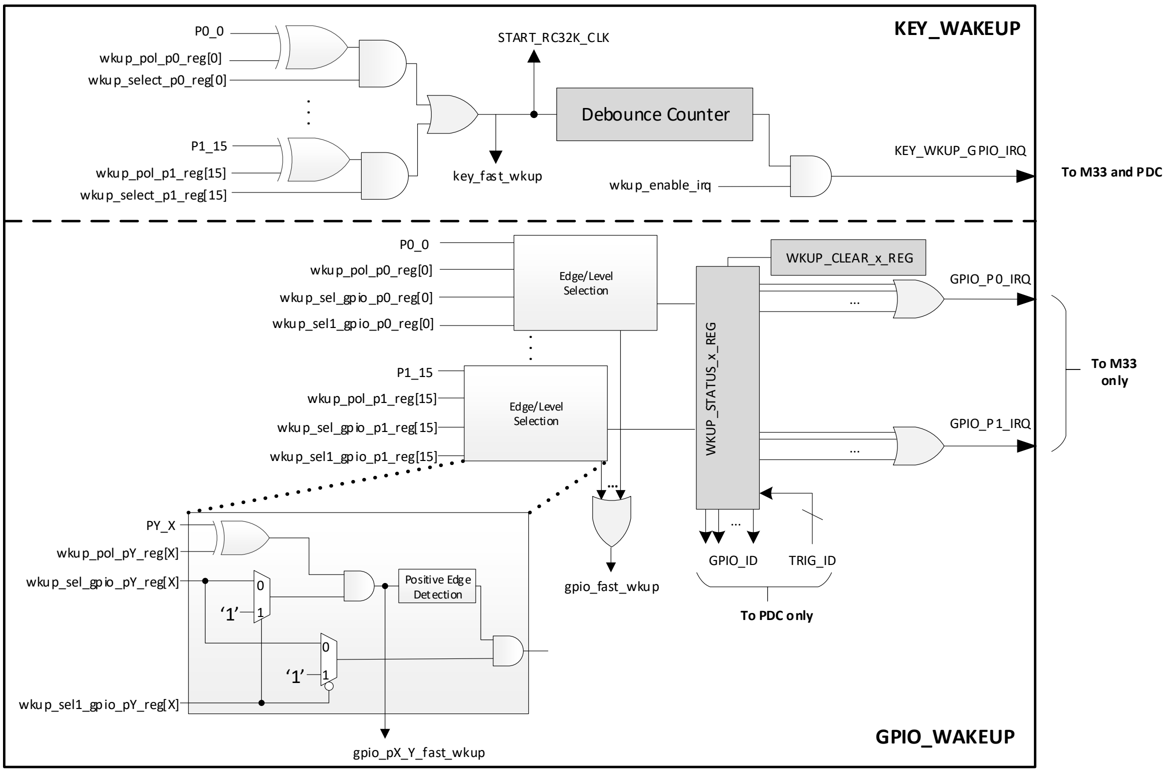

Figure 38. Wake-up controller block diagram

19.2 Architecture

The wake-up controller can monitor all GPIO lines for an event. A line of XOR gates defines the polarity of the signal to be monitored. Two parallel structures are implemented explicitly separated in Figure 38 by a dashed line:

- KEY_WAKEUP: It can be programmed to monitor a number of GPIOs regarding their edge or level. After a GPIO toggles, a debounce counter can be triggered to debounce the key press before generating an interrupt. The KEY_WKUP_GPIO_IRQ is generated towards the Cortex-M33 and the PDC (named as Debounced_IO in the PDC). The debouncing time can be programmed to be up to 63 ms. If debouncing is selected to be 0, the interrupt is issued on an edge detection, but the external signal needs to be at least 2 x RC32/512K clock periods asserted considering the system being active.

This circuit can wake up the system from Deep or Extended (clocked) Sleep. Since the interrupt is kept asserted until acknowledged by software, the Cortex-M33 can receive it after its power domain has been powered up by PDC. Of course, a respective PDC entry needs to be in place for waking up Cortex-M33.

When the system is active, the circuit can keep on being used as a button press indicator with debouncing. Note that, if multiple GPIOs have been toggled there is no indication which one has generated the interrupt.

- GPIO_WAKEUP: This circuit can be programmed to monitor edge (positive or negative) or level sensitive triggers and latch them into a status register, so the source is known by the application. The default behavior is level sensitive. This status register is delivered to the PDC in form of a signal bus for being used as a trigger for the PDC entries. Moreover, an OR of each GPIO port signal forms a level interrupt towards the Cortex-M33, namely GPIO_P0_IRQ and GPIO_P1_IRQ.

  Since both interrupt lines are kept asserted until acknowledged by software, the M33 can receive them after its power domain has been powered up by PDC. Of course, a respective PDC entry needs to be in place for waking up Cortex-M33.

  When the system is active, the circuit can keep on being used as an interrupt generator towards Cortex-M33 storing the GPIO state that has caused the interrupt in the first place.

  Figure 39 and Figure 40 show the IRQ generation based on the configuration of the WKUP_SEL_GPIO_PY_REG and WKUP_SEL1_GPIO_PY_REG bits.

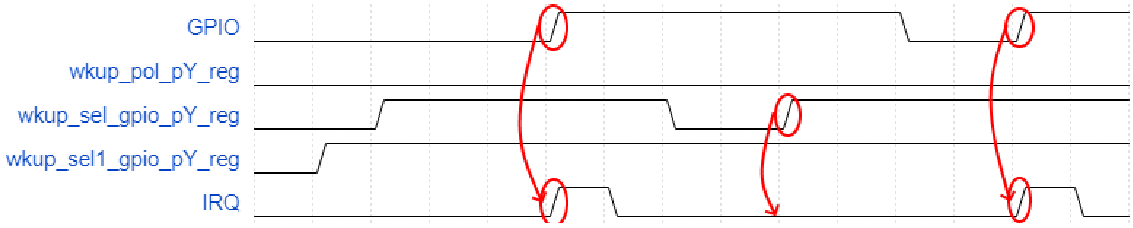

Figure 39. Edge sensitive GPIO

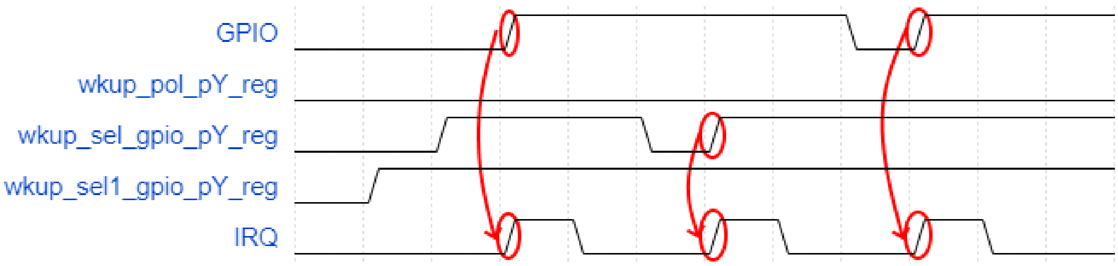

Figure 40. Level sensitive GPIO

NOTE
The circuits KEY_WAKEUP and GPIO_WAKEUP can wake up the system from any clocked sleep (Deep or Extended) but not from Hibernation.

19.3 Programming

To program the wake-up controller:

1. Enable the Wake-up controller by setting the CLK_TMR_REG[WAKEUPCT_ENABLE] bit.
2. Set up triggering polarity (WKUP_POL_Px_REG).
3. If debouncing is needed, define debounce time in WKUP_CTRL_REG[WKUP_DEB_VALUE].
4. Register PDC and/or KEY Interrupts:
   a. Add PDC Entries (needed when the device goes to sleep).

   b. Clear any latched values of the GPIOs (WKUP_CLEAR_Px_REG).
   c. Add wake-up events (WKUP_SELECT_Px_REG).
   d. Add GPIO interrupts (WKUP_SELx_GPIO_Py_REG).
5. If needed, add the corresponding ISRs and enable the interrupts (KEY_WKUP_GPIO_IRQn, GPIO_P0_IRQn, GPIO_P1_IRQn, PDC_IRQn).

20 Wake-Up from Hibernation

20.1 Introduction

The wake-up from hibernation controller wakes up the system when the device enters the (clockless) hibernation mode. It consists of a very low count of gates directly powered from VBAT and as a result the device can reach ultra-low hibernation current. To keep the power consumption as low as possible, only two GPIOs can wake up the system (P0_14, P1_04).

Figure 38 shows the wake-up from hibernation function.

Features

- Ultra-low hibernation power.
- Designed for shipping mode.
- Two wake-up triggers with configurable polarity.

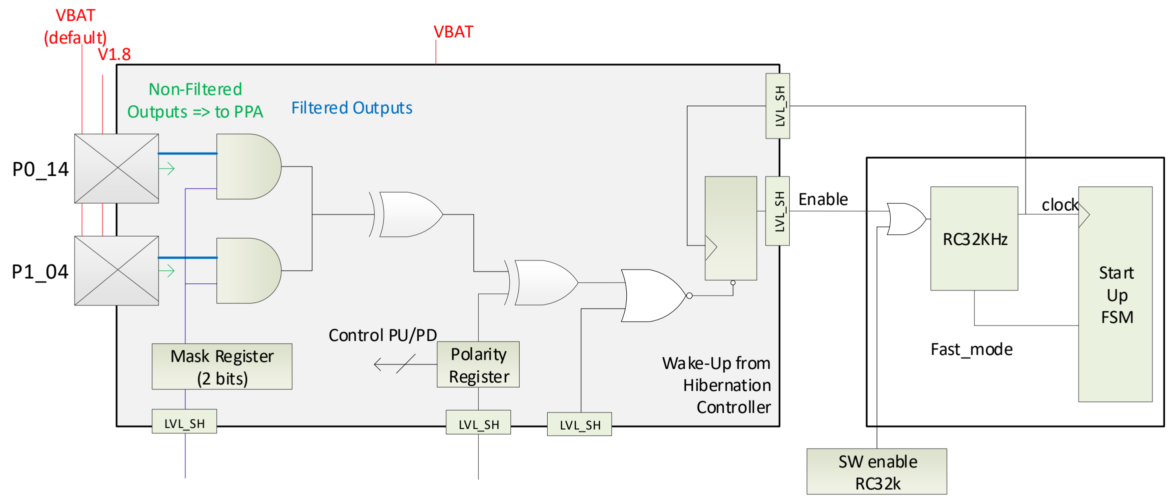

Figure 41. Wake-up from hibernation controller block diagram

20.2 Architecture

The wake-up from hibernation controller (Grey box) resides in the Always ON power domain (PD_AON) and it is powered directly from VBAT. The LVL_SHs are level shifters (low to high or high to low). The output enable signal is produced by controlling the asynchronous reset of a flip-flop. This signal enables the RC32/512K clock. When RC32/512K is enabled, the flip-flop is also clocked and, as a result, is cleared.

As soon as the RC32/512K is enabled, the system's Start-Up FSM begins to bring up the system.

Only two GPIOs are used as wake-up triggers. Figure 38 shows these GPIOs that comprise Pull Up/Down resistors that can be controlled by setting the Polarity register (HIBERN_CTRL_REG[HIBERN_WKUP_POLARITY]). If the polarity is set to active low, the GPIO has to be externally pulled up.

20.3 Programming

To enter the Hibernation mode:

1. Select which pin to wake up from by configuring HIBERN_CTRL_REG[HIBERN_WKUP_MASK].
2. Select the polarity of the wake-up source by configuring HIBERN_CTRL_REG[HIBERN_WKUP_POLARITY].
3. Enable the enter to Hibernation mode on sleep functionality by setting the HIBERN_CTRL_REG[HIBERNATION_ENABLE] bit field.

21 General Purpose ADC

21.1 Introduction

The DA14592 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 and bipolar (differential) mode. The ADC has its own voltage regulator (LDO) of 0.9 V, which represents the full-scale reference voltage.

Features

- 10-bit dynamic ADC
- Maximum sampling rate, 1 Msample/s
- 128x averaging; conversion time 1 ms, up to 11.5 b ENOB
- Single-ended as well as differential input with two input scales
- Eight single-ended or four differential external input channels
- Battery monitoring function
- Chopper function
- Offset and zero scale adjust
- Common-mode input level adjust
- Selectable attenuators: 1x, 2x, 3x, and 4x
- DMA support.

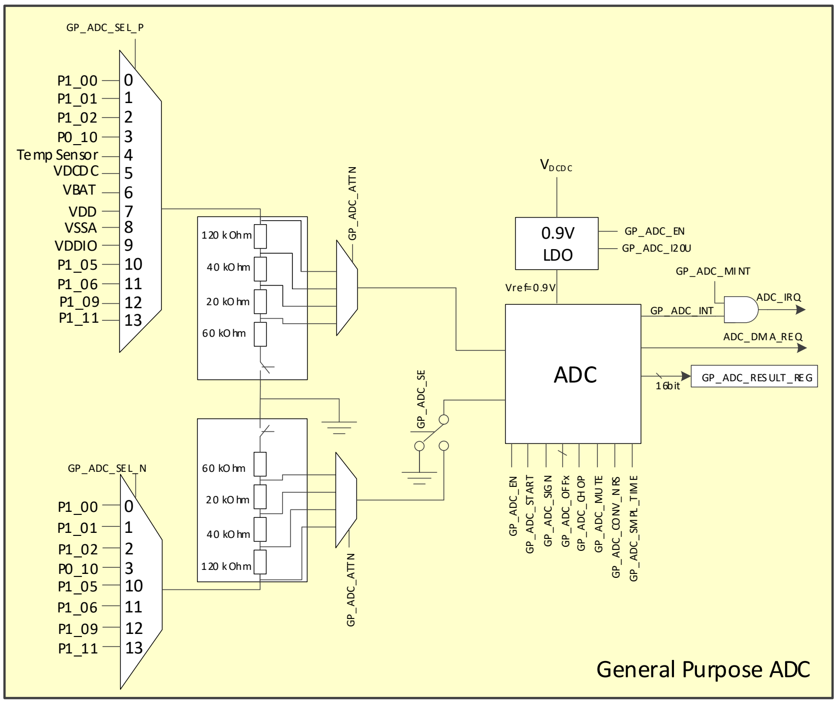

Figure 42. General-purpose ADC block diagram

21.2 Architecture

The ADC architecture shown in Figure 42 has the following sub-blocks:

- Analog to Digital converter (ADC)
  - ADC analog part internally clocked with 100 MHz.
  - ADC logic part clocked with the ADC_CLK which is the 16 MHz system clock (sys_clk).

- 0.9 V LDO for the ADC supply with a high PSRR enabled with GP_ADC_CTRL_REG[GP_ADC_EN].
- Configurable attenuator with 1×, 2×, 3×, and 4× attenuation controlled by GP_ADC_CTRL2_REG[GP_ADC_ATTN].
- APB Bus interface clocked with the APB clock. Control and status registers are available through registers GP_ADC_*
- Maskable Interrupt (ADC_IRQ) and DMA request (ADC_DMA_REQ).
- ADC input channel selector. Up to four GPIO ports, the battery and DCDC output (V_{BAT} and V_{DCDC}), the internal V_{DD}, and the analog ground level (AVS) can be measured.

21.2.1 Input channels

Table 87 summarizes the ADC input channels. The GPIO signals at the channels [3:0] and [13:10] can be monitored both single-ended and differentially. The signals at the 4-9 inputs can be monitored single-ended or differentially with respect to the GPIOs.

Table 87: ADC Input Channels

Channel	Signal	Description
3:0	GPIO [P1_00, P1_01, P1_02, P0_10]	General Purpose Inputs
4	Temperature Sensor	Temperature Sensor
5	V_{DCDC}	V_{DCDC} rail
6	V_{BAT}	V_{BAT} rail (3.6 V -> 0.9)
7	V_{DD}	V_{DD} rail for the digital power domain
8	VSS(A)	Reserved for test only
9	VDDIO	V_{DDIO} rail
13:10	GPIO [P1_05, P1_06, P1_09, P1_11]	General Purpose Inputs

Table 88 summarizes the voltage ranges which can be handled with the single-ended or differential operation for different attenuation values. The single-ended/differential mode is controlled by the GP_ADC_CTRL_REG[GP_ADC_SE] bit, and the attenuation is handled by the GP_ADC_CTRL2_REG[GP_ADC_ATTN] bit.

Table 88: GPADC External Input Channels and Voltage Range

GP_ADC_ATTN	GP_ADC_SE	Input Scale	Input Limits
0 (1 ×)	0	-0.9 V to +0.9 V	-1 V to +1 V
	1	0 V to +0.9 V	-0.1 V to 1 V
1 (2 ×)	0	-1.8 V to +1.8 V	-1.9 V to +1.9 V
	1	0 V to +1.8 V	-0.1 V to 1.9 V
2 (3 ×)	0	-2.7 V to +2.7 V	-2.8 V to +2.8 V
	1	0 V to +2.7 V	-0.1 V to 2.8 V
3 (4 ×)	0	-3.6 V to +3.6 V	-3.45 V to +3.45 V
	1	0 V to +3.6 V	-0.1 V to 3.45 V

21.2.2 Operating modes

Figure 43 shows the GPADC operation flow diagram.

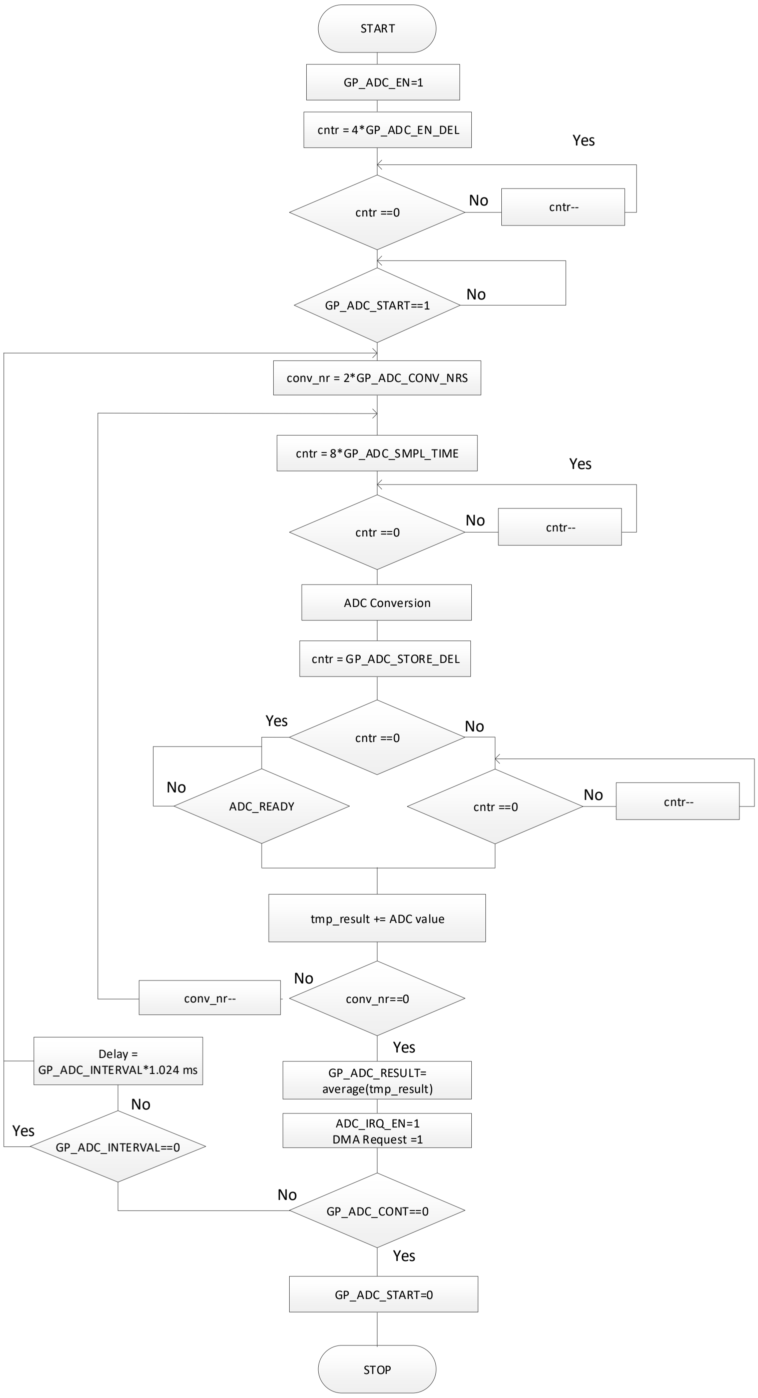

Figure 43. GPADC operation flow diagram

21.2.2.1 Enabling the ADC

Enabling/disabling of the ADC is triggered by configuring the GP_ADC_CTRL_REG[GP_ADC_EN] bit. When the bit is set to 1, first the LDO is enabled. Then after the delay value set in GP_ADC_CTRL3_REG[GP_ADC_EN_DEL] (typically 16 µs to account for the LDO settling time), the ADC is enabled, and an AD conversion can be started. See Table 89 for recommended values.

Table 89: ADC_LDO Start-Up Delay

f_{ADC_CLK}	GP_ADC_EN_DEL	$T_{ADC\_EN\_DEL}$
16 MHz	0x40	16 µs

Formula:

GP_ADC_EN_DEL = $\tau_{ADC\_EN\_DEL}$ × $f_{ADC\_CLK}/4$

This value must be rounded up to the nearest integer.

The GPADC is a dynamic ADC and consumes no static power, except for the **ADC_LDO** which consumes approximately 20 μA. Therefore, GP_ADC_EN must be set to 0 if the ADC is not used.

21.2.2.2 Manual mode

An AD conversion can be started by setting GP_ADC_START to 1. While a conversion is active, GP_ADC_START remains 1. When a conversion is finished, the hardware sets GP_ADC_START to 0 and GP_ADC_INT to 1 (interrupt), and GP_ADC_RESULT_REG contains the valid ADC value. While a conversion is active, writing 1 to GP_ADC_START does not start a new conversion. Software should always check that bit GP_ADC_START = 0 before starting a new conversion.

21.2.2.3 Continuous mode

Setting GP_ADC_CTRL_REG[GP_ADC_CONT] to 1 enables the continuous mode, which automatically starts a new AD conversion when the current conversion has been completed. The GP_ADC_START bit is only needed once to trigger the first conversion. As long as the continuous mode is active, GP_ADC_RESULT_REG always contains the latest ADC value.

To correctly terminate the continuous mode, it is required to disable the GP_ADC_CONT bit first and then wait until the GP_ADC_START bit is cleared to 0, so the ADC is in a defined state.

NOTE
Before making any changes to the ADC settings, users must disable the continuous mode by setting bit GP_ADC_CONT to 0 and waiting until bit GP_ADC_START = 0.

At full speed the ADC consumes approximately 50 to 60 μA. If the data rate is less than 100 ksample/s, the current consumption is in the order of 25 μA.

The time interval between two successive AD conversions is programmable with GP_ADC_CTRL3_REG[GP_ADC_INTERVAL] in steps of 1.024 ms. If GP_ADC_INTERVAL = 0, the conversion restarts immediately. If GP_ADC_INTERVAL is not zero, the ADC first synchronizes to the delay clock before starting the conversion. This can take up to 1 ms.

21.2.3 Conversion modes

21.2.3.1 AD conversion

Each AD conversion has three phases:

- Sampling
- Conversion
- Storage.

The AD conversion starts with the sampling phase. This phase ends after the time set in GP_ADC_CTRL2_REG[GP_ADC_SMPL_TIME] and triggers the conversion phase. If GP_ADC_CTRL2_REG[GP_ADC_STORE_DEL] = 0, handshaking is used, that is, the ADC result is stored when a conversion is finished. Otherwise, a fixed (programmable) delay is used, and the result is stored regardless of whether the conversion is finished or not.

The total conversion time of an AD conversion depends on various settings. In short, it is as follows.

$$T_{ADC} = \frac{N_{CONV} \cdot (N_{CYCL\_SMPL} + N_{CYLC\_STORE})}{f_{ADC\_CLK}} \quad (1)$$

Where

- N_{CONV} = the number of conversions. This is related to the value programmed in GP_ADC_CTRL2_REG[GP_ADC_CONV_NRS], following $2^{GP\_ADC\_CONV\_NRS}$. When GP_ADC_CTRL_REG[GP_ADC_CHOP] is set, the minimum value for N_{CONV} is always 2.
- N_{CYLC_SMPL} = the number of ADC_CLK cycles used for sampling, which is 8 × GP_ADC_CTRL2_REG[GP_ADC_SMPL_TIME].
- N_{CYCL_STORE} = the number of ADC_CLK cycles until the result is stored. When GP_ADC_CTRL2_REG[GP_ADC_STORE_DEL] = 0, handshaking is used. With handshaking, the number of ADC_CLK cycles is typically three. This value may spread from sample to sample and over temperature, otherwise the number of ADC_CLK cycles is GP_ADC_CTRL2_REG[GP_ADC_STORE_DEL] + 1.

Sampling phase

The sampling time can be programmed through GP_ADC_CTRL2_REG[GP_ADC_SMPL_TIME] and depends on the sampling time constant in combination with the desired sampling accuracy. This sampling time constant, τ_{ADC_SMPL} (Table 90), then depends on the output impedance of the source, the internal resistive dividers, and the internal sampling capacitor. And the number of required time constants is given by the natural logarithm of the desired accuracy, that is, ln(2^N_{BIT}). For N_{BIT} = 10-bit accuracy, seven-time constants are required.

Table 90: ADC Sampling Time Constant (τ_{ADC_SMPL})

ADC Input	τ_{ADC_SMPL}
GPADC0, GPADC1 (GP_ADC_ATTN = 0)	R_{OUT} × 0.5 pF (Differential Input) R_{OUT} × 1 pF (Single-Ended Input)
GPADC0, GPADC1 (GP_ADC_ATTN = 1)	(R_{OUT} + 160 kΩ) × 0.5 pF (Differential Input) (R_{OUT} + 160 kΩ) × 1 pF (Single-Ended Input)

Formula:

GP_ADC_SMPL_TIME = ln(2^N_{BIT}) × τ_{ADC_SMPL} × f_{ADC_CLK}/8

This value must be rounded up to the nearest integer.

Conversion and storage phase

One AD conversion typically takes around 125 ns with a 100 MHz clock. The result can be stored either by handshaking or after a fixed number of cycles (programmable).

- Handshake mode (GP_ADC_STORE_DEL = 0):

In handshake mode, the conversion result is available in GP_ADC_RESULT_REG after two sampling ADC_CLK cycles plus two conversion ADC_CLK cycles plus two ADC_CLK cycles for synchronization.

- Fixed delay mode (GP_ADC_STORE_DEL > 0):

In fixed delay mode, the conversion result is available in GP_ADC_RESULT_REG after the programmed storage delay, regardless of whether the conversion is ready or not. Note that when the delay is too short (that is, the conversion is not finished in the allocated time), the old (previous) ADC result is stored.

21.2.3.2 Averaging

To reduce noise and improve performance, multiple samples can be averaged out (assuming the time average of noise equals zero). This is handled by hardware and can be controlled by setting GP_ADC_CTRL2_REG[GP_ADC_CONV_NRS] to a non-zero value. The actual number of the consecutive samples taken is by $2^{GP\_ADC\_CONV\_NRS}$.

Because the internal noise also acts as a form of dither, the actual accuracy can be improved. Therefore, the ADC result is not truncated to 10-bit but stored as 16-bit left aligned, and truncation is left for the user. Table 91 shows the expected Effective Number of Bits (ENOB).

Table 91: ENOB in Oversampling Mode

GP_ADC_CONV_NRS	ENOB (Left Aligned) in GP_ADC_RESULT_REG
0	8
1	9
2	10
3	10
4	10
5	11
6	11
7	11

21.2.3.3 Chopper mode

Inherently, the ADC has a DC offset (E_{OFS}). When GP_ADC_CTRL_REG[GP_ADC_CHOP] is set to 1, the hardware triggers two consecutive AD conversions and flips the sign of the offset in-between. Summing the two samples effectively cancels out the inherent ADC offset. This method also smooths other non-ideal effects and is recommended for DC and the slowly changing signals.

When combined with averaging, every other AD conversion is taken with the opposite sign. Without averaging two AD conversions are always triggered.

Note that a DC offset causes saturation effects at zero scale or full scale. When chopping is used without offset calibration, non-linear behavior is introduced towards zero scale and full scale.

21.2.4 Additional settings

The hardware also supports pre-ADC attenuation through GP_ADC_CTRL2_REG[GP_ADC_ATTN]:

- Setting 0 disables the attenuator
- Setting 1 scales the input range by a factor of two
- Setting 2 scales the input range by a factor of three
- Setting 3 scales the input range by a factor of four.

With bit GP_ADC_CTRL_REG[GP_ADC_MUTE] = 1, the input is connected to 0.5 × ADC reference. So, the ideal ADC result should be 511.5. Any deviation from this is the ADC offset.

With bit GP_ADC_CTRL_REG[GP_ADC_SIGN] = 1, the sign of the offset is inverted. When chopper is used, the hardware alternates GP_ADC_SIGN = 0 and 1. This bit is typically only used for the offset calibration routine described in Section 21.2.6 and has no specific use to the end user.

21.2.5 Non-ideal effects

Besides Differential Non-Linearity (DNL) and Integral Non-Linearity (INL), each ADC has a gain error (linear) and an offset error (linear). The gain error (E_G) of the GPADC affects the effective input range. The offset error (E_{OFS}) causes the effective input scale to become non-centered. The offset error can be reduced by chopping and/or by offset calibration.

The ADC result also includes some noise. If the input signal itself is noise free (inductive effects included), the average noise level is ±1 LSB. Reducing noise effects can be done by taking more samples and calculating the average value. This can be done by programming GP_ADC_CTRL2_REG[GP_ADC_CONV_NRS] to a non-zero value.

With a "perfect" input signal (for example, 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. Because the DA14592 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]. The GP_ADC_I20U bit enables a constant 20 μA load current at the regulator output so that the current does not drop to zero. This, obviously, increases power consumption by 20 μA.

21.2.6 Offset calibration

A relative high offset error (E_{OFS}, up to 30 mV, so approximately 30 LSB) is caused by a very small dynamic comparator. This offset error can be cancelled with the chopping function, but it still causes unwanted saturation effects at zero scale or full scale. With GP_ADC_OFFP_REG and GP_ADC_OFFN_REG, the offset error can be compensated in the ADC network itself.

During production, the offset correction values are stored in the Information Page section, so you do not need to perform the run time calibration procedure. Only the use of the stored values is needed.

If you need to perform a run time calibration, follow the steps in Table 92. In this routine, 0x200 is the target mid-scale of the ADC.

Table 92: 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	GP_ADC_OFFP = 0x200 - adc_off_p GP_ADC_OFFN = 0x200 - adc_off_n

To increase the accuracy, it is recommended to set GP_ADC_CTRL2_REG[GP_ADC_SMPL_TIME] = 2 or 3 and GP_ADC_CTRL2_REG[GP_ADC_CONV_NRS] = 3 or 4 prior to this routine.

It is recommended to implement the above calibration routine during the initialization phase of DA14592. To verify the calibration results, check whether the GP_ADC_RESULT value is close to 0x200 while bit GP_ADC_CTRL_REG[GP_ADC_MUTE] = 1.

21.2.7 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.

21.2.8 Common mode adjustment

The common mode level of the differential signal must be 0.45 V = Full Scale/2 (or 1.35 V with GP_ADC_ATTN = 2, that is, 3× attenuation). If the common mode input level of 0.45 V cannot be achieved, the common mode level of the GPADC can be adjusted through GP_ADC_OFFP_REG and GP_ADC_OFFN_REG according to Table 93. The GPADC can tolerate a common mode margin of up to 50 mV.

Table 93: Common Mode Adjustment

CM Voltage (V_{ccm})	GP_ADC_OFFP = GP_ADC_OFFN
0.225 V	0x300

CM Voltage (V_{ccm})	GP_ADC_OFFP = GP_ADC_OFFN
0.450 V	0x200
0.675 V	0x100

Any other common mode levels between 0.0 V and 0.9 V can be calculated from Table 93. Offset calibration can be combined with common mode adjustment by replacing the 0x200 value in the offset calibration routine with the value required to get the appropriate common mode level.

21.2.9 Input impedance, inductance, and input settling

The GPADC has no input buffer stage. During the sampling phase, a capacitor of 0.5 pF in differential mode or 1 pF in single-ended mode is switched to the input line(s). The pre-charge of this capacitor is at midscale level, so the input impedance is infinite.

During the sampling phase, a certain settling time is required. A 10-bit accuracy requires at least seven-time constants τ_{ADC_SMPL}, determined by the output impedance of the input signal source, the internal resistive dividers, and the 0.5 pF or 1 pF sampling capacitor. See Table 90.

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 (single-ended mode) or from pin to pin (differential mode).

21.3 Programming

To program and use the GPADC:

1. Enable the GPADC by setting the GP_ADC_CTRL_REG[GP_ADC_EN] bit.
2. Set up the GPIO input (P0_x_MODE_REG[PID] = 16).
3. Select the input channel (GP_ADC_SEL_REG).
4. Select the sampling mode (differential or single ended) by writing the GP_ADC_CTRL_REG[GP_ADC_SE] bit.
5. Select between the manual mode and the continuous mode of sampling (GP_ADC_CTRL_REG[GP_ADC_CONT].
6. Set up extra options (see GP_ADC_CTRLx_REG description)
7. Start the conversion by setting GP_ADC_CTRL_REG[GP_ADC_START] bit.
8. Wait for GP_ADC_CTRL_REG[GP_ADC_START] to become 0 or interrupt being triggered (when used).
9. Clear the ADC interrupt by writing any value to GP_ADC_CLEAR_INT_REG.
10. Get the ADC result from the GP_ADC_RESULT_REG.

22 ΣΔ ADC

22.1 Introduction

The DA14592 is equipped with a ΣΔ ADC that supports two operation modes: the audio mode and the sensor mode. The audio mode delivers 11 effective number of bits (ENOB) at a rate of 16 ksamples/s. The sensor mode implements a third order filter that achieves 12.5 ENOB at 968 samples/s rate.

A programmable gain amplifier (PGA) is connected to two of the eight available channels for adjusting the input level of a microphone or sensor to the ADC input range.

The reference voltage can be selected between the internal 0.9 V reference or an external voltage reference (Note 1). It also supports VBAT measuring (Note 1).

Features

- 12.5 bits resolution in Sensor Mode, at 968 samples/s
- 11 bits resolution in Audio Mode, at 16 ksamples/s
- Single-ended as well as differential input with two input scales
- Programmable Gain Amplifier used in differential mode only
- Eight external input channels (two reserved for PGA)
- Battery monitoring function (Note 1).

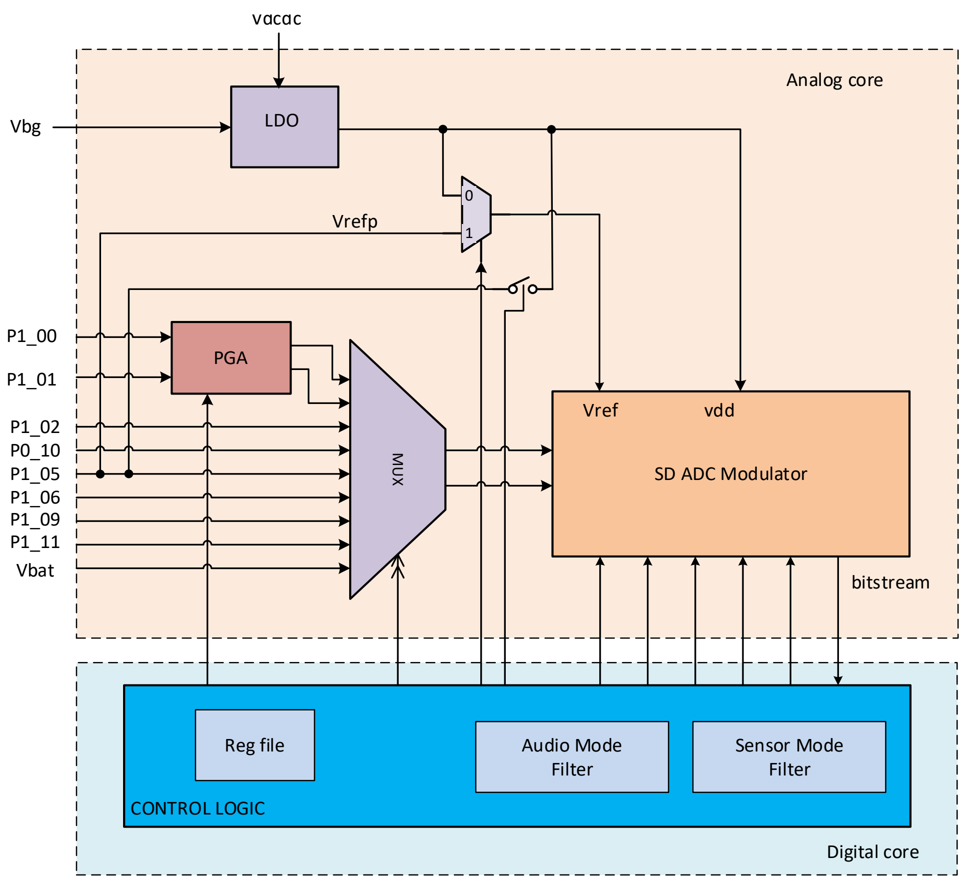

Figure 44. Block diagram of the ΣΔ ADC

Note 1 In sensor mode, available in FCQFN52 package.

22.2 Architecture

The block comprises an analogue part including the ΣΔ-Modulator and auxiliary circuitry, and a digital part containing the digital filter and control logic.

An eight-input multiplexer and a separate VBAT input is placed in front of the modulator. A reference selector is also added that allows an external reference voltage to be used. The internal reference voltage of 0.9 V is selected by default.

The digital block includes the digital filters and the control logic for analog part. Figure 45 shows the audio mode filter and data path.

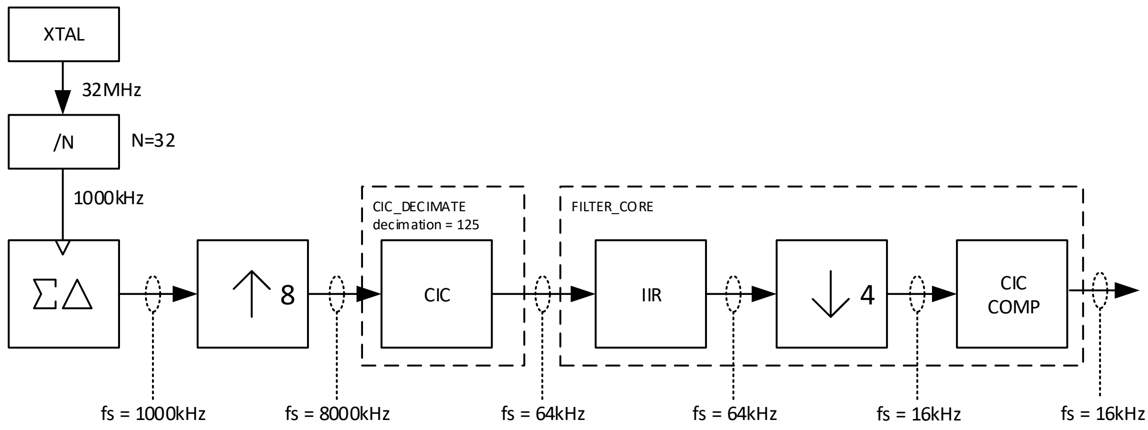

Figure 45. Audio mode data path

This filter delivers a 16-bit word at a 16 kHz rate. Note that out of the 16 bits the 5 LSBs are considered to be noise and should be discarded (ENOB = 11). The output of the filter is driven to the result register of the ΣΔ ADC block where it can be read by the CPU or a DMA engine. In parallel, it is connected to the APU block in one of the SRC inputs, enabling sampling conversion directly.

Figure 46 shows the third order sensor mode filter.

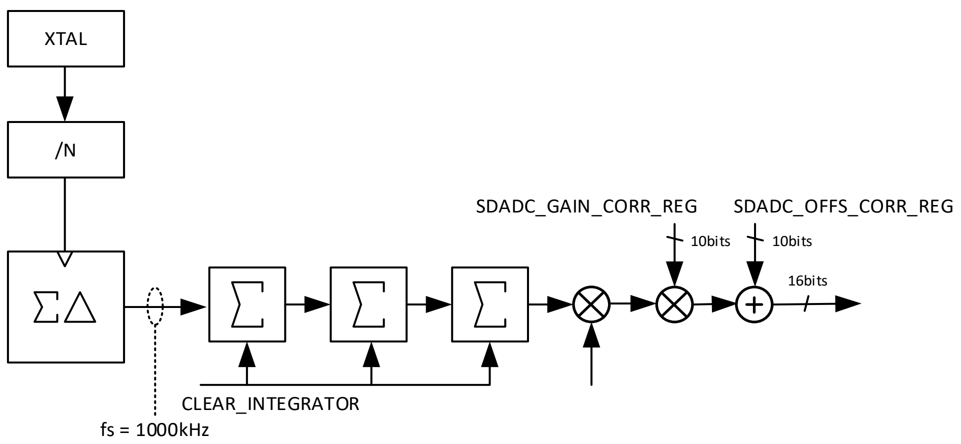

Figure 46. Sensor mode data path

This filter delivers a 16-bit word at a 1 kHz rate. Note that, out of the 16 bits, the 2.5 LSBs are considered to be noise and should be discarded (ENOB = 12.5). The later resolution is achieved by applied an oversampling rate of 1024 (SDADC_OSR = 2).

For reading the battery voltage an internal 4x attenuator is utilized.

An external reference voltage can be applied at pins SDADC_REF (P1_05) of no more than 0.9 V to improve precision. It is also possible to output the internal reference from the LDO to P1_05 and use it as a reference for the sensor. In this case, the external load current and capacitance should not exceed the drive capabilities of the internal LDO.

22.3 PGA

An internal PGA can be used to amplify a microphone input. The PGA is a pseudo-differential amplifier that can be used in two different configurations as shown in Figure 47 and Figure 48.

Figure 47 shows the differential configuration that consists of two single ended amplifiers that form a pseudo differential amplifier. The value of the resistors depends upon the gain setting. The common mode voltage of the amplifier is fixed to the Vref_int/2, which is approximately 450 mV.

Figure 48 shows the single ended configuration that uses the same amplifiers in a series configuration. The output of the first amplifier is inverted (gain = -1) by the second. In this way a differential output is obtained from a single ended input.

Since the PGA has a fixed input common mode level, the PGA is best used with AC coupling capacitors at the input.

The best performance is achieved when both PGA and ADC are configured in differential mode.

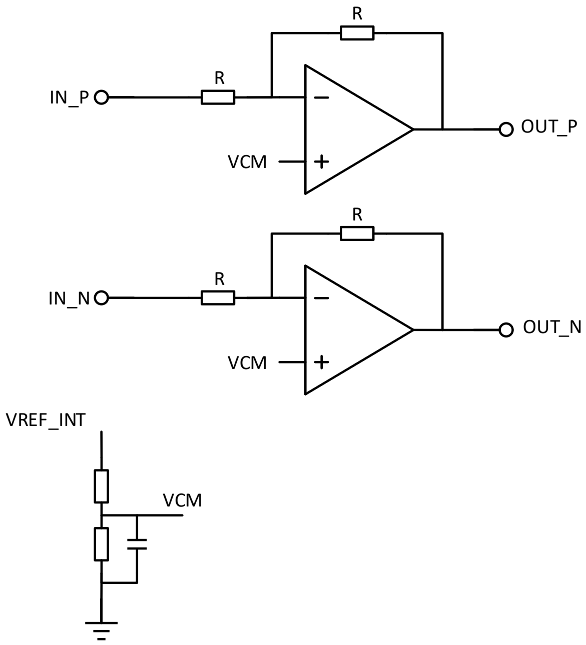

Figure 47. PGA differential configuration

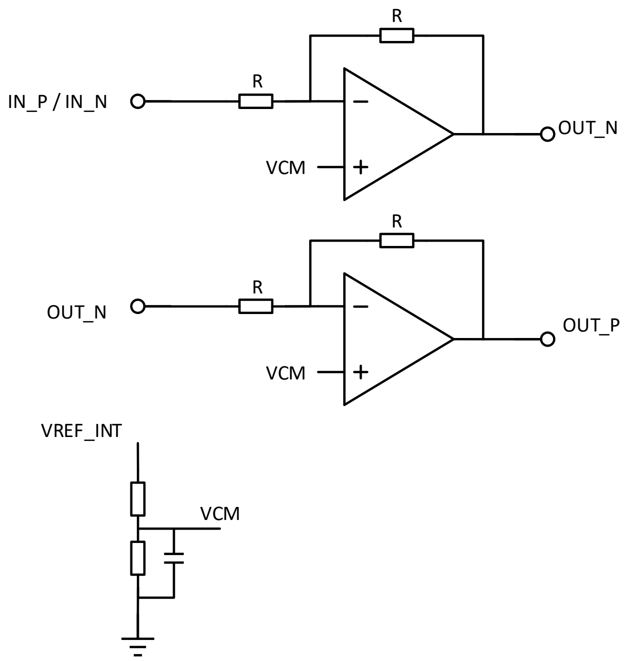

Figure 48. PGA single ended configuration

22.4 Programming

To program and use the ΣΔ AD Converter:

Sensor mode

1. Enable the SDADC block by setting the SDADC_CTRL_REG[SDADC_EN] bit.
2. Set up the GPIO input (Px_yy_MODE_REG[PID] = 16).
3. Select the input channel for the positive side (SDADC_CTRL_REG[SDADC_INP_SEL]) and the negative side (SDADC_CTRL_REG[SDADC_INN_SEL], negative side is ignored in single ended mode).
4. Select the Voltage reference (SDADC_CTRL_REG[SDADC_VREF_SEL]).
5. Select the sampling mode (differential, single ended) by writing the SDADC_CTRL_REG[GP_ADC_SE] bit.
6. Select between manual and continuous mode of sampling (SDADC_CTRL_REG[SDADC_CONT]).
7. Set up extra options (see SDADC_CTRL_REG description)
8. Start the conversion by setting SDADC_CTRL_REG[SDADC_START] bit.
9. Wait for SDADC_CTRL_REG[SDADC_START] to become 0 or interrupt being triggered (when used).
10. Clear the ADC interrupt by writing any value to SDADC_CLEAR_INT_REG.
11. Get the ADC result from the SDADC_RESULT_REG.

Audio mode

To configure the PGA:

1. Select the required PGA gain using the SDADC_PGA_CTRL_REG[PGA_GAIN] bit field.
2. Select between single ended and differential mode using the SDADC_PGA_CTRL_REG[PGA_SINGLE] bit field.
3. Enable PGA by setting the SDADC_PGA_CTRL_REG[PGA_EN] bit.

To start the audio processing stream:

1. Select the Audio filter (SDADC_CTRL_REG[SDAC_MODE] = 1).

2. Select the positive and negative channels for the PGA (SDADC_CTRL_REG[SDAC_INP_SEL] = 0 and SDADC_CTRL_REG[SDAC_INN_SEL] = 1).
3. Select the Voltage reference (SDADC_CTRL_REG[SDADC_VREF_SEL]).
4. Enable the interrupts by setting SDADC_CTRL_REG[SDAC_MINT] = 1.
5. Enable the SDADC block (SDADC_CTRL_REG[SDAC_EN] = 1).
6. Select the continuous mode, so a new 16-bit sample is available on the output on a 16-kHz rate (SDADC_CTRL_REG[SDAC_CONT] = 1).
7. Start the conversion by SDADC_CTRL_REG[SDAC_START] = 1.
8. Get the ADC result from the SDADC_RESULT_REG when interrupt is triggered.

To stop the audio processing stream:

1. Deselect continuous mode by SDADC_CTRL_REG[SDAC_CONT] = 0.
2. Wait until the SDAC_START bit is cleared.

23 Temperature Sensor

23.1 Introduction

The DA14592 features a built-in temperature sensor.

Features

- Temperature range -40 °C to 115 °C.
- Ambient temperature single point calibration reference value provided in IP sector.

23.2 Architecture

The temperature sensor can be read out through the GP_ADC.

Figure 49 shows the relationship between the actual ambient temperature and the calculated temperature from the GP_ADC readout, including possible inaccuracies in $T_{SENSE\_ACC\_IP}$ (offset) and TC_{SENSE} (angle).

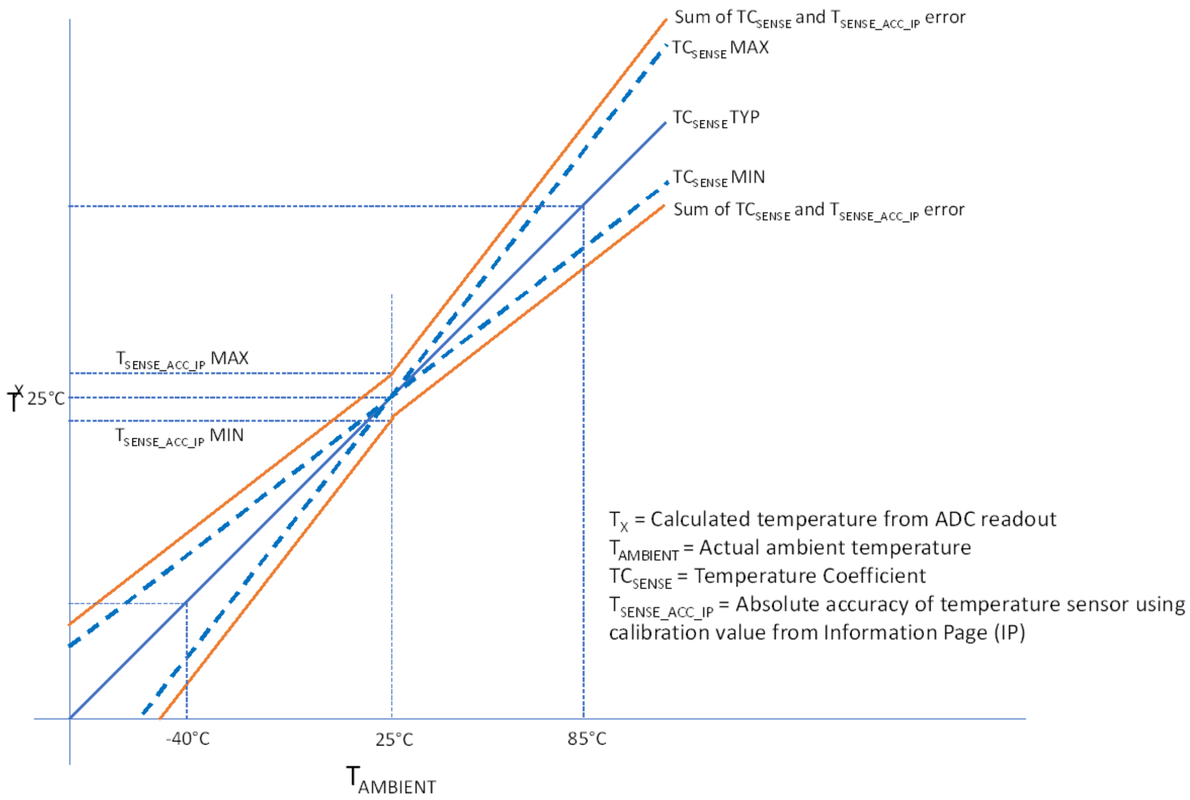

Figure 49. Temperature sensor behavior

 |

The recommended formula for single point calibrated temperature reading is as follows:

$$T_x = ADC_{IP\_CAL\_T} + (ADC_x - ADC_{IP\_CAL\_VAL})/(TC_{SENSE} \times 64)$$

Where:

- T_x = calculated single point calibrated die temperature in °C
- ADC_x = 16-bit GP_ADC_VAL readout (converted to decimal) at temperature T_x
- $ADC_{IP\_CAL\_VAL}$ = device calibration value recorded during production testing (based on the 16-bit readout, stored in IP sector)
- $ADC_{IP\_CAL\_T}$ = die temperature applied during device calibration (stored in IP sector)
- TC_{SENSE} = temperature coefficient in LSB/°C
- 64 = correction for 16-bit to 10-bit ADC values.

For uncalibrated temperature sensor measurements, $ADC_{IP\_CAL\_VAL}$ and $ADC_{IP\_CAL\_VAL\_T}$ can be replaced by the default value using the formula below:

$$T_x = 25 + (ADC_x - 43530) / (TC_{SENSE} \times 64)$$

Note that this is not recommended since it can result in large offsets.

NOTE
While measuring and/or calibration, the system's power dissipation should be kept the same, otherwise the measurement is affected by the internal thermal gradient.

23.3 Programming

There is a certain programming sequence required to read the temperature sensor. There are two reading options available:

- Absolute temperature (single-point calibration)
- Relative temperature.

23.3.1 Absolute temperature

A calibration value at $ADC_{IP\_CAL\_T}$ °C is stored in IP sector for absolute temperature measurements. When the calibration value from IP is used, the GP_ADC offset calibration settings should be used.

NOTE
Absolute temperature reading is indicative only.

- To read the production test calibration value:
  - Read $ADC_{IP\_CAL\_T}$: the content of $ADC_{IP\_CAL\_T}$ is in SDK_SECTION_TEMP_SENS_25C
  - Read $ADC_{IP\_CAL\_VAL}$: the content of $ADC_{IP\_CAL\_VAL}$ is in SDK_SECTION_TEMP_SENS_25C
- Use factory calibrated offset values for the GPADC
  - GP_ADC_OFFP_REG[GP_ADC_OFFP] = SDK_SECTION_GP_ADC_SINGLE_MODE, B4 - B5
  - GP_ADC_OFFN_REG[GP_ADC_OFFN] = SDK_SECTION_GP_ADC_SINGLE_MODE, B6 - B7
- To enable the temperature sensor:
  - GP_ADC_CTRL_REG[DIE_TEMP_EN] = 1
- Wait 25 µs for the temperature sensor to start up
- To set the advised ADC settings:
  - GP_ADC_CTRL_REG[GP_ADC_CHOP] = 1
  - GP_ADC_CTRL_REG[GP_ADC_SE] = 1
  - GP_ADC_CTRL_REG[GP_ADC_EN] = 1

   - GP_ADC_CTRL2_REG[GP_ADC_I20U] = 1
   - GP_ADC_SEL_REG[GP_ADC_SEL_P] = 4
   - GP_ADC_CTRL2_REG[GP_ADC_STORE_DEL] = 0
- To set sample time and averaging of the ADC sampling
   - GP_ADC_CTRL2_REG[GP_ADC_SMPL_TIME] = 5
   - GP_ADC_CTRL2_REG[GP_ADC_CONV_NRS] = 6
- To perform ADC conversion:
   - GP_ADC_CTRL_REG[GP_ADC_START] = 1
- To wait for the conversion to finish, read the register
   - GP_ADC_RESULT_REG[GP_ADC_VAL].

23.3.2 Relative temperature

For relative temperature measurements, single-point calibration is not needed. The programming sequence is presented below.

- To enable GP_ADC:
   - GP_ADC_CTRL_REG[DIE_TEMP_EN] = 1.
- Wait 25 μs for the temperature sensor to start up.
- To set the advised ADC settings:
   - GP_ADC_CTRL_REG[GP_ADC_CHOP] = 1.
   - GP_ADC_CTRL_REG[GP_ADC_SE] = 1.
   - GP_ADC_CTRL_REG[GP_ADC_EN] = 1.
   - GP_ADC_CTRL2_REG[GP_ADC_I20U] = 1.
   - GP_ADC_SEL_REG[GP_ADC_SEL_P] = 4.
   - GP_ADC_CTRL2_REG[GP_ADC_STORE_DEL] = 0.
- To set sample time and averaging of the ADC sampling:
   - GP_ADC_CTRL2_REG[GP_ADC_SMPL_TIME] = 5.
   - GP_ADC_CTRL2_REG[GP_ADC_CONV_NRS] = 6.
- To perform ADC conversion:
   - GP_ADC_CTRL_REG[GP_ADC_START] = 1.
- To wait for the conversion to finish, read the register:
   - GP_ADC_RESULT_REG[GP_ADC_VAL].

24 Audio Unit

24.1 Introduction

The Audio Unit (AU) is made up of two digital interfaces, namely: a PDM and PCM. It also has two Sampling Rate Converters (SRC1, SRC2) that are used for adjusting the sampling rate of audio samples between the two interfaces and memory.

The PDM interface provides a serial connection for one stereo or two mono input devices (for example, MEMS microphones) or output devices. The interface has a single clock PDM_CLK and one input/output PDM_DI/PDM_DO that can carry two channels in a time divided manner.

The PCM controller implements up to 48 kHz synchronous interface to external audio devices, ISDN circuits, and serial data interfaces. It enables master and slave modes, and supports I2S and TDM formats.

The AU has dedicated DMA channels for the PCM and SRC streams. The PCM data flow is further supported by an internal dedicated 8x32-bit FIFO that cannot be used in stereo mode.

Features

- Supported conversions:
  - SRC_IN (32 bits) to SRC_OUT (32 bits)
  - PDM_IN (1bit) to SRC_OUT (32 bits)
  - SRC_IN (32 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 (for example, PCM_FSC)
- Manual mode to generate interrupts at the programmed sample rate. Adjustment is done by software based on buffer pointers
- PCM (Master/Slave) interface:
  - PCM_FSC
    - Master/slave 4 kHz to 48 kHz
    - Strobe Length 1, 8, 16, 24, 32, 40, 48, and 64 bits
    - PCM_FSC before or on the first bit. (In Master mode)
  - 2x32 channels
  - Programmable slot delay up-to 31*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 32 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.

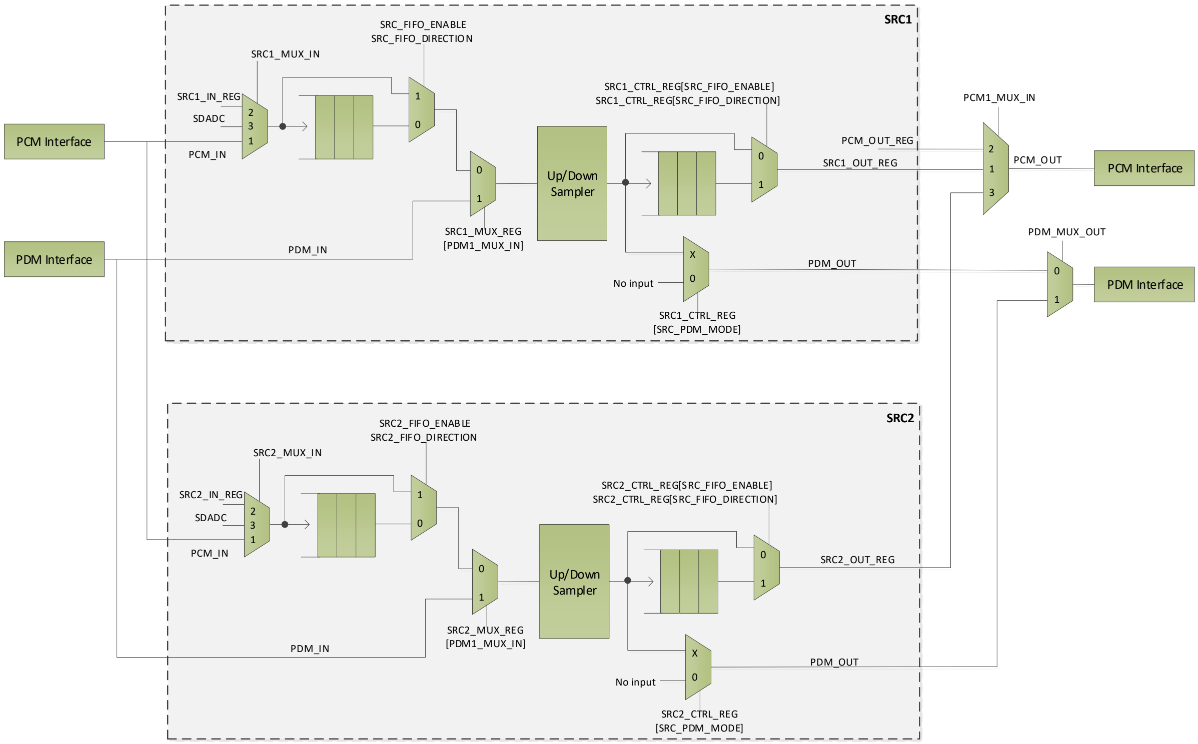

Figure 50. Audio unit block diagram

24.2 Architecture

24.2.1 Data paths

The SRC blocks convert two 32-bit channels into either a stereo pair, or two mono streams. PCM linear data pairs are transferred to SRCx_IN1/2; the output is 2x32-bit left-aligned on SRCx_OUT1/2. The two 1-bit PDM data inputs are received on PDM_IN and are converted to 2x32 bits, left-aligned to SRC_OUT.

The SRCx_IN input multiplexer (Figure 50) is controlled by SRC1_MUX_REG[PDM1_MUX_IN]. The input of these multiplexers comes from the audio interfaces or the SRCx_IN1/2_REG. The data to these registers is left-aligned, bits 31-8 are mapped on bits 23-0 of the SRC.

The 32 bits SRCs outputs can be read in SRCx_OUT1_REG and SRCx_OUT2_REG and can also be routed to the PCM interface. The input selection of these multiplexers is also controlled by SRC1_MUX_REG[PCM1_MUX_IN].

An 8x32-bit FIFO can be connected to the SRCs input or output data path when not in stereo mode. When the FIFO services the SRC input data path, an SRC input event triggers a FIFO write, while a CPU read access triggers a FIFO read. SRC_IN_IRQ is issued when the FIFO level increments to four samples. When the FIFO services the SRC output data path, an SRC output event triggers a FIFO read, while a CPU write access triggers a FIFO write. SRC_OUT_IRQ is issued when the FIFO level drops below five samples. FIFO operation can be completely disabled.

SRCs 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 SRCx_IN_FS_REG/SRCx_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 SRCx_IN_FS_REG/SRCx_OUT_FS_REG registers.

When PDM is used (input/output), SRCs operate in automatic mode. The sampling rate reported in SRCx_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.

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.

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_IN2/OUT2_REG for input and output directions respectively. These registers can be arranged as eight 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 31 x 8 bits 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) must be generated according to the required sample rate. There are two ways of generating the clock:

1. **The Fractional option**. Dividing the system clock by an integer and a fractional part (inserting jitter in the clock pulse train). This is programmed in 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 defines the numerator of the fraction as explained in the example of Table 94.

Table 94: 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 Table 95, with 8 bits, 16 bits, 32 bits, and 48 bits in various sample rates.

Table 95: Fractional and Integer Only Clock Divisors for Various PCM Frequencies and Sample Rates

			XTAL (kHz)					Doubler (kHz)				
			32000					64000				
				Fractional Option		Integer Only Option			Fractional Option		Integer Only Option	
Sample Rate (kHz)	Bits	Desired Bit Clock (kHz)	Desired Divider	PCM_DIV_REG	PCM_FDIV_REG	PCM_DIV_REG	Actual Wordsize (Bits)	Desired Divider	PCM_DIV_REG	PCM_FDIV_REG	PCM_DIV_REG	Actual Wordsize (Bits)
8	1*8	64	500	500		500	8	1000	1000		1000	8
8	1*16	128	250	250		250	16	500	500		500	16
8	1*24	192	166,667	166	2/3	160	25	333,333	333		320	25
8	1*32	256	125	125		125	32	250	250		250	32
8	2*8	128	250	250		250	8	500	500		500	8
8	2*16	256	125	125		125	16	250	250		250	16
8	2*24	384	83,333	83	1/3	80	25	166,667	166		160	25
8	2*32	512	62,5	62	1/2	50	40	125	125	0	125	32
16	1*8	128	250	250		250	8	500	500		500	8
16	1*16	256	125	125		125	16	250	250		250	16

			XTAL (kHz)					Doubler (kHz)				
			32000					64000				
				Fractional Option		Integer Only Option			Fractional Option		Integer Only Option	
Sample Rate (kHz)	Bits	Desired Bit Clock (kHz)	Desired Divider	PCM_DIV_REG	PCM_FDIV_REG	PCM_DIV_REG	Actual Wordsize (Bits)	Desired Divider	PCM_DIV_REG	PCM_FDIV_REG	PCM_DIV_REG	Actual Wordsize (Bits)
16	1*24	384	83,333	83	1/3	80	25	166,667	166		160	25
16	1*32	512	62,5	62	1/2	50	40	125	125	0	125	32
16	2*8	256	125	125		125	8	250	250		250	8
16	2*16	512	62,5	62	1/2	50	20	125	125	0	125	16
16	2*24	768	41,667	41	2/3	40	25	83,333	83		80	25
16	2*32	1024	31,25	31	1/4	25	40	62,5	62	1/2	50	40
32	1*8	256	125	125		125	8	250	250		250	8
32	1*16	512	62,5	62	1/2	50	20	125	125	0	125	16
32	1*24	768	41,667	41	2/3	40	25	83,333	83		80	25
32	1*32	1024	31,25	31	1/4	25	40	62,5	62	1/2	50	40

			XTAL (kHz)					Doubler (kHz)				
			32000					64000				
				Fractional Option		Integer Only Option			Fractional Option		Integer Only Option	
Sample Rate (kHz)	**Bits**	**Desired Bit Clock (kHz)**	**Desired Divider**	**PCM_DIV_REG**	**PCM_FDIV_REG**	**PCM_DIV_REG**	**Actual Wordsize (Bits)**	**Desired Divider**	**PCM_DIV_REG**	**PCM_FDIV_REG**	**PCM_DIV_REG**	**Actual Wordsize (Bits)**
32	2*8	512	62,5	62	1/2	50	10	125	125	0	125	8
32	2*16	1024	31,25	31	1/4	25	20	62,5	62	1/2	50	20
32	2*24	1536	20,833	20	5/6	20	25	41,667	41	2/3	40	25
32	2*32	2048	15,625	15	5/8	10	50	31,25	31	1/4	25	40
48	1*8	384	83,333	83	1/3	N/A	N/A	166,667	166		N/A	N/A
48	1*16	768	41,667	41	2/3	N/A	N/A	83,333	83		N/A	N/A
48	1*24	1152	27,778	27	7/9	N/A	N/A	55,556	55	5/9	N/A	N/A
48	1*32	1536	20,833	20	5/6	N/A	N/A	41,667	41	2/3	N/A	N/A
48	2*8	768	41,667	41	2/3	N/A	N/A	83,333	83		N/A	N/A
48	2*16	1536	20,833	20	5/6	N/A	N/A	41,667	41	2/3	N/A	N/A

			XTAL (kHz)					Doubler (kHz)				
			32000					64000				
				Fractional Option		Integer Only Option			Fractional Option		Integer Only Option	
Sample Rate (kHz)	Bits	Desired Bit Clock (kHz)	Desired Divider	PCM_DIV_REG	PCM_FDIV_REG	PCM_DIV_REG	Actual Wordsize (Bits)	Desired Divider	PCM_DIV_REG	PCM_FDIV_REG	PCM_DIV_REG	Actual Wordsize (Bits)
48	2*24	2304	13,889	13	8/9	N/A	N/A	27,778	27	7/9	N/A	N/A
48	2*32	3072	10,417	10	2/5	N/A	N/A	20,833	20	5/6	N/A	N/A

The yellow marked fields designate that the actual word size achieved in the Integer Only option, is larger than the required bits. The last clock pulses is 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 possible. A higher clock is generated (200 kHz), which results in a word of 25 bits. In this case, the last bit is 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 for example, 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] from 8 to 48 kHz.

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 3.072 MHz.

In slave mode, the main counter can be stopped and resumed on a PCM_FSC rising edge.

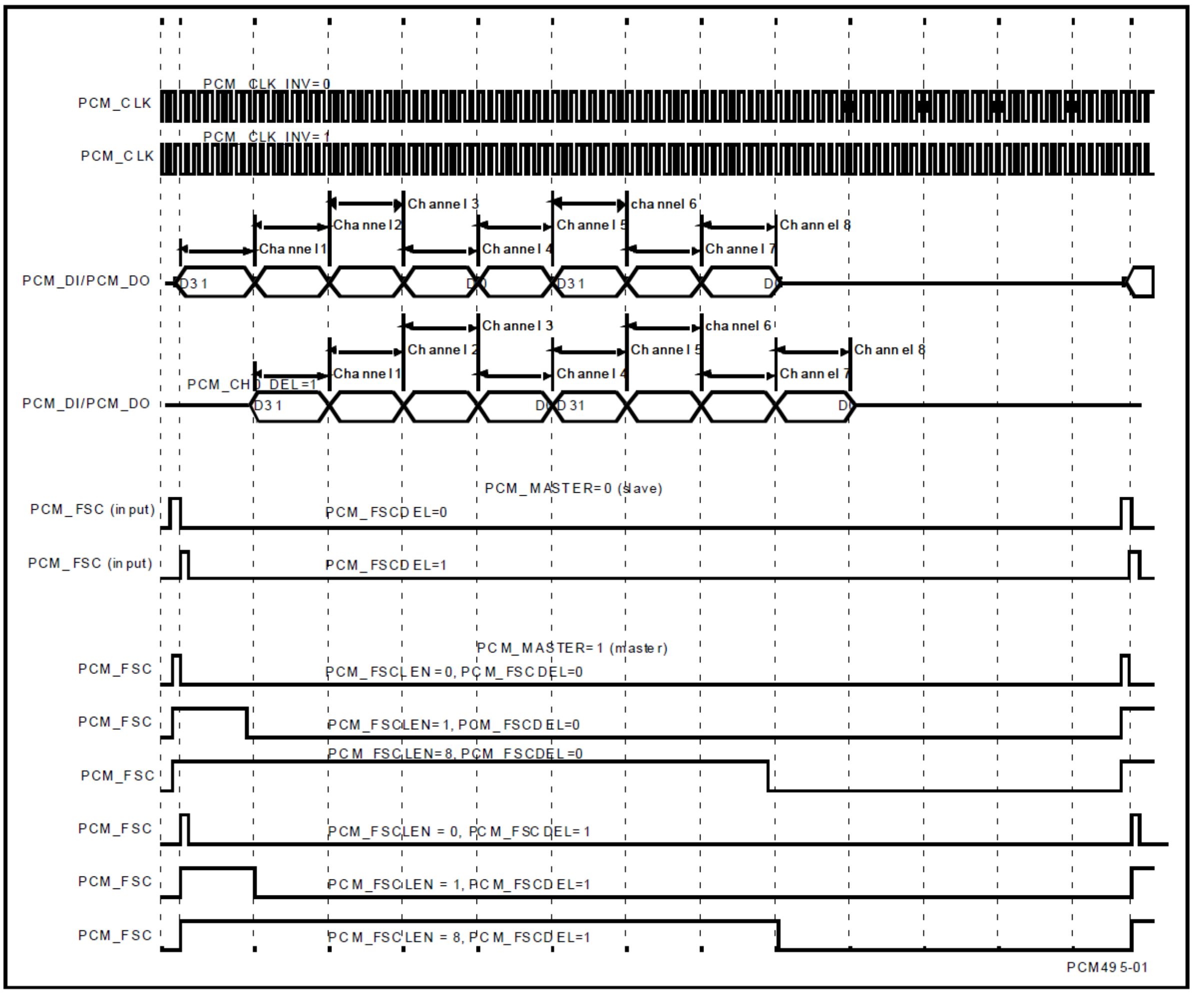

Figure 51. PCM interface formats

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_CH_DEL: 0 (no channel delay).

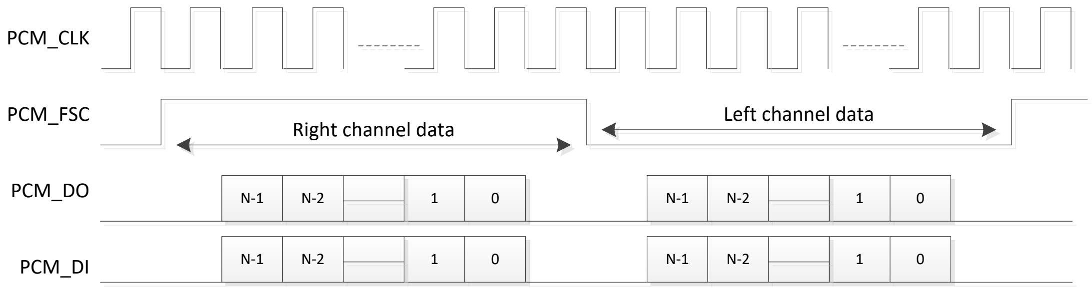

Figure 52. I2S mode

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 shown in Figure 53, 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, left and right alignment can also 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.

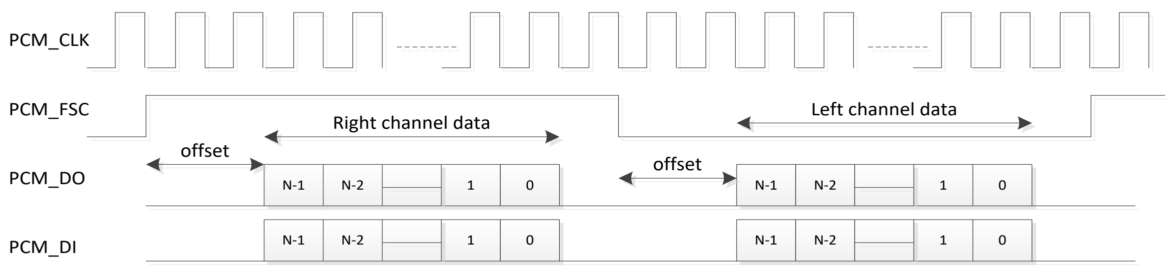

Figure 53. TDM mode (left justified mode)

NOTE
Offset is always in multiples of 8.

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 clocked 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: 1 (no bit delay)
- PCM_CLK_INV: 0 (output on rising edge)

- PCM_CH0_DEL: 0 (no delay)
- PCM_CLK_BIT: 1.

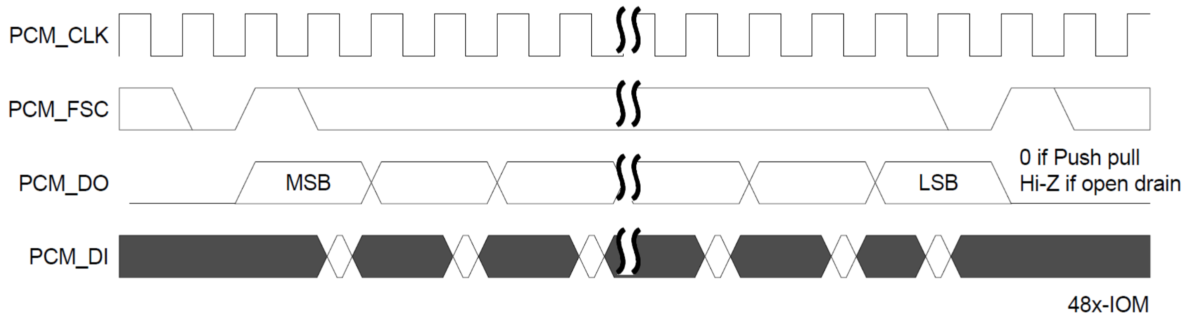

Figure 54. IOM format

24.2.4 PDM interface

The PDM comprises two 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 55.

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 = 32 and PID = 33 for DATA and CLK respectively in the Px_yz_MODE_REG.

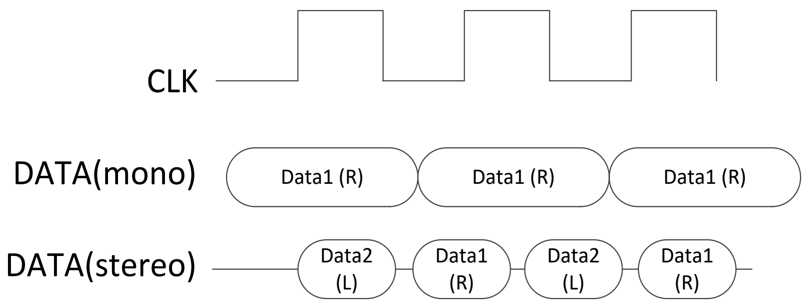

Figure 55. PDM mono/stereo formats

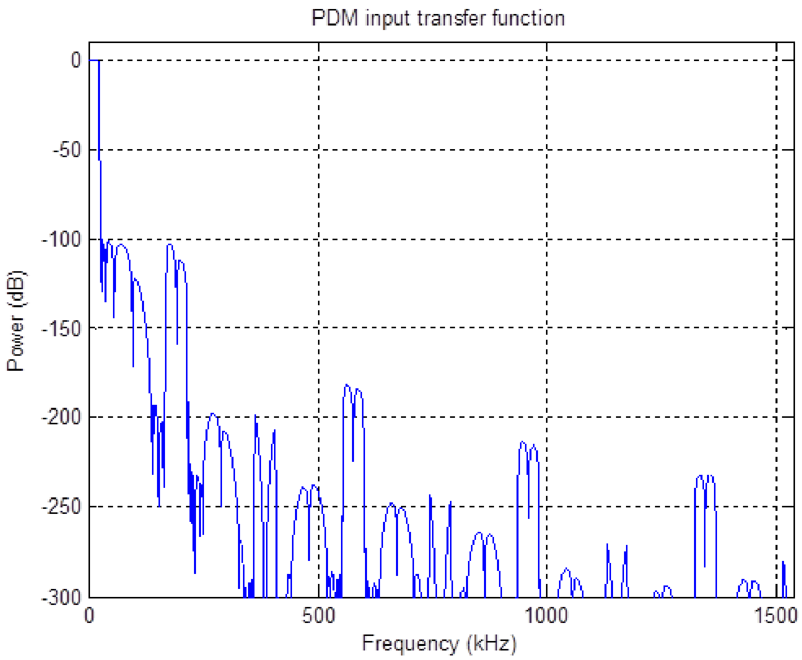

Figure 56. SRC PDM input transfer function

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 needs to be transferred to or 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. The channels that are reserved in the DMA can support the PCM, the SRCx (IN) and the SRCx (OUT) directions.

24.2.6 Interrupts

After a Sample Rate Conversion, the input up-sampler and output down-sampler, generate edge triggered interrupts on SRCx_IN_SYNC and SRCx_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 (there are no FIFOs included).

24.3 Programming

24.3.1 PDM input to PCM output

To configure the Audio Unit:

1. Configure the GPIOs functionality used for the PDM and PCM I/F by writing the appropriate Px_yy_MODE_REG[PID].
2. Configure the GPIOs direction (Px_yy_MODE_REG[PUPD]).
3. Configure PDM I/F:
   a. Configure as Master by setting the PDM_DIV_REG[PDM_MASTER_MODE] bit.
   b. Set PDM clock divider (PDM_DIV_REG[PDM_DIV]).
   c. Enable PDM (internal) block clock (PDM_DIV_REG[CLK_PDM_EN]).
4. Configure PCM I/F:
   a. Select the clock source (PCM_DIV_REG[PCM_SRC_SEL]).

   b. Set up PCM clock division (PCM_DIV_REG[PCM_DIV], PCM_FDIV_REG).
   c. Enable the clock (master mode) by setting the PCM_DIV_REG[CLK_PCM_EN] bit.
   d. Disable PCM (PCM1_CTRL_REG[PCM_EN] = 0).
   e. Set PCM Framesync divider (PCM1_CTRL_REG[PCM_FSC_DIV]).
   f. (PCM1_CTRL_REG[PCM_FSC_EDGE]).
   g. Set channel delay in multiples of 8 bits (PCM1_CTRL_REG[PCM_CH_DEL]).
   h. Set the number of clock cycles per data bit (PCM1_CTRL_REG[PCM_CLK_BIT]).
   i. Set polarity of PCM FSC (PCM1_CTRL_REG[PCM_FSCINV]).
   j. Set polarity of PCM CLK(PCM1_CTRL_REG[PCM_CLKINV]).
   k. Set PCM DO output mode (PCM1_CTRL_REG[PCM_PPOD]).
   l. Set PCM FSC start time (PCM1_CTRL_REG[PCM_FSCDEL]).
   m. Set PCM FSC data length (PCM1_CTRL_REG[PCM_FSCLEN]).
   n. Set PCM in Master mode (PCM1_CTRL_REG[PCM_MASTER] = 1).
5. Configure the Sample Rate Converter:
   a. Set the SRC clock divider (SRC_DIV_REG[SRC/2_DIV]).
   b. Enable the SRC block clock by setting the SRC_DIV_REG[CLK_SRC/2_EN] bit.
   c. Select the SRC input Up Sampling IIR filters setting according to the sample rate (SRCx_CTRL_REG[SRC_IN_DS]).
   d. Configure the SRC input sample rate (SRCx_IN_FS_REG).
   e. Select the SRC output Up Sampling IIR filters setting according to the sample rate (SRCx_CTRL_REG[SRC_OUT_US]).
   f. Configure the SRC output sample rate (SRCx_OUT_FS_REG).
   g. Select the PDM as input to SRC (SRCx_MUX_REG[PDM1_MUX_IN] = 1).
   h. Enable the SRC FIFO (SRCx_CTRL_REG[SRC_FIFO_ENABLE]) and set direction (SRCx_CTRL_REG[SRC_FIFO_DIRECTION] = 1). Note that in stero mode, FIFO cannot be used.
   i. Select the output to PCM (SRCx_MUX_REG[PCM1_MUX_IN] = 1).
   j. Set SRCx_MUX_REG[SRC1_MUX_IN] = 0.
   k. Set SRC input to Automatic conversion mode (SRCx_CTRL_REG[SRC_IN_AMODE] = 1).
6. Enable SRC (SRCx_CTRL_REG[SRC_EN] = 1).
7. Enable PCM (PCM1_CTRL_REG[PCM_EN] = 1).

25 I2C Interface

25.1 Introduction

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. It comprises 32 levels deep FIFO in both directions.

Features

- Two-wire I2C serial interface consists of a serial data line (SDA) and a serial clock (SCL)
- Three speeds are supported:
  - Standard mode (0 to 100 kbit/s)
  - Fast mode (<= 400 kbit/s)
  - High Speed mode (<= 3.4 Mbit/s)
- Clock synchronization
- 32 locations deep transmit/receive FIFOs (32 x 8-bit RX, 32 x 10-bit TX)
- 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.

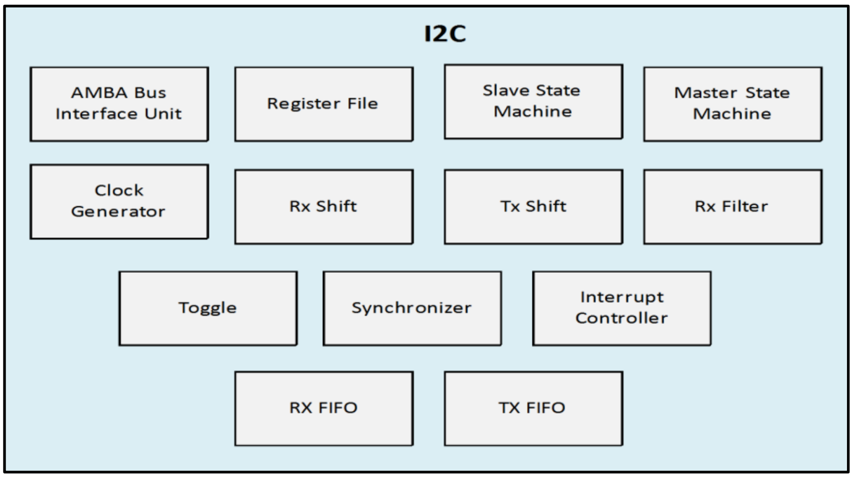

Figure 57. I2C controller block diagram

Figure 57 shows the I2C Controller block diagram. 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
  - Sup the data and hold the data.
- RX Shift. Takes data into the design and extracts it in byte format.
- TX Shift. Presents data supplied by CPU for transfer on the I2C bus.
- RX Filter. Detects the events in the bus; for example, start, stop and arbitration lost.
- Toggle. Generates pulses on both sides and toggles to transfer signals across clock domains.
- Synchronizer. Transfers signals from one clock domain to another.
- Interrupt Controller. Generates the raw interrupt and interrupt flags, allowing them to be set and cleared.
- RX FIFO/TX. Holds the RX FIFO and TX FIFO register banks and controllers, along with their status levels.

25.2 Architecture

25.2.1 I2C behavior

I2C can be controlled (through software) to be a 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 transmitting or receiving data to and from the master. Data acknowledgement is sent by the device that receives data, which can be master or slave. The I2C protocol allows multiple masters to reside on the I2C bus. It 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. Figure 58 shows this behavior.

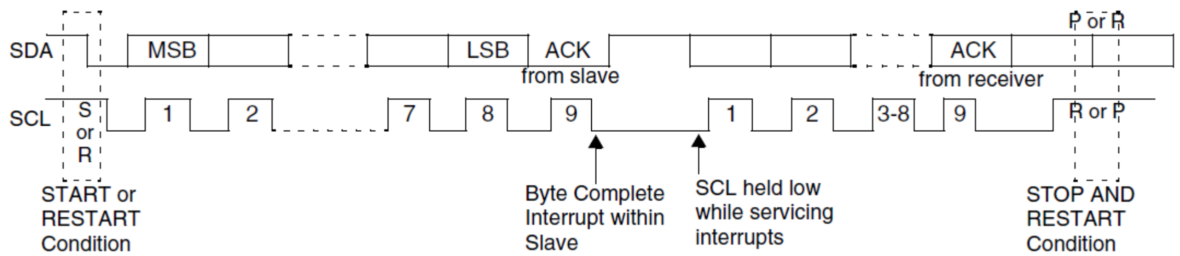

Figure 58. Data transfer on the I2C bus

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

25.2.1.1 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. Writing a 1 to I2C_DATA_CMD_REG[9] causes the i2c to generate a STOP condition on the I2C bus; a STOP condition is not issued if this bit is not set, even if the transmit FIFO is empty.

When operating as a slave, the I2C Controller does not generate START and STOP conditions, as per the protocol. However, if a read request is made to the I2C Controller, it holds the SCL line low until read data has been supplied to it. This stalls the I2C bus until read data is provided to the slave I2C Controller, or the I2C Controller slave is disabled by writing a 0 to I2C_ENABLE.

25.2.1.2 Combined formats

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

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

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

25.2.2 I2C protocols

The I2C Controller has the following protocols:

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

25.2.2.1 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 59 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.

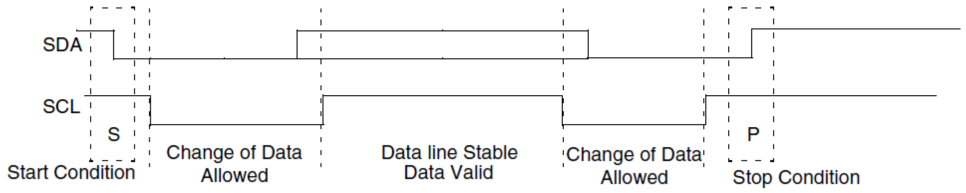

Figure 59. START and STOP conditions

NOTE
The signal transitions for the START/STOP conditions, as shown in Figure 59, reflect those observed at the output signals of the Master driving the I2C bus. Care should be taken when observing the SDA/SCL signals at the input signals of the Slave(s), because unequal line delays may result in an incorrect SDA/SCL timing relationship.

25.2.2.2 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 60. 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.

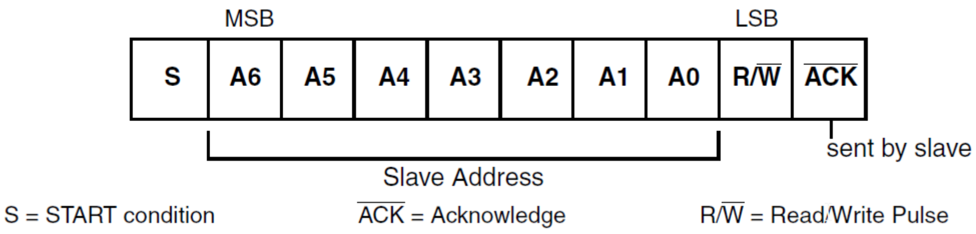

Figure 60. 7-bit address format

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 61 shows the 10-bit address format, and Table 96 defines the special purpose and reserved first byte addresses.

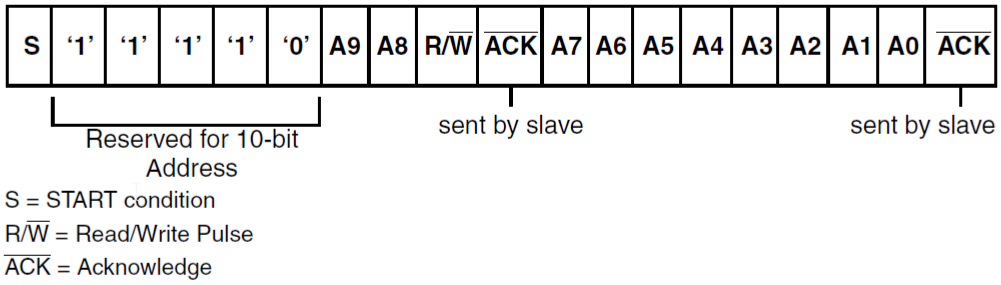

Figure 61. 10-bit address format

Table 96: 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, see "START BYTE Transfer Protocol 0000
0000 001	X	CBUS address. I2C Controller ignores these accesses
0000 010	X	Reserved

Slave Address	R/W Bits	Description
0000 011	X	Reserved
0000 1XX	X	High-speed master code. For more information, see "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.

25.2.2.3 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 62, then the slave-receiver responds to the master-transmitter with an acknowledge pulse after every byte of data is received.

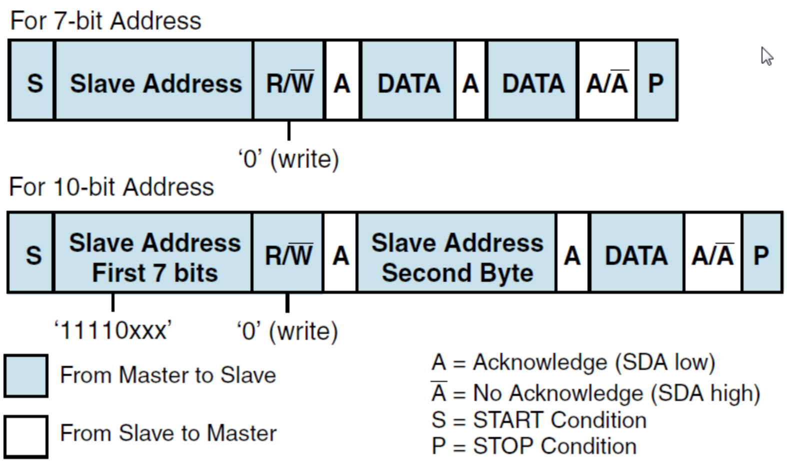

Figure 62. Master-transmitter protocol

Master-receiver and slave-transmitter

If the master is receiving data as shown in Figure 63 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.

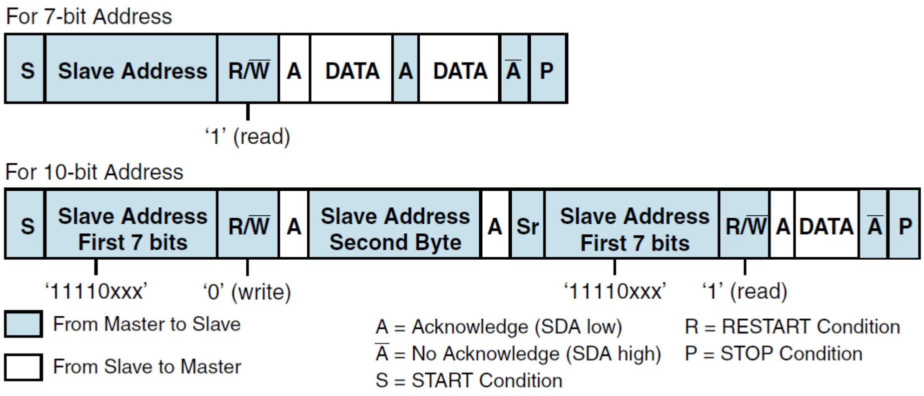

Figure 63. Master-receiver protocol

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 shown in Figure 64. 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.

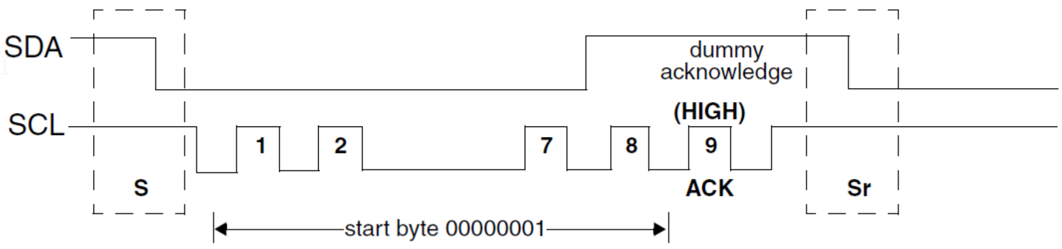

Figure 64. START BYTE transfer

The START BYTE procedure is as follows:

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

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

25.2.3 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 65 shows 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.

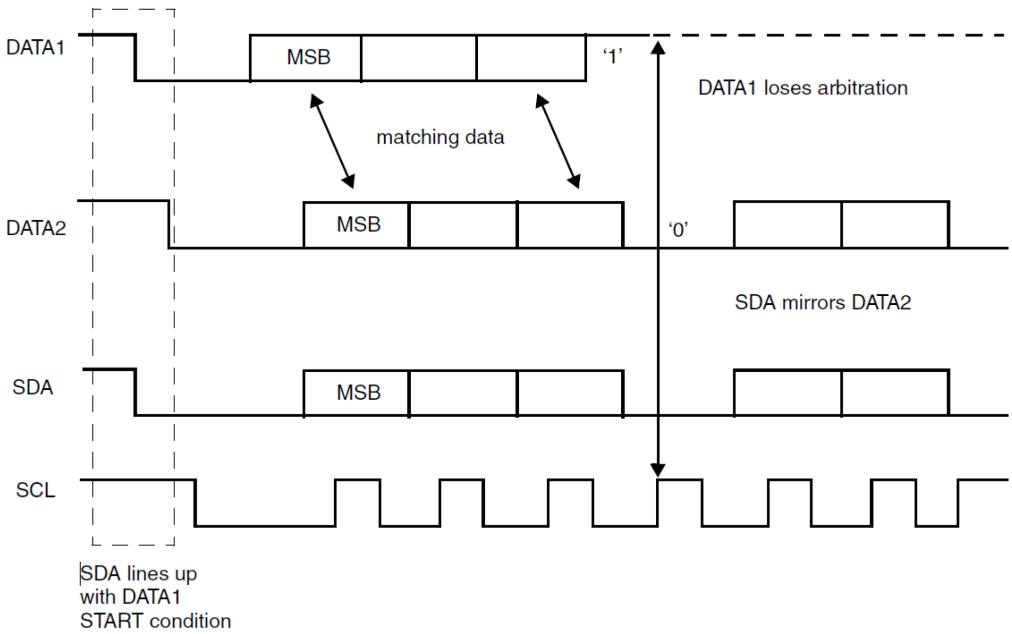

Figure 65. Multiple master arbitration

25.2.4 Clock synchronization

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

All masters then count off their high time, and the master with the shortest high time, transitions the SCL line to 0. The masters then count out their low time. 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 shown in Figure 66. Optionally, slaves may hold the SCL line low, to slow down the timing on the I2C bus.

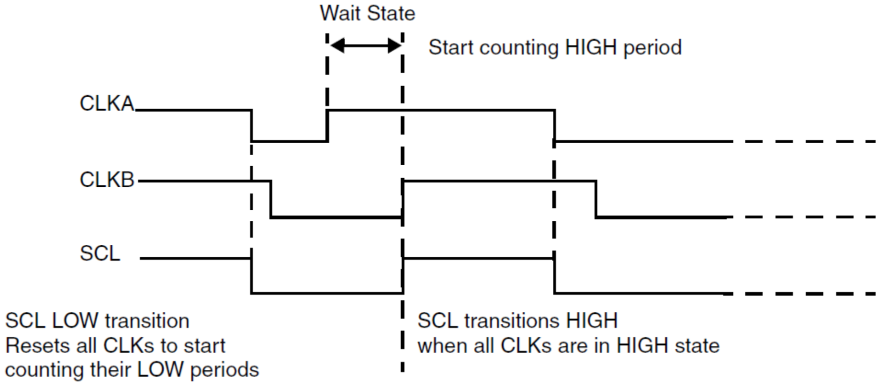

Figure 66. Multiple master clock synchronization

25.3 Programming

To configure and use the I2C Controllers:

1. Set up the GPIOs to be used for the I2C interface (Px_yy_MODE_REG[PID] = 14 to 15).
2. Configure I2C clock frequency.
   For CLK_COM_REG[I2C_CLK_SEL] = 0 (DivN clock):
   a. Standard mode (100 kbits/s) : I2C_CON_REG[I2C_SPEED] = 1.
   b. Full-speed mode (400 kbits/s) : I2C_CON_REG[I2C_SPEED] = 2.
   c. High-speed mode (<= 3.4 Mbit/s) : I2C_CON_REG[I2C_SPEED] = 3.
3. Set up the Controller as:
   a. Master: I2C_CON_REG[I2C_MASTER_MODE] = 1 and I2C_CON_REG[I2C_SLAVE_DISABLE] = 1.
   b. Slave: I2C_CON_REG[I2C_MASTER_MODE] = 0 and I2C_CON_REG[I2C_SLAVE_DISABLE] = 0.
4. Choose whether the controller starts its transfers in 7-bit or 10-bit addressing mode when acting as a master (I2C_CON_REG[I2C_10BITADDR_MASTER]) or when acting as a slave, whether the controller responds to 7-bit or 10-bit addresses (I2C_CON_REG[I2C_10BITADDR_SLAVE]).
5. Set the target slave address in:
   a. Master mode (I2C_TAR_REG[IC_TAR] = 0x55 (default)).
   b. Slave mode (I2C_SAR_REG[IC_SAR] = 0x55 (default)).
6. Set threshold level on RX and TX FIFO (I2C_RX_TL_REG, I2C_TX_TL_REG).
7. Enable the required interrupts (I2C_INTR_MASK_REG).
8. Enable the I2C Controller by setting the CLK_COM_REG[I2C_ENABLE] bit.
9. Read a byte:
   a. Prepare to transmit the read command byte (I2C_DATA_CMD_REG[I2C_CMD] = 1).
   b. Wait until TX FIFO is empty (I2C_STATUS_REG[TFE] = 1).
   c. Wait until the master has finished reading the byte from slave device (I2C_STATUS_REG[MST_ACTIVITY] = 0).
10. Write a byte:
   a. Prepare to transmit the write command byte (I2C_DATA_CMD_REG[I2C_CMD] = 0 and I2C_DATA_CMD_REG[I2C_DAT] = command byte).
   b. Wait until TX FIFO is empty (I2C_STATUS_REG[TFE] = 1).

c. Wait until the master has finished reading the response byte from slave device (I2C_STATUS_REG[MST_ACTIVITY] = 0).

26 UART

26.1 Introduction

The DA14592 contains two instances of this block: UART and UART2.

UART and UART2 are compliant to industry-standard 16550 and are used for serial communication with a peripheral. Data is written from a master (CPU) over the APB bus to the UART/2. It is converted to serial form and transmitted to the destination device. Serial data is also received by the UART/2 and stored for the master (CPU) to read back.

There is also DMA support on the UART blocks, so the internal FIFOs can be used. UART2 supports hardware flow control signals (RTS, CTS).

Features

- Dedicated 16 bytes Transmit and 16 bytes Receive FIFO for each UART
- Hardware flow control and 9-bit mode support (CTS/RTS, UART2)
- Shadow registers reduces software overhead and include a software programmable reset
- Transmitter Holding Register Empty (THRE) interrupt mode
- Functionality based on 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 x divisor) and the fractional part UART_DLF/16
- ISO7816 support (UART2).

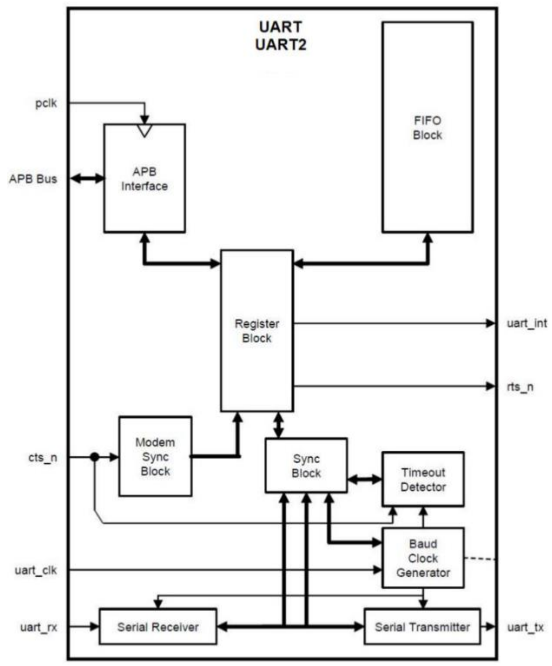

Figure 67. UART block diagram

26.2 Architecture

26.2.1 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 68.

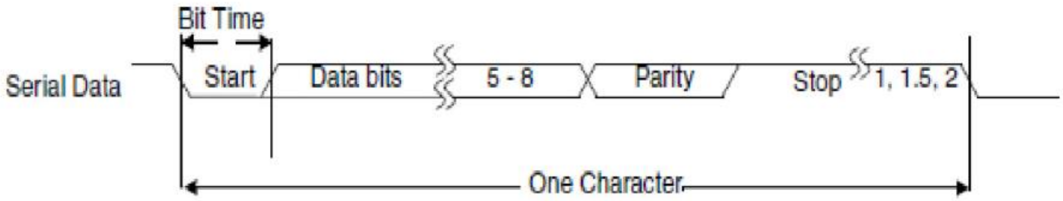

Figure 68. Serial data format

An additional parity bit may be added to the serial character. This bit appears after the last data bit, but before the stop bit(s) in the character structure. It provides 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 of 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 Bit Time 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 69 shows the sampling points of the first couple of bits in a serial character.

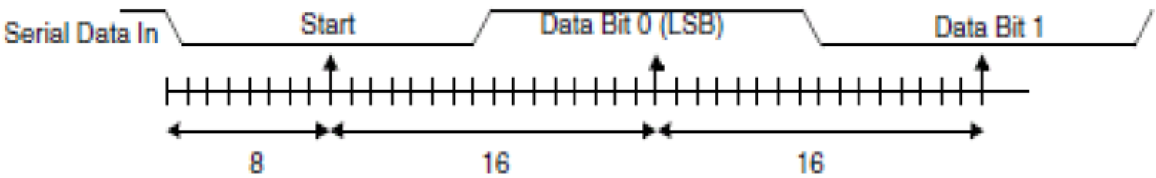

Figure 69. Receiver serial data sampling points

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). Table 97 and Table 98 show the registers settings for common baud rate values.

Table 97: UART/2 Baud Rate Generation on DIVN

Baud Rate (Note 1)	Divider	Divisor Latch	DLH Reg	DLL Reg	DLF Reg	Actual BR	Error %
1200	1666.667	1666.6875	6	130	11	1199.99	0.00
2400	833.333	833.3125	3	65	5	2400.06	0.00
4800	416.667	416.6875	1	160	11	4799.76	0.00

Baud Rate (Note 1)	Divider	Divisor Latch	DLH Reg	DLL Reg	DLF Reg	Actual BR	Error %
9600	208.333	208.3125	0	208	5	9600.96	0.01
14400	138.889	138.875	0	138	14	14401.44	0.01
19200	104.167	104.1875	0	104	3	19196.16	0.02
28800	69.444	69.4375	0	69	7	28802.88	0.01
38400	52.083	52.0625	0	52	1	38415.37	0.04
57600	34.722	34.75	0	34	12	57553.96	0.08
115200	17.361	17.375	0	17	6	115107.91	0.08
230400	8.681	8.6875	0	8	11	230215.83	0.08
460800	4.340	4.3125	0	4	5	463768.12	0.64
921600	2.170	2.1875	0	2	3	914285.71	0.79
1000000	2	2	0	2	0	1000000	0.00

Note 1 Values are valid for UART CLK = 32 MHz (divN_clk).

Table 98: UART/2 Baud Rate Generation on Doubler

Baud Rate (Note 1)	Divider	Divisor Latch	DLH Reg	DLL Reg	DLF Reg	Actual BR	Error %
1200	3333.333	3333.3125	13	5	5	1200.01	0.00
2400	1666.667	1666.6875	6	130	11	2399.97	0.00
4800	833.333	833.3125	3	65	5	4800.12	0.00
9600	416.667	416.6875	1	160	11	9599.52	0.00
14400	277.778	277.75	1	21	12	14401.44	0.01
19200	208.333	208.3125	0	208	5	19201.92	0.01
28800	138.889	138.875	0	138	14	28802.88	0.01
38400	104.167	104.1875	0	104	3	38392.32	0.02
57600	69.444	69.4375	0	69	7	57605.76	0.01
115200	34.722	34.75	0	34	12	115107.91	0.08
230400	17.361	17.375	0	17	6	230215.83	0.08
460800	8.681	8.6875	0	8	11	460431.65	0.08
921600	4.340	4.3125	0	4	5	927536.23	0.64
1000000	4	4	0	4	0	1000000	0.00
3000000	1.333333333	1.3125	0	1	5	3047619.048	1.59

Note 1 Values are valid for CLK_COM_REG[UART/2_CLK_SEL] = 1 and sys_clk = 64 MHz. For CLK_COM_REG[UART/2_CLK_SEL] = 0, see Table 97.

26.2.2 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.

26.2.3 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. Table 99 shows these interrupt types in more detail.

Table 99: 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 four character times and there is at least one 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	Busy detect	Line Control Register was written while the UART is busy (RX or TX line is low).	Reading the UART status register.

26.2.4 Programmable THRE interrupt

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

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

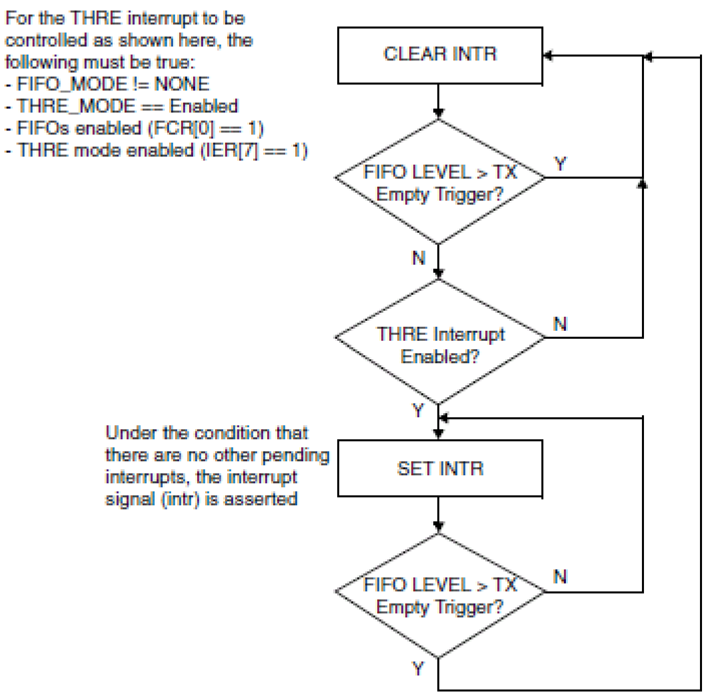

Figure 70. Flowchart of interrupt generation for programmable THRE interrupt mode

This threshold level is programmed into FCR[5:4]. The available empty thresholds are: empty, 2, 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. The flow then becomes, "fill transmitter FIFO whenever an interrupt occurs and there is data to transmit", instead of waiting until the FIFO is completely empty. Waiting until the FIFO is empty, causes a performance hit whenever the system is too busy to respond immediately.

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

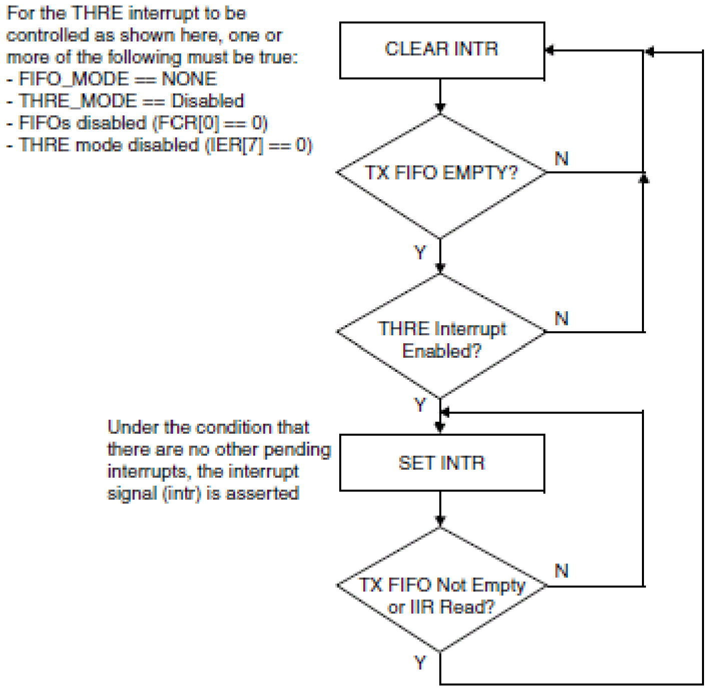

Figure 71. Flowchart of interrupt generation when not in programmable THRE interrupt mode

26.2.5 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 supports a host burst mode where the host increments its address, but still accesses the same Receive buffer register.
- UART_STHR supports 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.

26.2.6 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)*.

26.3 Programming

To configure and use the UART controllers:

1. Set up the GPIOs to be used for the UART interface (Px_yy_MODE_REG[PID] = 1 to 6).
2. Select the UART clock (CLK_COM_REG[UART/2_CLK_SEL]).
3. Enable the selected UART by setting the CLK_COM_REG[UARTx_ENABLE] bit.

4. Enable access to Divisor Latch Registers (DLL and DLH) by setting the UARTx_LCR_REG[UART_DLAB] bit.
5. Set the desired baud rate. To calculate the registers values for the desired baud rate, use the formula: Divisor = UART CLK / (16 x Baud rate).
    a. UARTx_IER_DLH_REG: High byte of the Divisor integer part.
    b. UARTx_RBR_THR_DLL_REG: Low byte of the Divisor integer part.
    c. UARTx_DLF_REG: The fractional part of the Divisor.
6. Configure the brake control bit, parity, number of stop bits and data length (UARTx_LCR_REG).
7. Enable and configure the FIFO (UARTx_IIR_FCR_REG).
8. Configure the generated interrupts, if needed (UARTx_IER_DLH_REG).
9. Send a byte:
    a. Check if Transmit Hold Register (THR) is empty (UARTx_LSR_REG[UART_THRE]).
    b. Load the byte to THR (UARTx_RBR_THR_DLL_REG).
    c. Check if the byte was transmitted (UARTx_LSR_REG[UART_TEMT]).
10. Receive a byte:
    a. Wait until serial data is ready (UARTx_LSR_REG[UART_DR]).
    b. Read the incoming byte from the THR (UARTx_RBR_THR_DLL_REG).

27 Smart Card Interface

27.1 Introduction

The DA14592 features a Smart Card interface implemented using the UART2 block. The UART2 block supports asynchronous protocol smartcards as defined in the ISO 7816-3 (Class B and C) standard.

Features

- ISO/IEC 7816-3 (Class B and C) compliant
- UART line with flow control signals
- Interrupt line
- DMA support
- An error signal support with indication for character repetition
- Inverse and direct convention
- Guard time.

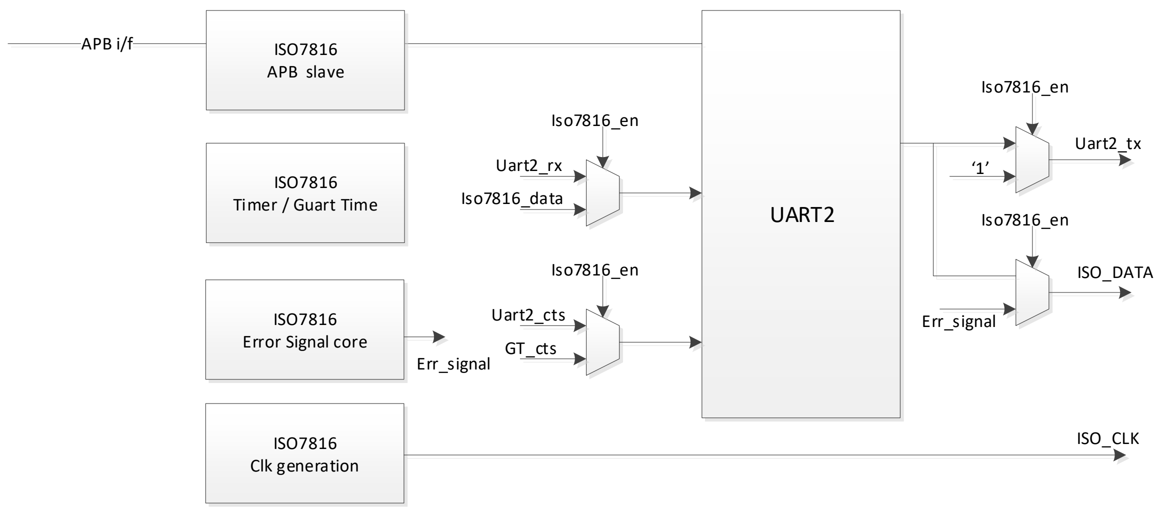

Figure 72. Smart card (ISO7816-3) block diagram

27.2 Architecture

27.2.1 ISO7816-3 clock generation

The ISO7816-3 block operates on a system-based clock with a 50% duty cycle. The clock frequency is calculated using the formula:

F_{SCLK}/[2 * (UART2_CTRL_REG[ISO7816_CLK_DIV] + 1)]

The UART2_CTRL_REG[ISO7816_CLK_DIV] value can be updated at any time, whether the ISO7816-3 clock is enabled or not. The operation of the clock can be checked at any given time by polling the UART2_CTRL_REG[ISO7816_CLK_STATUS] bit and disabled by clearing the ISO7816_CLK_EN[ISO7816_CLK_EN] bit. The UART2_CTRL_REG[ISO7816_CLK_LEVEL] reflects the logical level of the clock while it has been disabled.

27.2.2 ISO7816-3 timer – guard time

The ISO7816-3 block is equipped with an ISO7816-3 Timer/Guard Timer and can operate in two modes:

- UART2_TIMER_REG[ISO7816_TIM_MODE] = 0.
  The timer is clocked with the clock of the ISO7816 module and starts when UART2_TIMER_REG[ISO7816_TIM_EN] = 1. It counts from 0 to UART2_TIMER_REG[ISO7816_TIM_MAX] value. The value of the timer can be read from UART2_TIMER_REG[ISO_TIM_MAX] register field.
  At the point where the top value is reached, the timer stops and UART2_IRQ_STATUS_REG[ISO7816_TIM_EXPIRED_IRQ] is set. If the UART2_CTRL_REG[ISO7816_TIM_EXPIRED_IRQMASK] is 1, an interrupt is generated. The interrupt is cleared when UART2_TIMER_REG[ISO7816_TIM_EN] is set.
- UART2_TIMER_REG[ISO7816_TIM_MODE] = 1.
  The timer is clocked with the 1/16 of the UART bit clock (1/16 etu). If the UART fractional divider has been set, the 1/16 of the UART bit clock does not have a fixed value. If the UART2_TIMER_REG [ISO7816_TIM_EN] bit is set, the timer starts counting from 0 to UART2_TIMER_REG[ISO_TIM_MAX] value, each time a start bit is transmitted by the UART.
  If the UART2_TIMER_REG[ISO_TIM_MAX] value is set equal to 16*GuartTime-1, the timer counts the minimum delay between the leading edges of two consecutive characters (Guard time). In case of no FIFO mode and GT>12 etu, the expiration of the timer indicates that the module is allowed to send the next character (Guard Time elapsed). In case of FIFO mode and GT>12 etu, the UART2_CTRL_REG[ISO7816_AUTO_GT] bit can be set so the module is able to send the next character automatically each time the Guard Time is reached.

27.2.3 ISO7816-3 error detection

The ISO7816-3 block is designed to support the functionality described in section 7.3 of ISO7816-3 specification document.

Transmit phase

The transmitter checks the ISO7816 data level for 11 etu sampling time after a character's leading edge is detected. The module samples the data line at 11 etu time and creates two types of interrupts:

- The ISO7816_ERR_TX_TIME_IRQ interrupt is created each time a character is sent.
- The ISO7816_ERR_TX_VALUE_IRQ interrupt is created each time the receiver sends an error.

The software must check if an ISO7816_ERR_TX_VALUE_IRQ interrupt is generated to retransmit the character. The IRQs are generated at the same time.

Receive phase

The receiver holds the data line level Low between 1 etu (minimum) and 2 etu (maximum) at 10.5 etu time. The error detection circuit uses the UART parity check and creates an error signal to the transmitter at the proper time. The error signal pulse width and offset can be configured using the UART2_ERR_CTRL_REG register. Furthermore, the ISO7816-3 block can hold the UART_Rx signal High during the error signal transmitting to prevent UART errors from happening.

27.3 Programming

To configure the Smart Card Controller:

1. Set up the GPIOs to be used for the ISO7816-3 interface (Px_yy_MODE_REG[PID] = 7 to 8). Reset (output) and card insert (input) signals shall be configured as GPIOs (Px_yy_MODE_REG[PID] = 0).
2. Enable UART2 by setting the CLK_COM_REG[UART2_ENABLE] bit.
3. Set up UART2 clock (CLK_COM_REG[UART2_CLK_SEL]).
4. Enable the ISO7816-3 module by setting the UART2_CONFIG_REG[ISO7816_ENABLE] bit.

5. Set up the ISO7816-3 clock (UART2_CTRL_REG[ISO7816_CLK_DIV]).
6. Initialize UART2 as follows:
    a. Configure FIFO, if needed (UART2_IIR_FCR_REG, UART2_SRT_REG, UART2_STET_REG, UART2_SFE_REG). Note that if Error detection is enabled, the TX FIFO shall remain disabled.
    b. Configure ISO7816 convention.
7. Select data length (UART2_LCR_REG[UART_DLS]):
    a. Select the number of stop bits (UART2_LCR_REG[UART_STOP]).
    b. Enable/disable parity (UART2_LCR_REG[UART_PEN]).
    c. Select even or odd parity (UART2_LCR_REG[UART_EPS]).
    d. If needed, enable Error detection (UART2_CONFIG_REG[ISO7816_ERR_SIG_EN]) and configure the error pulse width and offset (UART2_ERR_CTRL_REG).
    e. Select direct/inverse convention (UART2_CONFIG_REG[ISO7816_CONVENTION]).
8. Configure the baud rate. To calculate the registers values for the desired baud rate, use the formula: Divisor = Fi * (ISO7816_CLK_DIV + 1)/(8 * Di). For Fi and Di values, see ISO/IEC 7816-3 Standard Specification, table 7 and table 8:
    a. Enable access to the Divisor Latch register: UART2_LCR_REG[UART_DLAB] = 1.
    b. Set the High byte of the Divisor integer part (UART2_IER_DLH_REG).
    c. Set the Low byte of the Divisor integer part (UART2_RBR_THR_DLL_REG).
    d. Set the fractional part of the Divisor (UART2_DLF_REG).
    e. Configure the generated interrupts, if needed (UART2_CTRL_REG).
9. Send a byte:
    a. Set up the Guard Timer (UART2_TIMER_REG).
    b. In case TX FIFO is enabled, check if FIFO is full (UART2_USR_REG[UART_TFNF]) or check if Transmit Hold Register (THR) is empty (UART2_LSR_REG[UART_THRE]).
    c. Load the byte to THR (UART2_RBR_THR_DLL_REG).
    d. Check if the byte was transmitted (UART2_LSR_REG[UART_TEMT]).
10. Receive a byte:
    a. Set up the Wait Time (UART2_TIMER_REG).
    b. Wait until serial data is ready (UART2_LSR_REG[UART_DR]) or timer expiration.
    c. Read the received byte from the THR (UART2_RBR_THR_DLL_REG).

28 SPI Interface

28.1 Introduction

This controller implements the Serial Peripheral Interface (SPI™)[1] for Master and Slave modes. The serial interface can transmit and receive from four to up to 32 bits in Master/Slave mode. The controller comprises separate TX and RX FIFOs and DMA handshake support. Slave mode clock speed is independent from the system clock speed. Moreover, master's clock speed can be as fast as the system's clock speed. The controller can generate an interrupt upon data threshold reached in the TX or RX FIFOs.

Features

- Slave and Master mode.
- From 4-bit to up to 32-bit operation.
- SPI Master clock line speed up to 32 MHz.
- SPI mode 0, 1, 2, and 3 support (clock edge and phase).
- Built-in separate 8-bit wide and 4-byte deep RX/TX FIFOs for continuous SPI bursts.
- Maskable interrupt generation based on TX or RX FIFO thresholds.
- DMA support.

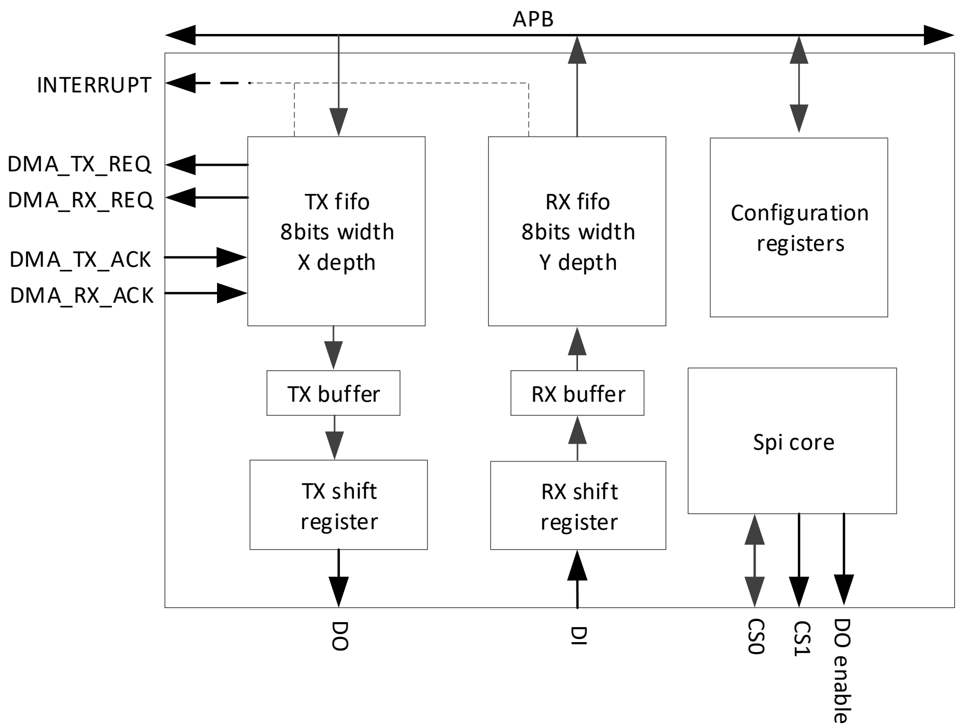

Figure 73. SPI block diagram

1 SPI is a trademark of Motorola, Inc.

28.2 Architecture

The SPI controller is an APB peripheral operating on the apb_clk clock. It contains a front end which is clocked by the spi_clk clock and is responsible for the serialization/deserialization of the data in the RX and TX streams.

Two separate FIFOs, each of eight bits wide and four bytes deep, are used to store data for RX and TX streams. Since a SPI word can be configured to be from four bits to up to 32 bits, one to four FIFO positions can be written/read at the same time. FIFOs contain logic implementing programmable thresholds comparison.

The SPI controller supports DMA requests and interrupt generation based on the FIFO thresholds. If enabled, a DMA request and/or interrupt is asserted with whether TX_FIFO level is low or RX_FIFO level is high.

The SPI interface supports all four modes of operation and Table 100 shows the corresponding polarity (CPOL) and phase (CPHA) of the SPI clock (SPI_CLK).

Table 100: SPI Modes Configuration and SCK States

SPI Mode	CPOL	CHPA	TX SPI_CLK	RX SPI_CLK	Idle SPI_CLK
0	0	0	Falling edge	Rising edge	Low
1	0	1	Rising edge	Falling edge	Low
2	1	0	Rising edge	Falling edge	High
3	1	1	Falling edge	Rising edge	High

To read from or to write to an external single byte Flash device in the SPI Master mode, a byte swap mechanism is implemented to allow for a proper placement of the bytes in a 16-bit word for the DMA to write to/read from the internal RAM. More specifically, when the SPI controller is configured as a master with DMA support and a 16-bit word width so that the bus utilization is increased compared to reading from an 8-bit device, the byte swap mechanism brings the least significant byte read and place it in the most significant byte in the 16-bit word. The controller automatically swaps the bytes to allow for placing the first byte read in the least significant byte of the 16-bit word. This feature is programmable via SPI_CTRL_REG[SPI_SWAP_BYTES].

The SPI controller can operate at the highest speed (32 MHz on the SPI_CLK line) in a special Master mode. The clock of the controller is then either the XTAL32M or the RC32M and can be used for fast booting from external Flash devices that support this frequency.

28.2.1 SPI timing

Figure 2 shows the timing of the SPI interface when the SPI controller is in Slave mode.

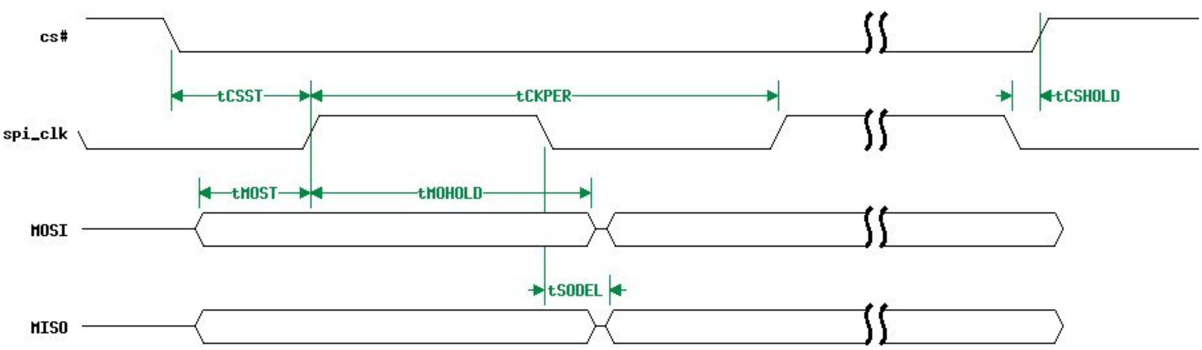

Figure 74. SPI Slave mode timing (CPOL = 0, CPHA = 0)

Table 101: SPI Timing Parameters

Parameter	Description	Typ	Unit
t_{CKPER}	spi_clk clock period	No constraints	ns
t_{CSST}	CS active time before the first edge of spi_clk	1 spi_clk cycle	
t_{CSHOLD}	CS non-active time after the last edge of spi_clk	1 spi_clk cycle	
t_{MOST}	Master input data latching setup time	5	ns
t_{MOHOLD}	Master input data hold time	0	ns
t_{SODEL}	Slave output data delay	15 (Note 1)	ns

Note 1 Typical conditions with VDDIO = 1.8 V, VDD = 0.9 V, and 15 pF load.

28.3 Programming

28.3.1 Master mode

To configure the SPI controller in Master mode:

1. Set the appropriate GPIO ports in SPI clock mode (output), SPI Chip Select mode (output), SPI Data Out mode (output), and SPI Data In mode (input).
2. Enable SPI clock by setting CLK_COM_REG[SPI_ENABLE] = 1.
3. Reset SPI FIFO by setting SPI_CTRL_REG[SPI_FIFO_RESET] = 1.
4. Set the SPI clock frequency by programming SPI_CLOCK_REG[SPI_CLK_DIV]. If SPI_CLK_DIV is not equal to 0x7F, SPI_CLK = module_clk/2 × (SPI_CLK_DIV + 1). If SPI_CLK_DIV = 0x7F, SPI_CLK = module_clk.
5. Set the SPI mode (CPOL or CPHA) by programming SPI_CONFIG_REG[SPI_MODE].
6. Set the SPI controller in Master mode by setting SPI_CONFIG_REG[SPI_SLAVE_EN] = 0.
7. Define the SPI word length (from 4-bit to 32-bit) by programming SPI_CONFIG_REG[SPI_WORD_LENGTH]. SPI_WORD_LENGTH = word length - 1.

To execute a SPI Read/Write command:

1. It is possible to configure the SPI module to capture data at the next clock edge when the slave device does not produce the data at the correct clock edge by setting: SPI_CTRL_REG[SPI_CAPTURE_AT_NEXT_EDGE] = 1. Otherwise, set SPI_CTRL_REG[SPI_CAPTURE_AT_NEXT_EDGE] = 0.
2. Release FIFO reset by setting SPI_CTRL_REG[SPI_FIFO_RESET] = 0.
3. Enable SPI TX path by setting SPI_CTRL_REG[SPI_TX_EN] = 1.
4. Enable SPI RX path by setting SPI_CTRL_REG[SPI_RX_EN] = 1.
5. Enable the SPI chip select by programming the SPI_CS_CONFIG_REG[SPI_CS_SELECT] = 1 or 2. This option allows the master to select the slave that is connected to the GPIO that has the function of SPI_EN or SPI_EN2. (Additionally, setting SPI_CS_CONFIG_REG[SPI_CS_SELECT] = 4 allows any GPIO to operate as chip select. The transmission is activated when this GPIO is set low using the Px_yy_DATA_REG or Px_yy_RESET_DATA_REG)
6. Enable the SPI controller by setting SPI_CTRL_REG[SPI_EN] = 1.
7. Write to TX FIFO by programming SPI_FIFO_WRITE_REG[SPI_FIFO_WRITE]. Write access is permitted only when SPI_FIFO_STATUS_REG[SPI_TX_FIFO_FULL] = 0.
8. Read from RX FIFO by programming SPI_FIFO_READ_REG[SPI_FIFO_READ]. Read is permitted only when SPI_FIFO_STATUS_REG[SPI_RX_FIFO_EMPTY] = 0.
9. To disable the SPI chip select, set SPI_CS_CONFIG_REG[SPI_CS_SELECT] = 0 to deselect the slave and set SPI_CTRL_REG[SPI_FIFO_RESET] = 1 to reset the SPI FIFO (if SPI_CS_CONFIG_REG[SPI_CS_SELECT] = 4 is used, the slave is deselected by setting the respective GPIO to high).

28.3.2 Slave mode

To configure the SPI controller in Slave mode:

1. Set the appropriate GPIO ports in SPI clock mode (input), SPI Chip Select mode (input), SPI Data Out mode (output), and SPI Data In mode (input). The SPI_CS_CONFIG_REG[SPI_CS_SELECT] bitfield must be set to 1 and the selected GPIO PID should be set to 12 (SPI_EN).
2. Enable SPI clock by setting CLK_COM_REG[SPI_ENABLE] = 1.
3. Reset SPI FIFO by setting SPI_CTRL_REG[SPI_FIFO_RESET] = 1.
4. Set the SPI mode (CPOL or CPHA) by programming SPI_CONFIG_REG[SPI_MODE].
5. Set the SPI module in slave controller by setting SPI_CONFIG_REG[SPI_SLAVE_EN] = 1.
6. Define the SPI word length (from 4-bit to 32-bit) by programming SPI_CONFIG_REG[SPI_WORD_LENGTH]. SPI_WORD_LENGTH = word length - 1.

To execute a SPI Read/Write command:

1. Set SPI FIFO in normal operation by setting SPI_CTRL_REG[SPI_FIFO_RESET] = 0.
2. Enable SPI TX path by setting SPI_CTRL_REG[SPI_TX_EN] = 1.
3. Enable SPI RX path by setting SPI_CTRL_REG[SPI_RX_EN] = 1.
4. Enable the SPI controller by setting SPI_CTRL_REG[SPI_EN] = 1.
5. Write the first data byte directly to TX buffer by programming the SPI_TXBUFFER_FORCE_L_REG[SPI_TXBUFFER_FORCE_L].
6. Write the rest of the data to TX FIFO by programming SPI_FIFO_WRITE_REG[SPI_FIFO_WRITE]. Write access is permitted only if SPI_FIFO_STATUS_REG[SPI_TX_FIFO_FULL] = 0.
7. Read from RX FIFO by programming SPI_FIFO_READ_REG[SPI_FIFO_READ]. Read is permitted only if SPI_FIFO_STATUS_REG[SPI_RX_FIFO_EMPTY] = 0.

29 Quadrature Decoder

29.1 Introduction

The DA14592 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. It can also be programmed to simply count rising/falling edges on any of the channel pairs. This block can be used to wake up the chip as soon as there is a movement from the connected external device. Figure 75 shows the block diagram of the quadrature decoder.

Features

- Three 16-bit signed counters that provide the step count and direction on each of the axes (X, Y, and Z) and one 8-bit counter counting the overall edges from all the three counters.
- Programmable system clock sampling at a maximum of 32 MHz.
- APB interface for control and programming.
- Programmable source from the GPIOs.
- Digital filter on the channel inputs to avoid spikes.

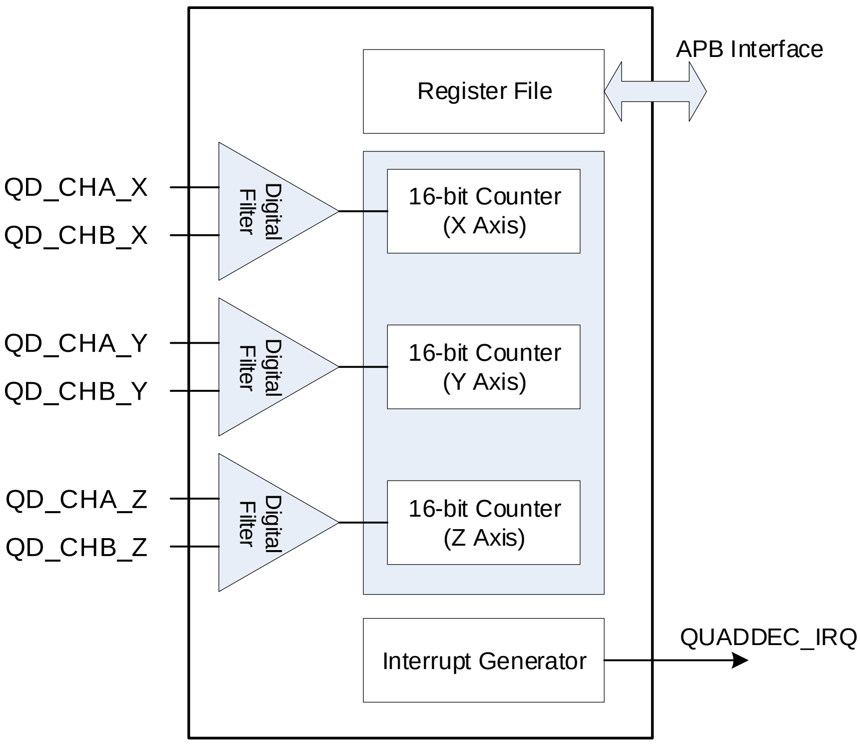

Figure 75. Quadrature decoder block diagram

29.2 Architecture

Channels are expected to provide a pulse train with 90 degrees rotation as shown in Figure 76 and Figure 77.

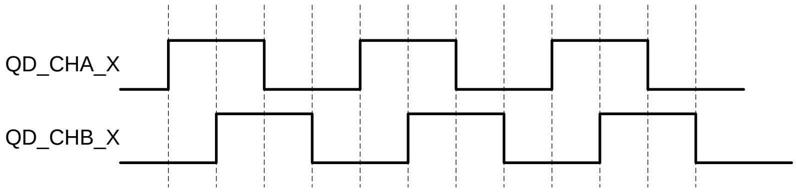

Figure 76. Moving forward on axis X

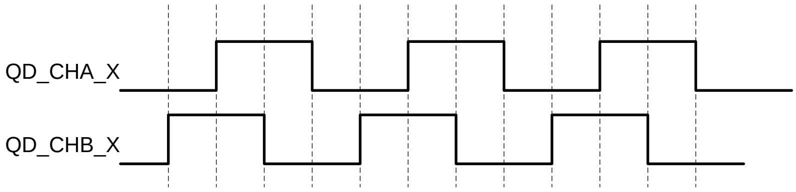

Figure 77. Moving backwards on axis X

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.

You can choose which GPIOs to use for the channels by programming the QDEC_CTRL2_REG register. The block supports two modes of operation: quadrature counting and edges counting. The quadrature counting mode reads the patterns of successive pulses as in Figure 76 and Figure 77, while the edges counting mode simply counts all positive and negative edges on any of the two channels of a pair.

NOTE
If two edges happen at the same time, the counter only counts one.

The digital filter eliminates spikes shorter than three clock periods. It is followed by an edge detection circuitry as shown in Figure 78.

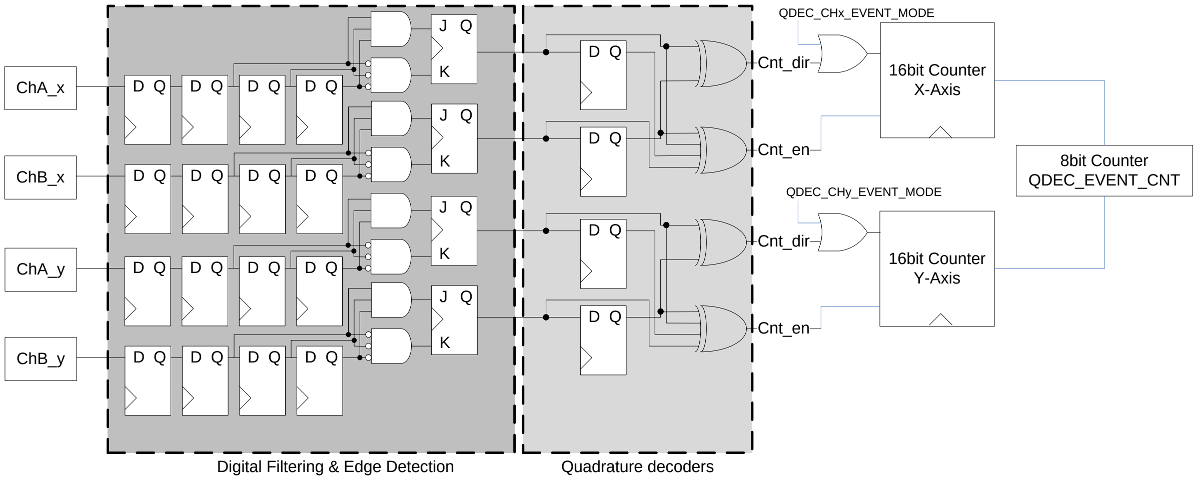

Figure 78. Digital filtering and edge detection circuit

A counter for its dedicated axis holds the movement events of the channels. When a channel is disabled, the counter is reset. The counters are accessible through the APB bus.

QDEC_EVENT_CNT gathers all edges on all channels regardless of the mode of operation. If two edges happen at the same time, this counter only is increased by one.

The quadrature decoder operates on the DIVN clock. The QDEC_CLOCKDIV register defines the number of clock cycles during which the decoding logic samples the data on the channel inputs. The division is automatically disabled when the system goes to sleep. In sleep, the quadrature decoder operates on the RCLP clock.

29.3 Programming

To program the quadrature decoder for actual quadrature counting or edge counting:

1. Enable the clock of the block by writing at CLK_AMBA_REG[QDEC_CLK_ENABLE].
2. Configure the clock frequency by configuring the QDEC_CLOCKDIV_REG.
3. Define which pin pairs represent the different channels for the X, Y, and Z axes or the GPIOs from which the edges are counted. Configure such information at QDEC_CHX_PORT_SEL, QDEC_CHY_PORT_SEL, and QDEC_CHZ_PORT_SEL registers.
4. Configure the interrupt threshold upon which an interrupt is generated at QDEC_CTRL_REG[QDEC_IRQ_THRES]. Note that the interrupt threshold is based on the value of QDEC_EVENT_CNT_REG that keeps on counting after the interrupt is generated.
5. Define the mode of operation by configuring the respective QDEC_CHx_EVENT_MODE field in QDEC_CTRL2_REG.
6. Wait for the interrupt and then read X, Y, and Z values at QDEC_XCNT_REG, QDEC_YCNT_REG, and QDEC_ZCNT_REG (in the quadrature counting case) or the QDEC_EVENT_CNT_REG (in the edges counting case).
7. Clear the interrupt (by writing at QDEC_CTRL_REG[QDEC_IRQ_STATUS]) and the edge counter (by writing at QDEC_CTRL_REG[QDEC_EVENT_CNT_CLR) if needed.

30 Real Time Clock

30.1 Introduction

The DA14592 is equipped with a Real Time Clock (RTC) that provides complete clock and calendar information with automatic time units' adjustment and easy configuration.

Features

- Complete time of day clock: 12/24 hour, hours, minutes, seconds, and hundredths.
- Calendar function: day of week, date of month, month, year, century, leap year compensation, and year 2000 compliant.
- Alarm function: month, date, hour, minute, second, and hundredths resolution.
- Event interrupt on any calendar or time unit.
- Available during sleep.
- Granularity of 10 ms.

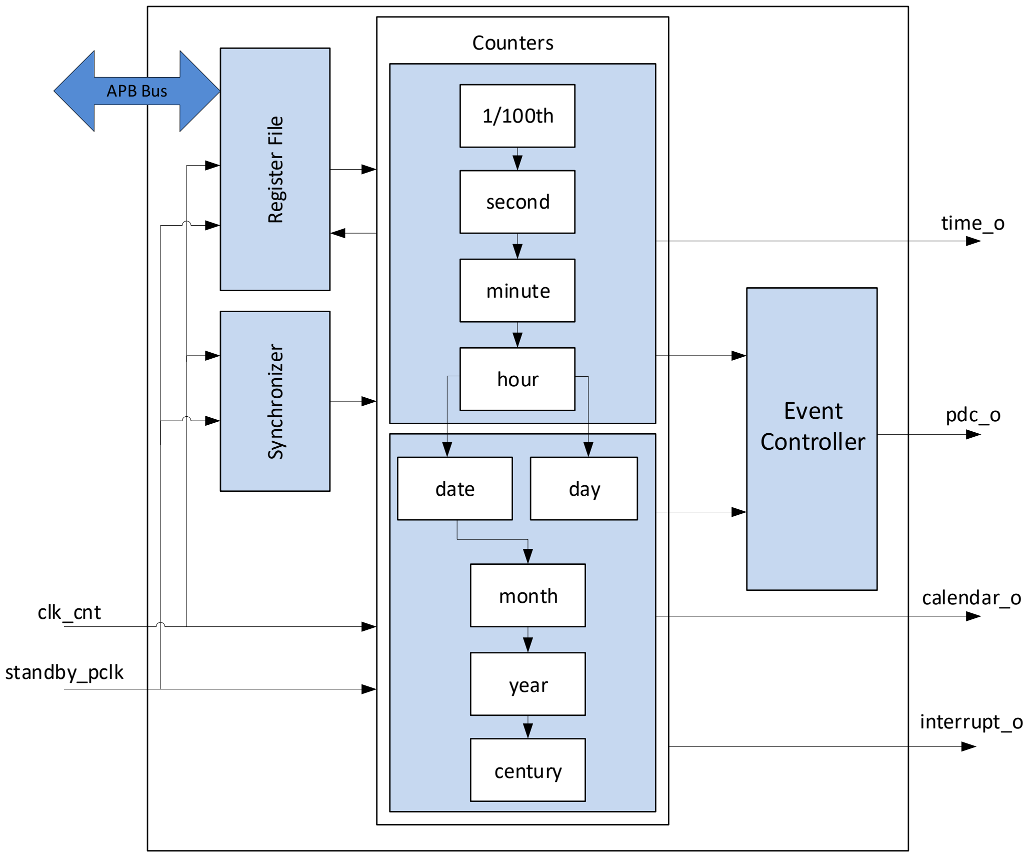

Figure 79. Real time clock block diagram

30.2 Architecture

Figure 79 shows the architecture of the Real Time Clock.

The RTC supports a year range from 1900 to 2999 as well as full month, date, minute, second, and hundredth of second ranges. It also supports hour ranges of 0 to 23 (24-hour format), or 1 to 12 with a.m./p.m. flag (12-hour format).

Alarms can be generated in two ways: as a one-time alarm, or as a recurring alarm. In addition to alarms, the RTC can detect when a particular event occurs. Each field of the calendar and time counter can generate an event when it rolls over. For example, an event can be generated every new month, new week, new day, new half day (12-hour mode), new minute, or new second. Both alarms and events can generate an interrupt. All the interrupts can be set, enabled, disabled, or masked at any time.

The RTC block has been enhanced with RTC_EVENT_CONTROLLER which comprises a 13-bit counter that counts down starting from a configurable value (programmed in RTC_PDC_EVENT_PERIOD). This operation can be enabled/disabled by RTC_PDC_EVENT_EN.

Every time this counter reaches 0, a signal towards the PDC is asserted. The same signal is also driven to the Cortex-M33 RTC_EVENT interrupt line. The signal is automatically de-asserted after the RTC_PDC_EVENT_CLEAR_REG is read.

The counter is accessible by the Cortex-M33 CPU (RTC_PDC_EVENT_CNT_REG).

30.3 Programming

To configure the RTC:

1. Configure the 100 Hz RTC granularity if needed:
    a. Based on the selected LP clock (for example, 32768 kHz), set the CLK_RTCDIV_REG[RTC_DIV_INT] = 327 (= 0x147).
    This value should be equal to the integer divisor part of the formula $F_{LP\_CLK}/100$ = **327**.680.
    b. Based on the selected LP clock (for example, 32768 kHz), set the CLK_RTCDIV_REG[RTC_DIV_FRAC] = 680 (= 0x2A8).
    This value should be equal to the fractional divisor part of the formula $F_{LP\_CLK}/100$ = 327.**680**.
    c. To achieve better accuracy of the divisor, configure the denominator for the fractional division accordingly (CLK_RTCDIV_REG[RTC_DIV_DENOM]).
    d. Enable the 100 Hz RTC granularity by setting the CLK_RTCDIV_REG [RTC_DIV_ENABLE] bit.
2. Enable the time functionality by clearing the RTC_CONTROL_REG[RTC_TIME_DISABLE].
3. Enable the calendar functionality by clearing the RTC_CONTROL_REG[RTC_CAL_DISABLE].
4. Choose between 12 or 24 hours based mode (RTC_HOUR_MODE_REG[RTC_HMS]).
5. Configure the time (RTC_TIME_REG).
6. Configure the date (RTC_CALENDAR_REG).
7. Set up a time alarm if needed (RTC_ALARM_ENABLE_REG).
8. Set up a calendar alarm if needed (RTC_CALENDAR_ALARM_REG).
9. Enable the configured alarms (RTC_ALARM_ENABLE_REG[RTC_ALARM_xxxx_EN]).
10. Configure the interrupt generation when an alarm happens (RTC_INTERRUPT_ENABLE_REG). Disable the interrupt generation with RTC_INTERRUPT_DISABLE_REG.
11. Configure the event flag generation when an alarm happens (RTC_EVENT_FLAGS_REG).
12. Choose if software reset resets the RTC (RTC_KEEP_RTC_REG[RTC_KEEP]).
13. Set up the Event Controller to trigger the PDC:
    a. Set up the event period (RTC_PDC_EVENT_PERIOD_REG[RTC_PDC_EVENT_PERIOD]).
    b. Enable the PDC to trigger (RTC_EVENT_CTRL_REG[RTC_PDC_EVENT_EN]).

31 General Purpose Timers

31.1 Introduction

The Timers block contains four identical 24-bit wide timers: Timer, Timer2, Timer3, and Timer4. They are software controlled, programmable, and you can use them for various tasks. Timer, Timer2, and Timer3 are in the timer power domain (PD_TIM), while Timer4 resides in the communications power domain (PD_COM).

Features

- Four 24-bit general purpose timers.
- Pulse Width Modulated signal (PWM) per Timer block.
- Two channels for the capture input triggered by GPIOs (timer has four channels).
- One-shot pulse with programmable pulse width (only valid for Timer and Timer2).
- 5-bit clock pre-scaler.
- Selectable system or sleep clock source.
- Up/down counting capability with the free running mode.
- Active while the system is in the Sleep mode.
- Dedicated interrupt line per timer (timer supports one extra interrupt line for complex capture events).

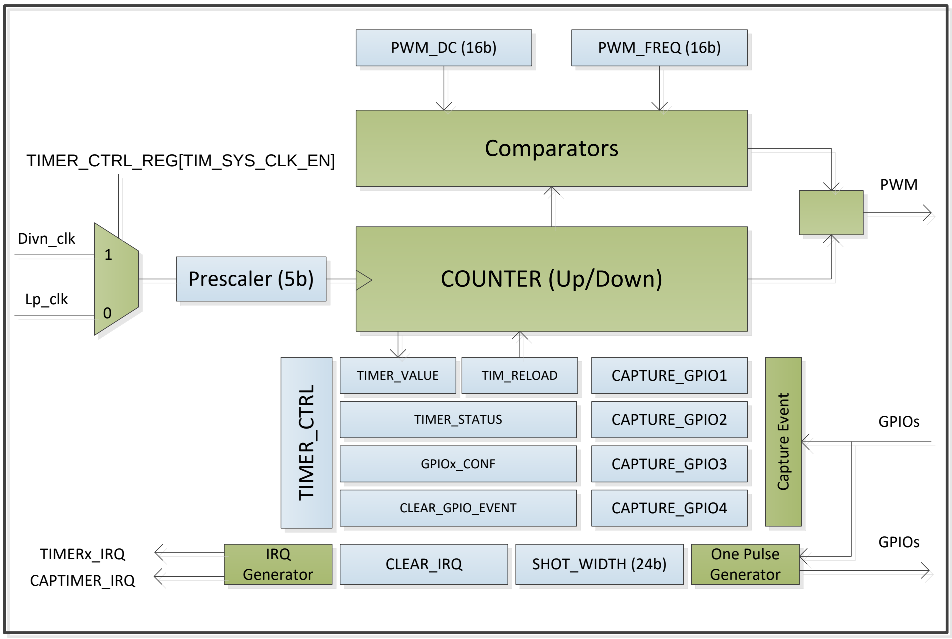

Figure 80. General-purpose timer block diagram

31.2 Architecture

There are four instances of the same Timer block (Timer, Timer2, Timer3, and Timer4) in the system with minor modifications between the instances. Figure 80 shows the timer block diagram. The only differences between the actual instances:

- Timer, Timer2, and Timer3 are parts of the PD_TIM domain, while Timer 4 is a part of the PD_COM power domain.
- Timer3 and Timer4 do not support the one-shot feature.

- Timer supports four channels (GPIOs) for capturing events, while the rest supports two channels. Timer also has a separate interrupt line dedicated to these channel activities.
- Timer and Timer2 can drive specific GPIOs, even when the system is in Sleep mode.

Each block has a configurable free-running up/down counter that operates either on the system clock (sys_clk) or the low power clock (lp_clk) with a 5-bit pre-scaler. This results in a minimum frequency of 1 kHz and a maximum of 32 MHz.

Each block has a PWM generator which does not affect the actual timer running. The PWM generator can define the frequency and the duty cycle of the generated PWM signal by further dividing the clock by maximum (2^{16}+1) resulting in a minimum speed of 0.5 kHz. A 16-bit PWM duty cycle enables 256 steps for defining the pulse width. Only 50% duty cycle is possible for the minimum PWM frequency.

Capturing an edge of a GPIO is another feature of the Timer block. Rising and falling edge detection on any configurable GPIO can trigger a Timer's snapshot stored in a separate register. Another (or the same) GPIO can be configured for capturing a second edge, hence storing a second snapshot in a second register. The difference of these two registers indicates the time distance of the two triggering events, hence it can be used for measuring the frequency of a signal or the timing interval between two interrupts coming from external devices.

One-shot is also supported. Upon a trigger configured as an input on a GPIO, another GPIO serves as an output delivering a pulse of configurable width. This is basically a PWM reply in hardware.

Table 102 summarizes the supported Timers block features.

Table 102: Timers Block Features Overview

Feature	Timer	Timer2	Timer3	Timer4
Free-running counter	✓	✓	✓	✓
PWM generation	✓	✓	✓	✓
Edge detection counter	✓	✓	✓	✓
PWM when in sleep mode	P0_12	P0_10	×	×
Event capturing channels	4	2	2	2
Dedicated event capture IRQ	✓	×	×	×
One-shot	✓	✓	✓	✓

There are specific GPIOs that can be configured to be input events to Timer block. Table 103 presents the actual configuration.

Table 103: Timers Block Input GPIOs

Timer Input	Value	GPIO
TIMER_GPIOx_CONF_REG	0-32	0: Disabled, 1-16: P0_0 to P0_15, 17-32: P1_0 to P1_15
TIMER2_GPIOx_CONF_REG	0-32	0: Disabled, 1-16: P0_0 to P0_15, 17-32: P1_0 to P1_15
TIMER3_GPIOx_CONF_REG	0-32	0: Disabled, 1-16: P0_0 to P0_15, 17-32: P1_0 to P1_15
TIMER4_GPIOx_CONF_REG	0-32	0: Disabled, 1-16: P0_0 to P0_15, 17-32: P1_0 to P1_15

31.3 Programming

To program GP timers:

1. Clear TIMERx_CTRL_REG and set the clock source:
   a. LP Clock: TIMERx_CTRL_REG = 0x00.
   b. DivN Clock: TIMERx_CTRL_REG = 0x80.

2. Clear pending Timer IRQs (TIMERx_CLEAR_IRQ_REG)
3. Enable the timer's clock (TIMERx_CTRL_REG[TIM_CLK_EN]).
4. Select the timer's pre-scaler (TIMERx_SETTINGS_REG[TIM_PRESCALER]).
5. Set the timer reload or max value (or phase delay in one shot mode) (TIMERx_SETTINGS_REG[TIM_RELOAD]).
6. Wait until TIMERx_STATUS_REG[TIM_TIMER_BUSY] is cleared
7. Select the timer's mode (TIMERx_CTRL_REG[TIM_ONESHOT_MODE_EN]).
   a. One-shot mode (TIMER/TIMER2):
      i. Set the shot phase duration (TIMER/2_SHOTWIDTH_REG).
      ii. Select the one-shot/primary capture event input (TIMER/2_GPIO1_CONF_REG) and trigger polarity (TIMER/2_CTRL_REG[TIM_IN1_EVENT_FALL_EN]).
      iii. Select the secondary capture event input (TIMER/2_GPIO2_CONF_REG) and trigger polarity (TIMER/2_CTRL_REG[TIM_IN2_EVENT_FALL_EN]).
   b. Counter mode (all Timers):
      i. Set the timer's up/down direction (TIMERx_CTRL_REG[TIM_COUNT_DOWN_EN]).
      ii. Enable the free run mode if timer counts upwards by setting the TIMERx_CTRL_REG[TIM_FREE_RUN_MODE_EN] bit.
      iii. Set up the capture events input (TIMERx_GPIOy_CONF_REG) and trigger polarity (TIMERx_CTRL_REG[TIM_INx_EVENT_FALL_EN]). TIMER supports up to four capture events inputs.
8. Select if event on the selected input GPIOs creates a CAPTIM interrupt (TIMERx_CTRL_REG[TIM_CAP_GPIOy_IRQ_EN]).
9. Configure PWM:
   a. Set up the frequency (TIMERx_PWM_FREQ_REG[TIM_PWM_FREQ]).
   b. Configure the duty cycle (TIMERx_PWM_FREQ_REG [TIM_PWM_DC]).
   c. Wait until TIMERx_STATUS_REG[TIM_PWM_BUSY] is cleared
   d. Set up GPIO as the timer's output (Px_yy_MODE_REG[PUPD] = 3, Px_yy_MODE_REG[PID] = 24 to 29, see P0_00_MODE_REG description).
      TIMER and TIMER2 PWM outputs can be activated at specific pins (P0_12 and P0_10) during sleep mode by setting the CLK_TMR_REG[TMRx_PWM_AON_MODE] bits.
10. Enable the timer's interrupt by setting the TIMERx_CTRL_REG[TIM_IRQ_EN] bit.
11. Enable the timer by setting the TIMERx_CTRL_REG[TIM_EN] bit.

32 Watchdog Timers

32.1 Introduction

The DA14592 contains two watchdog timers:

- The System watchdog that is in the Sleep power domain.
- The CMAC watchdog that is in the radio power domain.

They protect the system from getting stuck because of software problems in the application processor (Cortex-M33), and in the CMAC area (Cortex-M0+). They are clocked by internally generated clocks, RC32/512K, or RCX.

CMAC watchdog is accessible by Cortex-M33 if the radio power domain is powered up.

Features

- System watchdog
  - A 13-bit down-counter running on either RC32/512K or RCX further divides these clocks by 320 and can operate for 82 seconds or three minutes depending on the clock. 512 kHz are automatically downscaled to 32 kHz.
  - Internal programmable clock sources are RC32/512K or RCX, where RC32/512K is default. If RCX is used as a sleep clock, RC32/512K might be turned off.
  - Generates:
    - NMI to Cortex-M33 if counter reaches 0.
    - Hardware reset if counter reaches -16.
  - Protected by a lock bit to avoid freeze by mistake.
  - Automatically frozen when Cortex-M33 is in the debug mode.
- CMAC watchdog
  - A 13-bit down-counter running on the lp_clk further divides this clock by 32.
  - Generates:
    - Early notification interrupt to Cortex-M0+ if counter reaches 16.
    - CMAC2SYS_IRQ to Cortex-M33 if counter reaches 0.
    - Hardware reset if the counter reaches -16.
  - Protected by a lock bit to avoid freeze by mistake.
  - Automatically frozen when Cortex-M0+ is in the debug mode.

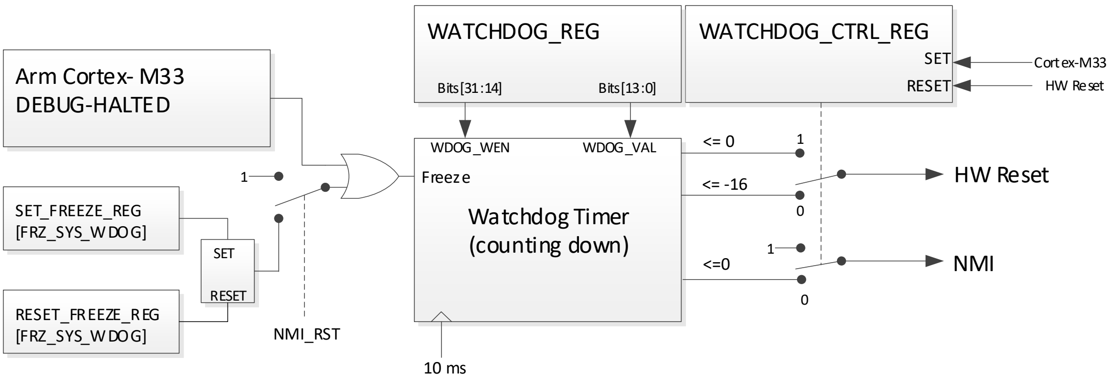

Figure 81. System watchdog block diagram

32.2 Architecture

The system watchdog is supplied by the sleep power domain (PD_SLP) and is automatically enabled as soon as the system powers up. It is decremented by one every 10 ms assuming the default source clock is RC32/512K. You can access the timer value through the WATCHDOG_REG, which is set to the max value at reset. This results in a maximum watchdog time-out of ~82 seconds. If the RCX is used as a source clock, the time-out time is even longer depending on the RCX frequency.

During the write access, WATCHDOG_REG bits [31-14] must be 0. This provides extra filtering for a software run-away writing all ones to WATCHDOG_REG. If the watchdog reaches 0, the counter value gets a negative value setting bit 8. The counter sequence is 1, 0, 1FFF16 (-1), 1FFE16(-2), ...,1FF016 (-16).

If WATCHDOG_CTRL_REG[NMI_RST] = 0, the watchdog timer generates an NMI to Cortex-M33 when the watchdog timer reaches 0 and a hardware reset when the counter becomes less or equal to -16. The NMI handler must write any value > -16 to the WATCHDOG_REG to prevent the generation of a WDOG reset within 16x10 = 160 ms.

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

The system watchdog can be frozen by Cortex-M33. It is always frozen automatically when the debugger is attached, and the Cortex-M33 CPU is halted during debugging. However, even if the watchdog is frozen by Cortex-M33, when Cortex-M33 gets into any sleep mode (PD_SYS is turned off), the system watchdog is automatically resumed to operate during sleep.

The CMAC watchdog is basically the same circuit with the following amendments:

- It resides in the radio power domain (PD_RAD). It becomes active as soon as the radio power domain is enabled.
- Its clock source is the block clock. A 1 ms period pulse is fed to the CMAC as an enable signal to the watchdog counter. Hence, this watchdog can reach maximum 16 ms before reaching 0.
- It has one extra notification compared to the system watchdog:
  - When 16 is reached, an interrupt is issued to the Cortex-M0+ (CMAC).
  - When 0 is reached, an interrupt is issued to the Cortex-M33.
  - When -16 is reached, a hardware reset is triggered.

This watchdog is also automatically halted when the debugger is attached and the Cortex-M0+ is halted during debugging.

32.3 Programming

32.3.1 System watchdog

To program the system watchdog timer:

1. Switch to the RCX clock as source if needed (CLK_RCX_REG[RCX_ENABLE]).
2. Freeze the watchdog by setting the SET_FREEZE_REG[FRZ_SYS_WDOG] bit (optionally).
3. Select NMI and reset events (WATCHDOG_CTRL_REG[NMI_RST]).
4. Wait until WATCHDOG_CTRL_REG[WRITE_BUSY] = 0.
5. Enable writing of the watchdog timer (WATCHDOG_REG[WDOG_WEN] = 0]).
6. Write the watchdog timer reload value (WATCHDOG_REG[WDOG_VAL], see register description).
7. Resume the watchdog (RESET_FREEZE_REG[FRZ_SYS_WDOG] = 1), if frozen.

32.3.2 CMAC watchdog

There is no specific sequence of steps for the CMAC watchdog since it gets automatically enabled. Some general points:

- The CMAC watchdog is automatically enabled when the radio power domain is activated.
- CMAC takes care of reloading the watchdog timer frequently. M33 intervention is not necessary.
- M33 can freeze/unfreeze the CMAC watchdog (SET_FREEZE_REG[FRZ_SYS_WDOG]) and reload the watchdog timer (CM_WDOG_REG[CM_WDOG_CNT]) if needed.
- The CMAC watchdog automatically freezes when Cortex-M0+ is halted through SWD.
- When the system power domain is powered down, it is not possible to freeze the CMAC watchdog with software.

33 Input/Output Ports

33.1 Introduction

The DA14592 has a software-configurable input/output (I/O) pin assignment organized into ports Port 0 and Port 1.

Features

- 16 pins on each Port (Port 0 and Port 1, including M33_SWCLK, M33_SWDIO, CMAC_SWCLK, and CMAC_SWDIO).
- Fully programmable pin assignment (PPA).
- Selectable pull-up and pull-down resistors of 25 kΩ per pin.
- Programmable open-drain functionality.
- Pull-up voltage at VBAT or V18.
- Pins can retain their last state when system enters in the Extended or Deep Sleep mode.

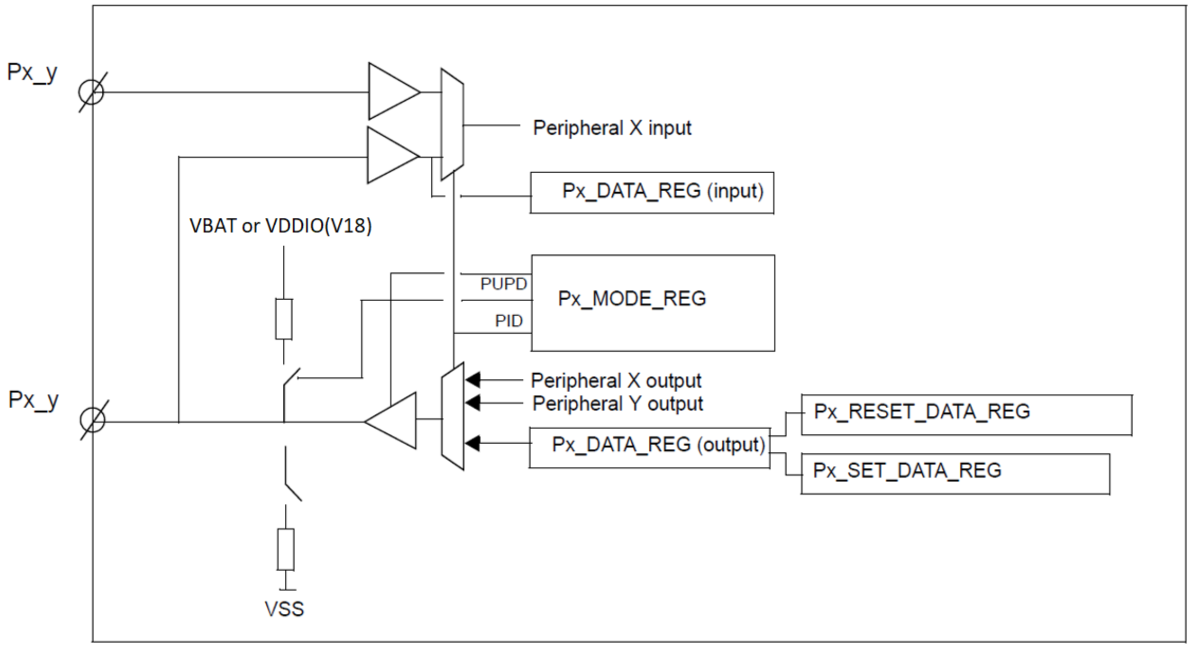

Figure 82. Port P0 and P1 with programmable pin assignment

33.2 Architecture

33.2.1 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 signals can be freely mapped to any I/O port bit by setting Px_yy_MODE_REG[5:0]

```
0x00 to 0x3B: Peripheral IO ID (PID)
```

Refer to the Px_yy_MODE_REGs for an overview of the available PIDs. Analog, GPADC, and SDADC signals have the fixed pin assignment to limit interference with the digital domain. The same applies to Quadrature Decoder and QSPI RAM Controller signals. The M33_SWD interface is mapped on P0_06 and P0_07, and the CMAC_SWD interface is mapped on P0_08 and P0_09.

33.2.1.1 Priority

The firmware has the possibility to assign the same peripheral output to more than one pin. You must make a unique assignment.

If you assign more than one input signals to a peripheral input, the left most pin in the lowest port pin number has priority. For example, P0_00_MODE_REG has priority over P0_01_MODE_REG.

33.2.1.2 Direction control

The port direction is controlled by setting Pxy_MODE_REG[9:8].

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

33.2.2 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.

33.2.2.1 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. It returns the current state on each port pin even if the output direction is selected, regardless of the programmed PID, and unless the analog function is selected. In this case, it reads 0. The Cortex-M33 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.

33.2.2.2 Port set data output register

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

33.2.2.3 Port reset data output register

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

33.2.3 Fixed assignment functionality

There are certain signals that have a fixed mapping on specific general purpose IOs. Table 104 shows this assignment.

Table 104: Fixed Assignment of Specific Signals

GPIO	SWD (Note 1)	QSPI_RAM	QUADDEC	Clocks (Note 2)	Analog (Note 3)	Special
P0_00		QSPI_CLK	QD_XYZ_CHA			
P0_01		QSPI_CS	QD_XYZ_CHB			
P0_02		QSPI_D0	QD_XYZ_CHA			
P0_03		QSPI_D1	QD_XYZ_CHB			
P0_04		QSPI_D2	QD_XYZ_CHA			
P0_05		QSPI_D3	QD_XYZ_CHB			
P0_06	M33_SWDIO					
P0_07	M33_SWCLK					
P0_08	M0_SWDIO		QD_XYZ_CHA	DIVN		
P0_09	M0_SWCLK		QD_XYZ_CHB			
P0_10			QD_XYZ_CHA		GPADC3 SDADC3	Timer2.PWM (Note 5)
P0_11			QD_XYZ_CHB	XTAL32M		Bangap_Enable (Note 4)
P0_12			PPA	LP_CLK		Timer.PWM (Note 5)
P0_13						UART Boot TX
P0_14						Hibernation wake up 1
P0_15						UART Boot RX

GPIO	SWD (Note 1)	QSPI_RAM	QUADDEC	Clocks (Note 2)	Analog (Note 3)	Special
P1_00			QD_XYZ_CHA		PGA_Inp GPADC0 SDADC0	
P1_01			QD_XYZ_CHB		PGA_Inm GPADC1 SDADC1	
P1_02			QD_XYZ_CHB	RC32M	GPADC2/SDADC2	WokenUp (Note 4)
P1_03			QD_XYZ_CHA			
P1_04			QD_XYZ_CHA			Hibernation wake up 2
P1_05			QD_XYZ_CHB		GPADC4 SDADC4 SDADC_REFp SDADC_INT_REF	
P1_06			QD_XYZ_CHA		GPADC5 SDADC5 SDADC_REFm	
P1_07			QD_XYZ_CHB			
P1_08			QD_XYZ_CHA			
P1_09			QD_XYZ_CHB		GPADC6 SDADC6	
P1_10			QD_XYZ_CHA			
P1_11			QD_XYZ_CHB		GPADC7 SDADC7	
P1_12			QD_XYZ_CHA			
P1_15			QD_XYZ_CHB			

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

Note 2 Enable specific clock outputs through GPIO_CLK_SEL_REG register.

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

Note 4 PMU_CTRL_REG[MAP_BANDGAP_EN].

Note 5 Timer.PWM and Timer.PWM2 signals are available during sleep (CLK_TMR_REG[TMR_PWM/2_AON_MODE]).

33.2.4 GPIO state retention while sleeping

Before setting the system to any sleep modes, the state of the pads needs to be retained. It is done to avoid external components being affected by the GPIOs changing states when the system goes to sleep. Also, it is done to avoid any floating driving signals from shut-off power domains leading to increasing power dissipation.

Always-on latches automatically latch the state of the pads by setting the corresponding PAD_LATCH_EN bit in the Px_RESET_PAD_LATCH_REG. These bits latch the digital control signals going into the pad and latch the data output separately for each pin. Hence, if the pad has been set as an output driving high, it retains its precise state. Px_SET_PAD_LATCH_REG is used to unlatch the pins.

Figure 83 and Figure 84 show the signals in red are latched.

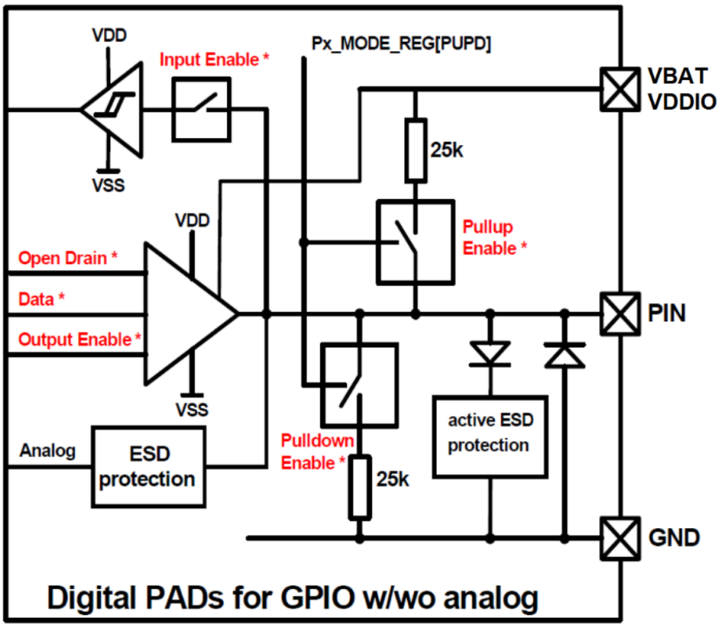

Figure 83. Latching of digital pad signals

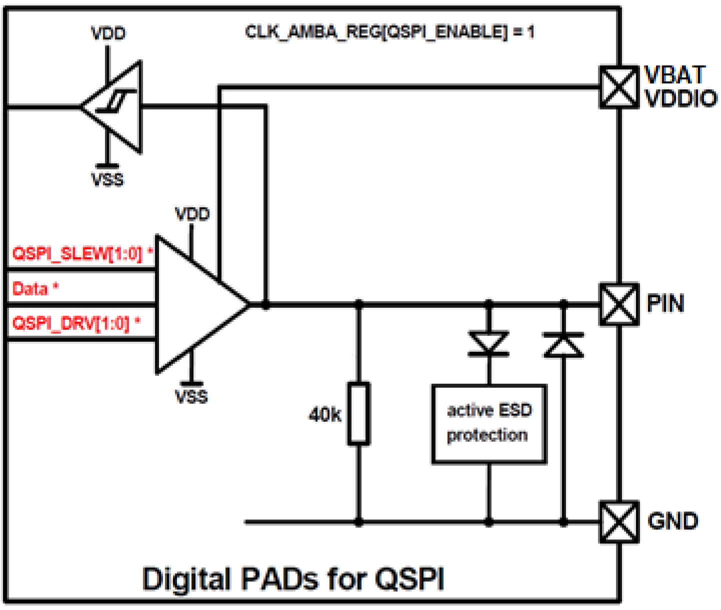

Figure 84. Latching of QSPI pad signals

After the system waking up, the software must disable the latching by setting the corresponding PAD_LATCH_EN bits of Px_PAD_LATCH_REG so that all pads can be accessed and controlled again.

For QSPI pads, the QSPI controller overwrites the pad latching as soon as the clock of the controller is enabled.

Note
The QSPI (GPIO) pads input voltage must not exceed VDDIO voltage level since there is no backdrive protection on those pads.

34 Radio

34.1 Introduction

The radio transceiver provides a 103 dB RF link budget for reliable wireless communications. All RF blocks are supplied by on-chip low dropout regulators (LDOs). The bias scheme is programmable and optimized for the minimum power consumption. Figure 85 shows the radio block diagram. It comprises the Receiver, Transmitter, Synthesizer, RX/TX combiner block, and Biasing LDOs.

Features

- Single-ended RFIO interface, 50 Ω matched.
- Alignment-free operation.
- Configurable transmit output power.
- Ultra-low power consumption.
- Fast frequency tuning minimizes overhead.

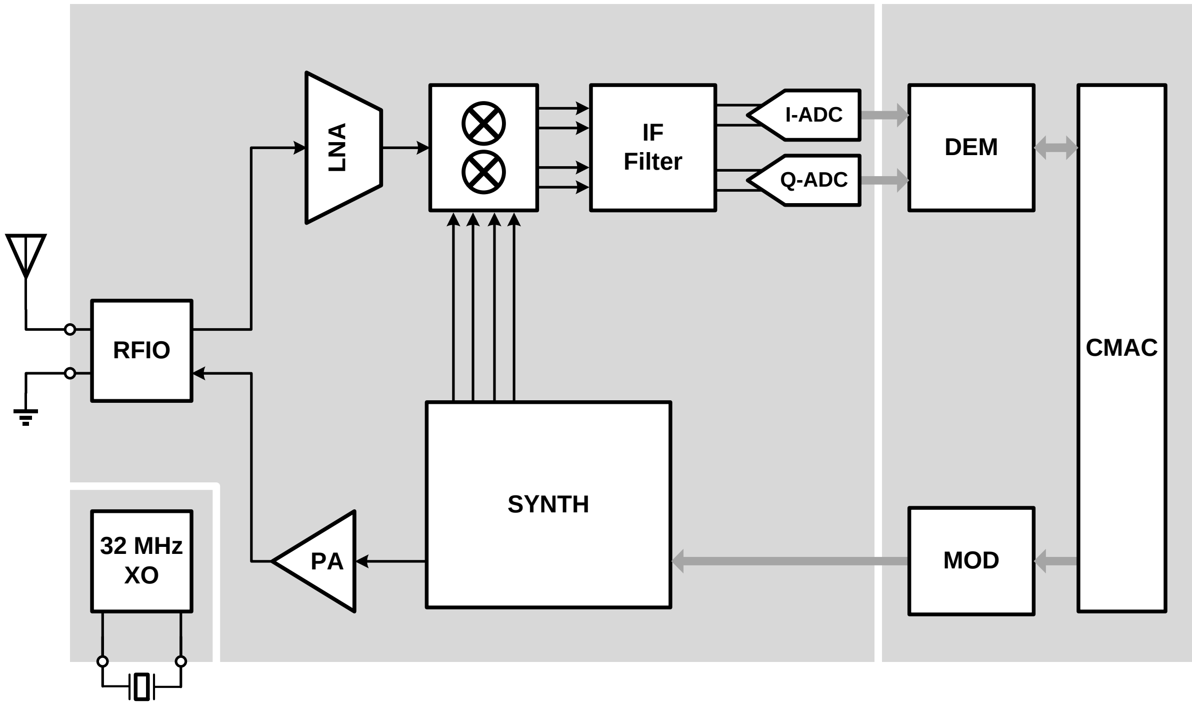

Figure 85. Radio block diagram

34.2 Architecture

34.2.1 Receiver

The RX front-end consists of a selective matching network, a low noise amplifier (LNA), and an image rejection down the conversion mixer. The intermediate frequency (IF) part of the receiver comprises a filter with a programmable gain. The AGC controls LNA and IF Filter gains. This provides the necessary signal conditioning prior to digitalization. The digital demodulator block (DEM) provides a synchronous bit stream.

34.2.2 Synthesizer

The RF synthesizer generates the quadrature LO signal for the mixer. It also generates the modulated TX output signal. DCO runs at twice the required frequency, and a dedicated divide-by-two circuit generates the 2.4 GHz signals in the required phase relations. The reference frequency is the 32 MHz crystal clock. The 2-point modulation performs the modulation of the TX frequency.

34.2.3 Transmitter

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

34.2.4 RFIO

The RX/TX combiner block is a unique feature of the DA14592. It makes sure that the received power is applied to LNA with minimum losses towards RFPA. In the TX mode, LNA poses a minimal load for RFPA, and its input pins are protected from the RFPA. In both modes, the single-ended RFIO port is matched to 50 Ω to provide the simplest possible interfacing to the antenna on the printed circuit board.

34.2.5 Biasing

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

34.2.6 RF monitoring

The radio is equipped with a monitoring block and its responsibilities are:

- To acquire the data provided by the functional RF units and other various analog resources
- To pack the data in words of 32 bits (when necessary)
- To store words in the system memory to achieve the production test of the corresponding blocks.

The data can be the output of the Demodulator (I and Q) and the data provided by GPADC.

35 Registers

This section contains a detailed view of the DA14592 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 Cortex-M33 and Cortex-M0+ can be found in the following documents, available on the Arm website:

Devices generic user guide:
arm_cortex_m33_dgug_100235_0002_00_en.pdf
DUI0662B_cortex_m0p_r0p1_dgug.pdf

Technical reference manual:
Cortex_m33_trm_100230_0002_00_en.pdf
DDI0484C_cortex_m0p_r0p1_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).

35.1 AMBA bus registers

Table 105: Register map SYSB

Address	Register	Description
0x50060400	QSPI_ARB_REG	ICM_S1 Priorities Register
0x50060404	BRIDGE_REG	H2H CMAC Synchronization Register
0x50060408	FLASH_ARB_REG	FPU/ICM_S2 Register

Table 106: QSPI_ARB_REG (0x50060400)

Bit	Mode	Symbol/Description	Reset
7:6	R/W	AHB_DMA_PRIO Priority AHB_DMA layer system bus 0x0: Highest priority 0x1: Second priority 0x2: Third priority 0x3: Fourth priority	0x3
5:4	R/W	AHB_CPUC_PRIO Priority AHB_CPUC layer system bus 0x0: Highest priority 0x1: Second priority 0x2: Third priority 0x3: Fourth priority	0x2
3:2	R/W	AHB_CPUS_PRIO Priority AHB_CPUS layer system bus 0x0: Highest priority 0x1: Second priority 0x2: Third priority 0x3: Fourth priority	0x1
1:0	R/W	AHB_CMAC_PRIO	0x0

Bit	Mode	Symbol/Description	Reset
		Priority AHB_CMAC layer system bus 0x0: Highest priority 0x1: Second priority 0x2: Third priority 0x3: Fourth priority	

Table 107: BRIDGE_REG (0x50060404)

Bit	Mode	Symbol/Description	Reset
1	R/W	SYNC_BYPASS CMAC bridge, internal synchronization stages are bypassed. 0: Disabled 1: Enabled	0x0
0	R/W	BRIDGE_BYPASS CMAC bridge is bypassed only allowed when cmac hclk equals system hclk 0: Disabled 1: Enabled	0x0

Table 108: FLASH_ARB_REG (0x50060408)

Bit	Mode	Symbol/Description	Reset
0	R/W	ENABLE_SEQ Arbitration flash memory 0: Arbitration is carried out on every transfer 1: Arbitration is carried out at the beginning of every burst transfer	0x0

Table 109: Register map DW

Address	Register	Description
0x30020000	AHB_DMA_PL1_REG	AHB-DMA layer priority level for RFTP (AHB DMA layer only)
0x30020004	AHB_DMA_PL2_REG	AHB-DMA layer priority level for GPDMA (AHB DMA layer only)
0x30020008	AHB_DMA_PL3_REG	AHB-DMA layer Priority level for CRYPTO-DMA (AHB DMA layer only)
0x30020048	AHB_DMA_DFLT_MASTER_REG	Default master ID number (AHB DMA layer only)
0x3002004C	AHB_DMA_WTEN_REG	Weighted-Token Arbitration Scheme Enable (AHB DMA layer only)
0x30020050	AHB_DMA_TCL_REG	Master clock refresh period (AHB DMA layer only)
0x30020054	AHB_DMA_CCLM1_REG	RFTP Master clock tokens (AHB DMA layer only)
0x30020058	AHB_DMA_CCLM2_REG	GPDMA Master clock tokens (AHB DMA layer only)
0x3002005C	AHB_DMA_CCLM3_REG	CRYPTO-DMA Master clock tokens (AHB DMA layer only)
0x30020090	AHB_DMA_VERSION_REG	Version ID (AHB DMA layer only)

Table 110: AHB_DMA_PL1_REG (0x30020000)

Bit	Mode	Symbol/Description	Reset
31:4	R	- Reserved	0x0
3:0	R/W	AHB_DMA_PL1 Arbitration priority for master RFPT. 0: lowest, 15: highest.	0xF

Table 111: AHB_DMA_PL2_REG (0x30020004)

Bit	Mode	Symbol/Description	Reset
31:4	R/W	- Reserved	0x0
3:0	R/W	AHB_DMA_PL2 Arbitration priority for master GPDMA. 0: lowest, 15: highest.	0xE

Table 112: AHB_DMA_PL3_REG (0x30020008)

Bit	Mode	Symbol/Description	Reset
31:4	R	- Reserved	0x0
3:0	R/W	AHB_DMA_PL3 Arbitration priority for master CRYPTO-DMA. 0: lowest, 15: highest.	0xD

Table 113: AHB_DMA_DFLT_MASTER_REG (0x30020048)

Bit	Mode	Symbol/Description	Reset
31:4	R	- Reserved	0x0
3:0	R/W	AHB_DMA_DFLT_MASTER Default master ID number register. The default master is the master that is granted by the bus when no master has requested ownership. 0: Dummy master 1: RFPT 2: GEN-DMA 3: CRYPTO-DMA 4: Reserved	0x0

Table 114: AHB_DMA_WTEN_REG (0x3002004C)

Bit	Mode	Symbol/Description	Reset
31:1	R	- Reserved	0x0
0	R/W	AHB_DMA_WTEN	0x0

Bit	Mode	Symbol/Description	Reset
		Weighted-token arbitration scheme enable. 0: Disabled 1: Enabled	

Table 115: AHB_DMA_TCL_REG (0x30020050)

Bit	Mode	Symbol/Description	Reset
31:16	R	- Reserved	0x0
15:0	R/W	AHB_DMA_TCL Master clock refresh period, counting clock cycles. An arbitration period is defined over this number of tokens. When a new arbitration period starts, the master counters are reloaded. Recommended value is the sum of the AHB_DMA_CCLMx_REG values plus two tokens for each master, that is plus six.	0xFFFF

Table 116: AHB_DMA_CCLM1_REG (0x30020054)

Bit	Mode	Symbol/Description	Reset
31:16	R	- Reserved	0x0
15:0	R/W	AHB_DMA_CCLM Number of tokens (counted in AHB clock cycles) that a master can use on the bus before it has to arbitrate on a bus master with low priority and having tokens. Masters with tokens remaining have priority over masters that have used all of their tokens. User should configure all the token values ensuring that the sum does not exceed the total allocated number of tokens. If a value of zero is configured, then the bus is deemed to have infinite tokens and will always operate in the upper-tier of arbitration.	0xF

Table 117: AHB_DMA_CCLM2_REG (0x30020058)

Bit	Mode	Symbol/Description	Reset
31:16	R	- Reserved	0x0
15:0	R/W	AHB_DMA_CCLM See AHB_DMA_CCLM1_REG	0xF

Table 118: AHB_DMA_CCLM3_REG (0x3002005C)

Bit	Mode	Symbol/Description	Reset
31:16	R	- Reserved	0x0
15:0	R/W	AHB_DMA_CCLM AHB_DMA_CCLM1_REG	0xF

Table 119: AHB_DMA_VERSION_REG (0x30020090)

Bit	Mode	Symbol/Description	Reset
31:0	R	AHB_DMA_VERSION System bus version ID	0x3231332A

35.2 Memory controller registers

Table 120: Register map MEMCTRL

Address	Register	Description
0x50060004	MEM_PRIO_REG	Priority Control Register
0x50060008	MEM_STALL_REG	Maximum Stall cycles Control Register
0x5006000C	MEM_STATUS_REG	Memory Arbiter Status Register
0x50060010	MEM_STATUS2_REG	RAM cells Status Register
0x50060020	CMI_CODE_BASE_REG	CMAC code Base Address Register
0x50060024	CMI_DATA_BASE_REG	CMAC data Base Address Register
0x50060028	CMI_SHARED_BASE_REG	CMAC shared data Base Address Register
0x50060074	BUSY_SET_REG	BSR Set Register
0x50060078	BUSY_RESET_REG	BSR Reset Register
0x5006007C	BUSY_STAT_REG	BSR Status Register

Table 121: MEM_PRIO_REG (0x50060004)

Bit	Mode	Symbol/Description	Reset
17:15	R/W	- Reserved	0x0
14:12	R/W	- Reserved	0x0
11:9	R/W	- Reserved	0x0
8:6	R/W	AHB3_PRIO Priority of the AHB3 interface for the arbitration that is performed per memory cell. Six priority classes are supported, from priority class 0 (lowest priority) to priority class 5 (highest priority). The AHB3 interface can participate in one of the priority classes by setting the corresponding value: from 0 up to 5. The values 6 and 7 are reserved.	0x0
5:3	R/W	AHB2_PRIO Priority of the AHB2 interface for the arbitration that is performed per memory cell. Six priority classes are supported, from priority class 0 (lowest priority) to priority class 5 (highest priority). The AHB2 interface can participate in one of the priority classes by setting the corresponding value: from 0 up to 5. The values 6 and 7 are reserved.	0x0
2:0	R/W	AHB_PRIO	0x0

Bit	Mode	Symbol/Description	Reset
		Priority of the AHB interface for the arbitration that is performed per memory cell. Six priority classes are supported, from priority class 0 (lowest priority) to priority class 5 (highest priority). The AHB interface can participate in one of the priority classes by setting the corresponding value: from 0 up to 5. The values 6 and 7 are reserved.	

Table 122: MEM_STALL_REG (0x50060008)

Bit	Mode	Symbol/Description	Reset
23:20	R/W	- Reserved	0xF
19:16	R/W	- Reserved	0xF
15:12	R/W	- Reserved	0xF
11:8	R/W	AHB3_MAX_STALL Maximum allowed number of stall cycles for the AHB3 interface. If exceeded, the interface will get top priority (above high priority). Valid for a single access so the next access (of a burst) might end up in the queue for the same number of wait cycles. 0: do not use, not feasible and can block other interfaces 1: max 1 stall cycle 15: max 15 stall cycles	0xF
7:4	R/W	AHB2_MAX_STALL Maximum allowed number of stall cycles for the AHB2 interface. If exceeded, the interface will get top priority (above high priority). Valid for a single access so the next access (of a burst) might end up in the queue for the same number of wait cycles. 0: do not use, not feasible and can block other interfaces 1: max 1 stall cycle 15: max 15 stall cycles	0xF
3:0	R/W	AHB_MAX_STALL Maximum allowed number of stall cycles for the AHB interface. If exceeded, the interface will get top priority (above high priority). Valid for a single access so the next access (of a burst) might end up in the queue for the same number of wait cycles. 0: do not use, not feasible and can block other interfaces 1: max 1 stall cycle 15: max 15 stall cycles	0xF

Table 123: MEM_STATUS_REG (0x5006000C)

Bit	Mode	Symbol/Description	Reset
21	W	MTB_CLEAR_READY Writing 1 clears MTB_NOT_READY bit.	0x0
20	R	MTB_NOT_READY 0: Normal operation	0x0

Bit	Mode	Symbol/Description	Reset
		1: MTB access performed which could not be handled right away (interface does not allow wait cycles)	
19:16	R	AHB3_WR_BUFF_CNT The maximum number of arbiter clock cycles that an AHB3 access has been buffered.	0x0
15:12	R	AHB2_WR_BUFF_CNT The maximum number of arbiter clock cycles that an AHB2 access has been buffered.	0x0
11:8	R	AHB_WR_BUFF_CNT The maximum number of arbiter clock cycles that an AHB access has been buffered.	0x0
6	W	AHB3_CLR_WR_BUFF Writing 1 clears AHB3_WR_BUFF_CNT.	0x0
5	W	AHB2_CLR_WR_BUFF Writing 1 clears AHB2_WR_BUFF_CNT.	0x0
4	W	AHB_CLR_WR_BUFF Writing 1 clears AHB_WR_BUFF_CNT.	0x0
2	R	AHB3_WRITE_BUFF 0: No AHB3 write access is buffered. 1: Currently a single AHB3 write access is buffered in the arbiter.	0x0
1	R	AHB2_WRITE_BUFF 0: No AHB2 write access is buffered. 1: Currently a single AHB2 write access is buffered in the arbiter.	0x0
0	R	AHB_WRITE_BUFF 0: No AHB write access is buffered. 1: Currently a single AHB write access is buffered in the arbiter.	0x0

Table 124: MEM_STATUS2_REG (0x50060010)

Bit	Mode	Symbol/Description	Reset
2	RW1C	RAM3_OFF_BUT_ACCESS Reading 1 indicates RAM3 was off but still access was performed. Writing 1 will clear the status back to 0.	0x0
1	RW1C	RAM2_OFF_BUT_ACCESS Reading 1 indicates RAM2 was off but still access was performed. Writing 1 will clear the status back to 0.	0x0
0	RW1C	RAM1_OFF_BUT_ACCESS Reading 1 indicates RAM1 was off but still access was performed. Writing 1 will clear the status back to 0.	0x0

Table 125: CMI_CODE_BASE_REG (0x50060020)

Bit	Mode	Symbol/Description	Reset
18:8	R/W	CMI_CODE_BASE_ADDR The complete register value is a location to System RAM (limited to this area), where the CMAC code segment is remapped.	0x0
7:0	R	- Reserved	0x0

Table 126: CMI_DATA_BASE_REG (0x50060024)

Bit	Mode	Symbol/Description	Reset
18:2	R/W	CMI_DATA_BASE_ADDR The complete register value is a location to System RAM (limited to this area), where the CMAC data segment is remapped. Typically, this value is higher than CMI_CODE_BASE_REG.	0x0
1:0	R	- Reserved	0x0

Table 127: CMI_SHARED_BASE_REG (0x50060028)

Bit	Mode	Symbol/Description	Reset
18:2	R/W	CMI_SHARED_BASE_ADDR The complete register value is a location to System RAM (limited to this area), where the CMAC shared data segment is remapped. Typically, this value is higher than CMI_DATA_BASE_REG.	0x0
1:0	R	- Reserved	0x0

Table 128: BUSY_SET_REG (0x50060074)

Bit	Mode	Symbol/Description	Reset
31:30	WS	BUSY_SPARE Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
29:28	WS	BUSY_SPARE1 Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
27:26	WS	BUSY_TIMER4 Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
25:24	WS	BUSY_TIMER3 Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0).	0x0

Bit	Mode	Symbol/Description	Reset
		Reading returns 0 to allow read/modify/write to the register.	
23:22	WS	BUSY_TIMER2 Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
21:20	WS	BUSY_TIMER Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
19:18	WS	BUSY_PDM Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
17:16	WS	BUSY_PCM Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
15:14	WS	BUSY_SRC2 Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
13:12	WS	BUSY_SRC Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
11:10	WS	BUSY_SDADC Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
9:8	WS	BUSY_GPADC Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
7:6	WS	BUSY_I2C Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
5:4	WS	BUSY_SPI Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
3:2	WS	BUSY_UART2 Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	0x0
1:0	WS	BUSY_UART	0x0

Bit	Mode	Symbol/Description	Reset
		Writing a non-zero value to this field sets the corresponding BUSY bit, but only if it was not claimed (BUSY=0). Reading returns 0 to allow read/modify/write to the register.	

Table 129: BUSY_RESET_REG (0x50060078)

Bit	Mode	Symbol/Description	Reset
31:30	RW1C	BUSY_SPARE Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
29:28	RW1C	BUSY_SPARE1 Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
27:26	RW1C	BUSY_TIMER4 Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
25:24	RW1C	BUSY_TIMER3 Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
23:22	RW1C	BUSY_TIMER2 Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
21:20	RW1C	BUSY_TIMER Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
19:18	RW1C	BUSY_PDM Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
17:16	RW1C	BUSY_PCM Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
15:14	RW1C	BUSY_SRC2 Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
13:12	RW1C	BUSY_SRC Clear the BUSY bitfield, by writing the master code which has claimed to this field	0x0

Bit	Mode	Symbol/Description	Reset
		Reading returns 0 to allow read/modify/write to the register.	
11:10	RW1C	BUSY_SDADC Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
9:8	RW1C	BUSY_GPADC Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
7:6	RW1C	BUSY_I2C Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
5:4	RW1C	BUSY_SPI Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
3:2	RW1C	BUSY_UART2 Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0
1:0	RW1C	BUSY_UART Clear the BUSY bitfield, by writing the master code which has claimed to this field Reading returns 0 to allow read/modify/write to the register.	0x0

Table 130: BUSY_STAT_REG (0x5006007C)

Bit	Mode	Symbol/Description	Reset
31:30	R	BUSY_SPARE A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
29:28	R	BUSY_SPARE1 A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
27:26	R	BUSY_TIMER4 A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
25:24	R	BUSY_TIMER3 A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
23:22	R	BUSY_TIMER2 A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
21:20	R	BUSY_TIMER	0x0

Bit	Mode	Symbol/Description	Reset
		A non-zero value indicates the resource is busy. The value represents which master is using it.	
19:18	R	BUSY_PDM A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
17:16	R	BUSY_PCM A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
15:14	R	BUSY_SRC2 A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
13:12	R	BUSY_SRC A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
11:10	R	BUSY_SDADC A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
9:8	R	BUSY_GPADC A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
7:6	R	BUSY_I2C A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
5:4	R	BUSY_SPI A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
3:2	R	BUSY_UART2 A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0
1:0	R	BUSY_UART A non-zero value indicates the resource is busy. The value represents which master is using it.	0x0

35.3 eFlash controller registers

Table 131: Register map FCU

Address	Register	Description
0x50060100	FLASH_CTRL_REG	Flash control register
0x50060104	FLASH_PTNVH1_REG	NVSTR1 hold time register
0x50060108	FLASH_PTPROG_REG	Flash programming time register
0x5006010C	FLASH_PTERASE_REG	Page Erase time register
0x50060110	FLASH_PTME_REG	Mass Erase time register

Address	Register	Description
0x50060114	FLASH_PTWK_SP_REG	Wake-up time of sleep to standby register
0x50060118	FLASH_PTERASE_SEG_REG	Page erase segment time register
0x5006011C	FLASH_RTERASE_TOT_CNT_REG	Total erase time counter register
0x50060120	FLASH_RTERASE_SEG_CNT_REG	Segment erase time counter register

Table 132: FLASH_CTRL_REG (0x50060100)

Bit	Mode	Symbol/Description	Reset
25	R	DISCHARGE_STAT Flash discharging status 1: Discharging state 0: Normal operation	0x0
24	R/W	DFT_EN 1: Enable DFT logic, programming Flash through pins	0x0
23	R/W	VDD_LEVEL_VALUE If VDD_LEVEL_FORCE is set to 0, this register field has no effect. If VDD_LEVEL_FORCE is set to 1, then if VDD_LEVEL_VALUE is: 1: sets vdd_level value to 1 (Should be used if VDD > 1.08 V. Results in Flash input LVEN = 0) 0: sets vdd_level_value to 0 (Should be used if VDD < 1.08 V. Results in Flash input LVEN = 1)	0x0
22	R/W	VDD_LEVEL_FORCE Used with VDD_LEVEL_VALUE to override the vdd_level FCU input signal, to set the LVEN Flash input signal. This register does not affect the actual VDD level. It only affects the indication to the FCU. 1: Override vdd_level input and force vdd_level to VDD_LEVEL_VALUE 0: Use vdd_level input and do not use VDD_LEVEL_VALUE register field	0x0
21	W	ERASE_RESUME Resumes erase on writing 1.	0x0
20	R	ERASE_SUSPEND_STAT Erase Suspend status 1: FLASH erase is suspended. Check FLASH_RTERASE_TOT_CNT_REG for the remaining time needed for the erase to complete. 0: FLASH erase is not suspended. Check PROG_ERS status bit for the erase cycle status.	0x0
19	R/W	ERASE_SUSPEND_MODE Choose mode of erase suspend operation: 1: When a read operation is issued, FCU stalls the read access until the erase segment is completed. 0: When a read operation is issued, erase is immediately suspended and erase segment time is not subtracted from total erase time	0x0

Bit	Mode	Symbol/Description	Reset
18	R/W	ERASE_SUSPEND_EN 1: Enable erase suspend functionality	0x0
17	R	SLEEP Sleep mode status 1: FCU is in sleeping mode. This register's value is cleared on the next read access, and the FCU automatically wakes up. 0: FCU is in standby mode.	0x1
16	R/W	DMA_EN 1: Enable DMA handshake when writing to the FCU. It also disables IRQ generation. 0: Disable DMA handshake when writing to the FCU. IRQ generation remains enabled.	0x0
15	RWS	BUS_ERROR_EN 1: Bus error response enabled. 0: Bus error response disabled. The BUS_ERROR status register is not affected by this register's value. This bit can only be set!! and is reset by RSTn.	0x0
14	R	BUS_ERROR Flash bus error status 1: Bus error occurred in the last FCU AHB access. Register gets cleared on next FCU AHB access. 0: Bus error did not occur in the last FCU AHB access.	0x0
13	W	IRQ_CLEAR Clears the IRQ on writing 1	0x0
12:10	R/W	WAIT_CYCLES Number of wait cycles to be programmed in case HCLK is faster than the worst case access time (Tacc) specified in the Flash memory datasheet. 0: 0 wait cycles 1: 1 wait cycle 2: 2 wait cycles (default) 3: 3 wait cycles 4: 4 wait cycles 5: 5 wait cycles 6: 6 wait cycles 7: 7 wait cycles Note: When dcore is running at 0.9 V, worst case Tacc = 65 ns, when running at 1.2 V, worst case Tacc = 23 ns. The combinatorial path from cell to DFF is 5.66 ns. Use cases: HCLK = 64 MHz, VDD = 1.2 V -> 1 wait cycles HCLK = 32 MHz, VDD = 0.9 V -> 2 wait cycles (default conditions on power up) HCLK = 32 MHz, VDD = 1.2 V -> 0 wait cycles HCLK = 16 MHz, VDD = 0.9 V -> 1 wait cycles HCLK = 16 MHz, VDD = 1.2 V -> 0 wait cycles HCLK = slower than above -> 0 wait cycles Warning: User must not use 0 wait cycles with 64 MHz clk.	0x3

Bit	Mode	Symbol/Description	Reset
9	RWS	FLASH_RPROT If 1 the FLASH cannot be read. This bit can only be **set** and is reset by RSTn.	0x0
8	RWS	FLASH_WPROT If 1 the FLASH cannot be written/erased. This bit can only be **set** and is reset by RSTn.	0x0
7	R/W	FLASH_PROT If 1 the FLASH cannot be read, written, or erased.	0x0
6	R	PROG_RMIN FLASH read mode inhibit. 1: Read mode may not be selected after a write, page erase or mass erase of the selected FLASH as long as this bit is 1. 0: Read mode may be selected.	0x0
5	R	PROG_ERS FLASH erase status 1: Erase cycle in progress 0: Erase cycle ready	0x0
4	R	PROG_WRS FLASH write status 1: Write cycle in progress 0: Write cycle ready	0x0
3	R/W	PROG_SEL Select FLASH access mode: 0: FLASH: Read access mode selected 1: FLASH: Erase/write access mode selected Must be set back to 0 to reduce power consumption.	0x0
2	W	SLEEP_MODE Puts flash in sleep mode for lower leakage current on writing 1. Automatically wakes up on read access.	0x0
1:0	R/W	PROG_MODE Select FLASH programming mode 00: No write or erase to flash(es) possible. 01: Write page 10: Erase page. Address first written to select page. 11: Erase one of the flash blocks completely (bulk erase) If the address that starts the erase belongs to the info block, info block and main block are mass erased. If the address that starts the erase belongs to the main block, only the main block is erased. To only erase the info block, a page erase of the info block must be done.	0x0

Table 133: FLASH_PTNVH1_REG (0x50060104)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	PTNVH1 Program flash NVSTR1 hold time. T = PTNVH1 x 1 μs = 100 μs (min)	0x64

Table 134: FLASH_PTPROG_REG (0x50060108)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	PTPROG Program flash programming time. T = PTPROG x 1 μs = 8 μs (min)	0x8

Table 135: FLASH_PTERASE_REG (0x5006010C)

Bit	Mode	Symbol/Description	Reset
23:0	R/W	PTERASE Program flash page Erase time. T = PTERASE x 1 μs = 80 ms (min) This register can be programmed with values from 1 up to 262143.	0x13880

Table 136: FLASH_PTME_REG (0x50060110)

Bit	Mode	Symbol/Description	Reset
23:0	R/W	PTME Program flash mass Erase time. T = PTME x 1 μs = 80 ms (min) This register can be programmed with values from 1 up to 262143.	0x13880

Table 137: FLASH_PTWK_SP_REG (0x50060114)

Bit	Mode	Symbol/Description	Reset
7:0	R/W	PTWK_SP Program flash wake-up time of sleep to standby T = PTWK_SP x 1 μs = 3 μs	0x3

Table 138: FLASH_PTERASE_SEG_REG (0x50060118)

Bit	Mode	Symbol/Description	Reset
23:0	R/W	PTERASE_SEG Program flash segment of page erase time for suspend erase. T = PTERASE_SEG x 1 μs = 1 ms This register can be programmed with values from 1 up to 262143. Note: PTERASE should be a multiple of PTERASE_SEG	0x3E8

Table 139: FLASH_RTERASE_TOT_CNT_REG (0x5006011C)

Bit	Mode	Symbol/Description	Reset
23:0	R	RTERASE_TOT_CNT Returns total erase time counter value.	0x0

Table 140: FLASH_RTERASE_SEG_CNT_REG (0x50060120)

Bit	Mode	Symbol/Description	Reset
23:0	R	RTERASE_SEG_CNT Returns segment erase time counter value.	0x0

35.4 SPI Flash/RAM registers

Table 141: Register map QSPIC

Address	Register	Description
0x34000000	QSPIC_CTRLBUS_REG	SPI Bus control register for the Manual mode
0x34000004	QSPIC_CTRLMODE_REG	Mode control register
0x34000008	QSPIC_RECVDATA_REG	Received data for the Manual mode
0x3400000C	QSPIC_BURSTCMDA_REG	The way of reading in Auto mode (command register A)
0x34000010	QSPIC_BURSTCMDB_REG	The way of reading in Auto mode (command register B)
0x34000014	QSPIC_STATUS_REG	The status register of the QSPI controller
0x34000018	QSPIC_WRITEDATA_REG	Write data to SPI Bus for the Manual mode
0x3400001C	QSPIC_READDATA_REG	Read data from SPI Bus for the Manual mode
0x34000020	QSPIC_DUMMYDATA_REG	Send dummy clocks to SPI Bus for the Manual mode
0x34000024	QSPIC_ERASECTRL_REG	Erase control register
0x34000028	QSPIC_ERASECMDA_REG	The way of erasing in Auto mode (command register A)
0x3400002C	QSPIC_ERASECMDB_REG	The way of erasing in Auto mode (command register B)
0x34000030	QSPIC_BURSTBRK_REG	Read break sequence in Auto mode
0x34000034	QSPIC_STATUSCMD_REG	The way of reading the status of external device in Auto mode
0x34000038	QSPIC_CHCKERASE_REG	Check erase progress in Auto mode
0x3400003C	QSPIC_GP_REG	General purpose QSPIC register

Address	Register	Description
0x34000040	QSPIC_AWRITECMD_REG	The way of writing in Auto mode when the external device is a serial SRAM
0x34000044	QSPIC_MEMBLEN_REG	External memory burst length configuration

Table 142: QSPIC_CTRLBUS_REG (0x34000000)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0x0
4	W	QSPIC_DIS_CS Write 1 to disable the chip select (active low) when the controller is in Manual mode.	0x0
3	W	QSPIC_EN_CS Write 1 to enable the chip select (active low) when the controller is in Manual mode.	0x0
2	W	QSPIC_SET_QUAD Write 1 to set the bus mode in Quad mode when the controller is in Manual mode.	0x0
1	W	QSPIC_SET_DUAL Write 1 to set the bus mode in Dual mode when the controller is in Manual mode.	0x0
0	W	QSPIC_SET_SINGLE Write 1 to set the bus mode in Single SPI mode when the controller is in Manual mode.	0x0

Table 143: QSPIC_CTRLMODE_REG (0x34000004)

Bit	Mode	Symbol/Description	Reset
31:17	-	- Reserved	0x0
16	R/W	QSPIC_CLK_FREE_EN Controls the behavior of the QSPI_SCK when the QSPI_CS is high and the QSPIC_CS_MD = 1. 0: Is produced one QSPI_SCK clock pulse after each 0 to 1 transition in the QSPI_CS. 1: The QSPI_SCK clock remains always active, while the QSPI_CS is inactive. This setting has meaning only when the QSPIC_CS_MD = 1.	0x0
15	R/W	QSPIC_CS_MD Controls the clock edge with which is produced the QSPI_CS signal. 0: The QSPI_CS is produced with the rising edge of the QSPI_SCK. The QSPI_SCK is always inactive while the QSPI_CS is high. 1: The QSPI_CS is produced with the falling edge of the QSPI_SCK. The behavior of the QSPI_SCK while the QSPI_CS is high, is controlled by the QSPIC_CLK_FREE_EN.	0x0

Bit	Mode	Symbol/Description	Reset
14	R/W	QSPIC_SRAM_EN Defines the type of the external device that is connected on the QSPIC controller 0: The external memory device is a serial Flash 1: The external memory device is a serial SRAM When the external device is a serial SRAM, the erase suspend/ resume functionality of the controller is disabled. In this case the writing of the QSPIC_ERASECTRL_REG[QSPIC_ERASE_EN] bit has no effect. Also, the memory space where the external device is mapped, is considered as writable.	0x0
13	R/W	QSPIC_USE_32BA Controls the length of the address that the external memory device uses. 0: The external memory device uses 24 bits address. 1: The external memory device uses 32 bits address. The controller uses this bit in order to decide the number of the address bytes that has to transfer to the external device during Auto mode.	0x0
12	R/W	QSPIC_FORCENSEQ_EN Controls the way in which a burst request from the AMBA bus is addressed by the QSPI controller. 0: The controller translates a burst access on the AMBA bus as a burst access on the QSPI bus. That results in the minimum number of command/address phases. 1: The controller will split a burst access on the AMBA bus into a number of single accesses on the QSPI bus. That results to a separate command for each beat of the burst. For example, a 4-beat word incremental AMBA read access will be split into four different sequences on the QSPI bus: command/address/extra clock/read data. The QSPI_CS will be low only for the time that is needed for each of these single accesses. This configuration bit is useful when the clock frequency of the QSPI bus is much higher than the clock of the AMBA bus. In this case the interval for which the CS remains low is minimized, achieving lower power dissipation with respect of the case where the QSPIC_FORCENSEQ_EN = 0, at cost of performance.	0x0
11:9	R/W	QSPIC_PCLK_MD Controls the read pipe clock delay relative to the falling edge of QSPI_SCK. Refer to QSPI Timing for timing parameters	0x0
8	R/W	QSPIC_RPIPE_EN Controls the use of the data read pipe. 0: The read pipe is disabled; the sampling clock is defined according to the QSPIC_RXD_NEG setting. 1: The read pipe is enabled. The delay of the sampling clock is defined according to the QSPI_PCLK_MD setting. (Recommended)	0x0
7	R/W	QSPIC_RXD_NEG Defines the clock edge that is used for the capturing of the received data, when the read pipe is not active (QSPIC_RPIPE_EN = 0). 0: Sampling of the received data with the positive edge of the QSPI_SCK 1: Sampling of the received data with the negative edge of the QSPI_SCK The internal QSPI_SCK clock that is used by the controller for the capturing of the received data has a skew in respect of the QSPI_SCK that is received by the external memory device. To improve the timing requirements of the read path, the controller supports a read pipe register with programmable clock delay. See also the QSPIC_RPIPE_EN register.	0x0

Bit	Mode	Symbol/Description	Reset
6	R/W	QSPIC_HRDY_MD This configuration bit is useful when the frequency of the QSPI clock is much lower than the clock of the AMBA bus, to not lock the AMBA bus for a long time. 0: Adds wait states via hready signal when an access is performed on QSPIC_WRITEDATA, QSPIC_READDATA and QSPIC_DUMMYDATA registers. It is not necessary to check the QSPIC_BUSY of the QSPIC_STATUS_REG. 1: The controller does not add wait states via the hready signal, when the access is performed on QSPIC_WRITEDATA, QSPIC_READDATA and QSPIC_DUMMYDATA registers. The QSPIC_BUSY bit of the QSPIC_STATUS_REG must be checked to detect the completion of the requested access. It is applicable only when the controller is in Manual mode. In the case of the Auto mode, the controller always adds wait states via the hready signal.	0x0
5	R/W	QSPIC_IO3_DAT The value of QSPI_IO3 pad if QSPI_IO3_OEN is 1	0x0
4	R/W	QSPIC_IO2_DAT The value of QSPI_IO2 pad if QSPI_IO2_OEN is 1	0x0
3	R/W	QSPIC_IO3_OEN QSPI_IO3 output enable. Use this only in SPI or Dual SPI mode to control /HOLD signal. When the Auto Mode is selected (QSPIC_AUTO_MD = 1) and the QUAD SPI is used, set this bit to zero. 0: The QSPI_IO3 pad is input. 1: The QSPI_IO3 pad is output.	0x0
2	R/W	QSPIC_IO2_OEN QSPI_IO2 output enable. Use this only in SPI or Dual SPI mode to control /WP signal. When the Auto Mode is selected (QSPIC_AUTO_MD = 1) and the QUAD SPI is used, set this bit to zero. 0: The QSPI_IO2 pad is input. 1: The QSPI_IO2 pad is output.	0x0
1	R/W	QSPIC_CLK_MD Mode of the generated QSPI_SCK clock 0: Use Mode 0 for the QSPI_CLK. The QSPI_SCK is low when QSPI_CS is high. 1: Use Mode 3 for the QSPI_CLK. The QSPI_SCK is high when QSPI_CS is high. See also the QSPIC_CS_MD register and the QSPIC_CLK_FREE_EN register.	0x0
0	R/W	QSPIC_AUTO_MD Mode of operation 0: The Manual Mode is selected. 1: The Auto Mode is selected. During an erasing the QSPIC_AUTO_MD goes in read only mode (see QSPIC_ERASE_EN)	0x0

Table 144: QSPIC_RECVDATA_REG (0x34000008)

Bit	Mode	Symbol/Description	Reset
31:0	R	QSPIC_RECVDATA This register contains the received data when the QSPIC_READDATA_REG register is used in Manual mode, to retrieve data from the external memory device and QSPIC_HRDY_MD=1 and QSPIC_BUSY=0.	0x0

Table 145: QSPIC_BURSTCMDA_REG (0x3400000C)

Bit	Mode	Symbol/Description	Reset
31:30	R/W	QSPIC_DMY_TX_MD It describes the mode of the SPI bus during the Dummy bytes phase. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
29:28	R/W	QSPIC_EXT_TX_MD It describes the mode of the SPI bus during the Extra Byte phase. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
27:26	R/W	QSPIC_ADR_TX_MD It describes the mode of the SPI bus during the address phase. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
25:24	R/W	QSPIC_INST_TX_MD It describes the mode of the SPI bus during the instruction phase. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
23:16	R/W	QSPIC_EXT_BYTE The value of an extra byte which will be transferred after address (only if QSPIC_EXT_BYTE_EN= 1). Usually this is the Mode Bits in Dual/Quad SPI I/O instructions.	0x0
15:8	R/W	QSPIC_INST_WB Instruction Value for Wrapping Burst. This value is the selected instruction when QSPIC_WRAP_MD is equal to 1 and the access is a wrapping burst of length and size described by the bit fields QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE respectively.	0x0
7:0	R/W	QSPIC_INST Instruction Value for Incremental Burst or Single read access. This value is the selected instruction at the cases of incremental burst or single read	0x0

Bit	Mode	Symbol/Description	Reset
		access. Also this value is used when a wrapping burst is not supported (QSPIC_WRAP_MD)	

Table 146: QSPIC_BURSTCMDB_REG (0x34000010)

Bit	Mode	Symbol/Description	Reset
31:16	-	- Reserved	0x0
15	R/W	QSPIC_DMY_FORCE By setting this bit, the number of dummy bytes is forced to be equal to 3. In this case the QSPIC_DMY_NUM field is overruled and has no function. 0: The number of dummy bytes is controlled by the QSPIC_DMY_NUM field 1: Three dummy bytes are used. The QSPIC_DMY_NUM is overruled.	0x0
14:12	R/W	QSPIC_CS_HIGH_MIN Between the transmission of two different instructions to the flash memory, the QSPI bus stays in idle state (QSPI_CS high) for at least this number of QSPI_SCK clock cycles. See the QSPIC_ERS_CS_HI and the QSPIC_WR_CS_HIGH_MIN registers for some exceptions.	0x0
11:10	R/W	QSPIC_WRAP_SIZE It describes the selected data size of a wrapping burst (QSPIC_WRAP_MD). 0x0: Byte access (8-bits) 0x1: Half word access (16 bits) 0x2: Word access (32-bits) 0x3: Reserved	0x0
9:8	R/W	QSPIC_WRAP_LEN It describes the selected length of a wrapping burst (QSPIC_WRAP_MD). 0x0: 4 beat wrapping burst 0x1: 8 beat wrapping burst 0x2: 16 beat wrapping burst 0x3: Reserved	0x0
7	R/W	QSPIC_WRAP_MD Wrap mode 0: The QSPIC_INST is the selected instruction at any access. 1: The QSPIC_INST_WB is the selected instruction at any wrapping burst access of length and size described by the registers QSPIC_WRAP_LEN and QSPIC_WRAP_SIZE respectively. In all other cases the QSPIC_INST is the selected instruction. Use this feature only when the serial FLASH memory supports a special instruction for wrapping burst access.	0x0
6	R/W	QSPIC_INST_MD Instruction mode 0: Transmit instruction at any burst access. 1: Transmit instruction only in the first access after the selection of Auto Mode.	0x0
5:4	R/W	QSPIC_DMY_NUM Number of Dummy Bytes	0x0

Bit	Mode	Symbol/Description	Reset
		0x0: Zero Dummy Bytes (Do not Send Dummy Bytes) 0x1: Send 1 Dummy Byte 0x2: Send 2 Dummy Bytes 0x3: Send 4 Dummy Bytes When QSPIC_DMY_FORCE is enabled, the QSPIC_DMY_NUM is overruled. In this case the number of dummy bytes is defined by QSPIC_DMY_FORCE and is equal to 3, independent of the value of QSPIC_DMY_NUM.	
3	R/W	QSPIC_EXT_HF_DS Extra Half Disable Output 0: If QSPIC_EXT_BYTE_EN=1, then transmit the complete QSPIC_EXT_BYTE 1: If QSPIC_EXT_BYTE_EN=1, then disable (hi-z) output during the transmission of bits [3:0] of QSPIC_EXT_BYTE	0x0
2	R/W	QSPIC_EXT_BYTE_EN Extra Byte Enable 0: Do not Send QSPIC_EXT_BYTE 1: Send QSPIC_EXT_BYTE	0x0
1:0	R/W	QSPIC_DAT_RX_MD It describes the mode of the SPI bus during the data phase. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0

Table 147: QSPIC_STATUS_REG (0x34000014)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	QSPIC_BUSY The status of the SPI Bus. 0: The SPI Bus is idle 1: The SPI Bus is active. Read data, write data, or dummy data activity is in progress. Has meaning only in Manual mode and only when QSPIC_HRDY_MD = 1.	0x0

Table 148: QSPIC_WRITEDATA_REG (0x34000018)

Bit	Mode	Symbol/Description	Reset
31:0	W	QSPIC_WRITEDATA Writing to this register generates a data transfer from the controller to the external memory device. The data written in this register is then transferred to the memory using the selected mode of the SPI bus (SPI, Dual SPI, Quad SPI). The data size of the access to this register can be 32-bits/16-bits/8-bits and is equal to the number of the transferred bits. This register has meaning only when the controller is in Manual mode.	0x0

Table 149: QSPIC_READDATA_REG (0x3400001C)

Bit	Mode	Symbol/Description	Reset
31:0	R	QSPIC_READDATA A read access at this register generates a data transfer from the external memory device to the QSPIC controller. The data is transferred using the selected mode of the SPI bus (SPI, Dual SPI, Quad SPI). The data size of the access to this register can be 32-bits/16-bits/8-bits and is equal to the number of the transferred bits. This register has meaning only when the controller is in Manual mode.	0x0

Table 150: QSPIC_DUMMYDATA_REG (0x34000020)

Bit	Mode	Symbol/Description	Reset
31:0	W	QSPIC_DUMMYDATA Writing to this register generates a number of clock pulses to the SPI bus. During the last clock of this activity in the SPI bus, the QSPI_IOx data pads are in hi-z state. The data size of the access to this register can be 32-bits/16-bits/8-bits. The number of generated pulses is equal to: (size of AHB bus access)/(size of SPI bus). The size of SPI bus is equal to 1, 2, or 4 for Single, Dual, or Quad SPI mode respectively. This register has meaning only when the controller is in Manual mode.	0x0

Table 151: QSPIC_ERASECTRL_REG (0x34000024)

Bit	Mode	Symbol/Description	Reset
31:28	-	- Reserved	0x0
27:25	R	QSPIC_ERS_STATE It shows the progress of sector/block erasing (read only). 0x0: No Erase. 0x1: Pending erase request 0x2: Erase procedure is running 0x3: Suspended Erase procedure 0x4: Finishing the Erase procedure 0x5…0x7: Reserved	0x0
24	R/W	QSPIC_ERASE_EN This bit has meaning only when the external device is a serial FLASH (QSPIC_SRAM_EN=0). During Manual mode (QSPIC_AUTO_MD = 0): This bit is in read-only mode. During Auto mode (QSPIC_AUTO_MD = 1). To request the erasing of the block/sector (QSPIC_ERS_ADDR, 12'b0) write 1 to this bit. This bit is cleared automatically with the end of erasing. Until the end of erasing the QSPIC_ERASE_EN remains in read-only mode. During the same period of time, the controller remains in Auto Mode (QSPIC_AUTO_MD goes in read-only mode). In the case where the external device is a serial SRAM (QSPIC_SRAM_EN = 1) this bit is in read-only mode.	0x0
23:4	R/W	QSPIC_ERS_ADDR	0x0

Bit	Mode	Symbol/Description	Reset
		Defines the address of the block/sector that is requested to be erased. If QSPIC_USE_32BA = 0 (24 bits addressing), bits QSPIC_ERASECTRL_REG[23-12] determine the block/ sector address bits [23-12]. QSPIC_ERASECTRL_REG[11-4] are ignored by the controller. If QSPIC_USE_32BA = 1 (32 bits addressing) bits QSPIC_ERASECTRL_REG[23-4] determine the block / sectors address bits [31:12]	
3:0	-	- Reserved	0x0

Table 152: QSPIC_ERASECMDA_REG (0x34000028)

Bit	Mode	Symbol/Description	Reset
31:24	R/W	QSPIC_RES_INST The code value of the erase resume instruction	0x0
23:16	R/W	QSPIC_SUS_INST The code value of the erase suspend instruction.	0x0
15:8	R/W	QSPIC_WEN_INST The code value of the write enable instruction.	0x0
7:0	R/W	QSPIC_ERS_INST The code value of the erase instruction.	0x0

Table 153: QSPIC_ERASECMDB_REG (0x3400002C)

Bit	Mode	Symbol/Description	Reset
31:30	-	- Reserved	0x0
29:24	R/W	QSPIC_RESSUS_DLY Defines a timer that counts the minimum allowed delay between an erase suspend command and the previous erase resume command (or the initial erase command). 0x00: Do not wait. The controller starts immediately to suspend the erase procedure. 0x01…0x3F: The controller waits for at least this number of 288 kHz clock cycles before the suspension of erasing. Time starts counting after the end of the previous erase resume command (or the initial erase command)	0x0
23:20	-	- Reserved	0x0
19:16	R/W	QSPIC_ERSRES_HLD The controller must stay without flash memory reading requests for this number of AMBA hclk clock cycles, before to perform the command of erase or erase resume. Allowable range: 0xF – 0x0	0x0
15	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
14:10	R/W	QSPIC_ERS_CS_HI After the execution of instructions: write enable, erase, erase suspend and erase resume, the QSPI_CS remains high for at least this number of QSPI_SCK clock cycles.	0x0
9:8	R/W	QSPIC_EAD_TX_MD The mode of the SPI Bus during the address phase of the erase instruction 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
7:6	R/W	QSPIC_RES_TX_MD The mode of the SPI Bus during the transmission of the resume instruction 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
5:4	R/W	QSPIC_SUS_TX_MD The mode of the SPI Bus during the transmission of the suspend instruction. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
3:2	R/W	QSPIC_WEN_TX_MD The mode of the SPI Bus during the transmission of the write enable instruction. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
1:0	R/W	QSPIC_ERS_TX_MD The mode of the SPI Bus during the instruction phase of the erase instruction 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0

Table 154: QSPIC_BURSTBRK_REG (0x34000030)

Bit	Mode	Symbol/Description	Reset
31:21	-	- Reserved	0x0
20	R/W	QSPIC_SEC_HF_DS	0x0

Bit	Mode	Symbol/Description	Reset
		Disable output during the transmission of the second half (QSPIC_BRK_WRD[3:0]). Setting this bit is only useful if QSPIC_BRK_EN =1 and QSPIC_BRK_SZ= 1. 0: The controller drives the SPI bus during the transmission of the QSPIC_BRK_WRD[3:0]. 1: The controller leaves the SPI bus in Hi-Z during the transmission of the QSPIC_BRK_WORD[3:0].	
19:18	R/W	QSPIC_BRK_TX_MD The mode of the SPI Bus during the transmission of the read break sequence. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
17	R/W	QSPIC_BRK_SZ The size of the read break sequence. 0: One byte (Send QSPIC_BRK_WRD[15:8]) 1: Two bytes (Send QSPIC_BRK_WRD[15:0])	0x0
16	R/W	QSPIC_BRK_EN Controls the application of a special command (read break sequence) that is used in order to force the device to abandon the continuous read mode. 0: The special command is not applied 1: The special command is applied This special command is applied by the controller to the external device under the following conditions: - the controller is in Auto mode - the QSPIC_INST_MD = 1 - the previous command that has been applied in the external device was read - the controller wants to apply to the external device a command different than the read.	0x0
15:0	R/W	QSPIC_BRK_WRD This is the value of a special command (read break sequence) that is applied by the controller to the external memory device, to force the memory device to abandon the continuous read mode.	0x0

Table 155: QSPIC_STATUSCMD_REG (0x34000034)

Bit	Mode	Symbol/Description	Reset
31:23	-	- Reserved	0x0
22	R/W	QSPIC_STSDLY_SEL Defines the timer which is used to count the delay that it has to wait before to read the FLASH Status Register, after an erase or an erase resume command. 0: The delay is controlled by the QSPIC_RESSTS_DLY which counts on the QSPI clock.	0x0

Bit	Mode	Symbol/Description	Reset
		1: The delay is controlled by the QSPIC_RESSUS_DLY which counts on the 288 kHz clock.	
21:16	R/W	QSPIC_RESSTS_DLY Defines a timer that counts the minimum required delay between the reading of the status register and of the previous erase or erase resume instruction. 0x00: Do not wait. The controller starts reading the Flash memory status register immediately. 0x01…0x3F: The controller waits for at least this number of QSPI_CLK cycles and afterwards it starts to reading the Flash memory status register. The timer starts to count after the end of the previous erase or erase resume command. The actual timer that will be used by the controller before the reading of the Flash memory status register is defined by the QSPIC_STSDLY_SEL.	0x0
15	R/W	QSPIC_BUSY_VAL Defines the value of the Busy bit which means that the flash is busy. 0: The flash is busy when the Busy bit is equal to 0. 1: The flash is busy when the Busy bit is equal to 1.	0x0
14:12	R/W	QSPIC_BUSY_POS Defines the bit of the Flash status register which represents the Busy bit (0x7 - 0x0).	0x0
11:10	R/W	QSPIC_RSTAT_RX_MD The mode of the SPI Bus during the reception phase of the read status instruction, where the value of status register is retrieved. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
9:8	R/W	QSPIC_RSTAT_TX_MD The mode of the SPI Bus during the instruction phase of the read status instruction. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
7:0	R/W	QSPIC_RSTAT_INST The code value of the read status instruction. It is transmitted during the instruction phase of the read status instruction.	0x0

Table 156: QSPIC_CHCKERASE_REG (0x34000038)

Bit	Mode	Symbol/Description	Reset
31:0	W	QSPIC_CHCKERASE Writing any value to this register during erasing, forces the controller to read the flash memory status register. Depending on the value of the Busy bit, it updates the QSPIC_ERASE_EN. This register has meaning only when the controller is in Auto mode and there is an erase in progress (QSPIC_ERASE_EN = 1). It has no meaning when the external device is a serial SRAM.	0x0

Table 157: QSPIC_GP_REG (0x3400003C)

Bit	Mode	Symbol/Description	Reset
4:3	R/W	QSPIC_PADS_SLEW QSPI pads slew rate control. Indicative values under certain conditions: 0x0: Rise = 1.7 V/ns, Fall = 1.9 V/ns (weak) 0x1: Rise = 2.0 V/ns, Fall = 2.3 V/ns 0x2: Rise = 2.3 V/ns, Fall = 2.6 V/ns 0x3: Rise = 2.4 V/ns, Fall = 2.7 V/ns (strong) Conditions: FLASH pin capacitance 6 pF, Vcc = 1.8 V, T = 25 °C and Idrive = 16 mA	0x0
2:1	R/W	QSPIC_PADS_DRV QSPI pads drive current 0x0: 4 mA 0x1: 8 mA 0x2: 12 mA 0x3: 16 mA	0x0
0	R/W	- Reserved	0x0

Table 158: QSPIC_AWRITECMD_REG (0x34000040)

Bit	Mode	Symbol/Description	Reset
31:19	-	- Reserved	0x0
18:14	R/W	QSPIC_WR_CS_HIGH_MIN After the execution of the write command, the QSPI_CS remains high for at least this number of QSPI_SCK clock cycles.	0x0
13:12	R/W	QSPIC_WR_DAT_TX_MD The mode of the SPI Bus during the data phase of the write command. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
11:10	R/W	QSPIC_WR_ADR_TX_MD The mode of the SPI Bus during the address phase of the write command. 0x0: Single SPI 0x1: Dual 0x2: Quad 0x3: Reserved	0x0
9:8	R/W	QSPIC_WR_INST_TX_MD The mode of the SPI Bus during the instruction phase of the write command. 0x0: Single SPI 0x1: Dual	0x0

Bit	Mode	Symbol/Description	Reset
		0x2: Quad 0x3: Reserved	
7:0	R/W	QSPIC_WR_INST This is the value of the instruction that is used, to be programmed the external SRAM device.	0x0

Table 159: QSPIC_MEMBLEN_REG (0x34000044)

Bit	Mode	Symbol/Description	Reset
31:14	-	- Reserved	0x0
13:4	R/W	QSPIC_T_CEM_CC Defines the maximum allowed time tCEM for which the QSPIC_CS can stay active (QSPI_CS = 0). It has meaning only when QSPIC_T_CEM_EN is equal to 1. See also the description of the QSPIC_T_CEM_EN for more details. The tCEM is expressed in number of qspi clock cycles and can be calculated as follows: tCEM / (qspi_clock_period) If the result of the above equation is higher than 0x3FF, use the value 0x3FF.	0x0
3	R/W	QSPIC_T_CEM_EN This bit enables the controlling of the maximum time tCEM for which the QSPI_CS remains active. It has meaning only when the Auto mode is active (QSPIC_AUTO_MD = 1) and the external device is a serial SRAM (QSPIC_SRAM_EN = 1). In the case where the external device is a serial Flash (QSPIC_SRAM_EN = 0) or the controller is in Manual mode (QSPIC_AUTO_MD = 0), this field has no any effect. This feature is useful when the external serial device is a dynamic RAM that requires refresh. If the refresh is applied only when the device is in the idle state (QSPI_CS = 1), the time for which the device remains in the active state (QSPI_CS = 0) should be limited by a maximum threshold. 0: There is no any constraint regarding the maximum allowed time for which QSPI_CS can stay active. This is the case also when QSPIC_SRAM_EN = 0 or QSPIC_AUTO_MD = 0. 1: There is a maximum allowed time interval tCEM for which QSPI_CS can stay active during a burst access (for reading or writing of data). For the controller, this is considered as equal to QSPIC_T_CEM_CC x qspi_clock_period. In the case where the data transfer requires QSPI_CS to stay active for more than QSPIC_T_CEM_CC QSPI clock cycles, the QSPI controller splits the access on the SPI bus in more than one bursts, by inserting inactive periods (QSPI_CS = 0) between them. This will cost extra clock cycles for the realization of the original access, due to the additional commands that are required in the SPI bus. The value in QSPIC_T_CEM_CC should be updated every time where the frequency of the QSPI clock is modified. The QSPI clock frequency should not be decreased more than a lowest frequency. This is the lowest frequency that enables to be performed a 32-bit word read and write access, without violating the tCEM timing requirement (the QSPI controller allows to be performed at least the transferring of one beat of the requested burst, independent of the QSPIC_T_CEM_CC limit).	0x0
2:0	R/W	QSPIC_MEMBLEN In this register, the expected behavior of the external memory device regarding the length of a burst operation is defined:	0x0

Bit	Mode	Symbol/Description	Reset
		0x0: The external memory device is capable of implementing incremental burst of unspecified length. 0x1: The external memory device implements a wrapping burst of length 4 bytes. 0x2: The external memory device implements a wrapping burst of length 8 bytes. 0x3: The external memory device implements a wrapping burst of length 16 bytes. 0x4: The external memory device implements a wrapping burst of length 32 bytes. 0x5: The external memory device implements a wrapping burst of length 64 bytes. 0x6 – 0x7: Reserved This setting is used by the QSPI controller when the Auto mode is enabled (QSPIC_AUTO_MD = 1) to handle the various burst requests of the AHB bus, in respect of the requirements of the external memory device. The external memory device may need to be configured by applying special instruction to define the kind of the burst operation. This can be implemented by applying this special instruction with the QSPI controller in Manual mode (QSPIC_AUTO_MD = 1). Refer to the datasheet of the external device for more information.	

35.5 Real time clock registers

Table 160: Register map RTC

Address	Register	Description
0x50000400	RTC_CONTROL_REG	RTC Control Register
0x50000404	RTC_HOUR_MODE_REG	RTC Hour Mode Register
0x50000408	RTC_TIME_REG	RTC Time Register
0x5000040C	RTC_CALENDAR_REG	RTC Calendar Register
0x50000410	RTC_TIME_ALARM_REG	RTC Time Alarm Register
0x50000414	RTC_CALENDAR_ALARM_REG	RTC Calendar Alarm Register
0x50000418	RTC_ALARM_ENABLE_REG	RTC Alarm Enable Register
0x5000041C	RTC_EVENT_FLAGS_REG	RTC Event Flags Register
0x50000420	RTC_INTERRUPT_ENABLE_REG	RTC Interrupt Enable Register
0x50000424	RTC_INTERRUPT_DISABLE_REG	RTC Interrupt Disable Register
0x50000428	RTC_INTERRUPT_MASK_REG	RTC Interrupt Mask Register
0x5000042C	RTC_STATUS_REG	RTC Status Register
0x50000430	RTC_KEEP_RTC_REG	RTC Keep RTC Register

Address	Register	Description
0x50000480	RTC_EVENT_CTRL_REG	RTC Event Control Register
0x50000488	RTC_PDC_EVENT_PERIOD_REG	RTC PDC Event Period Register
0x5000048C	RTC_PDC_EVENT_CLEAR_REG	RTC PDC Event Clear Register
0x50000494	RTC_PDC_EVENT_CNT_REG	RTC PDC Event Counter Register

Table 161: RTC_CONTROL_REG (0x50000400)

Bit	Mode	Symbol/Description	Reset
1	R/W	RTC_CAL_DISABLE When this field is set high the RTC stops incrementing the calendar value.	0x1
0	R/W	RTC_TIME_DISABLE When this field is set high the RTC stops incrementing the time value.	0x1

Table 162: RTC_HOUR_MODE_REG (0x50000404)

Bit	Mode	Symbol/Description	Reset
0	R/W	RTC_HMS When this field is set high the RTC operates in 12-hour clock mode; otherwise, times are in 24 hour clock format.	0x0

Table 163: RTC_TIME_REG (0x50000408)

Bit	Mode	Symbol/Description	Reset
31	R/W	RTC_TIME_CH The value in this register has altered since last read. Read and clear.	0x0
30	R/W	RTC_TIME_PM In 12 hour clock mode, indicates PM when set.	0x0
29:28	R/W	RTC_TIME_HR_T Hours tens. Represented in BCD digit (0-2).	0x0
27:24	R/W	RTC_TIME_HR_U Hours units. Represented in BCD digit (0-9).	0x0
23	-	- Reserved	0x0
22:20	R/W	RTC_TIME_M_T Minutes tens. Represented in BCD digit (0-5).	0x0
19:16	R/W	RTC_TIME_M_U Minutes units. Represented in BCD digit (0-9).	0x0
15	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
14:12	R/W	RTC_TIME_S_T Seconds tens. Represented in BCD digit (0-9).	0x0
11:8	R/W	RTC_TIME_S_U Seconds units. Represented in BCD digit (0-9).	0x0
7:4	R/W	RTC_TIME_H_T Hundredths of a second tens. Represented in BCD digit (0-9).	0x0
3:0	R/W	RTC_TIME_H_U Hundredths of a second units. Represented in BCD digit (0-9).	0x0

Table 164: RTC_CALENDAR_REG (0x5000040C)

Bit	Mode	Symbol/Description	Reset
31	R/W	RTC_CAL_CH The value in this register has altered since last read. Read and clear	0x0
30	-	- Reserved	0x0
29:28	R/W	RTC_CAL_C_T Century tens. Represented in BCD digit (1-2).	0x2
27:24	R/W	RTC_CAL_C_U Century units. Represented in BCD digit (0-9).	0x0
23:20	R/W	RTC_CAL_Y_T Year tens. Represented in BCD digit (0-9).	0x0
19:16	R/W	RTC_CAL_Y_U Year units. Represented in BCD digit (0-9).	0x0
15:14	-	- Reserved	0x0
13:12	R/W	RTC_CAL_D_T Date tens. Represented in BCD digit (0-3).	0x0
11:8	R/W	RTC_CAL_D_U Date units. Represented in BCD digit (0-9).	0x1
7	R/W	RTC_CAL_M_T Month tens. Represented in BCD digit (0-1).	0x0
6:3	R/W	RTC_CAL_M_U Month units. Represented in BCD digit (0-9).	0x1
2:0	R/W	RTC_DAY Day of the week (arbitrary) units. Represented in BCD digit (0-7).	0x7

Table 165: RTC_TIME_ALARM_REG (0x50000410)

Bit	Mode	Symbol/Description	Reset
31	-	- Reserved	0x0
30	R/W	RTC_TIME_PM In 12-hour clock mode, indicates PM when set.	0x0
29:28	R/W	RTC_TIME_HR_T Hours tens. Represented in BCD digit (0-2).	0x0
27:24	R/W	RTC_TIME_HR_U Hours units. Represented in BCD digit (0-9).	0x0
23	-	- Reserved	0x0
22:20	R/W	RTC_TIME_M_T Minutes tens. Represented in BCD digit (0-5).	0x0
19:16	R/W	RTC_TIME_M_U Minutes units. Represented in BCD digit (0-9).	0x0
15	-	- Reserved	0x0
14:12	R/W	RTC_TIME_S_T Seconds tens. Represented in BCD digit (0-9).	0x0
11:8	R/W	RTC_TIME_S_U Seconds units. Represented in BCD digit (0-9).	0x0
7:4	R/W	RTC_TIME_H_T Hundredths of a second tens. Represented in BCD digit (0-9).	0x0
3:0	R/W	RTC_TIME_H_U Hundredths of a second units. Represented in BCD digit (0-9).	0x0

Table 166: RTC_CALENDAR_ALARM_REG (0x50000414)

Bit	Mode	Symbol/Description	Reset
31:14	R/W	- Reserved	0x0
13:12	R/W	RTC_CAL_D_T Date tens. Represented in BCD digit (0-3).	0x0
11:8	R/W	RTC_CAL_D_U Date units. Represented in BCD digit (0-9).	0x0
7	R/W	RTC_CAL_M_T Month tens. Represented in BCD digit (0-1).	0x0
6:3	R/W	RTC_CAL_M_U Month units. Represented in BCD digit (0-9).	0x0
2:0	-	-	0x0

Bit	Mode	Symbol/Description	Reset
		Reserved	

Table 167: RTC_ALARM_ENABLE_REG (0x50000418)

Bit	Mode	Symbol/Description	Reset
5	R/W	RTC_ALARM_MNTH_EN Alarm on month enable. Enable to trigger alarm when data specified in Calendar Alarm Register (M_T and M_U) has been reached.	0x0
4	R/W	RTC_ALARM_DATE_EN Alarm on date enable. Enable to trigger alarm when data specified in Calendar Alarm Register (D_T and D_U) has been reached.	0x0
3	R/W	RTC_ALARM_HOUR_EN Alarm on hour enable. Enable to trigger alarm when data specified in Time Alarm Register (PM, HR_T and HR_U) has been reached.	0x0
2	R/W	RTC_ALARM_MIN_EN Alarm on minute enable. Enable to trigger alarm when data specified in Time Alarm Register (M_T and M_U) has been reached.	0x0
1	R/W	RTC_ALARM_SEC_EN Alarm on second enable. Enable to trigger alarm when data specified in Time Alarm Register (S_T and S_U) has been reached.	0x0
0	R/W	RTC_ALARM_HOS_EN Alarm on hundredths of a second enable. Enable to trigger alarm when data specified in Time Alarm Register (H_T and H_U) has been reached.	0x0

Table 168: RTC_EVENT_FLAGS_REG (0x5000041C)

Bit	Mode	Symbol/Description	Reset
6	R	RTC_EVENT_ALRM Alarm event flag. Indicate that alarm event occurred since the last reset.	0x0
5	R	RTC_EVENT_MNTH Month rolls over event flag. Indicate that month rolls over event occurred since the last reset.	0x0
4	R	RTC_EVENT_DATE Date rolls over event flag. Indicate that date rolls over event occurred since the last reset.	0x0
3	R	RTC_EVENT_HOUR Hour rolls over the event flag. Indicate that hour rolls over event occurred since the last reset.	0x0
2	R	RTC_EVENT_MIN Minute rolls over event flag. Indicate that minute rolls over event occurred since the last reset.	0x0
1	R	RTC_EVENT_SEC Second rolls over event flag. Indicate that second rolls over event occurred since the last reset.	0x0
0	R	RTC_EVENT_HOS	0x0

Bit	Mode	Symbol/Description	Reset
		Hundredths of a second event flag. Indicate that hundredths of a second rolls over event occurred since the last reset.	

Table 169: RTC_INTERRUPT_ENABLE_REG (0x50000420)

Bit	Mode	Symbol/Description	Reset
6	W	RTC_ALRM_INT_EN Interrupt on alarm enable. Enable to issue the interrupt when alarm event occurred.	0x0
5	W	RTC_MNTH_INT_EN Interrupt on month enable. Enable to issue the interrupt when month event occurred.	0x0
4	W	RTC_DATE_INT_EN Interrupt on date enable. Enable to issue the interrupt when date event occurred.	0x0
3	W	RTC_HOUR_INT_EN Interrupt on hour enable. Enable to issue the interrupt when hour event occurred.	0x0
2	W	RTC_MIN_INT_EN Interrupt on minute enable. Enable to issue the interrupt when minute event occurred.	0x0
1	W	RTC_SEC_INT_EN Interrupt on second enable. Enable to issue the interrupt when second event occurred.	0x0
0	W	RTC_HOS_INT_EN Interrupt on hundredths of a second enable. Enable to issue the interrupt when hundredths of a second event occurred.	0x0

Table 170: RTC_INTERRUPT_DISABLE_REG (0x50000424)

Bit	Mode	Symbol/Description	Reset
6	W	RTC_ALRM_INT_DIS Interrupt on alarm disable. Disable to issue the interrupt when alarm event occurred.	0x0
5	W	RTC_MNTH_INT_DIS Interrupt on month disable. Disable to issue the interrupt when month event occurred.	0x0
4	W	RTC_DATE_INT_DIS Interrupt on date disable. Disable to issue the interrupt when date event occurred.	0x0
3	W	RTC_HOUR_INT_DIS Interrupt on hour disable. Disable to issue the interrupt when hour event occurred.	0x0
2	W	RTC_MIN_INT_DIS	0x0

Bit	Mode	Symbol/Description	Reset
		Interrupt on minute disable. Disable to issue the interrupt when minute event occurred.	
1	W	RTC_SEC_INT_DIS Interrupt on second disable. Disable to issue the interrupt when second event occurred.	0x0
0	W	RTC_HOS_INT_DIS Interrupt on hundredths of a second disable. Disable to issue the interrupt when hundredths of a second event occurred.	0x0

Table 171: RTC_INTERRUPT_MASK_REG (0x50000428)

Bit	Mode	Symbol/Description	Reset
6	R	RTC_ALRM_INT_MSK Mask alarm interrupt. It can be cleared (set) by setting corresponding bit (ALRM) in Interrupt Enable Register (Interrupt Disable Register).	0x1
5	R	RTC_MNTH_INT_MSK Mask month interrupt. It can be cleared (set) by setting corresponding bit (MNTH) in Interrupt Enable Register (Interrupt Disable Register).	0x1
4	R	RTC_DATE_INT_MSK Mask date interrupt. It can be cleared (set) by setting corresponding bit (DATE) in Interrupt Enable Register (Interrupt Disable Register).	0x1
3	R	RTC_HOUR_INT_MSK Mask hour interrupt. It can be cleared (set) by setting corresponding bit (HOUR) in Interrupt Enable Register (Interrupt Disable Register).	0x1
2	R	RTC_MIN_INT_MSK Mask minute interrupt. It can be cleared (set) by setting corresponding bit (MIN) in Interrupt Enable Register (Interrupt Disable Register).	0x1
1	R	RTC_SEC_INT_MSK Mask second interrupt. It can be cleared (set) by setting corresponding bit (SEC) in Interrupt Enable Register (Interrupt Disable Register).	0x1
0	R	RTC_HOS_INT_MSK Mask hundredths of a second interrupt. It can be cleared (set) by setting corresponding bit (HOS) in Interrupt Enable Register (Interrupt Disable Register).	0x1

Table 172: RTC_STATUS_REG (0x5000042C)

Bit	Mode	Symbol/Description	Reset
3	R	RTC_VALID_CAL_ALM Valid Calendar Alarm. If cleared, then indicates that invalid entry occurred when writing to Calendar Alarm Register.	0x1
2	R	RTC_VALID_TIME_ALM Valid Time Alarm. If cleared, then indicates that invalid entry occurred when writing to Time Alarm Register.	0x1
1	R	RTC_VALID_CAL	0x1

Bit	Mode	Symbol/Description	Reset
		Valid Calendar. If cleared, then indicates that invalid entry occurred when writing to Calendar Register.	
0	R	RTC_VALID_TIME Valid Time. If cleared, then indicates that invalid entry occurred when writing to Time Register.	0x1

Table 173: RTC_KEEP_RTC_REG (0x50000430)

Bit	Mode	Symbol/Description	Reset
0	R/W	RTC_KEEP Keep RTC. When high, the time and calendar registers and any other registers which directly affect or are affected by the time and calendar registers are NOT reset when software reset is applied. When low, the software reset will reset every register except the keep RTC and control registers.	0x1

Table 174: RTC_EVENT_CTRL_REG (0x50000480)

Bit	Mode	Symbol/Description	Reset
1	R/W	RTC_PDC_EVENT_EN 0 = Event to PDC is disabled. No clear any pending event 1 = Even to PDC is enabled	0x0
0	R/W	- Reserved	0x0

Table 175: RTC_PDC_EVENT_PERIOD_REG (0x50000488)

Bit	Mode	Symbol/Description	Reset
12:0	R/W	RTC_PDC_EVENT_PERIOD RTC will send an event to PDC (if RTC_PDC_EVENT_EN = 1) every (RTC_PDC_EVENT_PERIOD + 1)*10 ms	0x0

Table 176: RTC_PDC_EVENT_CLEAR_REG (0x5000048C)

Bit	Mode	Symbol/Description	Reset
0	R	PDC_EVENT_CLEAR On read, PDC event is cleared	0x0

Table 177: RTC_PDC_EVENT_CNT_REG (0x50000494)

Bit	Mode	Symbol/Description	Reset
12:0	R	RTC_PDC_EVENT_CNT It gives the current value of the PDC event counter (0 to RTC_PDC_EVENT_PERIOD)	0x0

35.6 Quadrature decoder registers

Table 178: Register map QDEC

Address	Register	Description
0x50000500	QDEC_CTRL_REG	Quad Decoder control register
0x50000504	QDEC_XCNT_REG	Counter value of the X Axis
0x50000508	QDEC_YCNT_REG	Counter value of the Y Axis
0x5000050C	QDEC_CLOCKDIV_REG	Clock divider register
0x50000510	QDEC_CTRL2_REG	Quad Decoder port selection register
0x50000514	QDEC_ZCNT_REG	Counter value of the Z Axis
0x50000518	QDEC_EVENT_CNT_REG	Event counter register

Table 179: QDEC_CTRL_REG (0x50000500)

Bit	Mode	Symbol/Description	Reset
10:3	R/W	QDEC_IRQ_THRES Defines the number of events on either counter (X or Y or Z) that need to be reached before an interrupt is generated. Events are equal to QDEC_IRQ_THRES+1.	0x2
2	R/W	QDEC_IRQ_STATUS 1 = Interrupt has occurred 0 = No interrupt pending Writing 1 clears the pending interrupt	0x0
1	R0/WC	QDEC_EVENT_CNT_CLR Writing 1 QDEC_EVENT_CNT_REG is cleared	0x0
0	R/W	QDEC_IRQ_ENABLE 0 = Interrupt is masked 1 = Interrupt is enabled	0x0

Table 180: QDEC_XCNT_REG (0x50000504)

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

Table 181: QDEC_YCNT_REG (0x50000508)

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

Table 182: QDEC_CLOCKDIV_REG (0x5000050C)

Bit	Mode	Symbol/Description	Reset
10	R/W	QDEC_PRESCALER_EN 0 = no prescaler enabled 1 = in sleep and active mode, quadrature clock is divided by 2	0x0
9:0	R/W	QDEC_CLOCKDIV Contains the number of the input clock cycles minus one, that are required to generate one logic clock cycle. Clock divider is bypassed when system runs at LP_CLK	0x3E7

Table 183: QDEC_CTRL2_REG (0x50000510)

Bit	Mode	Symbol/Description	Reset
14	R/W	QDEC_CHZ_EVENT_MODE 0 = Normal quadrature counting 1 = Counts rising and falling edge of both ports (if both ports change at the same time, counter increases by 1)	0x1
13	R/W	QDEC_CHY_EVENT_MODE 0 = Normal quadrature counting 1 = Counts rising and falling edge of both ports (if both ports change at the same time, counter increases by 1)	0x1
12	R/W	QDEC_CHX_EVENT_MODE 0 = Normal quadrature counting 1 = Counts rising and falling edge of both ports (if both ports change at the same time, counter increases by 1)	0x1
11:8	R/W	QDEC_CHZ_PORT_SEL Defines which GPIOs are mapped on Channel Z 0: none 1: P0_00 -> CHZ_A, P0_01 -> CHZ_B 2: P0_02 -> CHZ_A, P0_03 -> CHZ_B 3: P0_04 -> CHZ_A, P0_05 -> CHZ_B 4: P0_08 -> CHZ_A, P0_09 -> CHZ_B 5: P0_10 -> CHZ_A, P0_11 -> CHZ_B 6: P1_00 -> CHZ_A, P1_02 -> CHZ_B 7: P1_03 -> CHZ_A, P1_05 -> CHZ_B 8: P1_06 -> CHZ_A, P1_07 -> CHZ_B 9: P1_08 -> CHZ_A, P1_09 -> CHZ_B 10: P1_10 -> CHZ_A, P1_11 -> CHZ_B 11: P1_12 -> CHZ_A, P1_01 -> CHZ_B 12: P1_04 -> CHZ_A, P1_15 -> CHZ_B 13-31: none	3
7:4	R/W	QDEC_CHY_PORT_SEL Defines which GPIOs are mapped on Channel Y 0: none 1: P0_00 -> CHY_A, P0_01 -> CHY_B 2: P0_02 -> CHY_A, P0_03 -> CHY_B 3: P0_04 -> CHY_A, P0_05 -> CHY_B	2

Bit	Mode	Symbol/Description	Reset
		4: P0_08 -> CHY_A, P0_09 -> CHY_B 5: P0_10 -> CHY_A, P0_11 -> CHY_B 6: P1_00 -> CHY_A, P1_02 -> CHY_B 7: P1_03 -> CHY_A, P1_05 -> CHY_B 8: P1_06 -> CHY_A, P1_07 -> CHY_B 9: P1_08 -> CHY_A, P1_09 -> CHY_B 10: P1_10 -> CHY_A, P1_11 -> CHY_B 11: P1_12 -> CHY_A, P1_01 -> CHY_B 12: P1_04 -> CHY_A, P1_15 -> CHY_B 13-31: none	
3:0	R/W	QDEC_CHX_PORT_SEL Defines which GPIOs are mapped on Channel X 0: none 1: P0_00 -> CHX_A, P0_01 -> CHX_B 2: P0_02 -> CHX_A, P0_03 -> CHX_B 3: P0_04 -> CHX_A, P0_05 -> CHX_B 4: P0_08 -> CHX_A, P0_09 -> CHX_B 5: P0_10 -> CHX_A, P0_11 -> CHX_B 6: P1_00 -> CHX_A, P1_02 -> CHX_B 7: P1_03 -> CHX_A, P1_05 -> CHX_B 8: P1_06 -> CHX_A, P1_07 -> CHX_B 9: P1_08 -> CHX_A, P1_09 -> CHX_B 10: P1_10 -> CHX_A, P1_11 -> CHX_B 11: P1_12 -> CHX_A, P1_01 -> CHX_B 12: P1_04 -> CHX_A, P1_15 -> CHX_B 13-31: none	1

Table 184: QDEC_ZCNT_REG (0x50000514)

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

Table 185: QDEC_EVENT_CNT_REG (0x50000518)

Bit	Mode	Symbol/Description	Reset
7:0	R	QDEC_EVENT_CNT Gives the number of events at all channels.	0x0

35.7 SPI controller registers

Table 186: Register map SPI

Address	Register	Description
0x50020200	SPI_CTRL_REG	SPI control register
0x50020204	SPI_CONFIG_REG	SPI control register
0x50020208	SPI_CLOCK_REG	SPI clock register

Address	Register	Description
0x5002020C	SPI_FIFO_CONFIG_REG	SPI FIFO configuration register
0x50020210	SPI_IRQ_MASK_REG	SPI interrupt mask register
0x50020214	SPI_STATUS_REG	SPI status register
0x50020218	SPI_FIFO_STATUS_REG	SPI RX/TX FIFO status register
0x5002021C	SPI_FIFO_READ_REG	SPI RX FIFO read register
0x50020220	SPI_FIFO_WRITE_REG	SPI TX FIFO write register
0x50020224	SPI_CS_CONFIG_REG	SPI cs configuration register
0x5002022C	SPI_TXBUFFER_FORCE_REG	SPI TX buffer force low value

Table 187: SPI_CTRL_REG (0x50020200)

Bit	Mode	Symbol/Description	Reset
7	R/W	SPI_SWAP_BYTES 0 = Normal operation 1 = LSB and MSB are swapped in the APB interface In case of 8-bit SPI interface, DMA/SPI can be configured in 16-bit mode to off load the bus. Enabling SPI_SWAP_BYTES bytes will read/write correctly	0x0
6	R/W	SPI_CAPTURE_AT_NEXT_EDGE 0 = SPI captures data at correct clock edge 1 = SPI captures data at the next clock edge. (only for Master mode and high clock)	0x0
5	R/W	SPI_FIFO_RESET 0 = FIFO normal operation 1 = FIFO in reset state	0x0
4	R/W	SPI_DMA_RX_EN Applicable only when SPI_RX_EN = 1 0 = No DMA request for RX 1 = DMA request when SPI_STATUS_RX_FULL = 1	0x0
3	R/W	SPI_DMA_TX_EN Applicable only when SPI_TX_EN = 1 0 = No DMA request for TX 1 = DMA request when SPI_STATUS_TX_EMPTY = 1	0x0
2	R/W	SPI_RX_EN 0 = RX path is disabled 1 = RX path is enabled Note: if SPI mode = 1 or SPI mode = 3 read-only is not supported	0x0
1	R/W	SPI_TX_EN 0 = TX path is disabled 1 = TX path is enabled	0x0

Bit	Mode	Symbol/Description	Reset
0	R/W	SPI_EN 0 = SPI module is disable 1 = SPI module is enable	0x0

Table 188: SPI_CONFIG_REG (0x50020204)

Bit	Mode	Symbol/Description	Reset
7	R/W	SPI_SLAVE_EN 0 = SPI module master mode 1 = SPI module slave mode	0x0
6:2	R/W	SPI_WORD_LENGTH Define the SPI word length = 1+ SPI_WORD_LENGTH (range 4 to 32)	0x0
1:0	R/W	SPI_MODE Define the SPI mode (CPOL, CPHA) 0 = new data on falling, capture on rising, Clk low in idle state 1 = new data on rising, capture on falling, Clk low in idle state 2 = new data on rising, capture on falling, Clk high in idle state 3 = new data on falling, capture on rising Clk high in idle state	0x0

Table 189: SPI_CLOCK_REG (0x50020208)

Bit	Mode	Symbol/Description	Reset
6:0	R/W	SPI_CLK_DIV Applicable only in master mode Defines the SPI clock frequency in master only mode SPI_CLK = module_clk/2*(SPI_CLK_DIV+1) when SPI_CLK_DIV not 0x7F if SPI_CLK_DIV = 0x7F then SPI_CLK = module_clk	0x0

Table 190: SPI_FIFO_CONFIG_REG (0x5002020C)

Bit	Mode	Symbol/Description	Reset
7:4	R/W	SPI_RX_TL Receive FIFO threshold level in bytes. Control the level of bytes in FIFO that triggers the RX_FULL interrupt. IRQ has occurred when FIFO level is more or equal to SPI_RX_TL+1. FIFO level is from 0 to 4	0x0
3:0	R/W	SPI_TX_TL Transmit FIFO threshold level in bytes. Control the level of bytes in FIFO that triggers the TX_EMPTY interrupt. IRQ has occurred when FIFO level is less or equal to SPI_TX_TL. FIFO level is from 0 to 4	0x0

Table 191: SPI_IRQ_MASK_REG (0x50020210)

Bit	Mode	Symbol/Description	Reset
1	R/W	SPI_IRQ_MASK_RX_FULL	0x0

Bit	Mode	Symbol/Description	Reset
		0 = FIFO RX full IRQ is masked 1 = FIFO RX full IRQ is enabled	
0	R/W	SPI_IRQ_MASK_TX_EMPTY 0 = FIFO TX empty IRQ is masked 1 = FIFO TX empty IRQ is enabled	0x0

Table 192: SPI_STATUS_REG (0x50020214)

Bit	Mode	Symbol/Description	Reset
1	R	SPI_STATUS_RX_FULL Auto clear 0 = RX FIFO level is less than SPI_RX_TL+1 1 = RX FIFO level is more or equal to SPI_RX_TL+1	0x0
0	R	SPI_STATUS_TX_EMPTY Auto clear 0 = TX FIFO level is larger than SPI_TX_TL 1 = TX FIFO level is less or equal to SPI_TX_TL	0x1

Table 193: SPI_FIFO_STATUS_REG (0x50020218)

Bit	Mode	Symbol/Description	Reset
15	R	SPI_TRANSACTION_ACTIVE In master mode 0 = SPI transaction is inactive 1 = SPI transaction is active	0x0
14	R	SPI_RX_FIFO_OVFL When 1, receive data is not written to FIFO because FIFO was full and interrupt is generated. It clears with SPI_CTRL_REG.SPI_FIFO_RESET	0x0
13	R	SPI_STATUS_TX_FULL 0 = TX FIFO is not full 1 = TX FIFO is full	0x0
12	R	SPI_STATUS_RX_EMPTY 0 = RX FIFO is not empty 1 = RX FIFO is empty	0x1
11:6	R	SPI_TX_FIFO_LEVEL Gives the number of bytes in TX FIFO	0x0
5:0	R	SPI_RX_FIFO_LEVEL Gives the number of bytes in RX FIFO	0x0

Table 194: SPI_FIFO_READ_REG (0x5002021C)

Bit	Mode	Symbol/Description	Reset
31:0	R	SPI_FIFO_READ	0x0

Bit	Mode	Symbol/Description	Reset
		Read from RX FIFO. Read access is permitted only if SPI_RX_FIFO_EMPTY = 0.	

Table 195: SPI_FIFO_WRITE_REG (0x50020220)

Bit	Mode	Symbol/Description	Reset
31:0	R0/W	SPI_FIFO_WRITE Write to TX FIFO. Write access is permit only if SPI_TX_FIFO_FULL is 0	0x0

Table 196: SPI_CS_CONFIG_REG (0x50020224)

Bit	Mode	Symbol/Description	Reset
2:0	R/W	SPI_CS_SELECT Control the cs output in master mode 0 = None slave device selected 1 = Selected slave device connected to GPIO with FUNC_MODE = SPI_EN 2 = Selected slave device connected to GPIO with FUNC_MODE = SPI_EN2 4 = Selected slave device connected to GPIO with FUNC_MODE = GPIO	0x0

Table 197: SPI_TXBUFFER_FORCE_REG (0x5002022C)

Bit	Mode	Symbol/Description	Reset
31:0	W	SPI_TXBUFFER_FORCE Write directly to the TX buffer. It must be used only in slave mode	0x0

35.8 I2C controller registers

Table 198: Register map I2C

Address	Register	Description
0x50020300	I2C_CON_REG	I2C Control Register
0x50020304	I2C_TAR_REG	I2C Target Address Register
0x50020308	I2C_SAR_REG	I2C Slave Address Register
0x5002030C	I2C_HS_MADDR_REG	I2C High Speed Master Mode Code Address Register
0x50020310	I2C_DATA_CMD_REG	I2C RX/TX Data Buffer and Command Register
0x50020314	I2C_SS_SCL_HCNT_REG	Standard Speed I2C Clock SCL High Count Register
0x50020318	I2C_SS_SCL_LCNT_REG	Standard Speed I2C Clock SCL Low Count Register
0x5002031C	I2C_FS_SCL_HCNT_REG	Fast Speed I2C Clock SCL High Count Register
0x50020320	I2C_FS_SCL_LCNT_REG	Fast Speed I2C Clock SCL Low Count Register

Address	Register	Description
0x50020324	I2C_HS_SCL_HCNT_REG	High Speed I2C Clock SCL High Count Register
0x50020328	I2C_HS_SCL_LCNT_REG	High Speed I2C Clock SCL Low Count Register
0x5002032C	I2C_INTR_STAT_REG	I2C Interrupt Status Register
0x50020330	I2C_INTR_MASK_REG	I2C Interrupt Mask Register
0x50020334	I2C_RAW_INTR_STAT_REG	I2C Raw Interrupt Status Register
0x50020338	I2C_RX_TL_REG	I2C Receive FIFO Threshold Register
0x5002033C	I2C_TX_TL_REG	I2C Transmit FIFO Threshold Register
0x50020340	I2C_CLR_INTR_REG	Clear Combined and Individual Interrupt Register
0x50020344	I2C_CLR_RX_UNDER_REG	Clear RX_UNDER Interrupt Register
0x50020348	I2C_CLR_RX_OVER_REG	Clear RX_OVER Interrupt Register
0x5002034C	I2C_CLR_TX_OVER_REG	Clear TX_OVER Interrupt Register
0x50020350	I2C_CLR_RD_REQ_REG	Clear RD_REQ Interrupt Register
0x50020354	I2C_CLR_TX_ABRT_REG	Clear TX_ABRT Interrupt Register
0x50020358	I2C_CLR_RX_DONE_REG	Clear RX_DONE Interrupt Register
0x5002035C	I2C_CLR_ACTIVITY_REG	Clear ACTIVITY Interrupt Register
0x50020360	I2C_CLR_STOP_DET_REG	Clear STOP_DET Interrupt Register
0x50020364	I2C_CLR_START_DET_REG	Clear START_DET Interrupt Register
0x50020368	I2C_CLR_GEN_CALL_REG	Clear GEN_CALL Interrupt Register
0x5002036C	I2C_ENABLE_REG	I2C Enable Register
0x50020370	I2C_STATUS_REG	I2C Status Register
0x50020374	I2C_TXFLR_REG	I2C Transmit FIFO Level Register
0x50020378	I2C_RXFLR_REG	I2C Receive FIFO Level Register
0x5002037C	I2C_SDA_HOLD_REG	I2C SDA Hold Time Length Register
0x50020380	I2C_TX_ABRT_SOURCE_REG	I2C Transmit Abort Source Register
0x50020388	I2C_DMA_CR_REG	DMA Control Register
0x5002038C	I2C_DMA_TDLR_REG	DMA Transmit Data Level Register
0x50020390	I2C_DMA_RDLR_REG	I2C Receive Data Level Register
0x50020394	I2C_SDA_SETUP_REG	I2C SDA Setup Register

Address	Register	Description
0x50020398	I2C_ACK_GENERAL_CALL_REG	I2C ACK General Call Register
0x5002039C	I2C_ENABLE_STATUS_REG	I2C Enable Status Register
0x500203A0	I2C_IC_FS_SPKLEN_REG	I2C SS and FS spike suppression limit Size
0x500203A4	I2C_IC_HS_SPKLEN_REG	I2C HS spike suppression limit Size

Table 199: I2C_CON_REG (0x50020300)

Bit	Mode	Symbol/Description	Reset
31:11	-	- Reserved	0x0
10	R	I2C_STOP_DET_IF_MASTER_ACTIVE In Master mode: 1 = Issues the STOP_DET interrupt only when master is active. 0 = Issues the STOP_DET irrespective of whether master is active or not.	0x0
9	R/W	I2C_RX_FIFO_FULL_HLD_CTRL This bit controls whether DW_apb_i2c should hold the bus when the RX FIFO is physically full to its RX_BUFFER_DEPTH 1 = Hold bus when RX_FIFO is full 0 = Overflow when RX_FIFO is full	0x0
8	R/W	I2C_TX_EMPTY_CTRL This bit controls the generation of the TX_EMPTY interrupt, as described in the IC_RAW_INTR_STAT register. 1 = Controlled generation of TX_EMPTY interrupt 0 = Default behavior of TX_EMPTY interrupt	0x0
7	R/W	I2C_STOP_DET_IFADDRESSED 1 = Slave issues STOP_DET interrupt only if addressed 0 = Slave issues STOP_DET interrupt always During a general call address, this slave does not issue the STOP_DET interrupt if STOP_DET_IF_ADDRESSED = 1'b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR).	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

Bit	Mode	Symbol/Description	Reset
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
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) 3 = High speed mode	0x3
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 200: I2C_TAR_REG (0x50020304)

Bit	Mode	Symbol/Description	Reset
31:12	-	- Reserved	0x0
11	R/W	SPECIAL On read 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 On write 1 = Enables programming of GENERAL_CALL or START_BYTE transmission 0 = Disables programming of GENERAL_CALL or START_BYTE transmission Writes to this register succeed only when IC_ENABLE[0] is set to 0.	0x0
10	R/W	GC_OR_START On read 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

Bit	Mode	Symbol/Description	Reset
		On write 1 = START byte transmission 0 = GENERAL_CALL byte transmission Writes to this register succeed only when IC_ENABLE[0] is set to 0.	
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 Writes to this register succeed only when IC_ENABLE[0] is set to 0.	0x55

Table 201: I2C_SAR_REG (0x50020308)

Bit	Mode	Symbol/Description	Reset
31:10	-	- Reserved	0x0
9:0	R/W	IC_SAR The IC_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect. Writes to this register succeed only when IC_ENABLE[0] is set to 0.	0x55

Table 202: I2C_HS_MADDR_REG (0x5002030C)

Bit	Mode	Symbol/Description	Reset
2:0	R/W	I2C_IC_HS_MAR This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.	0x1

Table 203: I2C_DATA_CMD_REG (0x50020310)

Bit	Mode	Symbol/Description	Reset
30:11	-	- Reserved	0x0
10	W	I2C_RESTART This bit controls whether a RESTART is issued before the byte is sent or received. 1 = If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether or	0x0

Bit	Mode	Symbol/Description	Reset
		not the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. 0 = If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead.	
9	W	I2C_STOP This bit controls whether a STOP is issued after the byte is sent or received. 1 = STOP is issued after this byte, regardless of whether or not the TX FIFO is empty. If the TX FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. 0 = STOP is not issued after this byte, regardless of whether or not the TX FIFO is empty. If the TX FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the TX FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the TX FIFO.	0x0
8	W	I2C_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	I2C_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 204: I2C_SS_SCL_HCNT_REG (0x50020314)

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.	0x91

Bit	Mode	Symbol/Description	Reset
		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.	

Table 205: I2C_SS_SCL_LCNT_REG (0x50020318)

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.	0xAB

Table 206: I2C_FS_SCL_HCNT_REG (0x5002031C)

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.	0x1A

Table 207: I2C_FS_SCL_LCNT_REG (0x50020320)

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.	0x32

Table 208: I2C_HS_SCL_HCNT_REG (0x50020324)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	IC_HS_SCL_HCNT	0x6

Bit	Mode	Symbol/Description	Reset
		This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period count for high speed.refer to "IC_CLK Frequency Configuration". The SCL High time depends on the loading of the bus. For 100 pF loading, the SCL High time is 60 ns; for 400 pF loading, the SCL High time is 120 ns. This register goes away and becomes read-only returning 0 s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] 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. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed.	

Table 209: I2C_HS_SCL_LCNT_REG (0x50020328)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	IC_HS_SCL_LCNT	0x10
		This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for high speed. For more information, refer to "IC_CLK Frequency Configuration". The SCL low time depends on the loading of the bus. For 100 pF loading, the SCL low time is 160 ns; for 400 pF loading, the SCL low time is 320 ns. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] 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 DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8.	

Table 210: I2C_INTR_STAT_REG (0x5002032C)

Bit	Mode	Symbol/Description	Reset
31:15	-	-	0x0
		Reserved	
14	R	R_SCL_STUCK_AT_LOW	0x0
		1 = R_SCL_STUCK_AT_LOW interrupt is active 0 = R_SCL_STUCK_AT_LOW interrupt is inactive	
13	R	R_MASTER_ON_HOLD	0x0
		Indicates whether master is holding the bus and TX FIFO is empty. Enabled only when I2C_DYNAMIC_TAR_UPDATE = 1 and IC_EMPTYFIFO_HOLD_MASTER_EN = 1.	
12	R	R_RESTART_DET	0x0

Bit	Mode	Symbol/Description	Reset
		Indicates whether a RESTART condition has occurred on the I2C interface when DW_apb_i2c is operating in Slave mode and the slave is being addressed. Enabled only when IC_SLV_RESTART_DET_EN = 1. Note: However, in high-speed mode or during a START BYTE transfer, the RESTART comes before the address field as per the I2C protocol. In this case, the slave is not the addressed slave when the RESTART is issued, therefore DW_apb_i2c does not generate the RESTART_DET interrupt.	
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
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	0x0

Bit	Mode	Symbol/Description	Reset
		register. This bit is set to 0 just after the processor reads the I2C_CLR_RD_REQ register	
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
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 211: I2C_INTR_MASK_REG (0x50020330)

Bit	Mode	Symbol/Description	Reset
31:15	-	- Reserved	0x0
14	R	M_SCL_STUCK_AT_LOW M_SCL_STUCK_AT_LOW Register field Reserved bits	0x0
13	R/W	M_MASTER_ON_HOLD These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register.	0x0
12	R/W	M_RESTART_DET These bits mask their corresponding interrupt status bits in the I2C_INTR_STAT register.	0x0

Bit	Mode	Symbol/Description	Reset
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 212: I2C_RAW_INTR_STAT_REG (0x50020334)

Bit	Mode	Symbol/Description	Reset
31:15	-	- Reserved	0x0
14	R	SCL_STUCK_AT_LOW CL_STUCK_AT_LOW Register field Reserved bits	0x0
13	R	MASTER_ON_HOLD	0x0

Bit	Mode	Symbol/Description	Reset
		Indicates whether master is holding the bus and TX FIFO is empty. Enabled only when I2C_DYNAMIC_TAR_UPDATE = 1 and IC_EMPTYFIFO_HOLD_MASTER_EN = 1.	
12	R	RESTART_DET Indicates whether a RESTART condition has occurred on the I2C interface when DW_apb_i2c is operating in Slave mode and the slave is being addressed. Enabled only when IC_SLV_RESTART_DET_EN=1. Note: However, in high-speed mode or during a START BYTE transfer, the RESTART comes before the address field as per the I2C protocol. In this case, the slave is not the addressed slave when the RESTART is issued, therefore DW_apb_i2c does not generate the RESTART_DET interrupt.	0x0
11	R	GEN_CALL Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling controller or when the CPU reads bit 0 of the I2C_CLR_GEN_CALL register. I2C Ctrl stores the received data in the RX buffer.	0x0
10	R	START_DET Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.	0x0
9	R	STOP_DET Indicates whether a STOP condition has occurred on the I2C interface regardless of whether controller is operating in slave or master mode.	0x0
8	R	ACTIVITY This bit captures I2C Ctrl activity and stays set until it is cleared. There are four ways to clear it: - 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 place. 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	0x0

Bit	Mode	Symbol/Description	Reset
		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	
4	R	TX_EMPTY This bit is set to 1 when the transmit buffer is at or below the threshold value set in the I2C_TX_TL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the IC_ENABLE bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with ic_en = 0, this bit is set to 0.	0x0
3	R	TX_OVER Set during transmit if the transmit buffer is filled to 32 and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared	0x0
2	R	RX_FULL Set when the receive buffer reaches or goes above the RX_TL threshold in the I2C_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (I2C_ENABLE[0] = 0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So, this bit is cleared once the I2C_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.	0x0
1	R	RX_OVER Set if the receive buffer is completely filled to 32 and an additional byte is received from an external I2C device. The controller acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (I2C_ENABLE[0] = 0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.	0x0
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 213: I2C_RX_TL_REG (0x50020338)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0x0
4:0	R/W	RX_TL Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in I2C_RAW_INTR_STAT register). The valid range is 0-31, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 214: I2C_TX_TL_REG (0x5002033C)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0x0
4:0	R/W	TX_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 215: I2C_CLR_INTR_REG (0x50020340)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_INTR Read this register to clear the combined interrupt, all individual interrupts, and the I2C_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the I2C_TX_ABRT_SOURCE register for an exception to clearing I2C_TX_ABRT_SOURCE	0x0

Table 216: I2C_CLR_RX_UNDER_REG (0x50020344)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_RX_UNDER Read this register to clear the RX_UNDER interrupt (bit 0) of the I2C_RAW_INTR_STAT register.	0x0

Table 217: I2C_CLR_RX_OVER_REG (0x50020348)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_RX_OVER Read this register to clear the RX_OVER interrupt (bit 1) of the I2C_RAW_INTR_STAT register.	0x0

Table 218: I2C_CLR_TX_OVER_REG (0x5002034C)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_TX_OVER Read this register to clear the TX_OVER interrupt (bit 3) of the I2C_RAW_INTR_STAT register.	0x0

Table 219: I2C_CLR_RD_REQ_REG (0x50020350)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_RD_REQ Read this register to clear the RD_REQ interrupt (bit 5) of the I2C_RAW_INTR_STAT register.	0x0

Table 220: I2C_CLR_TX_ABRT_REG (0x50020354)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_TX_ABRT Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the I2C_TX_ABRT_SOURCE register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the I2C_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.	0x0

Table 221: I2C_CLR_RX_DONE_REG (0x50020358)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_RX_DONE Read this register to clear the RX_DONE interrupt (bit 7) of the I2C_RAW_INTR_STAT register.	0x0

Table 222: I2C_CLR_ACTIVITY_REG (0x5002035C)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_ACTIVITY	0x0

Bit	Mode	Symbol/Description	Reset
		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	

Table 223: I2C_CLR_STOP_DET_REG (0x50020360)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_STOP_DET Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register.	0x0

Table 224: I2C_CLR_START_DET_REG (0x50020364)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_START_DET Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register.	0x0

Table 225: I2C_CLR_GEN_CALL_REG (0x50020368)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R	CLR_GEN_CALL Read this register to clear the GEN_CALL interrupt (bit 11) of I2C_RAW_INTR_STAT register.	0x0

Table 226: I2C_ENABLE_REG (0x5002036C)

Bit	Mode	Symbol/Description	Reset
31:3	-	- Reserved	0x0
2	R/W	I2C_TX_CMD_BLOCK In Master mode: 1 = Blocks the transmission of data on I2C bus even if TX FIFO has data to transmit. 0 = The transmission of data starts on I2C bus automatically, as soon as the first data is available in the TX FIFO.	0x0
1	R/W	I2C_ABORT	0x0

Bit	Mode	Symbol/Description	Reset
		The software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when ENABLE is already set; otherwise, the controller ignores any write to ABORT bit. The software cannot clear the ABORT bit once set. In response to an ABORT, the controller issues a STOP and flushes the TX FIFO after completing the current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation.	
0	R/W	I2C_EN 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 227: I2C_STATUS_REG (0x50020370)

Bit	Mode	Symbol/Description	Reset
31:11	-	- Reserved	0x0
10	R	LV_HOLD_RX_FIFO_FULL This bit indicates the BUS Hold in Slave mode due to RX FIFO is Full and an additional byte has been received 1 = Slave holds the bus due to RX FIFO is full 0 = Slave is not holding the bus or Bus hold is not due to RX FIFO is full	0x0
9	R	SLV_HOLD_TX_FIFO_EMPTY This bit indicates the BUS Hold in Slave mode for the Read request when the TX FIFO is empty. The Bus is in hold until the TX FIFO has data to Transmit for the read request. 1 = Slave holds the bus due to TX FIFO is empty 0 = Slave is not holding the bus or Bus hold is not due to TX FIFO is empty	0x0
8	R	MST_HOLD_RX_FIFO_FULL This bit indicates the BUS Hold in Master mode due to RX FIFO is Full and additional byte has been received 1 = Master holds the bus due to RX FIFO is full 0 = Master is not holding the bus or Bus hold is not due to RX FIFO is full	0x0
7	R	MST_HOLD_TX_FIFO_EMPTY the DW_apb_i2c master stalls the write transfer when TX FIFO is empty, and the last byte does not have the Stop bit set. This bit indicates the BUS hold when the master holds the bus because of the TX FIFO being empty, and the previous transferred command does not have the Stop bit set.	0x0

Bit	Mode	Symbol/Description	Reset
		1 = Master holds the bus due to TX FIFO is empty 0 = Master is not holding the bus or Bus hold is not due to TX FIFO is empty	
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 228: I2C_TXFLR_REG (0x50020374)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R	TXFLR	0x0

Bit	Mode	Symbol/Description	Reset
		Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Size is constrained by the TXFLR value	

Table 229: I2C_RXFLR_REG (0x50020378)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R	RXFLR Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. Size is constrained by the RXFLR value	0x0

Table 230: I2C_SDA_HOLD_REG (0x5002037C)

Bit	Mode	Symbol/Description	Reset
23:16	R/W	I2C_SDA_RX_HOLD Sets the required SDA hold time in units of ic_clk period when receiver.	0x0
15:0	R/W	I2C_SDA_TX_HOLD Sets the required SDA hold time in units of ic_clk period when transmitter.	0x1

Table 231: I2C_TX_ABRT_SOURCE_REG (0x50020380)

Bit	Mode	Symbol/Description	Reset
16	R	ABRT_USER_ABRT Master-Transmitter: This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1])	0x0
15	R	ABRT_SLVRD_INTX Slave-Transmitter: 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 1 = Slave trying to transmit to remote master in read mode 0 = Slave trying to transmit to remote master in read mode- scenario not present	0x0
14	R	ABRT_SLV_ARBLOST Slave-Transmitter: 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. 1 = Slave lost arbitration to remote master 0 = Slave lost arbitration to remote master - scenario not present	0x0
13	R	ABRT_SLVFLUSH_TXFIFO Slave-Transmitter: 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. 1 = Slave flushes existing data in TX-FIFO upon getting read command	0x0

Bit	Mode	Symbol/Description	Reset
		0 = Slave flushes existing data in TX-FIFO upon getting read command - scenario not present	
12	R	ARB_LOST Master-Transmitter or Slave-Transmitter: 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. 1 = Master or Slave-Transmitter lost arbitration 0 = Master or Slave-Transmitter lost arbitration - scenario not present	0x0
11	R	ABRT_MASTER_DIS Master-Transmitter or Master-Receiver: User tries to initiate a Master operation with the Master mode disabled. 1 = User initiating master operation when MASTER disabled 0 = User initiating master operation when MASTER disabled - scenario not present	0x0
10	R	ABRT_10B_RD_NORSTRT Master-Receiver: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode. 1 =Master trying to read in 10Bit addressing mode when RESTART disabled 0 =Master not trying to read in 10Bit addressing mode when RESTART disabled	0x0
9	R	ABRT_SBYTE_NORSTRT Master: 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 reasserted. 1: The restart is disabled (IC_RESTART_EN bit (I2C_CON[5]) = 0) and the user is trying to send a START Byte. 1 = User trying to send START byte when RESTART disabled 0 = User trying to send START byte when RESTART disabled- scenario not present	0x0
8	R	ABRT_HS_NORSTRT Master-Transmitter or Master-Receiver: 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 1 = User trying to switch Master to HS mode when RESTART disabled 0 = User trying to switch Master to HS mode when RESTART disabled - scenario not present	0x0
7	R	ABRT_SBYTE_ACKDET Master: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). 1 = ACK detected for START byte 0 = ACK detected for START byte - scenario not present	0x0
6	R	ABRT_HS_ACKDET Master: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).	0x0

Bit	Mode	Symbol/Description	Reset
		1 = HS Master code ACKed in HS Mode 0 = HS Master code ACKed in HS Mode - scenario not present	
5	R	ABRT_GCALL_READ Master-Transmitter: 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). 1 = GCALL is followed by read from bus 0 = GCALL is followed by read from bus - scenario not present	0x0
4	R	ABRT_GCALL_NOACK Master-Transmitter: The controller in master mode sent a General Call and no slave on the bus acknowledged the General Call. 1 = GCALL not ACKed by any slave 0 = GCALL not ACKed by any slave - scenario not present	0x0
3	R	ABRT_TXDATA_NOACK Master-Transmitter: 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). 1 = Transmitted data not ACKed by addressed slave 0 = Transmitted data non-ACKed by addressed slave - scenario not present	0x0
2	R	ABRT_10ADDR2_NOACK Master-Transmitter or Master-Receiver: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave. 1 = Byte 2 of 10-bit Address not ACKed by any slave 0 = This abort is not generated	0x0
1	R	ABRT_10ADDR1_NOACK Master-Transmitter or Master-Receiver: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. 1 = Byte 1 of 10-bit Address not ACKed by any slave 0 =This abort is not generated	0x0
0	R	ABRT_7B_ADDR_NOACK Master-Transmitter or Master-Receiver: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. 1 = This abort is generated because of NOACK for 7-bit address 0 = This abort is not generated	0x0

Table 232: I2C_DMA_CR_REG (0x50020388)

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	0x0

Bit	Mode	Symbol/Description	Reset
		Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled	

Table 233: I2C_DMA_TDLR_REG (0x5002038C)

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 234: I2C_DMA_RDLR_REG (0x50020390)

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 235: I2C_SDA_SETUP_REG (0x50020394)

Bit	Mode	Symbol/Description	Reset
15:8	-	- Reserved	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 236: I2C_ACK_GENERAL_CALL_REG (0x50020398)

Bit	Mode	Symbol/Description	Reset
15:1	-	- Reserved	0x0
0	R/W	ACK_GEN_CALL	0x0

Bit	Mode	Symbol/Description	Reset
		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. 1 = Generate ACK for a General Call 0 = Generate NACK for General Call	

Table 237: I2C_ENABLE_STATUS_REG (0x5002039C)

Bit	Mode	Symbol/Description	Reset
15:3	-	- Reserved	0x0
2	R	SLV_RX_DATA_LOST Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting of IC_ENABLE from 1 to 0. When read as 1, the controller is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. NOTE: If the remote I2C master terminates the transfer with a STOP condition before the controller has a chance to NACK a transfer, and IC_ENABLE has been set to 0, then this bit is also set to 1. When read as 0, the controller is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. NOTE: The CPU can safely read this bit when IC_EN (bit 0) is read as 0. 1 = Slave RX Data is lost 0 = Slave RX Data is not lost	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 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. 1 = Slave is disabled when it is active 0 = Slave is disabled when it is idle	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.	0x0

Bit	Mode	Symbol/Description	Reset
		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). 1 = I2C enabled 0 = I2C disabled	

Table 238: I2C_IC_FS_SPKLEN_REG (0x500203A0)

Bit	Mode	Symbol/Description	Reset
15:8	-	- Reserved	0x0
7:0	R/W	I2C_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

Table 239: I2C_IC_HS_SPKLEN_REG (0x500203A4)

Bit	Mode	Symbol/Description	Reset
7:0	R/W	I2C_HS_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[0] 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

35.9 Timers 1/2/3 registers

Table 240: Register map TIMER

Address	Register	Description
0x50010300	TIMER_CTRL_REG	Timer control register
0x50010304	TIMER_TIMER_VAL_REG	Timer counter value
0x50010308	TIMER_STATUS_REG	Timer status register
0x5001030C	TIMER_GPIO1_CONF_REG	Timer gpio1 selection
0x50010310	TIMER_GPIO2_CONF_REG	Timer gpio2 selection
0x50010314	TIMER_SETTINGS_REG	Timer reload value and Delay in shot mode

Address	Register	Description
0x50010318	TIMER_SHOTWIDTH_REG	Timer Shot duration in shot mode
0x50010320	TIMER_CAPTURE_GPIO1_REG	Timer value for event on GPIO1
0x50010324	TIMER_CAPTURE_GPIO2_REG	Timer value for event on GPIO2
0x50010328	TIMER_PRESCALER_VAL_REG	Timer prescaler counter value
0x5001032C	TIMER_PWM_CTRL_REG	Timer pwm frequency register
0x50010334	TIMER_GPIO3_CONF_REG	Timer gpio3 selection
0x50010338	TIMER_GPIO4_CONF_REG	Timer gpio4 selection
0x5001033C	TIMER_CAPTURE_GPIO3_REG	Timer value for event on GPIO1
0x50010340	TIMER_CAPTURE_GPIO4_REG	Timer value for event on GPIO1
0x50010344	TIMER_CLEAR_GPIO_EVENT_REG	Timer clear GPIO event register
0x50010348	TIMER_CLEAR_IRQ_REG	Timer clear interrupt

Table 241: TIMER_CTRL_REG (0x50010300)

Bit	Mode	Symbol/Description	Reset
31:15	-	- Reserved	0x0
14	R/W	TIM_CAP_GPIO4_IRQ_EN 0 = Event on GPIO4 does not create a CAPTIM interrupt 1 = Event on GPIO4 creates a CAPTIM interrupt	0x0
13	R/W	TIM_CAP_GPIO3_IRQ_EN 0 = Event on GPIO3 does not create a CAPTIM interrupt 1 = Event on GPIO3 creates a CAPTIM interrupt	0x0
12	R/W	TIM_CAP_GPIO2_IRQ_EN 0 = Event on GPIO2 does not create a CAPTIM interrupt 1 = Event on GPIO2 creates a CAPTIM interrupt	0x0
11	R/W	TIM_CAP_GPIO1_IRQ_EN 0 = Event on GPIO1 does not create a CAPTIM interrupt 1 = Event on GPIO1 creates a CAPTIM interrupt	0x0
10	R/W	TIM_IN4_EVENT_FALL_EN Event input 4 edge type 1 = Falling edge 0 = Rising edge	0x0
9	R/W	TIM_IN3_EVENT_FALL_EN	0x0

Bit	Mode	Symbol/Description	Reset
		Event input 3 edge type 1 = Falling edge 0 = Rising edge	
8	R/W	TIM_CLK_EN Timer clock enable 1 = Clock enabled 0 = Clock disabled	0x0
7	R/W	TIM_SYS_CLK_EN Select clock 1 = Timer uses the DIVN clock 0 = Timer uses the lp clock Note: When switching clock, the timer clock should be disabled (TIM_CLK_EN, bit 8)	0x0
6	R/W	TIM_FREE_RUN_MODE_EN Valid when timer counts up, if it is 1, timer does not zero when reaches to reload value. it becomes zero only when it reaches the max value.	0x0
5	R/W	TIM_IRQ_EN Interrupt mask 1 = Timer IRQ is unmasked 0 = Timer IRQ is masked	0x0
4	R/W	TIM_IN2_EVENT_FALL_EN Event input 2 edge type 1 = Falling edge 0 = Rising edge	0x0
3	R/W	TIM_IN1_EVENT_FALL_EN Event input 1 edge type 1 = Falling edge 0 = Rising edge	0x0
2	R/W	TIM_COUNT_DOWN_EN Timer count direction 1 = Down 0 = Up Note: Only change counter direction when timer is not enabled	0x0
1	R/W	TIM_ONESHOT_MODE_EN Timer mode 1 = One shot enabled 0 = Counter enabled	0x0
0	R/W	TIM_EN Timer enable 1 = On 0 = Off	0x0

Table 242: TIMER_TIMER_VAL_REG (0x50010304)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_TIMER_VALUE Gives the current timer value	0x0

Table 243: TIMER_STATUS_REG (0x50010308)

Bit	Mode	Symbol/Description	Reset
13	R	TIM_IN4_STATE Gives the logic level of the IN4	0x0
12	R	TIM_IN3_STATE Gives the logic level of the IN3	0x0
11	R	TIM_SWITCHED_TO_DIVN_CLK Indicates that timer clock has been switched to divn clock	0x0
10	R	TIM_PWM_BUSY Busy with synchronizing PWM_FREQ_REG and PWM_DC_REG. Do not write a new value to these registers when this bit is high	0x0
9	R	TIM_TIMER_BUSY Busy with synchronizing PRESCALER_REG, RELOAD_REG and SHOTWIDTH_REG. Do not write a new value to these registers when this bit is high	0x0
8	R	TIM_IRQ_STATUS IRQ status bit. When an irq has occurred, this bit is 1.	0x0
7	R	TIM_GPIO4_EVENT_PENDING When 1, GPIO4 event is pending.	0x0
6	R	TIM_GPIO3_EVENT_PENDING When 1, GPIO3 event is pending.	0x0
5	R	TIM_GPIO2_EVENT_PENDING When 1, GPIO2 event is pending.	0x0
4	R	TIM_GPIO1_EVENT_PENDING When 1, GPIO1 event is pending.	0x0
3:2	R	TIM_ONESHOT_PHASE OneShot phase 0 = Wait for event 1 = Delay phase 2 = Start Shot 3 = Shot phase	0x0
1	R	TIM_IN2_STATE Gives the logic level of the IN1	0x0
0	R	TIM_IN1_STATE Gives the logic level of the IN2	0x0

Table 244: TIMER_GPIO1_CONF_REG (0x5001030C)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO1_CONF Select one of the 32 GPIOs as IN1, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 245: TIMER_GPIO2_CONF_REG (0x50010310)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO2_CONF Select one of the 32 GPIOs as IN2, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 246: TIMER_SETTINGS_REG (0x50010314)

Bit	Mode	Symbol/Description	Reset
31:29	-	- Reserved	0x0
28:24	R/W	TIM_PRESCALER Defines the timer count frequency. CLOCK frequency/(TIM_PRESCALER + 1)	0x0
23:0	R/W	TIM_RELOAD Reload or max value in timer mode, Delay phase duration in oneshot mode. Actual delay is the register value plus synchronization time (3 clock cycles)	0x0

Table 247: TIMER_SHOTWIDTH_REG (0x50010318)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R/W	TIM_SHOTWIDTH Shot phase duration in oneshot mode	0x0

Table 248: TIMER_CAPTURE_GPIO1_REG (0x50010320)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
23:0	R	TIM_CAPTURE_GPIO1 Gives the Capture time for event on GPIO1	0x0

Table 249: TIMER_CAPTURE_GPIO2_REG (0x50010324)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO2 Gives the Capture time for event on GPIO2	0x0

Table 250: TIMER_PRESCALER_VAL_REG (0x50010328)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0x0
4:0	R	TIM_PRESCALER_VAL Gives the current prescaler counter value	0x0

Table 251: TIMER_PWM_CTRL_REG (0x5001032C)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	TIM_PWM_DC Defines the PWM duty cycle. TIM_PWM_DC/(TIM_PWM_FREQ + 1)	0x0
15:0	R/W	TIM_PWM_FREQ Defines the PWM frequency. Timer clock frequency/(TIM_PWM_FREQ + 1) Timer clock is clock after prescaler	0x0

Table 252: TIMER_GPIO3_CONF_REG (0x50010334)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO3_CONF Select one of the 32 GPIOs as IN3, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 253: TIMER_GPIO4_CONF_REG (0x50010338)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
5:0	R/W	TIM_GPIO4_CONF Select one of the 32 GPIOs as IN4, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 254: TIMER_CAPTURE_GPIO3_REG (0x5001033C)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO3 Gives the Capture time for event on GPIO3	0x0

Table 255: TIMER_CAPTURE_GPIO4_REG (0x50010340)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO4 Gives the Capture time for event on GPIO4	0x0

Table 256: TIMER_CLEAR_GPIO_EVENT_REG (0x50010344)

Bit	Mode	Symbol/Description	Reset
3	R0/W	TIM_CLEAR_GPIO4_EVENT 1 = Clear GPIO4 event. Return always 0	0x0
2	R0/W	TIM_CLEAR_GPIO3_EVENT 1 = Clear GPIO3 event. Return always 0	0x0
1	R0/W	TIM_CLEAR_GPIO2_EVENT 1 = Clear GPIO2 event. Return always 0	0x0
0	R0/W	TIM_CLEAR_GPIO1_EVENT 1 = Clear GPIO1 event. Return always 0	0x0

Table 257: TIMER_CLEAR_IRQ_REG (0x50010348)

Bit	Mode	Symbol/Description	Reset
0	R0/W	TIM_CLEAR_IRQ Write any value clear interrupt	0x0

Table 258: Register map TIMER2/3

Address	Register	Description
0x50010400	TIMER2_CTRL_REG	Timer control register
0x50010404	TIMER2_TIMER_VAL_REG	Timer counter value
0x50010408	TIMER2_STATUS_REG	Timer status register
0x5001040C	TIMER2_GPIO1_CONF_REG	Timer gpio1 selection
0x50010410	TIMER2_GPIO2_CONF_REG	Timer gpio2 selection
0x50010414	TIMER2_SETTINGS_REG	Timer reload value and Delay in shot mode
0x50010418	TIMER2_SHOTWIDTH_REG	Timer Shot duration in shot mode
0x50010420	TIMER2_CAPTURE_GPIO1_REG	Timer value for event on GPIO1
0x50010424	TIMER2_CAPTURE_GPIO2_REG	Timer value for event on GPIO2
0x50010428	TIMER2_PRESCALER_VAL_REG	Timer prescaler counter value
0x5001042C	TIMER2_PWM_CTRL_REG	Timer pwm frequency register
0x50010434	TIMER2_CLEAR_IRQ_REG	Timer clear interrupt
0x50010500	TIMER3_CTRL_REG	Timer control register
0x50010504	TIMER3_TIMER_VAL_REG	Timer counter value
0x50010508	TIMER3_STATUS_REG	Timer status register
0x5001050C	TIMER3_GPIO1_CONF_REG	Timer gpio1 selection
0x50010510	TIMER3_GPIO2_CONF_REG	Timer gpio2 selection
0x50010514	TIMER3_SETTINGS_REG	Timer reload value and Delay in shot mode
0x50010520	TIMER3_CAPTURE_GPIO1_REG	Timer value for event on GPIO1
0x50010524	TIMER3_CAPTURE_GPIO2_REG	Timer value for event on GPIO2
0x50010528	TIMER3_PRESCALER_VAL_REG	Timer prescaler counter value
0x5001052C	TIMER3_PWM_CTRL_REG	Timer pwm frequency register
0x50010534	TIMER3_CLEAR_IRQ_REG	Timer clear interrupt

Table 259: TIMER2_CTRL_REG (0x50010400)

Bit	Mode	Symbol/Description	Reset
31:9	-	- Reserved	0x0
8	R/W	TIM_CLK_EN Timer clock enable 1 = clock enabled 0 = clock disabled	0x0
7	R/W	TIM_SYS_CLK_EN Select clock 1 = Timer uses the DIVN clock 0 = Timer uses the lp clock NOTE: when switching clock, the timer clock should be disabled (TIM_CLK_EN, bit 8)	0x0
6	R/W	TIM_FREE_RUN_MODE_EN Valid when timer counts up, if it is 1, timer does not zero when reaches to reload value. it becomes zero only when it reaches the max value.	0x0
5	R/W	TIM_IRQ_EN Interrupt mask 1 = timer IRQ is unmasked 0 = timer IRQ is masked	0x0
4	R/W	TIM_IN2_EVENT_FALL_EN Event input 2 edge type 1 = falling edge 0 = rising edge	0x0
3	R/W	TIM_IN1_EVENT_FALL_EN Event input 1 edge type 1 = falling edge 0 = rising edge	0x0
2	R/W	TIM_COUNT_DOWN_EN Timer count direction 1 = down 0 = up NOTE: only change this bit when timer is disabled	0x0
1	R/W	TIM_ONESHOT_MODE_EN Timer mode 1 = One shot enabled 0 = Counter enabled	0x0
0	R/W	TIM_EN Timer enable 1 = On 0 = Off	0x0

Table 260: TIMER2_TIMER_VAL_REG (0x50010404)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_TIMER_VALUE Gives the current timer value	0x0

Table 261: TIMER2_STATUS_REG (0x50010408)

Bit	Mode	Symbol/Description	Reset
11	R	TIM_SWITCHED_TO_DIVN_CLK Indicates that timer clock has been switched to divn clock	0x0
10	R	TIM_PWM_BUSY Busy with synchronizing PWM_FREQ_REG and PWM_DC_REG. Do not write a new value to these registers when this bit is high.	0x0
9	R	TIM_TIMER_BUSY Busy with synchronizing PRESCALER_REG, RELOAD_REG and SHOTWIDTH_REG. Do not write a new value to these registers when this bit is high.	0x0
8	R	TIM_IRQ_STATUS IRQ status bit. When an irq has occurred, this bit is 1.	0x0
7:4	-	- Reserved	0x0
3:2	R	TIM_ONESHOT_PHASE OneShot phase 0 = Wait for event 1 = Delay phase 2 = Start Shot 3 = Shot phase	0x0
1	R	TIM_IN2_STATE Gives the logic level of the IN1	0x0
0	R	TIM_IN1_STATE Gives the logic level of the IN2	0x0

Table 262: TIMER2_GPIO1_CONF_REG (0x5001040C)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO1_CONF Select one of the 32 GPIOs as IN1, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 263: TIMER2_GPIO2_CONF_REG (0x50010410)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO2_CONF Select one of the 32 GPIOs as IN2, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 264: TIMER2_SETTINGS_REG (0x50010414)

Bit	Mode	Symbol/Description	Reset
31:29	-	- Reserved	0x0
28:24	R/W	TIM_PRESCALER Defines the timer count frequency. CLOCK frequency/(TIM_PRESCALER+1)	0x0
23:0	R/W	TIM_RELOAD Reload or max value in timer mode, Delay phase duration in oneshot mode. Actual delay is the register value plus synchronization time (3 clock cycles)	0x0

Table 265: TIMER2_SHOTWIDTH_REG (0x50010418)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R/W	TIM_SHOTWIDTH Shot phase duration in oneshot mode	0x0

Table 266: TIMER2_CAPTURE_GPIO1_REG (0x50010420)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO1 Gives the Capture time for event on GPIO1	0x0

Table 267: TIMER2_CAPTURE_GPIO2_REG (0x50010424)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO2 Gives the Capture time for event on GPIO2	0x0

Table 268: TIMER2_PRESCALER_VAL_REG (0x50010428)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0x0
4:0	R	TIM_PRESCALER_VAL Gives the current prescaler counter value	0x0

Table 269: TIMER2_PWM_CTRL_REG (0x5001042C)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	TIM_PWM_DC Defines the PWM duty cycle. TIM_PWM_DC/(TIM_PWM_FREQ + 1)	0x0
15:0	R/W	TIM_PWM_FREQ Defines the PWM frequency. Timer clock frequency/(TIM_PWM_FREQ + 1) Timer clock is clock after prescaler	0x0

Table 270: TIMER2_CLEAR_IRQ_REG (0x50010434)

Bit	Mode	Symbol/Description	Reset
0	R0/W	TIM_CLEAR_IRQ Write any value clear interrupt	0x0

Table 271: TIMER3_CTRL_REG (0x50010500)

Bit	Mode	Symbol/Description	Reset
31:9	-	- Reserved	0x0
8	R/W	TIM_CLK_EN Timer clock enable 1 = Clock enabled 0 = Clock disabled	0x0
7	R/W	TIM_SYS_CLK_EN Select clock 1 = Timer uses the DIVN clock 0 = Timer uses the lp clock	0x0
6	R/W	TIM_FREE_RUN_MODE_EN Valid when timer counts up, if it is 1, timer does not zero when reaches to reload value. It becomes zero only when it reaches the max value.	0x0
5	R/W	TIM_IRQ_EN Interrupt mask 1 = Timer IRQ is unmasked 0 = Timer IRQ is masked	0x0

Bit	Mode	Symbol/Description	Reset
4	R/W	TIM_IN2_EVENT_FALL_EN Event input 2 edge type 1 = Falling edge 0 = Rising edge	0x0
3	R/W	TIM_IN1_EVENT_FALL_EN Event input 1 edge type 1 = Falling edge 0 = Rising edge	0x0
2	R/W	TIM_COUNT_DOWN_EN Timer count direction 1 = Down 0 = Up	0x0
1	R/W	- Reserved	0x0
0	R/W	TIM_EN Timer enable 1 = On 0 = Off	0x0

Table 272: TIMER3_TIMER_VAL_REG (0x50010504)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_TIMER_VALUE Gives the current timer value	0x0

Table 273: TIMER3_STATUS_REG (0x50010508)

Bit	Mode	Symbol/Description	Reset
11	R	TIM_SWITCHED_TO_DIVN_CLK Indicates that timer clock has been switched to divn clock	0x0
10	R	TIM_PWM_BUSY Busy with synchronizing PWM_FREQ_REG and PWM_DC_REG. Do not write a new value to these registers when this bit is high.	0x0
9	R	TIM_TIMER_BUSY Busy with synchronizing PRESCALER_REG, RELOAD_REG and SHOTWIDTH_REG. Do not write a new value to these registers when this bit is high.	0x0
8	R	TIM_IRQ_STATUS IRQ status bit. When an irq has occurred, this bit is 1.	0x0
7:4	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
3:2	R	TIM_ONESHOT_PHASE OneShot phase 0 = Wait for event 1 = Delay phase 2 = Start Shot 3 = Shot phase	0x0
1	R	TIM_IN2_STATE Gives the logic level of the IN1	0x0
0	R	TIM_IN1_STATE Gives the logic level of the IN2	0x0

Table 274: TIMER3_GPIO1_CONF_REG (0x5001050C)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO1_CONF Select one of the 32 GPIOs as IN1, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 275: TIMER3_GPIO2_CONF_REG (0x50010510)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO2_CONF Select one of the 32 GPIOs as IN2, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 276: TIMER3_SETTINGS_REG (0x50010514)

Bit	Mode	Symbol/Description	Reset
31:29	-	- Reserved	0x0
28:24	R/W	TIM_PRESCALER Defines the timer count frequency. CLOCK frequency/(TIM_PRESCALER + 1)	0x0
23:0	R/W	TIM_RELOAD Reload or max value in timer mode. Actual delay is the register value plus synchronization time (3 clock cycles)	0x0

Table 277: TIMER3_CAPTURE_GPIO1_REG (0x50010520)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO1 Gives the Capture time for event on GPIO1	0x0

Table 278: TIMER3_CAPTURE_GPIO2_REG (0x50010524)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO2 Gives the Capture time for event on GPIO2	0x0

Table 279: TIMER3_PRESCALER_VAL_REG (0x50010528)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0x0
4:0	R	TIM_PRESCALER_VAL Gives the current prescaler counter value	0x0

Table 280: TIMER3_PWM_CTRL_REG (0x5001052C)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	TIM_PWM_DC Defines the PWM duty cycle. TIM_PWM_DC/(TIM_PWM_FREQ + 1)	0x0
15:0	R/W	TIM_PWM_FREQ Defines the PWM frequency. Timer clock frequency/(TIM_PWM_FREQ + 1) Timer clock is clock after prescaler	0x0

Table 281: TIMER3_CLEAR_IRQ_REG (0x50010534)

Bit	Mode	Symbol/Description	Reset
0	R0/W	TIM_CLEAR_IRQ Write any value clear interrupt	0x0

35.10 Timer 4 registers

Table 282: Register map TIMER4

Address	Register	Description
0x50020A00	TIMER4_CTRL_REG	Timer control register
0x50020A04	TIMER4_TIMER_VAL_REG	Timer counter value
0x50020A08	TIMER4_STATUS_REG	Timer status register
0x50020A0C	TIMER4_GPIO1_CONF_REG	Timer gpio1 selection
0x50020A10	TIMER4_GPIO2_CONF_REG	Timer gpio2 selection
0x50020A14	TIMER4_SETTINGS_REG	Timer reload value and Delay in shot mode
0x50020A20	TIMER4_CAPTURE_GPIO1_REG	Timer value for event on GPIO1
0x50020A24	TIMER4_CAPTURE_GPIO2_REG	Timer value for event on GPIO2
0x50020A28	TIMER4_PRESCALER_VAL_REG	Timer prescaler counter value
0x50020A2C	TIMER4_PWM_CTRL_REG	Timer pwm frequency register
0x50020A34	TIMER4_CLEAR_IRQ_REG	Timer clear interrupt

Table 283: TIMER4_CTRL_REG (0x50020A00)

Bit	Mode	Symbol/Description	Reset
31:9	-	- Reserved	0x0
8	R/W	TIM_CLK_EN Timer clock enable 1 = Clock enabled 0 = Clock disabled	0x0
7	R/W	TIM_SYS_CLK_EN Select clock 1 = Timer uses the DIVN clock 0 = Timer uses the lp clock	0x0
6	R/W	TIM_FREE_RUN_MODE_EN Valid when timer counts up, if it is 1, timer does not zero when reaches to reload value. it becomes zero only when it reaches the max value.	0x0
5	R/W	TIM_IRQ_EN Interrupt mask 1 = Timer IRQ is unmasked 0 = Timer IRQ is masked	0x0
4	R/W	TIM_IN2_EVENT_FALL_EN	0x0

Bit	Mode	Symbol/Description	Reset
		Event input 2 edge type 1 = Falling edge 0 = Rising edge	
3	R/W	TIM_IN1_EVENT_FALL_EN Event input 1 edge type 1 = Falling edge 0 = Rising edge	0x0
2	R/W	TIM_COUNT_DOWN_EN Timer count direction 1 = Down 0 = Up	0x0
1	R/W	- Reserved	0x0
0	R/W	TIM_EN Timer enable 1 = On 0 = Off	0x0

Table 284: TIMER4_TIMER_VAL_REG (0x50020A04)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_TIMER_VALUE Gives the current timer value	0x0

Table 285: TIMER4_STATUS_REG (0x50020A08)

Bit	Mode	Symbol/Description	Reset
11	R	TIM_SWITCHED_TO_DIVN_CLK Indicates that timer clock has been switched to divn clock	0x0
10	R	TIM_PWM_BUSY Busy with synchronizing PWM_FREQ_REG and PWM_DC_REG. Do not write a new value to these registers when this bit is high.	0x0
9	R	TIM_TIMER_BUSY Busy with synchronizing PRESCALER_REG, RELOAD_REG and SHOTWIDTH_REG. Do not write a new value to these registers when this bit is high.	0x0
8	R	TIM_IRQ_STATUS IRQ status bit. When an irq has occurred, this bit is 1.	0x0
7:4	-	- Reserved	0x0
3:2	R	TIM_ONESHOT_PHASE	0x0

Bit	Mode	Symbol/Description	Reset
		OneShot phase 0 = Wait for event 1 = Delay phase 2 = Start Shot 3 = Shot phase	
1	R	TIM_IN2_STATE Gives the logic level of the IN1	0x0
0	R	TIM_IN1_STATE Gives the logic level of the IN2	0x0

Table 286: TIMER4_GPIO1_CONF_REG (0x50020A0C)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO1_CONF Select one of the 32 GPIOs as IN1, Valid value 0-32. 1 for the first GPIO, 32 for the last GPIO. 0 Disable input	0x0

Table 287: TIMER4_GPIO2_CONF_REG (0x50020A10)

Bit	Mode	Symbol/Description	Reset
31:6	-	- Reserved	0x0
5:0	R/W	TIM_GPIO2_CONF Select one of the 32 GPIOs as IN2, Valid value 0-32. 1 for the first gpio, 32 for the last gpio. 0 Disable input	0x0

Table 288: TIMER4_SETTINGS_REG (0x50020A14)

Bit	Mode	Symbol/Description	Reset
31:29	-	- Reserved	0x0
28:24	R/W	TIM_PRESCALER Defines the timer count frequency. CLOCK frequency/(TIM_PRESCALER + 1)	0x0
23:0	R/W	TIM_RELOAD Reload or max value in timer mode. Actual delay is the register value plus synchronization time (3 clock cycles)	0x0

Table 289: TIMER4_CAPTURE_GPIO1_REG (0x50020A20)

Bit	Mode	Symbol/Description	Reset
31:24	-	-	0x0

Bit	Mode	Symbol/Description	Reset
		Reserved	
23:0	R	TIM_CAPTURE_GPIO1 Gives the Capture time for event on GPIO1	0x0

Table 290: TIMER4_CAPTURE_GPIO2_REG (0x50020A24)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0x0
23:0	R	TIM_CAPTURE_GPIO2 Gives the Capture time for event on GPIO2	0x0

Table 291: TIMER4_PRESCALER_VAL_REG (0x50020A28)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0x0
4:0	R	TIM_PRESCALER_VAL Gives the current prescaler counter value	0x0

Table 292: TIMER4_PWM_CTRL_REG (0x50020A2C)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	TIM_PWM_DC Defines the PWM duty cycle. TIM_PWM_DC/(TIM_PWM_FREQ + 1)	0x0
15:0	R/W	TIM_PWM_FREQ Defines the PWM frequency. Timer clock frequency/(TIM_PWM_FREQ + 1) Timer clock is clock after prescaler	0x0

Table 293: TIMER4_CLEAR_IRQ_REG (0x50020A34)

Bit	Mode	Symbol/Description	Reset
0	R0/W	TIM_CLEAR_IRQ Write any value clear interrupt	0x0

35.11 DCDC converter registers

Table 294: Register map DCDC

Address	Register	Description
0x50000300	DCDC_CTRL_REG	

Table 295: DCDC_CTRL_REG (0x50000300)

Bit	Mode	Symbol/Description	Reset
20:17	R/W	DCDC_ILIM_SLP Sets the inductor current limit value in sleep mode.	0xF
16	R/W	DCDC_FIX_ILIM_SLP Sets a fixed inductor current limit in sleep to maximize amount of charge added per cycle and minimize duty cycle. 0x0: Automatic current limit setting remains active in sleep mode 0x1: Current limit fixed in sleep mode, value set with DCDC_ILIM_SLP field	0x1
15:12	R/W	DCDC_ILIM_MAX Maximum value for automatic inductor peak current limit control. 0x0: 6 mA 0x1: 12 mA 0x2: 18 mA 0x3: 24 mA 0x4: 30 mA 0x5: 36 mA 0x6: 42 mA 0x7: 48 mA 0x8: 54 mA 0x9: 60 mA 0xA: 66 mA 0xB: 72 mA 0xC: 78 mA 0xD: 84 mA 0xE: 90 mA 0xF: 96 mA (default)	0xF
11:8	R/W	DCDC_ILIM_MIN Minimum value for automatic inductor peak current limit control. 0x0: 6 mA 0x1: 12 mA 0x2: 18 mA 0x3: 24 mA 0x4: 30 mA (default) 0x5: 36 mA 0x6: 42 mA 0x7: 48 mA 0x8: 54 mA 0x9: 60 mA 0xA: 66 mA 0xB: 72 mA 0xC: 78 mA 0xD: 84 mA 0xE: 90 mA 0xF: 96 mA	0x4
7:6	R/W	DCDC_OK_CLR_CNT Number of subsequent V_NOK events needed to reset VDCD_OK.	0x2

Bit	Mode	Symbol/Description	Reset
		0x0: 2 0x1: 4 0x2: 8 (default) 0x3: 15	
5:3	R/W	DCDC_TIMEOUT Switch timeout, go to next state if either switch is active for longer than this setting. 0x0: Disabled 0x1: 0.25 µs 0x2: 0.50 µs 0x3: 0.75 µs 0x4: 1.00 µs (default) 0x5: 1.25 µs 0x6: 1.50 µs 0x7: 1.75 µs	0x4
2:1	R/W	DCDC_CLK_DIV Idle clock divider sets rate at which the output is monitored when the converter is idle. 0x0: Divide by 4 0x1: Divide by 8 0x2: Divide by 16 0x3: Divide by 32	0x1
0	R/W	DCDC_ENABLE Enables hardware control of the DCDC converter. 0x0: DCDC converter disabled 0x1: DCDC converter under hardware control	0x0

35.12 UART registers

Table 296: Register map UART

Address	Register	Description
0x50020000	UART_RBR_THR_DLL_REG	Receive Buffer Register
0x50020004	UART_IER_DLH_REG	Interrupt Enable Register
0x50020008	UART_IIR_FCR_REG	Interrupt Identification Register/FIFO Control Register
0x5002000C	UART_LCR_REG	Line Control Register
0x50020010	UART_MCR_REG	Modem Control Register
0x50020014	UART_LSR_REG	Line Status Register
0x5002001C	UART_SCR_REG	Scratchpad Register
0x50020030	UART_SRBR_STHR0_REG	Shadow Receive/Transmit Buffer Register
0x50020034	UART_SRBR_STHR1_REG	Shadow Receive/Transmit Buffer Register
0x50020038	UART_SRBR_STHR2_REG	Shadow Receive/Transmit Buffer Register

Address	Register	Description
0x5002003C	UART_SRBR_STHR3_REG	Shadow Receive/Transmit Buffer Register
0x50020040	UART_SRBR_STHR4_REG	Shadow Receive/Transmit Buffer Register
0x50020044	UART_SRBR_STHR5_REG	Shadow Receive/Transmit Buffer Register
0x50020048	UART_SRBR_STHR6_REG	Shadow Receive/Transmit Buffer Register
0x5002004C	UART_SRBR_STHR7_REG	Shadow Receive/Transmit Buffer Register
0x50020050	UART_SRBR_STHR8_REG	Shadow Receive/Transmit Buffer Register
0x50020054	UART_SRBR_STHR9_REG	Shadow Receive/Transmit Buffer Register
0x50020058	UART_SRBR_STHR10_REG	Shadow Receive/Transmit Buffer Register
0x5002005C	UART_SRBR_STHR11_REG	Shadow Receive/Transmit Buffer Register
0x50020060	UART_SRBR_STHR12_REG	Shadow Receive/Transmit Buffer Register
0x50020064	UART_SRBR_STHR13_REG	Shadow Receive/Transmit Buffer Register
0x50020068	UART_SRBR_STHR14_REG	Shadow Receive/Transmit Buffer Register
0x5002006C	UART_SRBR_STHR15_REG	Shadow Receive/Transmit Buffer Register
0x5002007C	UART_USR_REG	UART Status register.
0x50020080	UART_TFL_REG	Transmit FIFO Level
0x50020084	UART_RFL_REG	Receive FIFO Level.
0x50020088	UART_SRR_REG	Software Reset Register.
0x50020090	UART_SBCR_REG	Shadow Break Control Register
0x50020094	UART_SDMAM_REG	Shadow DMA Mode
0x50020098	UART_SFE_REG	Shadow FIFO Enable
0x5002009C	UART_SRT_REG	Shadow RCVR Trigger
0x500200A0	UART_STET_REG	Shadow TX Empty Trigger
0x500200A4	UART_HTX_REG	Halt TX
0x500200A8	UART_DMASA_REG	DMA Software Acknowledge
0x500200C0	UART_DLF_REG	Divisor Latch Fraction Register
0x500200F8	UART_UCV_REG	Component Version
0x500200FC	UART_CTR_REG	Component Type Register

Table 297: UART_RBR_THR_DLL_REG (0x50020000)

Bit	Mode	Symbol/Description	Reset
31:8	-	-	0x0

Bit	Mode	Symbol/Description	Reset
		Reserved	
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 DLL is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data. Divisor Latch (High): (DLH) (Note: This register is placed in UART_IER_DLH_REG with offset 0x4) Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock frequency divided by sixteen times the value of the baud rate divisor, as follows: 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 occur. Also, once the DLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.	0x0

Table 298: UART_IER_DLH_REG (0x50020004)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
7	R/W	PTIME_DLH7	0x0

Bit	Mode	Symbol/Description	Reset
		Interrupt Enable Register: PTIME, Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[7] of the 8 bit DLH register.	
6:5	R/W	DLH6_5 Divisor Latch (High): Bit[6:5] of the 8 bit DLH register	0x0
4	R/W	ELCOLR_DLH4 Interrupt Enable Register: (read only) ELCOLR, this bit controls the method for clearing the status in the LSR register. This is applicable only for Overrun Error, Parity Error, Framing Error, and Break Interrupt status bits. Always 0 = LSR status bits are cleared either on reading RX FIFO (RBR Read) or on reading LSR register. Divisor Latch (High): Bit[4] of the 8 bit DLH register	0x0
3	R/W	EDSSI_DLH3 Interrupt Enable Register: reserved Divisor Latch (High): Bit[3] of the 8 bit DLH register	0x0
2	R/W	ELSI_DLH2 Interrupt Enable Register: ELSI, Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[2] of the 8 bit DLH register.	0x0
1	R/W	ETBEI_DLH1 Interrupt Enable Register: ETBEI, Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[1] of the 8 bit DLH register.	0x0
0	R/W	ERBFI_DLH0 Interrupt Enable Register: ERBFI, Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFO's enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled Divisor Latch (High): Bit[0] of the 8 bit DLH register.	0x0

Table 299: UART_IIR_FCR_REG (0x50020008)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
7:0	R/W	IIR_FCR On Read 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[5:4],Reserved	0x1

Bit	Mode	Symbol/Description	Reset
		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. On Write 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.	

Table 300: UART_LCR_REG (0x5002000C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
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 the initial baud rate setup 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.	0x0
5	-	-	0x0

Bit	Mode	Symbol/Description	Reset
		Reserved	
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 301: UART_MCR_REG (0x50020010)

Bit	Mode	Symbol/Description	Reset
31:7	-	- Reserved	0x0
6	R/W	- Reserved	0x0
5	R/W	- Reserved	0x0
4	R/W	UART_LB LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.	0x0

Bit	Mode	Symbol/Description	Reset
		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.	
3	R/W	- Reserved	0x0
2	R/W	- Reserved	0x0
1	R/W	- Reserved	0x0
0	R/W	- Reserved	0x0

Table 302: UART_LSR_REG (0x50020014)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x1

Bit	Mode	Symbol/Description	Reset
		the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.	
4	R	UART_BI Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. 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. 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 , that is 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
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.	0x0

Bit	Mode	Symbol/Description	Reset
		0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit.	
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 303: UART_SCR_REG (0x5002001C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 304: UART_SRBR_STHR0_REG (0x50020030)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 305: UART_SRBR_STHR1_REG (0x50020034)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 306: UART_SRBR_STHR2_REG (0x50020038)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		(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.	

Table 307: UART_SRBR_STHR3_REG (0x5002003C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 308: UART_SRBR_STHR4_REG (0x50020040)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 309: UART_SRBR_STHR5_REG (0x50020044)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 310: UART_SRBR_STHR6_REG (0x50020048)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 311: UART_SRBR_STHR7_REG (0x5002004C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.	

Table 312: UART_SRBR_STHR8_REG (0x50020050)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 313: UART_SRBR_STHR9_REG (0x50020054)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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-	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 314: UART_SRBR_STHR10_REG (0x50020058)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 315: UART_SRBR_STHR11_REG (0x5002005C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 316: UART_SRBR_STHR12_REG (0x50020060)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 317: UART_SRBR_STHR13_REG (0x50020064)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 318: UART_SRBR_STHR14_REG (0x50020068)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 319: UART_SRBR_STHR15_REG (0x5002006C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 320: UART_USR_REG (0x5002007C)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	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.	0x0

Bit	Mode	Symbol/Description	Reset
		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.	
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 UART is idle or inactive. 0 = UART is idle or inactive 1 = 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 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 UART. This is because 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 321: UART_TFL_REG (0x50020080)

Bit	Mode	Symbol/Description	Reset
4:0	R	UART_TRANSMIT_FIFO_LEVEL Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO.	0x0

Table 322: UART_RFL_REG (0x50020084)

Bit	Mode	Symbol/Description	Reset
4:0	R	UART_RECEIVE_FIFO_LEVEL Receive FIFO Level. This indicates the number of data entries in the receive FIFO.	0x0

Table 323: UART_SRR_REG (0x50020088)

Bit	Mode	Symbol/Description	Reset
31:3	-	- Reserved	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 324: UART_SBCR_REG (0x50020090)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	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.	0x0

Table 325: UART_SDMAM_REG (0x50020094)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		Mode bit gets updated. This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals. 0 = mode 0 1 = mode 1	

Table 326: UART_SFE_REG (0x50020098)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	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 327: UART_SRT_REG (0x5002009C)

Bit	Mode	Symbol/Description	Reset
31:2	-	- Reserved	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 328: UART_STET_REG (0x500200A0)

Bit	Mode	Symbol/Description	Reset
31:2	-	- Reserved	0x0
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	0x0

Bit	Mode	Symbol/Description	Reset
		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	

Table 329: UART_HTX_REG (0x500200A4)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R/W	UART_HALT_TX This register is used to halt transmissions, 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 not enabled, the setting of the halt TX register has no effect on operation.	0x0

Table 330: UART_DMASA_REG (0x500200A8)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	W	UART_DMASA This register is used 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 331: UART_DLF_REG (0x500200C0)

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 332: UART_UCV_REG (0x500200F8)

Bit	Mode	Symbol/Description	Reset
31:0	R	UART_UCV Component Version	0x34303 12A

Table 333: UART_CTR_REG (0x500200FC)

Bit	Mode	Symbol/Description	Reset
31:0	R	UART_CTR Component Type Register	0x44570110

35.13 UART2 registers

Table 334: Register map UART2

Address	Register	Description
0x50020100	UART2_RBR_THR_DLL_REG	Receive Buffer Register
0x50020104	UART2_IER_DLH_REG	Interrupt Enable Register
0x50020108	UART2_IIR_FCR_REG	Interrupt Identification Register/FIFO Control Register
0x5002010C	UART2_LCR_REG	Line Control Register
0x50020110	UART2_MCR_REG	Modem Control Register
0x50020114	UART2_LSR_REG	Line Status Register
0x50020118	UART2_MSR_REG	Modem Status Register
0x5002011C	UART2_CONFIG_REG	ISO7816 Config Register
0x50020130	UART2_SRBR_STHR0_REG	Shadow Receive/Transmit Buffer Register
0x50020134	UART2_SRBR_STHR1_REG	Shadow Receive/Transmit Buffer Register
0x50020138	UART2_SRBR_STHR2_REG	Shadow Receive/Transmit Buffer Register
0x5002013C	UART2_SRBR_STHR3_REG	Shadow Receive/Transmit Buffer Register
0x50020140	UART2_SRBR_STHR4_REG	Shadow Receive/Transmit Buffer Register
0x50020144	UART2_SRBR_STHR5_REG	Shadow Receive/Transmit Buffer Register
0x50020148	UART2_SRBR_STHR6_REG	Shadow Receive/Transmit Buffer Register
0x5002014C	UART2_SRBR_STHR7_REG	Shadow Receive/Transmit Buffer Register
0x50020150	UART2_SRBR_STHR8_REG	Shadow Receive/Transmit Buffer Register
0x50020154	UART2_SRBR_STHR9_REG	Shadow Receive/Transmit Buffer Register
0x50020158	UART2_SRBR_STHR10_REG	Shadow Receive/Transmit Buffer Register
0x5002015C	UART2_SRBR_STHR11_REG	Shadow Receive/Transmit Buffer Register

Address	Register	Description
0x50020160	UART2_SRBR_STHR12_REG	Shadow Receive/Transmit Buffer Register
0x50020164	UART2_SRBR_STHR13_REG	Shadow Receive/Transmit Buffer Register
0x50020168	UART2_SRBR_STHR14_REG	Shadow Receive/Transmit Buffer Register
0x5002016C	UART2_SRBR_STHR15_REG	Shadow Receive/Transmit Buffer Register
0x5002017C	UART2_USR_REG	UART Status register.
0x50020180	UART2_TFL_REG	Transmit FIFO Level
0x50020184	UART2_RFL_REG	Receive FIFO Level.
0x50020188	UART2_SRR_REG	Software Reset Register.
0x5002018C	UART2_SRTS_REG	Shadow Request to Send
0x50020190	UART2_SBCR_REG	Shadow Break Control Register
0x50020194	UART2_SDMAM_REG	Shadow DMA Mode
0x50020198	UART2_SFE_REG	Shadow FIFO Enable
0x5002019C	UART2_SRT_REG	Shadow RCVR Trigger
0x500201A0	UART2_STET_REG	Shadow TX Empty Trigger
0x500201A4	UART2_HTX_REG	Halt TX
0x500201A8	UART2_DMASA_REG	DMA Software Acknowledge
0x500201C0	UART2_DLF_REG	Divisor Latch Fraction Register
0x500201C4	UART2_RAR_REG	Receive Address Register
0x500201C8	UART2_TAR_REG	Transmit Address Register
0x500201CC	UART2_LCR_EXT	Line Extended Control Register
0x500201E0	UART2_CTRL_REG	ISO7816 Control Register
0x500201E4	UART2_TIMER_REG	ISO7816 Timer Register
0x500201E8	UART2_ERR_CTRL_REG	ISO7816 Error Signal Control Register
0x500201EC	UART2_IRQ_STATUS_REG	ISO7816 Interrupt Status Register
0x500201F8	UART2_UCV_REG	Component Version
0x500201FC	UART2_CTR_REG	Component Type Register

Table 335: UART2_RBR_THR_DLL_REG (0x50020100)

Bit	Mode	Symbol/Description	Reset
8	R/W	RBR_THR_9BIT When 9BIT_DATA_EN, On read: Receive Buffer bit 8 - On write: Transmit Buffer bit 8 when LCR_EXT[3] = 1	0x0
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	0x0

Bit	Mode	Symbol/Description	Reset
		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 DLL is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data. Divisor Latch (High): (DLH) (Note: This register is placed in UART_IER_DLH_REG with offset 0x4) Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may be accessed only when the DLAB bit (LCR[7]) is set. The output baud rate is equal to the serial clock frequency divided by sixteen times the value of the baud rate divisor, as follows: 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 occur. Also, once the DLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data.	

Table 336: UART2_IER_DLH_REG (0x50020104)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
7	R/W	PTIME_DLH7 Interrupt Enable Register: PTIME, Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[7] of the 8 bit DLH register.	0x0
6:5	R/W	DLH6_5 Divisor Latch (High): Bit[6:5] of the 8 bit DLH register	0x0

Bit	Mode	Symbol/Description	Reset
4	R/W	ELCOLR_DLH4 Interrupt Enable Register: ELCOLR (read only), this bit controls the method for clearing the status in the LSR register. This is applicable only for Overrun Error, Parity Error, Framing Error, and Break Interrupt status bits. 0 = LSR status bits are cleared either on reading RX FIFO (RBR Read) or on reading LSR register. Divisor Latch (High): Bit[4] of the 8 bit DLH register	0x0
3	R/W	EDSSI_DLH3 Interrupt Enable Register: EDSSI, Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[3] of the 8 bit DLH register	0x0
2	R/W	ELSI_DLH2 Interrupt Enable Register: ELSI, Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[2] of the 8 bit DLH register.	0x0
1	R/W	ETBEI_DLH1 Interrupt Enable Register: ETBEI, Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled Divisor Latch (High): Bit[1] of the 8 bit DLH register.	0x0
0	R/W	ERBFI_DLH0 Interrupt Enable Register: ERBFI, Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFO's enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled Divisor Latch (High): Bit[0] of the 8 bit DLH register.	0x0

Table 337: UART2_IIR_FCR_REG (0x50020108)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
7:0	R/W	IIR_FCR On Read 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[5:4],Reserved 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. On Write 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	0x1

Bit	Mode	Symbol/Description	Reset
		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 signaling 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.	

Table 338: UART2_LCR_REG (0x5002010C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
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 the initial baud rate setup 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.	0x0
5	R/W	UART_SP Stick Parity. (writeable only when UART is not busy USR[0] is 0); otherwise always writable and always readable. This bit is used to force parity value. When PEN, EPS and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled.	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

Bit	Mode	Symbol/Description	Reset
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 are 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 339: UART2_MCR_REG (0x50020110)

Bit	Mode	Symbol/Description	Reset
31:7	-	- Reserved	0x0
6	R/W	- Reserved	0x0
5	R/W	UART_AFCE Auto Flow Control Enable. 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. 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) are looped back to the inputs, internally.	0x0

Bit	Mode	Symbol/Description	Reset
3	R/W	- Reserved	0x0
2	R/W	- Reserved	0x0
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, 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	- Reserved	0x0

Table 340: UART2_LSR_REG (0x50020114)

Bit	Mode	Symbol/Description	Reset
8	R	UART_ADDR_RCVD Address Received Bit. If 9Bit data mode (LCR_EXT[0]=1) is enabled, this bit is used to indicate the 9th bit of the receive data is set to 1. This bit can also be used to indicate whether the incoming character is address or data. 1 = Indicates the character is address. 0 = Indicates the character is data. In the FIFO mode, since the 9th bit is associated with a character received, it is revealed when the character with the 9th bit set to 1 is at the top of the FIFO. Reading the LSR clears the 9BIT. Note: User needs to ensure that interrupt gets cleared (reading LSR register) before the next address byte arrives. If there is a delay in clearing the interrupt, then Software will not be able to distinguish between multiple address related interrupt.	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	0x1

Bit	Mode	Symbol/Description	Reset
		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.	
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
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. 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 that is 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.	0x0

Bit	Mode	Symbol/Description	Reset
		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.	
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 341: UART2_MSR_REG (0x50020118)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	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).	0x1
3:1	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 342: UART2_CONFIG_REG (0x5002011C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
7:3	R/W	ISO7816_SCRATCH_PAD This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART Ctrl.	0x0
2	R/W	ISO7816_ENABLE 0: Normal UART 1: ISO7816 Enabled	0x0
1	R/W	ISO7816_ERR_SIG_EN 0: Error Signal feature disabled 1: Error Signal feature enabled	0x0
0	R/W	ISO7816_CONVENTION 0: Direct convention 1: Inverse convention	0x0

Table 343: UART2_SRBR_STHR0_REG (0x50020130)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 344: UART2_SRBR_STHR1_REG (0x50020134)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 345: UART2_SRBR_STHR2_REG (0x50020138)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 346: UART2_SRBR_STHR3_REG (0x5002013C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 347: UART2_SRBR_STHR4_REG (0x50020140)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0
7:0	R/W	SRBR_STHRx	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 348: UART2_SRBR_STHR5_REG (0x50020144)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 349: UART2_SRBR_STHR6_REG (0x50020148)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 350: UART2_SRBR_STHR7_REG (0x5002014C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		(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.	

Table 351: UART2_SRBR_STHR8_REG (0x50020150)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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: UART2_SRBR_STHR9_REG (0x50020154)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 353: UART2_SRBR_STHR10_REG (0x50020158)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 354: UART2_SRBR_STHR11_REG (0x5002015C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
7:0	R/W	SRBR_STHRx	0x0
		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.	

Table 355: UART2_SRBR_STHR12_REG (0x50020160)

Bit	Mode	Symbol/Description	Reset
31:8	-	-	0x0
		Reserved	
7:0	R/W	SRBR_STHRx	0x0
		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	

Bit	Mode	Symbol/Description	Reset
		Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.	

Table 356: UART2_SRBR_STHR13_REG (0x50020164)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 357: UART2_SRBR_STHR14_REG (0x50020168)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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-	0x0

Bit	Mode	Symbol/Description	Reset
		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.	

Table 358: UART2_SRBR_STHR15_REG (0x5002016C)

Bit	Mode	Symbol/Description	Reset
31:8	-	- Reserved	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 359: UART2_USR_REG (0x5002017C)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	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	0x0

Bit	Mode	Symbol/Description	Reset
		1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.	
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 because 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 360: UART2_TFL_REG (0x50020180)

Bit	Mode	Symbol/Description	Reset
4:0	R	UART_TRANSMIT_FIFO_LEVEL Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO.	0x0

Table 361: UART2_RFL_REG (0x50020184)

Bit	Mode	Symbol/Description	Reset
4:0	R	UART_RECEIVE_FIFO_LEVEL Receive FIFO Level. This indicates the number of data entries in the receive FIFO.	0x0

Table 362: UART2_SRR_REG (0x50020188)

Bit	Mode	Symbol/Description	Reset
31:3	-	- Reserved	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 363: UART2_SRTS_REG (0x5002018C)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
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, (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 364: UART2_SBCR_REG (0x50020190)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	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 perform 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.	0x0

Table 365: UART2_SDMAM_REG (0x50020194)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	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 signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals. 0 = mode 0 1 = mode 1	0x0

Table 366: UART2_SFE_REG (0x50020198)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	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 367: UART2_SRT_REG (0x5002019C)

Bit	Mode	Symbol/Description	Reset
31:2	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
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 368: UART2_STET_REG (0x500201A0)

Bit	Mode	Symbol/Description	Reset
31:2	-	- Reserved	0x0
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 369: UART2_HTX_REG (0x500201A4)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	R/W	UART_HALT_TX This register is use to halt transmissions, 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 not enabled, the setting of the halt TX register has no effect on operation.	0x0

Table 370: UART2_DMASA_REG (0x500201A8)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0x0
0	W	UART_DMASA This register is used 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 371: UART2_DLF_REG (0x500201C0)

Bit	Mode	Symbol/Description	Reset
31:4	-	- Reserved	0x0
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 372: UART2_RAR_REG (0x500201C4)

Bit	Mode	Symbol/Description	Reset
7:0	R/W	UART_RAR This is an address matching register during receive mode. If the 9-th bit is set in the incoming character, then the remaining 8-bits will be checked against this register value. If the match happens then sub-sequent characters with 9-th bit set to 0 will be treated as data byte until the next address byte is received. Note: - This register is applicable only when 'ADDR_MATCH'(LCR_EXT[1] and 'DLS_E' (LCR_EXT[0]) bits are set to 1. RAR should be programmed only when UART is not busy.	0x0

Table 373: UART2_TAR_REG (0x500201C8)

Bit	Mode	Symbol/Description	Reset
7:0	R/W	UART_TAR This is an address matching register during transmit mode. If DLS_E (LCR_EXT[0]) bit is enabled, then UART will send the 9-bit character with 9-th bit set to 1 and remaining 8-bit address will be sent from this register provided 'SEND_ADDR' (LCR_EXT[2]) bit is set to 1. Note: - This register is used only to send the address. The normal data should be sent by programming THR register. - Once the address is started to send on the DW_apb_uart serial lane, then 'SEND_ADDR' bit will be auto-cleared by the hardware.	0x0

Table 374: UART2_LCR_EXT (0x500201CC)

Bit	Mode	Symbol/Description	Reset
3	R/W	UART_TRANSMIT_MODE Transmit mode control bit. This bit is used to control the type of transmit mode during 9-bit data transfers. 1 = In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding Register (STHR) are 9-bit wide. The user needs to ensure that the THR/STHR register is written correctly for address/data. Address: 9th bit is set to 1, Data: 9th bit is set to 0. Note: Transmit address register (TAR) is not applicable in this mode of operation. 0 = In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding register (STHR) are 8-bit wide. The user needs to program the address into Transmit Address Register (TAR) and data into the THR/STHR register. SEND_ADDR bit is used as a control knob to indicate the uart on when to send the address.	0x0
2	R/W	UART_SEND_ADDR Send address control bit. This bit is used as a control knob for the user to determine when to send the address during transmit mode. 1 = 9-bit character will be transmitted with 9-th bit set to 1 and the remaining 8-bits will match to what is being programmed in "Transmit Address Register". 0 = 9-bit character will be transmitted with 9-th bit set to 0 and the remaining 8-bits will be taken from the TXFIFO which is programmed through 8-bit wide THR/STHR register. Note: 1. This bit is auto-cleared by the hardware, after sending out the address character. User is not expected to program this bit to 0. 2. This field is applicable only when DLS_E bit is set to 1 and TRANSMIT_MODE is set to 0.	0x0
1	R/W	UART_ADDR_MATCH Address Match Mode.This bit is used to enable the address match feature during receive. 1 = Address match mode; uart will wait until the incoming character with 9-th bit set to 1. And further checks to see if the address matches with what is programmed in "Receive Address Match Register". If match is found, then sub-sequent characters will be treated as valid data and DW_apb_uart starts receiving data. 0 = Normal mode; DW_apb_uart will start to receive the data and 9-bit character will be formed and written into the receive RXFIFO. User is responsible to read the data and differentiate b/n address and data. Note: This field is applicable only when DLS_E is set to 1.	0x0
0	R/W	UART_DLS_E Extension for DLS. This bit is used to enable 9-bit data for transmit and receive transfers.	0x0

Table 375: UART2_CTRL_REG (0x500201E0)

Bit	Mode	Symbol/Description	Reset
31:12	-	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
11	R/W	ISO7816_AUTO_GT 0: UART sends when TX data is available 1: UART sends new character after guard time	0x0
10	R/W	ISO7816_ERR_TX_VALUE_IRQMASK 0: ERR_TX_VALUE IRQ is masked 1: ERR_TX_VALUE IRQ is enabled	0x0
9	R/W	ISO7816_ERR_TX_TIME_IRQMASK 0: ERR_TX_TIME IRQ is masked 1: ERR_TX_TIME IRQ is enabled	0x0
8	R/W	ISO7816_TIM_EXPIRED_IRQMASK 0: Timer expired IRQ is masked 1: Timer expired IRQ is enabled	0x0
7	R	ISO7816_CLK_STATUS 0: ISO7816 clock is stopped 1: ISO7816 clock is running	0x0
6	R/W	ISO7816_CLK_LEVEL 0: ISO7816 clock level low when stopped 1: ISO7816 clock level high when stopped	0x0
5	R/W	ISO7816_CLK_EN 0: ISO7816 clock disabled 1: ISO7816 clock enabled	0x0
4:0	R/W	ISO7816_CLK_DIV ISO7816 clk freq = sclk/(2*(ISO7816_CLK_DIV+1)	0x0

Table 376: UART2_TIMER_REG (0x500201E4)

Bit	Mode	Symbol/Description	Reset
31:18	-	- Reserved	0x0
17	R/W	ISO7816_TIM_MODE 0: Timer will count up to max value then stops. Timer has to be disabled and enabled again to restart. Timer is clocked with the ISO7816 clock 1: Timer will count guard time. ISO7816_TIM_MAX has to be 16*GuardTime-1	0x0
16	R/W	ISO7816_TIM_EN 0: Timer is disabled 1: Timer is enabled	0x0
15:0	R/W	ISO7816_TIM_MAX On write: timer will count from 0 to ISO7816_TIM_MAX On read: gives the current timer value	0x0

Table 377: UART2_ERR_CTRL_REG (0x500201E8)

Bit	Mode	Symbol/Description	Reset
31:9	-	- Reserved	0x0
8:4	R/W	ISO7816_ERR_PULSE_WIDTH When Error Signal feature is enable and receive mode, it gives the width of the error signal in 1/16 etu	0x10
3:0	R/W	ISO7816_ERR_PULSE_OFFSET When Error Signal feature is enable and receive mode, it gives the offset of the error signal in 1/16 etu from the 9.6 etu	0xE

Table 378: UART2_IRQ_STATUS_REG (0x500201EC)

Bit	Mode	Symbol/Description	Reset
31:3	-	- Reserved	0x0
2	R/W	ISO7816_ERR_TX_VALUE_IRQ On read 1: If error signal is enabled and in transmit mode, module generates IRQ when receiver does not receive correctly the character On Write 1: Clear IRQ	0x0
1	R/W	ISO7816_ERR_TX_TIME_IRQ On read 1: If error signal is enabled and in transmit mode, module generates IRQ when it checks the error signal On Write 1: Clear IRQ	0x0
0	R	ISO7816_TIM_EXPIRED_IRQ On read 1: When Timer is expired. Timer has to be disabled to clear the IRQ. When sclk is lower than pclk then this bit has to be checked if it's cleared before return form the IRQ Handler	0x0

Table 379: UART2_UCV_REG (0x500201F8)

Bit	Mode	Symbol/Description	Reset
31:0	R	UART_UCV Component Version	0x34303 12A

Table 380: UART2_CTR_REG (0x500201FC)

Bit	Mode	Symbol/Description	Reset
31:0	R	UART_CTR Component Type Register	0x44570 110

35.14 Silicon version registers

Table 381: Register map Version

Address	Register	Description
0x50050200	CHIP_ID1_REG	Chip identification register 1.
0x50050204	CHIP_ID2_REG	Chip identification register 2.
0x50050208	CHIP_ID3_REG	Chip identification register 3.
0x5005020C	CHIP_ID4_REG	Chip identification register 4.
0x50050210	CHIP_SWC_REG	Software compatibility register.
0x50050214	CHIP_REVISION_REG	Chip revision register.
0x500502F8	CHIP_TEST1_REG	Chip test register 1.
0x500502FC	CHIP_TEST2_REG	Chip test register 2.

Table 382: CHIP_ID1_REG (0x50050200)

Bit	Mode	Symbol/Description	Reset
7:0	R	CHIP_ID1 First character of device type in ASCII.	0x32

Table 383: CHIP_ID2_REG (0x50050204)

Bit	Mode	Symbol/Description	Reset
7:0	R	CHIP_ID2 Second character of device type in ASCII.	0x36

Table 384: CHIP_ID3_REG (0x50050208)

Bit	Mode	Symbol/Description	Reset
7:0	R	CHIP_ID3 Third character of device type in ASCII.	0x33

Table 385: CHIP_ID4_REG (0x5005020C)

Bit	Mode	Symbol/Description	Reset
7:0	R	CHIP_ID4 Fourth character of device type in ASCII.	0x34

Table 386: CHIP_SWC_REG (0x50050210)

Bit	Mode	Symbol/Description	Reset
3:0	R	CHIP_SWC SoftWare Compatibility (SWC) code.	0x2

Bit	Mode	Symbol/Description	Reset
		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.	

Table 387: CHIP_REVISION_REG (0x50050214)

Bit	Mode	Symbol/Description	Reset
7:0	R	CHIP_REVISION Chip version, corresponds with type number in ASCII. 0x41 = A, 0x42 = B	0x41

Table 388: CHIP_TEST1_REG (0x500502F8)

Bit	Mode	Symbol/Description	Reset
7:0	R	CHIP_LAYOUT_REVISION Chip layout revision, corresponds with type number in ASCII. 0x41 = A, 0x42 = B	0x45

Table 389: CHIP_TEST2_REG (0x500502FC)

Bit	Mode	Symbol/Description	Reset
3:0	R	CHIP_METAL_OPTION Chip metal option value.	0x0

35.15 Wake-up controller registers

Table 390: Register map WakeUp

Address	Register	Description
0x50000100	WKUP_CTRL_REG	Control register for the wake-up counter
0x50000108	WKUP_RESET_IRQ_REG	Reset wake-up interrupt
0x50000114	WKUP_SELECT_P0_REG	Select which inputs from P0 port can trigger wake-up counter
0x50000118	WKUP_SELECT_P1_REG	Select which inputs from P1 port can trigger wake-up counter
0x50000128	WKUP_POL_P0_REG	Select the sensitivity polarity for each P0 input
0x5000012C	WKUP_POL_P1_REG	Select the sensitivity polarity for each P1 input
0x5000013C	WKUP_STATUS_P0_REG	Event status register for P0
0x50000140	WKUP_STATUS_P1_REG	Event status register for P1
0x50000148	WKUP_CLEAR_P0_REG	Clear event register for P0
0x5000014C	WKUP_CLEAR_P1_REG	Clear event register for P1

Address	Register	Description
0x50000154	WKUP_SEL_GPIO_P0_REG	Enable fast wake-up and enable GPIO_P0_IRQ
0x50000158	WKUP_SEL_GPIO_P1_REG	Enable fast wake-up and enable GPIO_P1_IRQ
0x5000015C	WKUP_SEL1_GPIO_P0_REG	Configure to generate level or edge sensitive IRQ on P0 events
0x50000160	WKUP_SEL1_GPIO_P1_REG	Configure to generate level or edge sensitive IRQ on P1 events

Table 391: WKUP_CTRL_REG (0x50000100)

Bit	Mode	Symbol/Description	Reset
15:8	-	- Reserved	0x0
7	R/W	WKUP_ENABLE_IRQ 0: no interrupt will be enabled 1: if you have an event an IRQ will be generated	0x0
6	R/W	WKUP_SFT_KEYHIT 0 = no effect 1 = emulate key hit. First make this bit 0 before any new key hit can be sensed.	0x0
5:0	R/W	WKUP_DEB_VALUE Wake-up debounce time. If set to 0, no debouncing will be done. Debounce time: N*1 ms. N = 1...63 Note: Depending on the time key is pressed, debounce time can be less than N*1 ms. To make sure that debounce time is bigger than the programmed time, program this register field to ("desired value"+1). So, if at least 2 ms of debounce time is required, program the register to 3.	0x0

Table 392: WKUP_RESET_IRQ_REG (0x50000108)

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 393: WKUP_SELECT_P0_REG (0x50000114)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_SELECT_P0 0: input P0_xx is not enabled for wake-up event 1: input P0_xx is enabled for wake-up event	0x0

Table 394: WKUP_SELECT_P1_REG (0x50000118)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_SELECT_P1 0: input P1_xx is not enabled for wake-up event 1: input P1_xx is enabled for wake-up event	0x0

Table 395: WKUP_POL_P0_REG (0x50000128)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_POL_P0 0: enabled input P0_xx will give an event if that input goes high 1: enabled input P0_xx will give an event if that input goes low	0x0

Table 396: WKUP_POL_P1_REG (0x5000012C)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_POL_P1 0: enabled input P1_xx will give an event if that input goes high 1: enabled input P1_xx will give an event if that input goes low	0x0

Table 397: WKUP_STATUS_P0_REG (0x5000013C)

Bit	Mode	Symbol/Description	Reset
15:0	R	WKUP_STAT_P0 Contains the latched value of any toggle of the GPIOs Port P0. WKUP_STAT_P0[0] -> P0_00.	0x0

Table 398: WKUP_STATUS_P1_REG (0x50000140)

Bit	Mode	Symbol/Description	Reset
15:0	R	WKUP_STAT_P1 Contains the latched value of any toggle of the GPIOs Port P1. WKUP_STAT_P1[0] -> P1_00.	0x0

Table 399: WKUP_CLEAR_P0_REG (0x50000148)

Bit	Mode	Symbol/Description	Reset
15:0	W	WKUP_CLEAR_P0 Clear latched value of the GPIOs P0 when corresponding bit is 1	0x0

Table 400: WKUP_CLEAR_P1_REG (0x5000014C)

Bit	Mode	Symbol/Description	Reset
15:0	W	WKUP_CLEAR_P1 Clear latched value of the GPIOs P1 when corresponding bit is 1	0x0

Table 401: WKUP_SEL_GPIO_P0_REG (0x50000154)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_SEL_GPIO_P0 0: No GPIO_P0_IRQ on input P0_x. Fast wake-up is not enabled if the corresponding WKUP_SEL1_GPIO_P0_REG[x] is 0 too. 1: GPIO_P0_IRQ will be generated on P0_x input event. If WKUP_SEL1_GPIO_P0_REG[x] is 0, IRQ generation is level sensitive. If WKUP_SEL1_GPIO_P0_REG[x] is 1, IRQ generation is edge sensitive (only if there is a change on P0_x input). Fast wake-up from the corresponding P0_x input is enabled.	0x0

Table 402: WKUP_SEL_GPIO_P1_REG (0x50000158)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_SEL_GPIO_P1 0: No GPIO_P1_IRQ on input P1_x. Fast wake-up is not enabled if the corresponding WKUP_SEL1_GPIO_P1_REG[x] is 0 too. 1: GPIO_P1_IRQ will be generated on P1_x input event. If WKUP_SEL1_GPIO_P1_REG[x] is 0, IRQ generation is level sensitive. If WKUP_SEL1_GPIO_P1_REG[x] is 1, IRQ generation is edge sensitive (only if there is a change on P1_x input). Fast wake-up from the corresponding P1_x input is enabled.	0x0

Table 403: WKUP_SEL1_GPIO_P0_REG (0x5000015C)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_SEL1_GPIO_P0 0 (level sensitive): If WKUP_SEL_GPIO_P0_REG[x] is 1, generate GPIO_P0_IRQ based on P0_x level. Fast wake-up is not enabled if the corresponding WKUP_SEL_GPIO_P0_REG[x] is 0 too. 1 (edge sensitive): If WKUP_SEL_GPIO_P0_REG[x] is 1, GPIO_P0_IRQ will be generated only on rising/falling (defined by WKUP_POL_P0_REG) edge on P0_x. Fast wake-up from the corresponding P0_x input is enabled.	0x0

Table 404: WKUP_SEL1_GPIO_P1_REG (0x50000160)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	WKUP_SEL1_GPIO_P1 0 (level sensitive): If WKUP_SEL_GPIO_P1_REG[x] is 1, generate GPIO_P1_IRQ based on P1_x level. Fast wake-up is not enabled if the corresponding WKUP_SEL_GPIO_P1_REG[x] is 0 too. 1 (edge sensitive): If WKUP_SEL_GPIO_P1_REG[x] is 1, GPIO_P1_IRQ will be generated only on rising/falling (defined by WKUP_POL_P1_REG) edge on P1_x. Fast wake-up from the corresponding P1_x input is enabled.	0x0

35.16 Watchdog controller registers

Table 405: Register map WDOG

Address	Register	Description
0x50000700	WATCHDOG_REG	Watchdog timer register.
0x50000704	WATCHDOG_CTRL_REG	Watchdog control register.

Table 406: WATCHDOG_REG (0x50000700)

Bit	Mode	Symbol/Description	Reset
31:14	R0/W	WDOG_WEN Bit [31:14] = 0 = Write enable for Watchdog timer Else, Write disable. This filter prevents unintentional presetting the watchdog with a software run-away.	0x0
13	R/W	WDOG_VAL_NEG 0 = Watchdog timer value is positive. 1 = Watchdog timer value is negative.	0x0

12:0	R/W	WDOG_VAL
		Write: Watchdog timer reload value. Note that all bits [31-14] must be 0 to reload this register.
		Read: Actual Watchdog timer value. Decremented by 1 every ~10 ms (RCLP) or ~21 ms (RCX), that is the Watchdog timer clock tick.
		Bit 13 indicates a negative counter value. 2, 1, 0, $3FFF_{16}$, $3FFE_{16}$ and so forth. 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 =$3FF0_{16}$ -> WDOG reset -> reload $1FFF_{16}$
		If WATCHDOG_CTRL_REG[NMI_RST] = 1 then
		if WDOG_VAL <= 0 -> WDOG reset -> reload $1FFF_{16}$
		Note 1: The programmed value WDOG_VAL is updated in the (independent) Watchdog timer at the second next RCLP or RCX clock tick.
		Note 2: Select RCLP (32 kHz) or RCX (15 kHz) with CLK_RCX_REG[RCX_ENABLE] for the Watchdog clock. The RCLP is selected by default. See for more info the description of CLK_RCX_REG[RCX_ENABLE].
		Note 3: If WATCHDOG_CTRL_REG[NMI_RST] = 0, the time between the NMI generation and the WDOG reset generation is 15 Watchdog timer clock ticks.
		Note 4: The internal Watchdog timer clock tick is the Watchdog clock divided by a fixed value of 320.
			0x1FFF

Table 407: WATCHDOG_CTRL_REG (0x50000704)

Bit	Mode	Symbol/Description	Reset
31:4	R	- Reserved	0x0
3	R	WRITE_BUSY 0 = A new WATCHDOG_REG[WDOG_VAL] can be written.	0x0

Bit	Mode	Symbol/Description	Reset
		1 = No new WATCHDOG_REG[WDOG_VAL] can be written. **Note:** It takes some time before the programmed WDOG_VAL is updated in the (independent) Watchdog timer. During this time, it is not possible to write a new value to WATCHDOG_REG[WDOG_VAL].	
2	R/W	WDOG_FREEZE_EN 0 = Watchdog timer cannot be frozen when NMI_RST = 0. 1 = Watchdog timer can be frozen/resumed using SET_FREEZE_REG[FRZ_WDOG]/ RESET_FREEZE_REG[FRZ_WDOG] when NMI_RST = 0. **Note:** Although this bit is retained during sleep, the SET_FREEZE_REG[FRZ_SYS_WDOG] is not, so the watchdog cannot be frozen during CM33 sleep.	0x1
1	R/W	- Reserved	0x1
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 cannot be frozen by software. Note that this bit can only be set to 1 by software and only be reset with a WDOG (SYS) reset or software reset. The watchdog is always frozen when the Cortex-M33 is halted in DEBUG State.	0x0

35.17 General purpose/ΣΔ ADC registers

Table 408: Register map GPADC

Address	Register	Description
0x50020400	SDADC_CTRL_REG	Sigma Delta ADC Control Register
0x50020404	SDADC_PGA_CTRL_REG	Sigma Delta ADC PGA Control Registers
0x5002040C	SDADC_GAIN_CORR_REG	Sigma Delta ADC Gain Correction Register
0x50020410	SDADC_OFFS_CORR_REG	Sigma Delta ADC Offset Correction Register
0x50020414	SDADC_CLEAR_INT_REG	Sigma Delta ADC Clear Interrupt Register
0x50020418	SDADC_RESULT_REG	Sigma Delta ADC Result Register
0x5002041C	SDADC_AUDIO_FILT_REG	Sigma Delta ADC Audio Filter Register
0x50040900	GP_ADC_CTRL_REG	General Purpose ADC Control Register
0x50040904	GP_ADC_CTRL2_REG	General Purpose ADC Second Control Register
0x50040908	GP_ADC_CTRL3_REG	General Purpose ADC Third Control Register
0x5004090C	GP_ADC_SEL_REG	General Purpose ADC Input Selection Register

Address	Register	Description
0x50040910	GP_ADC_OFFP_REG	General Purpose ADC Positive Offset Register
0x50040914	GP_ADC_OFFN_REG	General Purpose ADC Negative Offset Register
0x5004091C	GP_ADC_CLEAR_INT_REG	General Purpose ADC Clear Interrupt Register
0x50040920	GP_ADC_RESULT_REG	General Purpose ADC Result Register

Table 409: SDADC_CTRL_REG (0x50020400)

Bit	Mode	Symbol/Description	Reset
20	R/W	SDADC_VREF_TO_PAD Connect internal SDADC reference voltage VREF (0.9 V) to P1[5] 0: Disconnected (default) 1: Connected	0x0
19	R/W	SDADC_MODE Select the mode of operation for the SDADC 0: Sensor mode 1: Audio mode	0x0
18	R/W	SDADC_DMA_EN 0: DMA functionality disabled 1: DMA functionality enabled	0x0
17	R/W	SDADC_MINT 0: Disable (mask) SDADC_ADC_INT. 1: Enable SDADC_ADC_INT to ICU.	0x0
16	R	SDADC_INT 1: AD conversion ready and has generated an interrupt. Must be cleared by writing any value to SDADC_CLEAR_INT_REG.	0x0
15	R	SDADC_LDO_OK 1: Internal LDO is ready for use	0x0
14:13	R/W	SDADC_VREF_SEL Select the reference input: 00: VREFN=internal VREFP=internal 01: VREFN=internal VREFP=external (pin P1[5]) 10: VREFN=external (pin P1[6]) VREFP=internal 11: VREFN=external (pin P1[6]) VREFP=external(pin P1[5])	0x0
12	R/W	SDADC_CONT 0: Manual ADC mode, a single result will be generated after setting the SDADC_START bit. 1: Continuous ADC mode, new ADC results will be constantly stored in SDADC_RESULT_REG. Still SDADC_START has to be set to start the execution. Wait for SDADC_START to become zero after clearing the SDADC_CONT bit to stop the continuous mode.	0x0
11:10	R/W	SDADC_OSR Oversample Rate in sensor mode. SDADC_MODE = 0x1 0: 256x	0x0

Bit	Mode	Symbol/Description	Reset
		1: 512x 2: 1024x 3: 2048x The OSR in audio mode is fixed: OSR = 62.5	
9	R/W	SDADC_SE 0: Differential mode 1: Single ended mode (Input selection negative side is ignored)	0x0
8:6	R/W	SDADC_INN_SEL Input selection of negative side. 0: ADC0/P1[00] 1: ADC1/P1[01] 2: ADC2/P1[02] 3: ADC3/P0[10] 4: ADC4/P1[05] 5: ADC5/P1[06] 6: ADC6/P1[09] 7: ADC7/P1[11]	0x0
5:2	R/W	SDADC_INP_SEL Input selection of positive side. 0: ADC0/P1[00] 1: ADC1/P1[01] 2: ADC2/P1[02] 3: ADC3/P0[10] 4: ADC4/P1[05] 5: ADC5/P1[06] 6: ADC6/P1[09] 7: ADC7/P1[11] 8: VBAT (via 4x attenuator, INN connected to ground)	0x0
1	R/W	SDADC_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 SDADC_INT bit will be set. It is not allowed to write this bit while it is not (yet) zero.	0x0
0	R/W	SDADC_EN 0: LDO is off and ADC is disabled. 1: LDO, bias currents and modulator are enabled.	0x0

Table 410: SDADC_PGA_CTRL_REG (0x50020404)

Bit	Mode	Symbol/Description	Reset
11:9	R/W	PGA_GAIN Select the PGA gain select: 0: -12 dB 1: -6 dB 2: 0 dB 3: 6 dB 4: 12 dB	0x0

Bit	Mode	Symbol/Description	Reset
		5: 18 dB 6: 24 dB 7: 30 dB	
8:7	R/W	PGA_SINGLE Use PGA in single ended mode 0: Differential mode (default) 1: Use N-branch as single ended mode 2: Use P-branch as single ended mode 3: Reserved (do not use)	0x0
6	R/W	PGA_MUTE Mute the PGA output 0: Unmuted (default) 1: Mute	0x0
5:3	R/W	PGA_BIAS Configure the PGA bias control: 0: 0.40 x Ibias 1: 0.44 x Ibias 2: 0.50 x Ibias 3: 0.57 x Ibias 4: 0.66 x Ibias 5: 0.80 x Ibias 6: 1.00 x Ibias 7: 1.33 x Ibias	0x4
2	R/W	PGA_SHORTIN PGA input short 0: Normal mode (default) 1: Short PGA inputs	0x0
1:0	R/W	PGA_EN PGA enable: 00: both branches of PGA disabled 01: Positive branch of PGA enabled, Negative branch disabled 10: Positive branch of PGA disabled, Negative branch enabled 11: Both branches of PGA enabled	0x0

Table 411: SDADC_GAIN_CORR_REG (0x5002040C)

Bit	Mode	Symbol/Description	Reset
9:0	R/W	SDADC_GAIN_CORR Gain adjust	0x0

Table 412: SDADC_OFFS_CORR_REG (0x50020410)

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

Bit	Mode	Symbol/Description	Reset
9:0	R/W	SDADC_OFFS_CORR Offset adjust	0x0

Table 413: SDADC_CLEAR_INT_REG (0x50020414)

Bit	Mode	Symbol/Description	Reset
31:0	R0/W	SDADC_CLR_INT Writing any value to this register will clear the ADC_INT interrupt. Reading returns 0.	0x0

Table 414: SDADC_RESULT_REG (0x50020418)

Bit	Mode	Symbol/Description	Reset
15:0	R	SDADC_VAL Returns up to 16 bits linear value of the last AD conversion. The effective resolution depends on the OSR used.	0x0

Table 415: SDADC_AUDIO_FILT_REG (0x5002041C)

Bit	Mode	Symbol/Description	Reset
20:0	R/W	SDADC_CIC_OFFSET Constant CIC offset	0x0

Table 416: GP_ADC_CTRL_REG (0x50040900)

Bit	Mode	Symbol/Description	Reset
12	R/W	DIE_TEMP_EN Enables the die-temperature sensor. Output can be measured on GPADC input 4.	0x0
11	R/W	- Reserved	0x0
10	R/W	GP_ADC_LDO_HOLD 0: GPADC LDO tracking bandgap reference 1: GPADC LDO hold sampled bandgap reference	0x0
9	R/W	GP_ADC_CHOP 0: Chopper mode off 1: Chopper mode enabled. Takes two samples with opposite GP_ADC_SIGN to cancel the internal offset voltage of the ADC; Highly recommended for DC-measurements.	0x0
8	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
7	R/W	GP_ADC_MUTE	0x0

Bit	Mode	Symbol/Description	Reset
		0: Normal operation 1: Mute ADC input. Takes sample at mid-scale (to determine the internal offset and/or noise of the ADC with regards to VDD_REF which is also sampled by the ADC).	
6	R/W	GP_ADC_SE 0: Differential mode 1: Single-ended mode	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_DMA_EN 0: DMA functionality disabled 1: DMA functionality enabled	0x0
2	R/W	GP_ADC_CONT 0: Manual ADC mode, a single result will be generated after setting the GP_ADC_START bit. 1: Continuous ADC mode, new ADC results will be constantly stored in GP_ADC_RESULT_REG. Still GP_ADC_START has to be set to start the execution. The time between conversions is configurable with GP_ADC_INTERVAL.	0x0
1	R/W	GP_ADC_START 0: ADC conversion ready. 1: If a 1 is written, the ADC starts a conversion. After the conversion this bit will be set to 0 and the GP_ADC_INT bit will be set. It is not allowed to write this bit while it is not (yet) zero.	0x0
0	R/W	GP_ADC_EN 0: LDO is off and ADC is disabled. 1: LDO is turned on and afterwards the ADC is enabled.	0x0

Table 417: GP_ADC_CTRL2_REG (0x50040904)

Bit	Mode	Symbol/Description	Reset
15:13	R/W	GP_ADC_STORE_DEL 0: Data is stored after handshake synchronization 1: Data is stored 2 ADC_CLK cycles after internal start trigger 2: Data is stored 3 ADC_CLK cycles after internal start trigger 7: Data is stored 8 ADC_CLK cycles after internal start trigger	0x0
12:9	R/W	GP_ADC_SMPL_TIME 0: The sample time (switch is closed) is two ADC_CLK cycles 1: The sample time is 1*8 ADC_CLK cycles 2: The sample time is 2*8 ADC_CLK cycles 15: The sample time is 15*8 ADC_CLK cycles	0x1
8:6	R/W	GP_ADC_CONV_NRS	0x0

Bit	Mode	Symbol/Description	Reset
		0: 1 sample is taken or 2 in case ADC_CHOP is active. 1: 2 samples are taken. 2: 4 samples are taken. 7: 128 samples are taken.	
5:3	R/W	- Reserved	0x0
2	R/W	GP_ADC_I20U 1: Adds 20 µA constant load current at the ADC LDO to minimize ripple on the reference voltage of the ADC.	0x0
1:0	R/W	GP_ADC_ATTN 0: No attenuator (input voltages up to 0.9 V allowed) 1: Enabling 2x attenuator (input voltages up to 1.8 V allowed) 2: Enabling 3x attenuator (input voltages up to 2.7 V allowed) 3: Enabling 4x attenuator (input voltages up to 3.6 V allowed) Enabling the attenuator requires a longer sampling time.	0x0

Table 418: GP_ADC_CTRL3_REG (0x50040908)

Bit	Mode	Symbol/Description	Reset
15:8	R/W	GP_ADC_INTERVAL Defines the interval between two ADC conversions in case GP_ADC_CONT is set. 0: No extra delay between two conversions. 1: 1.024 ms interval between two conversions. 2: 2.048 ms interval between two conversions. 255: 261.12 ms interval between two conversions.	0x0
7:0	R/W	GP_ADC_EN_DEL Defines the delay for enabling the ADC after enabling the LDO. 0: Not allowed 1: 4x ADC_CLK period. n: n*4x ADC_CLK period.	0x40

Table 419: GP_ADC_SEL_REG (0x5004090C)

Bit	Mode	Symbol/Description	Reset
7:4	R/W	GP_ADC_SEL_P ADC positive input selection. 0: ADC0 (P1[0]) 1: ADC1 (P1[1]) 2: ADC2 (P1[2]) 3: ADC3 (P0[10]) 4: Temperature Sensor 5: VDCDC 6: VBAT 7: VDDD 8: VSSA	0x0

Bit	Mode	Symbol/Description	Reset
		9: VDDIO A: ADC4 (P1[5]) B: ADC5 (P1[6]) C: ADC6 (P1[9]) D: ADC7 (P1[11]) E: TESTBUS[0] F: TESTBUS[1]	
3:0	R/W	GP_ADC_SEL_N ADC negative input selection. Differential only (GP_ADC_SE=0). 0: ADC0 (P1[0]) 1: ADC1 (P1[1]) 2: ADC2 (P1[2]) 3: ADC3 (P0[10]) A: ADC4 (P1[5]) B: ADC5 (P1[6]) C: ADC6 (P1[9]) D: ADC7 (P1[11]) E: TESTBUS[0] F: TESTBUS[1]	0x0

Table 420: GP_ADC_OFFP_REG (0x50040910)

Bit	Mode	Symbol/Description	Reset
9:0	R/W	GP_ADC_OFFP Offset adjust of the "positive" array of ADC-network. Effective if one of the following conditions holds: ADC_SE = 0 ADC_SE = 1 and ADC_SIGN = 0 ADC_SE = 1 and ADC_CHOP = 1	0x200

Table 421: GP_ADC_OFFN_REG (0x50040914)

Bit	Mode	Symbol/Description	Reset
9:0	R/W	GP_ADC_OFFN Offset adjust of the "negative" array of ADC-network. Effective if one of the following conditions holds: ADC_SE = 0 ADC_SE_1 and ADC_SIGN = 1 ADC_SE = 1 and ADC_CHOP = 1	0x200

Table 422: GP_ADC_CLEAR_INT_REG (0x5004091C)

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 423: GP_ADC_RESULT_REG (0x50040920)

Bit	Mode	Symbol/Description	Reset
15:0	R	GP_ADC_VAL Returns 10 up to 16 bits linear value of the last AD conversion. The upper 10 bits are always valid, the lower 6 bits are only valid in case oversampling has been applied. Two samples result in one extra bit and 64 samples result in six extra bits.	0x0

35.18 General purpose I/O registers

Table 424: Register map GPIO

Address	Register	Description
0x50020600	P0_DATA_REG	P0 Data input/output Register
0x50020604	P1_DATA_REG	P1 Data input/output Register
0x50020608	P0_SET_DATA_REG	P0 Set port pins Register
0x5002060C	P1_SET_DATA_REG	P1 Set port pins Register
0x50020610	P0_RESET_DATA_REG	P0 Reset port pins Register
0x50020614	P1_RESET_DATA_REG	P1 Reset port pins Register
0x50020618	P0_00_MODE_REG	P0_00 Mode Register
0x5002061C	P0_01_MODE_REG	P0_01 Mode Register
0x50020620	P0_02_MODE_REG	P0_02 Mode Register
0x50020624	P0_03_MODE_REG	P0_03 Mode Register
0x50020628	P0_04_MODE_REG	P0_04 Mode Register
0x5002062C	P0_05_MODE_REG	P0_05 Mode Register
0x50020630	P0_06_MODE_REG	P0_06 Mode Register
0x50020634	P0_07_MODE_REG	P0_07 Mode Register
0x50020638	P0_08_MODE_REG	P0_08 Mode Register
0x5002063C	P0_09_MODE_REG	P0_09 Mode Register
0x50020640	P0_10_MODE_REG	P0_10 Mode Register
0x50020644	P0_11_MODE_REG	P0_11 Mode Register
0x50020648	P0_12_MODE_REG	P0_12 Mode Register
0x5002064C	P0_13_MODE_REG	P0_13 Mode Register
0x50020650	P0_14_MODE_REG	P0_14 Mode Register
0x50020654	P0_15_MODE_REG	P0_15 Mode Register
0x50020658	P1_00_MODE_REG	P1_00 Mode Register
0x5002065C	P1_01_MODE_REG	P1_01 Mode Register
0x50020660	P1_02_MODE_REG	P1_02 Mode Register
0x50020664	P1_03_MODE_REG	P1_03 Mode Register
0x50020668	P1_04_MODE_REG	P1_04 Mode Register
0x5002066C	P1_05_MODE_REG	P1_05 Mode Register
0x50020670	P1_06_MODE_REG	P1_06 Mode Register

Address	Register	Description
0x50020674	P1_07_MODE_REG	P1_07 Mode Register
0x50020678	P1_08_MODE_REG	P1_08 Mode Register
0x5002067C	P1_09_MODE_REG	P1_09 Mode Register
0x50020680	P1_10_MODE_REG	P1_10 Mode Register
0x50020684	P1_11_MODE_REG	P1_11 Mode Register
0x50020688	P1_12_MODE_REG	P1_12 Mode Register
0x5002068C	P1_13_MODE_REG	P1_13 Mode Register
0x50020690	P1_14_MODE_REG	P1_14 Mode Register
0x50020694	P1_15_MODE_REG	P1_15 Mode Register
0x500206A0	GPIO_CLK_SEL_REG	Select which clock to map on ports P0/P1
0x500206A4	P0_WEAK_CTRL_REG	Weak Pads Control Register
0x500206A8	P1_WEAK_CTRL_REG	Weak Pads Control Register

Table 425: P0_DATA_REG (0x50020600)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	P0_DATA Set P0 output register when written; Returns the value of P0 port when read	0x142

Table 426: P1_DATA_REG (0x50020604)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	P1_DATA Set P1 output register when written; Returns the value of P1 port when read	0x0

Table 427: P0_SET_DATA_REG (0x50020608)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P0_SET Writing a 1 to P0[y] sets P0[y] to 1. Writing 0 is discarded; Reading returns 0	0x0

Table 428: P1_SET_DATA_REG (0x5002060C)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P1_SET Writing a 1 to P1[y] sets P1[y] to 1. Writing 0 is discarded; Reading returns 0	0x0

Table 429: P0_RESET_DATA_REG (0x50020610)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P0_RESET Writing a 1 to P0[y] sets P0[y] to 0. Writing 0 is discarded; Reading returns 0	0x0

Table 430: P1_RESET_DATA_REG (0x50020614)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P1_RESET Writing a 1 to P1[y] sets P1[y] to 0. Writing 0 is discarded; Reading returns 0	0x0

Table 431: P0_00_MODE_REG (0x50020618)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID Function of port: 0: GPIO (see also the PUPD bit-field) 1: UART_RX 2: UART_TX 3: UART2_RX 4: UART2_TX 5: UART2_CTSN 6: UART2_RTSN 7: ISO_CLK 8: ISO_DATA 9: SPI_DI 10: SPI_DO 11: SPI_CLK 12: SPI_EN 13: SPI_EN2 14: I2C_SCL 15: I2C_SDA 16: ADC (dedicated pins, see also the "Input/Output Ports" chapter of Datasheet)	0x0

Bit	Mode	Symbol/Description	Reset
		17: PCM_DI 18: PCM_DO 19: PCM_FSC 20: PCM_CLK 21: PDM_DATA 22: PDM_CLK 23: CLOCK (see also GPIO_CLK_SEL_REG for the dedicated pins mapping of supported clocks) 24: TIM_PWM 25: TIM2_PWM 26: TIM_1SHOT 27: TIM2_1SHOT 28: TIM3_PWM 29: TIM4_PWM 30: COEX_EXT_ACT 31: COEX_SMART_ACT 32: COEX_SMART_PRI 33: PORT0_DCF 34: PORT1_DCF 35: PORT2_DCF 36: PORT3_DCF 37: PORT4_DCF 38: CMAC_DIAG0 39: CMAC_DIAG1 40: CMAC_DIAG2 41: CMAC_DIAG3 42: CMAC_DIAG4 43: CMAC_DIAG5 44: CMAC_DIAG6 45: CMAC_DIAG7 46: CMAC_DIAG8 47: CMAC_DIAG9 48: CMAC_DIAG10 49: CMAC_DIAG11 50: CMAC_DIAG12 51: CMAC_DIAG13 52: CMAC_DIAG14 53: CMAC_DIAG15 54: Reserved 55: Reserved 56: Reserved 57: Reserved 58: Reserved 59: Reserved 60: Reserved 61: Reserved 62: Reserved 63: Reserved	

Table 432: P0_01_MODE_REG (0x5002061C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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	0x1
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 433: P0_02_MODE_REG (0x50020620)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 434: P0_03_MODE_REG (0x50020624)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 435: P0_04_MODE_REG (0x50020628)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 436: P0_05_MODE_REG (0x5002062C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 437: P0_06_MODE_REG (0x50020630)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	0x0
9:8	R/W	PUPD	0x1

Bit	Mode	Symbol/Description	Reset
		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	
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 438: P0_07_MODE_REG (0x50020634)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 439: P0_08_MODE_REG (0x50020638)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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.	0x1
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 440: P0_09_MODE_REG (0x5002063C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 441: P0_10_MODE_REG (0x50020640)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 442: P0_11_MODE_REG (0x50020644)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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

Bit	Mode	Symbol/Description	Reset
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 443: P0_12_MODE_REG (0x50020648)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 444: P0_13_MODE_REG (0x5002064C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 445: P0_14_MODE_REG (0x50020650)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD	0x0

Bit	Mode	Symbol/Description	Reset
		0: Push pull 1: Open drain	
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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 446: P0_15_MODE_REG (0x50020654)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 447: P1_00_MODE_REG (0x50020658)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 448: P1_01_MODE_REG (0x5002065C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 449: P1_02_MODE_REG (0x50020660)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 450: P1_03_MODE_REG (0x50020664)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	0x0
9:8	R/W	PUPD	0x2

Bit	Mode	Symbol/Description	Reset
		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	
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 451: P1_04_MODE_REG (0x50020668)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 452: P1_05_MODE_REG (0x5002066C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 453: P1_06_MODE_REG (0x50020670)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 454: P1_07_MODE_REG (0x50020674)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 455: P1_08_MODE_REG (0x50020678)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	-	0x0

Bit	Mode	Symbol/Description	Reset
		Reserved	
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 456: P1_09_MODE_REG (0x5002067C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 457: P1_10_MODE_REG (0x50020680)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 458: P1_11_MODE_REG (0x50020684)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull	0x0

Bit	Mode	Symbol/Description	Reset
		1: Open drain	
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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 459: P1_12_MODE_REG (0x50020688)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 460: P1_13_MODE_REG (0x5002068C)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID	0x0

Bit	Mode	Symbol/Description	Reset
		See P0_00_MODE_REG[PID]	

Table 461: P1_14_MODE_REG (0x50020690)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 462: P1_15_MODE_REG (0x50020694)

Bit	Mode	Symbol/Description	Reset
10	R/W	PPOD 0: Push pull 1: Open drain	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
7:6	R	- Reserved	0x0
5:0	R/W	PID See P0_00_MODE_REG[PID]	0x0

Table 463: GPIO_CLK_SEL_REG (0x500206A0)

Bit	Mode	Symbol/Description	Reset
9	R/W	DIVN_OUTPUT_EN DIVN output enable bit-field. When set, it enables the mapping of DIVN clock on dedicated GPIO (P0_08). The specific GPIO must be configured as GPIO output.	0x0
8	R/W	RC32M_OUTPUT_EN	0x0

Bit	Mode	Symbol/Description	Reset
		RC32M output enable bit-field. When set, it enables the mapping of RC32M clock on dedicated GPIO (P1_02). The specific GPIO must be configured as GPIO output.	
7	R/W	XTAL32M_OUTPUT_EN XTAL32M output enable bit-field. When set, it enables the mapping of XTAL32M clock on dedicated GPIO (P0_11). The specific GPIO must be configured as GPIO output.	0x0
6	R/W	- Reserved	0x0
5	R/W	- Reserved	0x0
4	R/W	- Reserved	0x0
3	R/W	FUNC_CLOCK_EN If set, it enables the mapping of the selected clock signal, according to FUNC_CLOCK_SEL bit-field.	0x0
2:0	R/W	FUNC_CLOCK_SEL Select which clock to map when PID = FUNC_CLOCK. 0x0: XTAL32K 0x1: RCLP 0x2: RCX 0x3: XTAL32M 0x4: RC32M 0x5: DIVN 0x6: Reserved 0x7: Reserved	0x0

Table 464: P0_WEAK_CTRL_REG (0x500206A4)

Bit	Mode	Symbol/Description	Reset
15:6	R/W	P0_LOWDRV 0 = P0_x port is driven with normal drive strength (default) 1 = P0_x port is driven with reduced drive strength	0x0

Table 465: P1_WEAK_CTRL_REG (0x500206A8)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	P1_LOWDRV 0 = P1_x port is driven with normal drive strength (default) 1 = P1_x port is driven with reduced drive strength	0x0

35.19 General purpose registers

Table 466: Register map GPREG

Address	Register	Description
0x50050300	SET_FREEZE_REG	Controls freezing of various timers/counters (incl. DMA).
0x50050304	RESET_FREEZE_REG	Controls unfreezing of various timers/counters (incl. DMA).
0x50050308	DEBUG_REG	Various debug information register.
0x5005030C	GP_STATUS_REG	General purpose system status register.
0x50050314	SCPU_FCU_TAG_REG	SysCPU FCU tag register.

Table 467: SET_FREEZE_REG (0x50050300)

Bit	Mode	Symbol/Description	Reset
31:11	R	- Reserved	0x0
10	R/W	FRZ_CMAC_WDOG If 1, the CMAC software Watchdog Timer is frozen, 0 is discarded.	0x0
9	R/W	FRZ_SWTIM4 If 1, the SW Timer4 is frozen, 0 is discarded.	0x0
8	R/W	FRZ_SWTIM3 If 1, the SW Timer3 is frozen, 0 is discarded.	0x0
7	R/W	- Reserved	0x0
6	R/W	FRZ_SWTIM2 If 1, the SW Timer2 is frozen, 0 is discarded.	0x0
5	R/W	FRZ_DMA If 1, the DMA is frozen, 0 is discarded.	0x0
4	R/W	- Reserved	0x0
3	R/W	FRZ_SYS_WDOG If 1, the SYS SW 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_RESERVED	0x0
1	R/W	FRZ_SWTIM If 1, the SW Timer is frozen, 0 is discarded.	0x0
0	R/W	FRZ_WKUPTIM If 1, the Wake-Up Timer is frozen, 0 is discarded.	0x0

Table 468: RESET_FREEZE_REG (0x50050304)

Bit	Mode	Symbol/Description	Reset
31:11	R	- Reserved	0x0
10	R/W	FRZ_CMAC_WDOG If 1, the CMAC SW Watchdog Timer continues, 0 is discarded.	0x0
9	R/W	FRZ_SWTIM4 If 1, the SW Timer4 continues, 0 is discarded.	0x0
8	R/W	FRZ_SWTIM3 If 1, the SW Timer3 continues, 0 is discarded.	0x0
7	R/W	- Reserved	0x0
6	R/W	FRZ_SWTIM2 If 1, the SW Timer2 continues, 0 is discarded.	0x0
5	R/W	FRZ_DMA If 1, the DMA continues, 0 is discarded.	0x0
4	R/W	- Reserved	0x0
3	R/W	FRZ_SYS_WDOG If 1, the SYS SW Watchdog Timer continues, 0 is discarded.	0x0
2	R/W	FRZ_RESERVED	0x0
1	R/W	FRZ_SWTIM If 1, the SW Timer continues, 0 is discarded.	0x0
0	R/W	FRZ_WKUPTIM If 1, the Wake Up Timer continues, 0 is discarded.	0x0

Table 469: DEBUG_REG (0x50050308)

Bit	Mode	Symbol/Description	Reset
31:16	R	- Reserved	0x0
15:10	R	- Reserved	0x0
9	R/W	- Reserved	0x0
8	R/W	CROSS_CPU_HALT_SENSITIVITY Select the cross CPU halt sensitivity. 0: Level triggered, 1: Pulse triggered. Note: This bit is retained.	0x1

Bit	Mode	Symbol/Description	Reset
7	R/W	SYS_CPUWAIT_ON_JTAG 1: Stall the processor core out of reset (**only after a wake-up from JTAG**). Debugger access continues when the core is stalled. When set to 0 again, the core resumes instruction execution. This feature is independent of the PDC (Power Domain Controller) settings. If this bit is set and there is SW/JTAG activity during deep sleep, the SYS CPU is stalled after the wake-up. Note: This bit is retained.	0x0
6	R/W	SYS_CPUWAIT 1: Stall the processor core out of reset (**always after a wake-up**). Debugger access continues when the core is stalled. When set to 0 again, the core resumes instruction execution. Note: This bit is retained.	0x0
5	R	CMAC_CPU_IS_HALTED 1: CMAC CPU is halted.	0x0
4	R	SYS_CPU_IS_HALTED 1: SYS CPU (Arm CM33) is halted.	0x0
3	R/W	HALT_CMAC_SYS_CPU_EN 1: Enable CMAC CPU halting to the SYS CPU (Arm CM33). Note 1: This bit is retained. Note 2: Set this bit to 0 before going into deep sleep to prevent unpredictable halting behavior after waking up.	0x0
2	R/W	HALT_SYS_CMAC_CPU_EN 1: Enable SYS CPU (Arm CM33) halting to the CMAC CPU. Note 1: This bit is retained. Note 2: Set this bit to 0 before going into deep sleep to prevent unpredictable halting behavior after waking up.	0x0
1	R/W	CMAC_CPU_FREEZE_EN 1: Enable Freezing on-chip peripherals (see Note 2) by the CMAC CPU. Note 1: This bit is retained. Note 2: See [RE]SET_FREEZE_REG for the specific on-chip peripherals.	0x0
0	R/W	SYS_CPU_FREEZE_EN 1: Enable Freezing on-chip peripherals (see Note 2) by the SYS CPU (Arm CM33). Default 1, freezing of the on-chip peripherals is enabled when the Cortex-M33 is halted in DEBUG State. If 0, freezing of the on-chip peripherals is only depending on [RE]SET_FREEZE_REG except the system watchdog timer. The system watchdog timer is always frozen when the Cortex-M33 is halted in the DEBUG state. Note 1: This bit is retained. Note 2: See [RE]SET_FREEZE_REG for the specific on-chip peripherals.	0x1

Table 470: GP_STATUS_REG (0x5005030C)

Bit	Mode	Symbol/Description	Reset
31:2	R	- Reserved	0x0

Bit	Mode	Symbol/Description	Reset
1	R/W	- Reserved	0x0
0	R/W	CAL_PHASE If 1, it designates that the chip is in Calibration Phase, that means the Information Page (IP) has been initially programmed but no Calibration has occurred.	0x0

Table 471: SCPU_FCU_TAG_REG (0x50050314)

Bit	Mode	Symbol/Description	Reset
31:2	R	- Reserved	0x0
1	R/W	SCPU_FCU_TAG_ALL_TRANS 0: Tag **only FCU/eFlash instruction fetches** from CPUC (default). 1: Tag **all FCU/eFlash transactions** from CPUC. Note 1: The (CM33) CPUS transactions are not tagged. Note 2: This bit is retained.	0x0
0	R/W	SCPU_FCU_TAG_EN For activation of FCU buffering by the SYS CPU (Arm CM33): 0: Disable the tagging of transfers as bufferable (default). 1: Enable the tagging of transfers as bufferable. Note: This bit is retained.	0x0

35.20 Power domain controller registers

Table 472: Register map PDC

Address	Register	Description
0x50000200	PDC_CTRL0_REG	PDC control register
0x50000204	PDC_CTRL1_REG	PDC control register
0x50000208	PDC_CTRL2_REG	PDC control register
0x5000020C	PDC_CTRL3_REG	PDC control register
0x50000210	PDC_CTRL4_REG	PDC control register
0x50000214	PDC_CTRL5_REG	PDC control register
0x50000218	PDC_CTRL6_REG	PDC control register
0x5000021C	PDC_CTRL7_REG	PDC control register
0x50000220	PDC_CTRL8_REG	PDC control register
0x50000224	PDC_CTRL9_REG	PDC control register
0x50000228	PDC_CTRL10_REG	PDC control register
0x5000022C	PDC_CTRL11_REG	PDC control register
0x50000280	PDC_ACKNOWLEDGE_REG	Clear a pending PDC bit
0x50000284	PDC_PENDING_REG	Shows any pending wake-up event

Address	Register	Description
0x5000028C	PDC_PENDING_CM33_REG	Shows any pending IRQ to CM33
0x50000290	PDC_PENDING_CMAC_REG	Shows any pending IRQ to CMAC
0x50000294	PDC_SET_PENDING_REG	Set a pending PDC bit
0x50000298	PDC_CONFIG_REG	PDC configuration register

Table 473: PDC_CTRL0_REG (0x50000200)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID Selects which individual bit from the selected bank is used for wake-up. For the peripheral banks, selected with TRIG_SELECT = 0x2 or 0x3, only the lower 4 bits are considered.	0x0
1:0	R/W	TRIG_SELECT Selects which bank is used as wake-up trigger When TRIG_SELECT is 0x0, selects GPIO port0 through the WAKEUP block. When TRIG_SELECT is 0x1, selects GPIO port1 through the WAKEUP block. When TRIG_SELECT is 0x2 or 0x3, selects the peripheral IRQ. Peripheral IRQ table: 0x0: Timer 0x1: Timer2 0x2: Timer3 0x3: Timer4 0x4: RTC Alarm/Rollover 0x5: RTC Timer 0x6: CMAC Timer OR wake-up from CMAC debugger 0x7: Reserved 0x8: XTAL32MRDY_IRQ	0x0

Bit	Mode	Symbol/Description	Reset
		0x9: RFDIAG_IRQ 0xA: CMAC2SYS_IRQ OR JTAG present OR Debounced IO 0xB: Reserved 0xC: FCU 0xD: Fast wake-up 0xE: Reserved 0xF: Software trigger only	

Table 474: PDC_CTRL1_REG (0x50000204)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 475: PDC_CTRL2_REG (0x50000208)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM IIf set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR	0x0

Bit	Mode	Symbol/Description	Reset
		If set, enables PD_TMR	
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 476: PDC_CTRL3_REG (0x5000020C)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 477: PDC_CTRL4_REG (0x50000210)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER	0x0

Bit	Mode	Symbol/Description	Reset
		If set, enables PD_PER	
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 478: PDC_CTRL5_REG (0x50000214)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 479: PDC_CTRL6_REG (0x50000218)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM	0x0

Bit	Mode	Symbol/Description	Reset
		If set, enables PD_COM for GPIO access.	
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 480: PDC_CTRL7_REG (0x5000021C)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 481: PDC_CTRL8_REG (0x50000220)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered	0x0

Bit	Mode	Symbol/Description	Reset
		0x3: Reserved	
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 482: PDC_CTRL9_REG (0x50000224)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 483: PDC_CTRL10_REG (0x50000228)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled.	0x0

Bit	Mode	Symbol/Description	Reset
		0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 484: PDC_CTRL11_REG (0x5000022C)

Bit	Mode	Symbol/Description	Reset
12:11	R/W	PDC_MASTER Chooses which master is triggered when waking up 0x0: entry is disabled. 0x1: PD_SYS is woken up and CM33 is triggered 0x2: PD_RAD is woken up and CMAC is triggered 0x3: Reserved	0x0
10	R/W	EN_COM If set, enables PD_COM for GPIO access.	0x0
9	R/W	EN_PER If set, enables PD_PER	0x0
8	R/W	EN_TMR If set, enables PD_TMR	0x0
7	R/W	EN_XTAL If set, the XTAL32M will be started	0x0
6:2	R/W	TRIG_ID For description, see PDC_CTRL0_REG.TRIG_ID	0x0
1:0	R/W	TRIG_SELECT For description, see PDC_CTRL0_REG.TRIG_SELECT	0x0

Table 485: PDC_ACKNOWLEDGE_REG (0x50000280)

Bit	Mode	Symbol/Description	Reset
4:0	W	PDC_ACKNOWLEDGE	0x0

Bit	Mode	Symbol/Description	Reset
		Writing to this field acknowledges the PDC IRQ request. The data controls which request is acknowledged	

Table 486: PDC_PENDING_REG (0x50000284)

Bit	Mode	Symbol/Description	Reset
11:0	R	PDC_PENDING Indicates which IRQ ids are pending	0x0

Table 487: PDC_PENDING_CM33_REG (0x5000028C)

Bit	Mode	Symbol/Description	Reset
11:0	R	PDC_PENDING Indicates which IRQ ids are pending towards the CM33	0x0

Table 488: PDC_PENDING_CMAC_REG (0x50000290)

Bit	Mode	Symbol/Description	Reset
11:0	R	PDC_PENDING Indicates which IRQ ids are pending towards the CMAC	0x0

Table 489: PDC_SET_PENDING_REG (0x50000294)

Bit	Mode	Symbol/Description	Reset
4:0	W	PDC_SET_PENDING Writing to this field sets the PDC wake-up request and IRQ. The data controls which request is acknowledged	0x0

Table 490: PDC_CONFIG_REG (0x50000298)

Bit	Mode	Symbol/Description	Reset
2	R/W	TRIG_SELECT_CONFIG 0: All triggers to all PDCs are let through 1: All triggers apart from FCU (0xC) are masked until the FCU is ready or powered off	0x0
1	R/W	PD_RAD_WKUP_CONFIG 0: PD_RAD is woken up without delay 1: PD_RAD is woken up only when the FCU is ready or powered off	0x0
0	R/W	PD_SYS_WKUP_CONFIG 0: PD_SYS is woken up without delay 1: PD_SYS is woken up only when the FCU is ready or powered off	0x0

35.21 DMA controller registers

Table 491: Register map DMA

Address	Register	Description
0x50060200	DMA0_A_START_REG	Source address register of DMA channel 0
0x50060204	DMA0_B_START_REG	Destination address register of DMA channel 0
0x50060208	DMA0_INT_REG	Interrupt length register of DMA channel 0
0x5006020C	DMA0_LEN_REG	Transfer length register of DMA channel 0
0x50060210	DMA0_CTRL_REG	Control register of DMA channel 0
0x50060214	DMA0_IDX_REG	Index pointer register of DMA channel 0
0x50060220	DMA1_A_START_REG	Source address register of DMA channel 1
0x50060224	DMA1_B_START_REG	Destination address register of DMA channel 1
0x50060228	DMA1_INT_REG	Interrupt length register of DMA channel 1
0x5006022C	DMA1_LEN_REG	Transfer length register of DMA channel 1
0x50060230	DMA1_CTRL_REG	Control register of DMA channel 1
0x50060234	DMA1_IDX_REG	Index pointer register of DMA channel 1
0x50060240	DMA2_A_START_REG	Source address register of DMA channel 2
0x50060244	DMA2_B_START_REG	Destination address register of DMA channel 2
0x50060248	DMA2_INT_REG	Interrupt length register of DMA channel 2
0x5006024C	DMA2_LEN_REG	Transfer length register of DMA channel 2
0x50060250	DMA2_CTRL_REG	Control register of DMA channel 2
0x50060254	DMA2_IDX_REG	Index pointer register of DMA channel 2
0x50060260	DMA3_A_START_REG	Source address register of DMA channel 3
0x50060264	DMA3_B_START_REG	Destination address register of DMA channel 3
0x50060268	DMA3_INT_REG	Interrupt length register of DMA channel 3
0x5006026C	DMA3_LEN_REG	Transfer length register of DMA channel 3
0x50060270	DMA3_CTRL_REG	Control register of DMA channel 3
0x50060274	DMA3_IDX_REG	Index pointer register of DMA channel 3
0x50060280	DMA4_A_START_REG	Source address register of DMA channel 4
0x50060284	DMA4_B_START_REG	Destination address register of DMA channel 4
0x50060288	DMA4_INT_REG	Interrupt length register of DMA channel 4
0x5006028C	DMA4_LEN_REG	Transfer length register of DMA channel 4
0x50060290	DMA4_CTRL_REG	Control register of DMA channel 4
0x50060294	DMA4_IDX_REG	Index pointer register of DMA channel 4

Address	Register	Description
0x500602A0	DMA5_A_START_REG	Source address register of DMA channel 5
0x500602A4	DMA5_B_START_REG	Destination address register of DMA channel 5
0x500602A8	DMA5_INT_REG	Interrupt length register of DMA channel 5
0x500602AC	DMA5_LEN_REG	Transfer length register of DMA channel 5
0x500602B0	DMA5_CTRL_REG	Control register of DMA channel 5
0x500602B4	DMA5_IDX_REG	Index pointer register of DMA channel 5
0x50060300	DMA_REQ_MUX_REG	DMA channels peripherals mapping register
0x50060304	DMA_INT_STATUS_REG	DMA Interrupt status register
0x50060308	DMA_CLEAR_INT_REG	DMA Interrupt clear register
0x5006030C	DMA_INT_MASK_REG	DMA Interrupt mask register
0x50060310	DMA_SET_INT_MASK_REG	DMA Set Interrupt mask register
0x50060314	DMA_RESET_INT_MASK_REG	DMA Reset Interrupt mask register

Table 492: DMA0_A_START_REG (0x50060200)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA0_A_START Source start address of channel 0	0x0

Table 493: DMA0_B_START_REG (0x50060204)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA0_B_START Destination start address of channel 0	0x0

Table 494: DMA0_INT_REG (0x50060208)

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 and only if DMA0_INT_REG has reached DMA0_IDX_REG and before DMA0_IDX_REG is incremented. The interrupt enable bit of this channel must be already enabled to let the channel's controller generate the interrupt (see also DMA_INT_MASK_REG and the SET/RESET interrupt registers).	0x0

Table 495: DMA0_LEN_REG (0x5006020C)

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

Table 496: DMA0_CTRL_REG (0x50060210)

Bit	Mode	Symbol/Description	Reset
16	R/W	DMA_EXCLUSIVE_ACCESS 0: DMA channel deasserts the bus request upon completion of the write transfer (burst or single-shot) 1: DMA channel keeps on requesting the bus upon completion of the write. This is effective only in Memory-to-Memory transfers (DREQ_MODE = 0) and results into requesting the bus continuously during the whole transfer to speed up its completion (default).	0x1
15	R/W	BUS_ERROR_DETECT 0: Ignores bus error response from the AHB bus, so DMA continues normally. 1: Detects the bus response and tracks any bus error that may occur during the transfer. If a bus error is detected, the channel completes the current read-write DMA cycle (either in burst or single transfers mode) and then closes the transfer, de-asserting DMA_ON bit automatically. It is noted that the respective bus error detection status bit of DMA_INT_STATUS_REG is automatically cleared as soon as the channel is switched on again to perform a new transfer.	0x1
14:13	R/W	BURST_MODE Enables the DMA read/write bursts according to the following configuration: 00: Bursts are disabled 01: Bursts of 4 are enabled 10: Bursts of 8 are enabled 11: Reserved	0x0
12	R/W	REQ_SENSE 0: DMA operates with level-sensitive peripheral requests (default) 1: DMA operates with (positive) edge-sensitive peripheral requests	0x0
11	R/W	DMA_INIT 0: DMA performs copy A1 to B1, A2 to B2, and so forth... 1: DMA performs copy of A1 to B1, B2, and so forth... 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
10	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 or DMA_EXCLUSIVE_ACCESS = 1, DMA_IDLE is don't care.	0x0

Bit	Mode	Symbol/Description	Reset
9:7	R/W	DMA_PRIO The priority level determines which DMA channel will be granted access for transferring data, in case more than one channel is 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
6	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
5	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
4	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
3	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
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. Note: If DMA_ON is disabled by software while the DMA channel is active, it cannot be enabled again until the channel has completed the last on-going read-write cycle and has stopped. Thus, the software has to check that the reading of DMA0_CTRL_REG.DMA_ON returns 0 before setting again the specific bit-field.	0x0

Table 497: DMA0_IDX_REG (0x50060214)

Bit	Mode	Symbol/Description	Reset
15:0	R	DMA0_IDX This (read-only) register determines the data items already transferred by the DMA channel. Hence, if its value is 1, then the DMA channel has already copied one data item and it is currently performing the next copy. If its value is 2, then two items have already been copied, and so on. When the transfer is completed (so when DMA0_CTRL_REG[DMA_ON] has been cleared) and DMA0_CTRL_REG[CIRCULAR] is not set, the register keeps its (last) value (which should be equal to DMA0_LEN_REG) and it is automatically reset to 0 upon starting a new transfer. In Circular mode, the register is automatically initialized to 0 as soon as the DMA channel starts-over again.	0x0

Table 498: DMA1_A_START_REG (0x50060220)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA1_A_START Source start address of channel 1	0x0

Table 499: DMA1_B_START_REG (0x50060224)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA1_B_START Destination start address of channel 1	0x0

Table 500: DMA1_INT_REG (0x50060228)

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 and only if DMA1_INT_REG has reached DMA1_IDX_REG and before DMA1_IDX_REG is incremented. The interrupt enable bit of this channel must be already enabled, to let the channel's controller generate the interrupt (see also DMA_INT_MASK_REG and the SET/RESET interrupt registers).	0x0

Table 501: DMA1_LEN_REG (0x5006022C)

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

Table 502: DMA1_CTRL_REG (0x50060230)

Bit	Mode	Symbol/Description	Reset
16	R/W	DMA_EXCLUSIVE_ACCESS	0x1

Bit	Mode	Symbol/Description	Reset
		0: DMA channel deasserts the bus request upon completion of the write transfer (burst or single-shot) 1: DMA channel keeps on requesting the bus upon completion of the write. This is effective only in Memory-to-Memory transfers (DREQ_MODE = 0) and results into requesting the bus continuously during the whole transfer to speed up its completion (default).	
15	R/W	BUS_ERROR_DETECT 0: Ignores bus error response from the AHB bus, so DMA continues normally. 1: Detects the bus response and tracks any bus error that may occur during the transfer. If a bus error is detected, the channel completes the current read-write DMA cycle (either in burst or single transfers mode) and then closes the transfer, de-asserting DMA_ON bit automatically. It is noted that the respective bus error detection status bit of DMA_INT_STATUS_REG is automatically cleared as soon as the channel is switched on again to perform a new transfer.	0x1
14:13	R/W	BURST_MODE Enables the DMA read/write bursts according to the following configuration: 00: Bursts are disabled 01: Bursts of 4 are enabled 10: Bursts of 8 are enabled 11: Reserved	0x0
12	R/W	REQ_SENSE 0: DMA operates with level-sensitive peripheral requests (default) 1: DMA operates with (positive) edge-sensitive peripheral requests	0x0
11	R/W	DMA_INIT 0: DMA performs copy A1 to B1, A2 to B2, and so forth... 1: DMA performs copy of A1 to B1, B2, and so forth... 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
10	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 or DMA_EXCLUSIVE_ACCESS = 1, DMA_IDLE is don't care.	0x0
9:7	R/W	DMA_PRIO The priority level determines which DMA channel will be granted access for transferring data, in case more than one channel is 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
6	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
5	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
4	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
3	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
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. NOTE: If DMA_ON is disabled by software while the DMA channel is active, it cannot be enabled again until the channel has completed the last on-going read-write cycle and has stopped. Thus, the software has to check that the reading of DMA1_CTRL_REG.DMA_ON returns 0 before setting again the specific bit-field.	0x0

Table 503: DMA1_IDX_REG (0x50060234)

Bit	Mode	Symbol/Description	Reset
15:0	R	DMA1_IDX This (read-only) register determines the data items already transferred by the DMA channel. Hence, if its value is 1, then the DMA channel has already copied one data item and it is currently performing the next copy. If its value is 2, then two items have already been copied, and so on. When the transfer is completed (so when DMA1_CTRL_REG[DMA_ON] has been cleared) and DMA1_CTRL_REG[CIRCULAR] is not set, the register keeps its (last) value (which should be equal to	0x0

Bit	Mode	Symbol/Description	Reset
		DMA1_LEN_REG) and it is automatically reset to 0 upon starting a new transfer. In Circular mode, the register is automatically initialized to 0 as soon as the DMA channel starts over again.	

Table 504: DMA2_A_START_REG (0x50060240)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA2_A_START Source start address of channel 2	0x0

Table 505: DMA2_B_START_REG (0x50060244)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA2_B_START Destination start address of channel 2	0x0

Table 506: DMA2_INT_REG (0x50060248)

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 and only if DMA2_INT_REG has reached DMA2_IDX_REG and before DMA2_IDX_REG is incremented. The interrupt enable bit of this channel must be already enabled to let the channel's controller generate the interrupt (see also DMA_INT_MASK_REG and the SET/RESET interrupt registers).	0x0

Table 507: DMA2_LEN_REG (0x5006024C)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	DMA2_LEN DMA channel's transfer length. DMA2_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 508: DMA2_CTRL_REG (0x50060250)

Bit	Mode	Symbol/Description	Reset
16	R/W	DMA_EXCLUSIVE_ACCESS 0: DMA channel deasserts the bus request upon completion of the write transfer (burst or single-shot) 1: DMA channel keeps on requesting the bus upon completion of the write. This is effective only in Memory-to-Memory transfers (DREQ_MODE = 0) and results into requesting the bus continuously during the whole transfer to speed up its completion (default).	0x1
15	R/W	BUS_ERROR_DETECT 0: Ignores bus error response from the AHB bus, so DMA continues normally.	0x1

Bit	Mode	Symbol/Description	Reset
		1: Detects the bus response and tracks any bus error that may occur during the transfer. If a bus error is detected, the channel completes the current read-write DMA cycle (either in burst or single transfers mode) and then closes the transfer, de-asserting DMA_ON bit automatically. It is noted that the respective bus error detection status bit of DMA_INT_STATUS_REG is automatically cleared as soon as the channel is switched on again to perform a new transfer.	
14:13	R/W	BURST_MODE Enables the DMA read/write bursts according to the following configuration: 00: Bursts are disabled 01: Bursts of 4 are enabled 10: Bursts of 8 are enabled 11: Reserved	0x0
12	R/W	REQ_SENSE 0: DMA operates with level-sensitive peripheral requests (default) 1: DMA operates with (positive) edge-sensitive peripheral requests	0x0
11	R/W	DMA_INIT 0: DMA performs copy A1 to B1, A2 to B2, and so forth... 1: DMA performs copy of A1 to B1, B2, and so forth... 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
10	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 or DMA_EXCLUSIVE_ACCESS = 1, DMA_IDLE is don't care.	0x0
9:7	R/W	DMA_PRIO The priority level determines which DMA channel will be granted access for transferring data, in case more than one channel is 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
6	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
5	R/W	AINC Enable increment of destination address.	0x0

Bit	Mode	Symbol/Description	Reset
		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)	
4	R/W	BINC Enable increment of destination address 0: Do not increment 1: Increment according to value of BW	0x0
3	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
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. Note: If DMA_ON is disabled by software while the DMA channel is active, it cannot be enabled again until the channel has completed the last on-going read-write cycle and has stopped. Thus, the software has to check that the reading of DMA2_CTRL_REG.DMA_ON returns 0 before setting again the specific bit-field.	0x0

Table 509: DMA2_IDX_REG (0x50060254)

Bit	Mode	Symbol/Description	Reset
15:0	R	DMA2_IDX This (read-only) register determines the data items already transferred by the DMA channel. Hence, if its value is 1, then the DMA channel has already copied one data item and it is currently performing the next copy. If its value is 2, then two items have already been copied, and so on. When the transfer is completed (so when DMA2_CTRL_REG[DMA_ON] has been cleared) and DMA2_CTRL_REG[CIRCULAR] is not set, the register keeps its (last) value (which should be equal to DMA2_LEN_REG) and it is automatically reset to 0 upon starting a new transfer. In Circular mode, the register is automatically initialized to 0 as soon as the DMA channel starts over again.	0x0

Table 510: DMA3_A_START_REG (0x50060260)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA3_A_START Source start address of channel 3	0x0

Table 511: DMA3_B_START_REG (0x50060264)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA3_B_START Destination start address of channel 3	0x0

Table 512: DMA3_INT_REG (0x50060268)

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 and only if DMA3_INT_REG has reached DMA3_IDX_REG and before DMA3_IDX_REG is incremented. The interrupt enable bit of this channel must be already enabled, to let the channel's controller generate the interrupt (see also DMA_INT_MASK_REG and the SET/RESET interrupt registers).	0x0

Table 513: DMA3_LEN_REG (0x5006026C)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	DMA3_LEN DMA channel's transfer length. DMA3_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 514: DMA3_CTRL_REG (0x50060270)

Bit	Mode	Symbol/Description	Reset
16	R/W	DMA_EXCLUSIVE_ACCESS 0: DMA channel deasserts the bus request upon completion of the write transfer (burst or single-shot) 1: DMA channel keeps on requesting the bus upon completion of the write. This is effective only in Memory-to-Memory transfers (DREQ_MODE = 0) and results into requesting the bus continuously during the whole transfer to speed up its completion (default).	0x1
15	R/W	BUS_ERROR_DETECT 0: Ignores bus error response from the AHB bus, so DMA continues normally. 1: Detects the bus response and tracks any bus error that may occur during the transfer. If a bus error is detected, the channel completes the current read-write DMA cycle (either in burst or single transfers mode) and then closes the transfer, de-asserting DMA_ON bit automatically. It is noted that the respective bus error detection status bit of DMA_INT_STATUS_REG is automatically cleared as soon as the channel is switched on again to perform a new transfer.	0x1
14:13	R/W	BURST_MODE Enables the DMA read/write bursts according to the following configuration: 00: Bursts are disabled 01: Bursts of 4 are enabled 10: Bursts of 8 are enabled 11: Reserved	0x0

Bit	Mode	Symbol/Description	Reset
12	R/W	REQ_SENSE 0: DMA operates with level-sensitive peripheral requests (default) 1: DMA operates with (positive) edge-sensitive peripheral requests	0x0
11	R/W	DMA_INIT 0: DMA performs copy A1 to B1, A2 to B2, and so forth... 1: DMA performs copy of A1 to B1, B2, and so forth... 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
10	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 or DMA_EXCLUSIVE_ACCESS = 1, DMA_IDLE is don't care.	0x0
9:7	R/W	DMA_PRIO The priority level determines which DMA channel will be granted access for transferring data, in case more than one channel is 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
6	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
5	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
4	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
3	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

Bit	Mode	Symbol/Description	Reset
2:1	R/W	BW	0x0
		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	
0	R/W	DMA_ON	0x0
		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. Note: If DMA_ON is disabled by software while the DMA channel is active, it cannot be enabled again until the channel has completed the last on-going read-write cycle and has stopped. Thus, the software has to check that the reading of DMA3_CTRL_REG.DMA_ON returns 0 before setting again the specific bit-field.	

Table 515: DMA3_IDX_REG (0x50060274)

Bit	Mode	Symbol/Description	Reset
15:0	R	DMA3_IDX	0x0
		This (read-only) register determines the data items already transferred by the DMA channel. Hence, if its value is 1, then the DMA channel has already copied one data item and it is currently performing the next copy. If its value is 2, then two items have already been copied, and so on. When the transfer is completed (so when DMA3_CTRL_REG[DMA_ON] has been cleared) and DMA3_CTRL_REG[CIRCULAR] is not set, the register keeps its (last) value (which should be equal to DMA3_LEN_REG) and it is automatically reset to 0 upon starting a new transfer. In Circular mode, the register is automatically initialized to 0 as soon as the DMA channel starts over again.	

Table 516: DMA4_A_START_REG (0x50060280)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA4_A_START	0x0
		Source start address of channel 4	

Table 517: DMA4_B_START_REG (0x50060284)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA4_B_START	0x0
		Destination start address of channel 4	

Table 518: DMA4_INT_REG (0x50060288)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	DMA4_INT	0x0

Bit	Mode	Symbol/Description	Reset
		Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if and only if DMA4_INT_REG has reached DMA4_IDX_REG and before DMA4_IDX_REG is incremented. The interrupt enable bit of this channel must be already enabled to let the channel's controller generate the interrupt (see also DMA_INT_MASK_REG and the SET/RESET interrupt registers).	

Table 519: DMA4_LEN_REG (0x5006028C)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	DMA4_LEN DMA channel's transfer length. DMA4_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 520: DMA4_CTRL_REG (0x50060290)

Bit	Mode	Symbol/Description	Reset
16	R/W	DMA_EXCLUSIVE_ACCESS 0: DMA channel deasserts the bus request upon completion of the write transfer (burst or single-shot) 1: DMA channel keeps on requesting the bus upon completion of the write. This is effective only in Memory-to-Memory transfers (DREQ_MODE = 0) and results into requesting the bus continuously during the whole transfer to speed up its completion (default).	0x1
15	R/W	BUS_ERROR_DETECT 0: Ignores bus error response from the AHB bus, so DMA continues normally. 1: Detects the bus response and tracks any bus error that may occur during the transfer. If a bus error is detected, the channel completes the current read-write DMA cycle (either in burst or single transfers mode) and then closes the transfer, de-asserting DMA_ON bit automatically. It is noted that the respective bus error detection status bit of DMA_INT_STATUS_REG is automatically cleared as soon as the channel is switched on again to perform a new transfer.	0x1
14:13	R/W	BURST_MODE Enables the DMA read/write bursts according to the following configuration: 00: Bursts are disabled 01: Bursts of 4 are enabled 10: Bursts of 8 are enabled 11: Reserved	0x0
12	R/W	REQ_SENSE 0: DMA operates with level-sensitive peripheral requests (default) 1: DMA operates with (positive) edge-sensitive peripheral requests	0x0
11	R/W	DMA_INIT 0: DMA performs copy A1 to B1, A2 to B2, and so forth... 1: DMA performs copy of A1 to B1, B2, and so forth... 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	0x0

Bit	Mode	Symbol/Description	Reset
		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.	
10	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 or DMA_EXCLUSIVE_ACCESS = 1, DMA_IDLE is don't care.	0x0
9:7	R/W	DMA_PRIO The priority level determines which DMA channel will be granted access for transferring data, in case more than one channel is 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
6	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
5	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
4	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
3	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
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	0x0

Bit	Mode	Symbol/Description	Reset
		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. Note: If DMA_ON is disabled by software while the DMA channel is active, it cannot be enabled again until the channel has completed the last on-going read-write cycle and has stopped. Thus, the software has to check that the reading of DMA4_CTRL_REG.DMA_ON returns 0 before setting again the specific bit-field.	

Table 521: DMA4_IDX_REG (0x50060294)

Bit	Mode	Symbol/Description	Reset
15:0	R	DMA4_IDX This (read-only) register determines the data items already transferred by the DMA channel. Hence, if its value is 1, then the DMA channel has already copied one data item and it is currently performing the next copy. If its value is 2, then two items have already been copied, and so on. When the transfer is completed (so when DMA4_CTRL_REG[DMA_ON] has been cleared) and DMA4_CTRL_REG[CIRCULAR] is not set, the register keeps its (last) value (which should be equal to DMA4_LEN_REG) and it is automatically reset to 0 upon starting a new transfer. In Circular mode, the register is automatically initialized to 0 as soon as the DMA channel starts over again.	0x0

Table 522: DMA5_A_START_REG (0x500602A0)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA5_A_START Source start address of channel 5	0x0

Table 523: DMA5_B_START_REG (0x500602A4)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	DMA5_B_START Destination start address of channel 5	0x0

Table 524: DMA5_INT_REG (0x500602A8)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	DMA5_INT Number of transfers until an interrupt is generated. The interrupt is generated after a transfer, if and only if DMA5_INT_REG has reached DMA5_IDX_REG and before DMA5_IDX_REG is incremented. The interrupt enable bit of this channel must be already enabled, to let the channel's controller generate the interrupt (see also DMA_INT_MASK_REG and the SET/RESET interrupt registers).	0x0

Table 525: DMA5_LEN_REG (0x500602AC)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	DMA5_LEN DMA channel's transfer length. DMA5_LEN of value 0, 1, 2, ... results into an actual transfer length of 1, 2, 3, ...	0x0

Table 526: DMA5_CTRL_REG (0x500602B0)

Bit	Mode	Symbol/Description	Reset
16	R/W	DMA_EXCLUSIVE_ACCESS 0: DMA channel deasserts the bus request upon completion of the write transfer (burst or single-shot) 1: DMA channel keeps on requesting the bus upon completion of the write. This is effective only in Memory-to-Memory transfers (DREQ_MODE = 0) and results into requesting the bus continuously during the whole transfer to speed up its completion (default).	0x1
15	R/W	BUS_ERROR_DETECT 0: Ignores bus error response from the AHB bus, so DMA continues normally. 1: Detects the bus response and tracks any bus error that may occur during the transfer. If a bus error is detected, the channel completes the current read-write DMA cycle (either in burst or single transfers mode) and then closes the transfer, de-asserting DMA_ON bit automatically. It is noted that the respective bus error detection status bit of DMA_INT_STATUS_REG is automatically cleared as soon as the channel is switched on again to perform a new transfer. NOTE: This bit-field is overruled to 1 when channel 5 is configured as trusted channel (in Secure Boot mode).	0x1
14:13	R/W	BURST_MODE Enables the DMA read/write bursts according to the following configuration: 00: Bursts are disabled 01: Bursts of 4 are enabled 10: Bursts of 8 are enabled 11: Reserved	0x0
12	R/W	REQ_SENSE 0: DMA operates with level-sensitive peripheral requests (default) 1: DMA operates with (positive) edge-sensitive peripheral requests	0x0
11	R/W	DMA_INIT 0: DMA performs copy A1 to B1, A2 to B2, and so forth... 1: DMA performs copy of A1 to B1, B2, and so forth... 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. NOTE: This bit-field is overruled to 0 when channel 5 is configured as trusted channel (in Secure Boot mode).	0x0
10	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.	0x0

Bit	Mode	Symbol/Description	Reset
		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 or DMA_EXCLUSIVE_ACCESS = 1, DMA_IDLE is don't care. NOTE: This bit-field is overruled to 0 when the DMA channel 5 is configured as trusted channel (in Secure Boot mode).	
9:7	R/W	DMA_PRIO The priority level determines which DMA channel will be granted access for transferring data, in case more than one channel is 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
6	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
5	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
4	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
3	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) Note: This bit-field is overruled to 0 when channel 5 is configured as trusted channel (in Secure Boot mode).	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	0x0

Bit	Mode	Symbol/Description	Reset
		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. Note: If DMA_ON is disabled by software while the DMA channel is active, it cannot be enabled again until the channel has completed the last on-going read-write cycle and has stopped. Thus, the software has to check that the reading of DMA5_CTRL_REG.DMA_ON returns 0 before setting again the specific bit-field.	

Table 527: DMA5_IDX_REG (0x500602B4)

Bit	Mode	Symbol/Description	Reset
15:0	R	DMA5_IDX This (read-only) register determines the data items already transferred by the DMA channel. Hence, if its value is 1, then the DMA channel has already copied one data item and it is currently performing the next copy. If its value is 2, then two items have already been copied, and so on. When the transfer is completed (so when DMA5_CTRL_REG[DMA_ON] has been cleared) and DMA5_CTRL_REG[CIRCULAR] is not set, the register keeps its (last) value (which should be equal to DMA5_LEN_REG) and it is automatically reset to 0 upon starting a new transfer. In Circular mode, the register is automatically initialized to 0 as soon as the DMA channel starts over again.	0x0

Table 528: DMA_REQ_MUX_REG (0x50060300)

Bit	Mode	Symbol/Description	Reset
11:8	R/W	DMA45_SEL Select which combination of peripherals are mapped on the DMA channels. The peripherals are mapped as pairs on two channels. The first DMA request is mapped on channel 4 and the second on channel 5. Please refer to the description of DMA01_SEL bit-field for the exact peripherals' mapping. NOTE: When channel 5 is configured as secure channel, it cannot support any peripheral requests, since DREQ_MODE is disabled automatically, overruling the corresponding bit-field of DMA5_CTRL_REG.	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. The first DMA request is mapped on channel 2 and the second on channel 3. Please refer to the description of DMA01_SEL bit-field for the exact peripherals' mapping.	0xF
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. The first DMA request is mapped on channel 0 and the second on channel 1 and the peripherals supported are listed below. Note that in the following list, the "rx" implies a Peripheral-to-Memory transfer and the "tx" a Memory-to-Peripheral transfer. 0x0: SPI_rx/SPI_tx 0x1: UART_rx/UART_tx 0x2: UART2_rx/UART2_tx	0xF

Bit	Mode	Symbol/Description	Reset
		0x3: I2C_rx/I2C_tx 0x4: PCM/PCM 0x5: SRC1_rx/SRC1_rx 0x6: SRC1_tx/SRC1_tx 0x7: SRC2_rx/SRC2_rx 0x8: SRC2_tx/SRC2_tx 0x9: SRC1_rx/SRC2_tx 0xA: SRC2_rx/SRC1_tx 0xB: SRC1_rx/SRC1_tx 0xC: SRC2_rx/SRC2_tx 0xD: GP_ADC/None 0xE: SDADC/Flash Controller (tx) 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 signal driven to the selected peripheral. Hence, if DMA01_SEL = DMA23_SEL, the channels 0 and 1 will provide the RX and TX DMA acknowledge signals for the selected peripheral. Consequently, it is suggested to assign the intended peripheral value to a unique selector field. Exceptions are SRC1 and SRC2, for which the mapping of the same peripheral option to more than one channel pairs may be intended (for example, for stereo mode and values 0xB and 0xC, translating to SRC_rx/SRC_tx and SRC2_rx/SRC2_tx).	

Table 529: DMA_INT_STATUS_REG (0x50060304)

Bit	Mode	Symbol/Description	Reset
13	R	DMA_BUS_ERR5 0: No bus error response is detected for channel 5 1: Bus error response detected for channel 5 Note: This bit-field is auto-clear and it is initialized to 0 as soon as a new transfer is started. It is also noted that when channel 5 is configured as "trusted" (in Secure Boot mode), this bit-field is overruled to 0 masking the bus error status reported to the user.	0x0
12	R	DMA_BUS_ERR4 0: No bus error response is detected for channel 4 1: Bus error response detected for channel 4 Note: This bit-field is auto-clear and it is initialized to 0 as soon as a new transfer is started.	0x0
11	R	DMA_BUS_ERR3 0: No bus error response is detected for channel 3 1: Bus error response detected for channel 3 Note: This bit-field is auto-clear and it is initialized to 0 as soon as a new transfer is started.	0x0
10	R	DMA_BUS_ERR2 0: No bus error response is detected for channel 2 1: Bus error response detected for channel 2 Note: This bit-field is auto-clear and it is initialized to 0 as soon as a new transfer is started.	0x0

Bit	Mode	Symbol/Description	Reset
9	R	DMA_BUS_ERR1 0: No bus error response is detected for channel 1 1: Bus error response detected for channel 1 Note: This bit-field is auto-clear and it is initialized to 0 as soon as a new transfer is started.	0x0
8	R	DMA_BUS_ERR0 0: No bus error response is detected for channel 0 1: Bus error response detected for channel 0 Note: This bit-field is auto-clear and it is initialized to 0 as soon as a new transfer is started.	0x0
7:6	R	- Reserved	0x0
5	R	DMA_IRQ_CH5 0: IRQ on channel 5 is not set 1: IRQ on channel 5 is set	0x0
4	R	DMA_IRQ_CH4 0: IRQ on channel 4 is not set 1: IRQ on channel 4 is set	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 530: DMA_CLEAR_INT_REG (0x50060308)

Bit	Mode	Symbol/Description	Reset
5	R0/W	DMA_RST_IRQ_CH5 Writing 1 will reset the status bit of DMA_INT_STATUS_REG for channel 5; writing 0 will have no effect	0x0
4	R0/W	DMA_RST_IRQ_CH4 Writing 1 will reset the status bit of DMA_INT_STATUS_REG for channel 4; writing 0 will have no effect	0x0
3	R0/W	DMA_RST_IRQ_CH3 Writing 1 will reset the status bit of DMA_INT_STATUS_REG for channel 3; writing 0 will have no effect	0x0
2	R0/W	DMA_RST_IRQ_CH2	0x0

Bit	Mode	Symbol/Description	Reset
		Writing 1 will reset the status bit of DMA_INT_STATUS_REG for channel 2; writing 0 will have no effect	
1	R0/W	DMA_RST_IRQ_CH1 Writing 1 will reset the status bit of DMA_INT_STATUS_REG for channel 1; writing 0 will have no effect	0x0
0	R0/W	DMA_RST_IRQ_CH0 Writing 1 will reset the status bit of DMA_INT_STATUS_REG for channel 0; writing 0 will have no effect	0x0

Table 531: DMA_INT_MASK_REG (0x5006030C)

Bit	Mode	Symbol/Description	Reset
5	R/W	DMA_IRQ_ENABLE5 0: disable interrupts on channel 5 1: enable interrupts on channel 5	0x0
4	R/W	DMA_IRQ_ENABLE4 0: disable interrupts on channel 4 1: enable interrupts on channel 4	0x0
3	R/W	DMA_IRQ_ENABLE3 0: disable interrupts on channel 3 1: enable interrupts on channel 3	0x0
2	R/W	DMA_IRQ_ENABLE2 0: disable interrupts on channel 2 1: enable interrupts on channel 2	0x0
1	R/W	DMA_IRQ_ENABLE1 0: disable interrupts on channel 1 1: enable interrupts on channel 1	0x0
0	R/W	DMA_IRQ_ENABLE0 0: disable interrupts on channel 0 1: enable interrupts on channel 0	0x0

Table 532: DMA_SET_INT_MASK_REG (0x50060310)

Bit	Mode	Symbol/Description	Reset
5	RWS	DMA_SET_IRQ_ENABLE5 Writing 1 will enable the IRQs in the DMA channel 5, writing 0 has no effect. Reading always returns 0.	0x0
4	RWS	DMA_SET_IRQ_ENABLE4 Writing 1 will enable the IRQs in the DMA channel 4, writing 0 has no effect. Reading always returns 0.	0x0
3	RWS	DMA_SET_IRQ_ENABLE3 Writing 1 will enable the IRQs in the DMA channel 3, writing 0 has no effect. Reading always returns 0.	0x0
2	RWS	DMA_SET_IRQ_ENABLE2	0x0

Bit	Mode	Symbol/Description	Reset
		Writing 1 will enable the IRQs in the DMA channel 2, writing 0 has no effect. Reading always returns 0.	
1	RWS	DMA_SET_IRQ_ENABLE1 Writing 1 will enable the IRQs in the DMA channel 1, writing 0 has no effect. Reading always returns 0..	0x0
0	RWS	DMA_SET_IRQ_ENABLE0 Writing 1 will enable the IRQs in the DMA channel 0, writing 0 has no effect. Reading always returns 0.	0x0

Table 533: DMA_RESET_INT_MASK_REG (0x50060314)

Bit	Mode	Symbol/Description	Reset
5	RWS	DMA_RESET_IRQ_ENABLE5 Writing 1 will disable the IRQs in the DMA channel 5, writing 0 has no effect. Reading always returns 0.	0x0
4	RWS	DMA_RESET_IRQ_ENABLE4 Writing 1 will disable the IRQs in the DMA channel 4, writing 0 has no effect. Reading always returns 0.	0x0
3	RWS	DMA_RESET_IRQ_ENABLE3 Writing 1 will disable the IRQs in the DMA channel 3, writing 0 has no effect. Reading always returns 0.	0x0
2	RWS	DMA_RESET_IRQ_ENABLE2 Writing 1 will disable the IRQs in the DMA channel 2, writing 0 has no effect. Reading always returns 0.	0x0
1	RWS	DMA_RESET_IRQ_ENABLE1 Writing 1 will disable the IRQs in the DMA channel 1, writing 0 has no effect. Reading always returns 0.	0x0
0	RWS	DMA_RESET_IRQ_ENABLE0 Writing 1 will disable the IRQs in the DMA channel 0, writing 0 has no effect. Reading always returns 0.	0x0

35.22 Clock generation controller registers

Table 534: Register map CRG

Address	Register	Description
0x50000000	CLK_AMBA_REG	HCLK, PCLK, divider and clock gates
0x5000000C	RST_CTRL_REG	Reset control register
0x50000010	CLK_RADIO_REG	Radio PLL control register
0x50000014	CLK_CTRL_REG	Clock control register
0x50000018	CLK_TMR_REG	Clock control for the timers
0x5000001C	CLK_SWITCH2XTAL_REG	Switches clock from RC32M to XTAL32M
0x50000020	PMU_CTRL_REG	Power Management Unit control register
0x50000024	SYS_CTRL_REG	System Control register

Address	Register	Description
0x50000028	SYS_STAT_REG	System status register
0x5000003C	CLK_RCLP_REG	32/512 kHz RC oscillator register
0x50000040	CLK_XTAL32K_REG	32 kHz XTAL oscillator register
0x50000044	CLK_RC32M_REG	Fast RC control register
0x50000048	CLK_RCX_REG	RCX-oscillator control register
0x5000004C	CLK_RTCDIV_REG	Divisor for RTC 100 Hz clock
0x50000050	BANDGAP_REG	bandgap trimming
0x50000070	P0_PAD_LATCH_REG	Control the state retention of the GPIO ports
0x50000074	P0_SET_PAD_LATCH_REG	Control the state retention of the GPIO ports
0x50000078	P0_RESET_PAD_LATCH_REG	Control the state retention of the GPIO ports
0x5000007C	P1_PAD_LATCH_REG	Control the state retention of the GPIO ports
0x50000080	P1_SET_PAD_LATCH_REG	Control the state retention of the GPIO ports
0x50000084	P1_RESET_PAD_LATCH_REG	Control the state retention of the GPIO ports
0x50000098	POR_PIN_REG	Selects a GPIO pin for POR generation
0x5000009C	POR_TIMER_REG	Time for POR to happen
0x500000A4	BIAS_VREF_SEL_REG	
0x500000BC	RESET_STAT_REG	Reset status register
0x500000C0	RAM_PWR_CTRL_REG	Control power state of System RAMS
0x500000CC	SECURE_BOOT_REG	Controls secure booting (only ROM software can write)
0x500000D0	BOD_CTRL_REG	BOD control register
0x500000D4	DISCHARGE_RAIL_REG	Immediate rail resetting. There is no LDO/DCDC gating
0x500000DC	ANA_STATUS_REG	Analog Signals Status Register
0x500000E0	POWER_CTRL_REG	Power control register
0x500000E4	POWER_LEVEL_REG	Power level settings
0x500000F0	HIBERN_CTRL_REG	Hibernation control register
0x500000F4	PMU_SLEEP_REG	Configures the sleep/wake-up strategy
0x500000FC	STARTUP_STATUS_REG	Startup Statemachine Status Register
0x50010000	XTAL32M_START_REG	Trim values for XTAL32M in START state of startup
0x50010004	XTAL32M_SETTLE_REG	Trim values for XTAL32M in SETTLE state of startup
0x50010008	XTAL32M_TRIM_REG	Trim values for XTAL32M in RUNNING state
0x5001000C	XTAL32M_CAP_MEAS_REG	Capacitance measure circuit control

Address	Register	Description
0x50010010	XTAL32M_FSM_REG	Startup state machine configuration
0x50010014	XTAL32M_CTRL_REG	Xtal32m control register
0x50010018	XTAL32M_IRQ_CTRL_REG	Xtal32m Interrupt control register
0x50010024	XTAL32M_STAT0_REG	XTAL32M status register
0x50010028	XTAL32M_IRQ_STAT_REG	XTAL32M IRQ status register
0x50010060	CLKDBLR_CTRL1_REG	Clock Doubler Control Logic control register 1
0x50010064	CLKDBLR_CTRL2_REG	Clock Doubler Control Logic control register 2
0x50010068	CLKDBLR_STATUS_REG	Clock Doubler Control Logic status register
0x50020504	CLK_COM_REG	Peripheral divider register
0x50020508	SET_CLK_COM_REG	Peripheral divider register SET register. Reads back 0x0000
0x5002050C	RESET_CLK_COM_REG	Peripheral divider register RESET register. Reads back 0x0000
0x50040C04	CLK_PER_REG	Peripheral divider register
0x50040C08	SET_CLK_PER_REG	Peripheral divider register SET register, reads 0x0000
0x50040C0C	RESET_CLK_PER_REG	Peripheral divider register RESET register, reads 0x0000
0x50050500	CLK_SYS_REG	Peripheral divider register

Table 535: CLK_AMBA_REG (0x50000000)

Bit	Mode	Symbol/Description	Reset
12	R/W	QSPI_ENABLE Clock enable for QSPI controller	0x0
11:10	R/W	QSPI_DIV QSPI divider 00 = divide by 1 01 = divide by 2 10 = divide by 4 11 = divide by 8	0x0
9	R/W	- Reserved	0x0
8	R/W	- Reserved	0x0
7	R/W	QDEC_CLK_ENABLE Clock enable for QDEC block	0x0
6	R/W	AES_CLK_ENABLE Clock enable for AES crypto block	0x0

Bit	Mode	Symbol/Description	Reset
5:4	R/W	PCLK_DIV APB interface clock, Cascaded with HCLK: 00 = divide hclk by 1 01 = divide hclk by 2 10 = divide hclk by 4 11 = divide hclk by 8	0x2
3	R/W	- Reserved	0x0
2:0	R/W	HCLK_DIV AHB interface and microprocessor clock. Source clock divided by: 000 = divide hclk by 1 001 = divide hclk by 2 010 = divide hclk by 4 011 = divide hclk by 8 1xx = divide hclk by 16	0x2

Table 536: RST_CTRL_REG (0x5000000C)

Bit	Mode	Symbol/Description	Reset
2	R/W	CMAC_CACHE_FLUSH_WITH_SW_RESET 0: Flush the CMAC Cache memory only at hardware reset. 1: Flush the CMAC Cache memory also at software reset.	0x0
1	R/W	SYS_CACHE_FLUSH_WITH_SW_RESET 0: Flush the System Cache memory only at hardware reset. 1: Flush the System Cache memory also at software reset.	0x0
0	R/W	GATE_RST_WITH_FCU 0: Reset is not gated with FCU write/erase 1: Reset delayed in case FCU write/erase is ongoing. Software reset is never gated so should not be set during FCU write/erase.	0x0

Table 537: CLK_RADIO_REG (0x50000010)

Bit	Mode	Symbol/Description	Reset
3	R/W	RFCU_ENABLE Enable the RF Control Unit clock	0x0
2	R/W	CMAC_SYNCH_RESET Force synchronous reset to CMAC core and Sleep Timer. Its effective only when both Radio and Timer Power Domains are powered, and the clocks are enabled. CMAC CPU and CMAC registers, including the retained ones, will be reset. It should be kept in reset for enough time to make sure that it will be captured by CMAC, Low Power and APB clocks.	0x1
1	R/W	CMAC_CLK_SEL Selects the clock source	0x0

Bit	Mode	Symbol/Description	Reset
		1 = DIV1 clock 0 = DIVN clock	
0	R/W	CMAC_CLK_ENABLE Enables the CMAC clock.	0x0

Table 538: CLK_CTRL_REG (0x50000014)

Bit	Mode	Symbol/Description	Reset
15	R	RUNNING_AT_DBLR64M Indicates that the DBLR64MHz clock is used as clock, and may not be switched off	0x0
14	R	RUNNING_AT_XTAL32M Indicates that the XTAL32M clock is used as clock, and may not be switched off	0x0
13	R	RUNNING_AT_RC32M Indicates that the RC32M clock is used as clock	0x1
12	R	RUNNING_AT_LP_CLK Indicates that either the LP_CLK is being used as clock	0x0
11:6	R	- Reserved	0x0
5	R/W	XTAL32M_DISABLE Setting this bit instantaneously disables the 32-MHz crystal oscillator. Also, after sleep/wake-up cycle, the oscillator will not be enabled. This bit may not be set to 1 when RUNNING_AT_XTAL32M is 1 to prevent deadlock. After resetting this bit, wait for XTAL32M_SETTLED or XTAL32M_TRIM_READY to become 1 before switching to XTAL32M clock source.	0x0
4	R/W	- Reserved	0x0
3:2	R/W	LP_CLK_SEL Sets the clock source of the Lower Power clock 0x0: RCLP 0x1: RCX 0x2: XTAL32K through the oscillator with an external Crystal. 0x3: XTAL32K through an external square wave generator (set PID of GPIO to FUNC_GPIO)	0x0
1:0	R/W	SYS_CLK_SEL Selects the clock source. 0x0: XTAL32M 0x1: RC32M 0x2: RCLP 0x3: DBLR64M	0x1

Table 539: CLK_TMR_REG (0x50000018)

Bit	Mode	Symbol/Description	Reset
2	R/W	TMR2_PWM_AON_MODE Maps Timer2 PWM onto P0_10. This state is preserved during deep sleep to allow PWM output on the pad during deep sleep.	0x0
1	R/W	TMR_PWM_AON_MODE Maps Timer1 PWM onto P0_12. This state is preserved during deep sleep to allow PWM output on the pad during deep sleep.	0x0
0	R/W	WAKEUPCT_ENABLE Enables the clock	0x0

Table 540: CLK_SWITCH2XTAL_REG (0x5000001C)

Bit	Mode	Symbol/Description	Reset
0	W	SWITCH2XTAL When writing to this register, the clock switch will happen from RC32M to XTAL32M. If any other clock is selected than RC32M, the selection is discarded.	0x0

Table 541: PMU_CTRL_REG (0x50000020)

Bit	Mode	Symbol/Description	Reset
11	R/W	RETAIN_CMAC_CACHE Selects the retainability of the cmac cache block during deep sleep. 1 is retainable 0 is power gated	0x0
10	R/W	AUD_SLEEP Put the AUDIO power domain (PD_AUD) in powerdown.	0x1
9	R/W	LP_CLK_OUTPUT_EN LP_CLK output enable bit: 0: Disabled 1: Map LP_CLK on GPIO P0[12]	0x0
8	R/W	- Reserved	0x0
7	R/W	RETAIN_CACHE Selects the retainability of the system cache block during deep sleep. 1 is retainable 0 is power gated	0x0
6	R/W	SYS_SLEEP Put the System powerdomain (PD_SYS) in powerdown. If this bit is 1, and there is no pending IRQ in the PDC for the M33, the PD_SYS will be switched off. Wake-up should be handled by the PDC.	0x0

Bit	Mode	Symbol/Description	Reset
5	R/W	RESET_ON_WAKEUP Perform a hardware reset after waking up. Booter will be started.	0x0
4	R/W	MAP_BANDGAP_EN Setting this bit will: - map bandgap_enable to P0_11 - map (wokenup OR cmac_slp_timer_expire) to P1_02	0x0
3	R/W	COM_SLEEP Put the Communications powerdomain (PD_COM) in powerdown	0x1
2	R/W	TIM_SLEEP Put the Timers Powerdomain (PD_TIM) in powerdown.	0x1
1	R/W	RADIO_SLEEP Put the digital part of the radio, including CMAC (PD_RAD) in powerdown	0x1
0	R/W	PERIPH_SLEEP Put the peripherals power domain (PD_PER) in powerdown	0x1

Table 542: SYS_CTRL_REG (0x50000024)

Bit	Mode	Symbol/Description	Reset
15	W	SW_RESET Writing 1 to this bit will generate a SW_RESET.	0x0
14:11	R	- Reserved	0x0
10	R/W	CACHERAM_MUX Controls accessibility of Cache RAM (incl. enabling/disabling of the Cache Controller): 0: The cache controller is disabled/bypassed, the cacheRAM is visible in the memory space, 1: The cache controller is enabled, the cacheRAM is not visible anymore in the memory space. Note: When the cache controller is enabled after or at reset, the Cache is first initialized (cleared/flushed) by writing all 0s to each line, one line per clock cycle to all SRAM modules. When the cache controller is disabled/bypassed the cache memory is cleared/flushed in the same way as during the first after reset initialization.	0x0
9	R/W	- Reserved	0x0
8	R/W	- Reserved	0x0
7	R/W	DEBUGGER_ENABLE Enable the debugger. This bit is set by the booter according to the OTP header. If not set, the SWDIO and SW_CLK can be used as GPIO ports.	0x0
6	R/W	- Reserved	0x0
5	R/W	-	0x1

Bit	Mode	Symbol/Description	Reset
		Reserved	
4	R/W	- Reserved	0x0
3	R/W	REMAP_INTVECT 0: Normal operation 1: If ARM is in address range 0 to 0x1FF then the address is remapped to SYS-RAM 0x0080.0000 to 0x0080.01FF. This allows to put the interrupt vector table to be placed in RAM while executing, for example, from QSPI.	0x0
2:0	R/W	REMAP_ADR0 Controls which memory is located at address 0x0000 for execution. 0x0: ROM 0x1: eFlash Note 1: When REMAP_ADR0 = 0x1, address 0x0 is mapped to EFLASH_REGION_BASE + EFLASH_REGION_OFFSET<<2 Note 2: When REMAP_ADR0 = 0x1, the CPU can only access the eFlash region [EFLASH_REGION_BASE + EFLASH_REGION_OFFSET<<2, EFLASH_REGION_SIZE] from the 0x00A00000 address range. The complete eFlash can be accessed via the 0x31000000 address range but only uncached.0x2: QSPI Flash cached (see also the CACHE_FLASH_REG.FLASH_REGION.* descriptions) Note 1: When REMAP_ADR0=0x2, address 0x0 is mapped to FLASH_REGION_BASE + FLASH_REGION_OFFSET<<2. Note 2: When REMAP_ADR0=0x2, the CPU can only access the Flash region [FLASH_REGION_BASE + FLASH_REGION_OFFSET<<2, FLASH_REGION_SIZE] from the 0x16000000 address range. The complete Flash can be accessed via the 0x32000000 address range but only uncached. 0x3: SYSRAMS un-cached 0x4: SYSRAM3 (for testing purposes only) 0x5: System Cache Data RAM un-cached (CACHERAM_MUX = 0) Note 1: DWord (64 bits) access is not supported by the Cache Data RAM interface in mirrored mode (only 32, 16 and 8 bits). Note 2: DMA access is not supported by the Cache Data RAM interface when REMAP_ADR0 = 0x5. 0x6: QSPI Flash un-cached (for verification only) 0x7: No remapping (only relevant for the CMAC Cache)	0x0

Table 543: SYS_STAT_REG (0x50000028)

Bit	Mode	Symbol/Description	Reset
15	R	POWER_IS_UP Indicates that the Startup state machine is finished, and all power regulation are in order. In Fast Wake-up mode, the software needs to wait for this signal before starting any heavy traffic.	0x1
14	R	DBG_IS_ACTIVE Indicates that a debugger is attached.	0x0
13	R	AUD_IS_UP Indicates that PD_AUD is functional	0x0
12	R	AUD_IS_DOWN	0x1

Bit	Mode	Symbol/Description	Reset
		Indicates that PD_AUD is in power down	
11	R	COM_IS_UP Indicates that PD_COM is functional	0x0
10	R	COM_IS_DOWN Indicates that PD_COM is in power down	0x1
9	R	TIM_IS_UP Indicates that PD_TIM is functional	0x0
8	R	TIM_IS_DOWN Indicates that PD_TIM is in power down	0x1
7	R	MEM_IS_UP Indicates that PD_MEM is functional	0x1
6	R	MEM_IS_DOWN Indicates that PD_MEM is in power down	0x0
5	R	SYS_IS_UP Indicates that PD_SYS is functional	0x1
4	R	SYS_IS_DOWN Indicates that PD_SYS is in power down	0x0
3	R	PER_IS_UP Indicates that PD_PER is functional	0x0
2	R	PER_IS_DOWN Indicates that PD_PER is in power down	0x1
1	R	RAD_IS_UP Indicates that PD_RAD is functional	0x0
0	R	RAD_IS_DOWN Indicates that PD_RAD is in power down	0x1

Table 544: CLK_RCLP_REG (0x5000003C)

Bit	Mode	Symbol/Description	Reset
6:3	R/W	RCLP_TRIM 0000 = lowest frequency 0111 = default 1111 = highest frequency	0x7
2	R/W	RCLP_LOW_SPEED_FORCE Puts the RCLP in 32-kHz mode.	0x0
1	R/W	RCLP_HIGH_SPEED_FORCE Puts the RCLP in 512-kHz mode 1: Frequency is 512 kHz	0x0
0	R/W	RCLP_DISABLE Disables the RCLP oscillator	0x0

Table 545: CLK_XTAL32K_REG (0x50000040)

Bit	Mode	Symbol/Description	Reset
12:9	R/W	XTAL32K_VDDX_TRIM Trim for the pre-regulator for the oscillator core, running on VDCDC.	0x8
8	R/W	- Reserved	0x0
7	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 to XTAL32Kp Keep this bit 0 with a crystal between XTAL32Kp and XTAL32Km	0x0
6:3	R/W	XTAL32K_CUR Bias current for the 32-kHz XTAL oscillator. 0000 is minimum, 1111 is maximum, 0011 is default. For each application there is an optimal setting for which the start-up behavior is optimal	0x5
2:1	R/W	XTAL32K_RBIAS Setting for the bias resistor. 00 is maximum, 11 is minimum. The preferred setting will be provided by Renesas.	0x3
0	R/W	XTAL32K_ENABLE Enables the 32kHz XTAL oscillator	0x0

Table 546: CLK_RC32M_REG (0x50000044)

Bit	Mode	Symbol/Description	Reset
11	R/W	- Reserved	0x0
10:7	R/W	RC32M_COSC C-adjust of RC-oscillator A higher value of COSC results in a lower frequency	0xF
6:5	R/W	RC32M_RANGE Coarse adjust A higher value of RANGE results in a higher frequency, values 2 and 3 are equal	0x0
4:1	R/W	RC32M_BIAS Bias adjustment	0x7
0	R/W	RC32M_ENABLE Enables the 32-MHz RC oscillator	0x0

Table 547: CLK_RCX_REG (0x50000048)

Bit	Mode	Symbol/Description	Reset
10:7	R/W	RCX_BIAS LDO bias current.	0xA

Bit	Mode	Symbol/Description	Reset
		0x0: minimum 0xF: maximum	
6	R/W	RCX_C0 Add unit capacitance to RC-time delay.	0x1
5:1	R/W	RCX_CADJUST Adjust capacitance part of RC-time delay. 0x00: minimum capacitance 0x1F: maximum capacitance	0x1F
0	R/W	RCX_ENABLE 0: Disable the RCX oscillator (watchdog runs at RCLP) 1: Enable the RCX oscillator (watchdog runs at RCX)	0x0

Table 548: CLK_RTCDIV_REG (0x5000004C)

Bit	Mode	Symbol/Description	Reset
21	R/W	RTC_RESET_REQ Reset request for the RTC module	0x0
20	R/W	RTC_DIV_ENABLE Enable for the 100 Hz generation for the RTC block	0x0
19	R/W	RTC_DIV_DENOM Selects the denominator for the fractional division: 0b0: 1000 0b1: 1024	0x0
18:10	R/W	RTC_DIV_INT Integer divisor part for RTC 100Hz generation	0x147
9:0	R/W	RTC_DIV_FRAC Fractional divisor part for RTC 100-Hz generation. If RTC_DIV_DENOM = 1, <RTC_DIV_FRAC> out of 1024 cycles will divide by <RTC_DIV_INT+1>, the rest is <RTC_DIV_INT> If RTC_DIV_DENOM = 0, <RTC_DIV_FRAC> out of 1000 cycles will divide by <RTC_DIV_INT+1>, the rest is <RTC_DIV_INT>	0x2A8

Table 549: BANDGAP_REG (0x50000050)

Bit	Mode	Symbol/Description	Reset
10:6	R/W	BGR_ITRIM Current trimming for bias	0x0
5	R/W	- Reserved	0x0
4:0	R/W	BGR_TRIM Trim register for bandgap	0x0

Table 550: P0_PAD_LATCH_REG (0x50000070)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	P0_LATCH_EN Direct write to the individual pad latching signals. Latches the control signals of the pads for state retention in powerdown mode. 0 = Control signals are retained 1 = Latch is transparent, pad can be recontrolled	0xFFFF

Table 551: P0_SET_PAD_LATCH_REG (0x50000074)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P0_SET_LATCH_EN Direct Set of the marked bits. Reading returns 0x0.	0x0

Table 552: P0_RESET_PAD_LATCH_REG (0x50000078)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P0_RESET_LATCH_EN Direct Reset of the marked bits. Reading returns 0x0.	0x0

Table 553: P1_PAD_LATCH_REG (0x5000007C)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	P1_LATCH_EN Direct write to the individual pad latching signals. Latches the control signals of the pads for state retention in powerdown mode. 0 = Control signals are retained 1 = Latch is transparent, pad can be recontrolled	0xFFFF

Table 554: P1_SET_PAD_LATCH_REG (0x50000080)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P1_SET_LATCH_EN Direct Set of the marked bits. Reading returns 0x0.	0x0

Table 555: P1_RESET_PAD_LATCH_REG (0x50000084)

Bit	Mode	Symbol/Description	Reset
15:0	R0/W	P1_RESET_LATCH_EN Direct Reset of the marked bits. Reading returns 0x0.	0x0

Table 556: POR_PIN_REG (0x50000098)

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	- Reserved	0x0
5:0	R/W	POR_PIN_SELECT 0x00: P0_00 ... 0x0F: P0_15 0x10: P1_00 ... 0x1F: P1_15 0x20 to 0x3E: reserved 0x3F: POR generation disabled	0x3F

Table 557: POR_TIMER_REG (0x5000009C)

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 558: BIAS_VREF_SEL_REG (0x500000A4)

Bit	Mode	Symbol/Description	Reset
7:4	R/W	BIAS_VREF_RF2_SEL HP: 0xA LP: 0x3 same coding as BIAS_VREF_RF1_SEL.	0x3
3:0	R/W	BIAS_VREF_RF1_SEL Vref_code | Vref_Voltage (mV) 0:900 1:930 2:960 3:990 4:1020 5:1050 6:1080 7:1110 8:1140 9:1170 10:1200	0xA

Bit	Mode	Symbol/Description	Reset
		11:1230 12:1260 13:1290 14:1320 15:1350	

Table 559: RESET_STAT_REG (0x500000BC)

Bit	Mode	Symbol/Description	Reset
5	R/W	CMAC_WDOGRESET_STAT Indicates that a CMAC-Watchdog timeout has happened. Note that it is also set when a POReset has happened.	0x1
4	R/W	SWD_HWRESET_STAT Indicates that a write to SWD_RESET_REG has happened. Note that it is also set when a POReset has happened.	0x1
3	R/W	WDOGRESET_STAT Indicates that a Watchdog timeout has happened. Note that it is also set when a POReset has happened.	0x1
2	R/W	SWRESET_STAT Indicates that a software reset has happened	0x1
1	R/W	HWRESET_STAT Indicates that a hardware reset has happened	0x1
0	R/W	PORESET_STAT Indicates that a PowerOn Reset has happened. All bitfields of RESET_STAT_REG should be read (to check the source of reset) and then cleared to 0, allowing thus the hardware to automatically set to 1 the proper bitfields during the next reset event.	0x1

Table 560: RAM_PWR_CTRL_REG (0x500000C0)

Bit	Mode	Symbol/Description	Reset
5:4	R/W	RAM3_PWR_CTRL See description of RAM1_PWR_CTRL.	0x0
3:2	R/W	RAM2_PWR_CTRL See description of RAM1_PWR_CTRL.	0x0
1:0	R/W	RAM1_PWR_CTRL Power state control of the individual RAMs. May only change when the memory isn't accessed. When PD_MEM_IS_UP: 0x0: Normal operation 0x1: Normal operation 0x2: Retained (no access possible) 0x3: Off (memory content corrupted) When PD_MEM_IS_DOWN: 0x0: Retained 0x1: Off (memory content corrupted)	0x0

Bit	Mode	Symbol/Description	Reset
		0x2: Retained 0x3: Off (memory content corrupted)	

Table 561: SECURE_BOOT_REG (0x500000CC)

Bit	Mode	Symbol/Description	Reset
8	RWS	PROT_INFO_PAGE Protection of eFlash info page 0: Write/erase of info page enabled 1: Write/erase of info page permanently disabled	0x0
7	RWS	- Reserved	0x0
6	RWS	SECURE_BOOT Follows the respective eFlash flag value 0: System is not supporting secure boot 1: System is a secure system supporting secure boot	0x0
5	RWS	FORCE_CMAC_DEBUGGER_OFF This bit will permanently disable the CMAC debugger	0x0
4	RWS	FORCE_M33_DEBUGGER_OFF This bit will permanently disable the M33 debugger	0x0
3	RWS	PROT_USER_APP_CODE This bit will permanently disable write/erase of user application code sector	0x0
2	RWS	PROT_VALID_KEY This bit will permanently disable read/erase of validation keys sector	0x0
1	RWS	PROT_APP_KEY This bit will permanently disable read/erase of application keys sector	0x0
0	RWS	PROT_CONFIG_SCRIPT This bit will permanently disable write/erase of configuration script sector	0x0

Table 562: BOD_CTRL_REG (0x500000D0)

Bit	Mode	Symbol/Description	Reset
8	R/W	BOD_VDDIO_MASK Mask the output of the VDDIO comparator. 0x0 - POR trigger due to BOD comparator enabled 0x1 - POR trigger due to BOD comparator blocked	0x1
7	R/W	BOD_VDCDC_MASK Mask the output of the VDCDC comparator 0x0 - POR trigger due to BOD comparator enabled 0x1 - POR trigger due to BOD comparator blocked	0x1
6	R/W	BOD_VDD_MASK Mask the output of the VDD comparator	0x0

Bit	Mode	Symbol/Description	Reset
		0x0 - POR trigger due to BOD comparator enabled 0x1 - POR trigger due to BOD comparator blocked	
5	R/W	- Reserved	0x0
4	R/W	BOD_DIS_VDDIO_COMP Disable the BOD for the VDDIO comparator. Preferred way is to use BOD_VDDIO_MASK = 0x1 instead	0x0
3	R/W	BOD_DIS_VDCDC_COMP Disable the BOD for the VDCDC comparator. Preferred way is to use BOD_VDCDC_MASK = 0x1 instead	0x0
2	R/W	BOD_DIS_VDD_COMP Disable the BOD for the VDD comparator. Preferred way is to use BOD_VDD_MASK = 0x1 instead	0x0
1:0	R/W	BOD_SEL_VDD_LVL Set the level for the Vdd BOD comparator. 0x0 - 0.78 V (Used for Vdd = 0.9 V) 0x1 - 1.05 V (Used for Vdd = 1.2 V) 0x2 - Undefined, do not use 0x3 - 0.70 V (Sleep mode) Note that the level is overruled by the hardware state machine when the chip is in sleep mode and set to 0x3. During wake-up, the level is set back to 0x0.	0x0

Table 563: DISCHARGE_RAIL_REG (0x500000D4)

Bit	Mode	Symbol/Description	Reset
2	R/W	RESET_VDD 1: Enables immediate discharging of the VDD rail. Note that the source is not disabled. 0: disable immediate discharging of the VDD rail.	0x0
1	R/W	RESET_VDCDC	0x0
0	R/W	RESET_VIO 1: Enables immediate discharging of the VDDIO rail. Note that the source is not disabled. 0: disable immediate discharging of the VDDIO rail.	0x0

Table 564: ANA_STATUS_REG (0x500000DC)

Bit	Mode	Symbol/Description	Reset
8	R	LDO_GPADC_OK When high LDO_GPADC is active	0x0
7	R	BOD_COMP_VEFLASH_OK COMP_VEFLASH_OK = 1 -> VDD > 1.08 V No filtering applied; software should read multiple times to filter glitches.	0x0

Bit	Mode	Symbol/Description	Reset
6	R	BOD_COMP_VDCDC_OK COMP_VDCDC_ = 1 -> VDCDC > 0.95 V	0x0
5	R	BOD_COMP_VDDIO_OK COMP_VDDIO = 1 -> VDDIO > 1.667 V	0x0
4	R	BOD_COMP_VDD_OK COMP_VDD_OK = 1 -> VDD > 1.125 V	0x0
3	R	LDO_IO_OK When high LDO_IO is active	0x0
2	R	LDO_LOW_OK When high LDO_LOW is active	0x0
1	R	LDO_CORE_OK When high LDO_CORE is active	0x0
0	R	BANDGAP_OK When high bandgap is active	0x0

Table 565: POWER_CTRL_REG (0x500000E0)

Bit	Mode	Symbol/Description	Reset
15	R/W	- Reserved	0x0
14	R/W	LDO_CORE_RET_VREF_ENABLE Enable the LDO_CORE_RET VREF buffer in the bandgap.	0x1
13	R/W	LDO_IO_RET_VREF_ENABLE Enable the LDO_IO_RET VREF buffer in the bandgap	0x1
12	R/W	LDO_VREF_HOLD_FORCE Forces LDO references in hold mode	0x0
11	R/W	DCDC_ENABLE_SLEEP Enable the DCDC in sleep mode when DCDC_ENABLE is also set in DCDC_CTRL_REG	0x0
10	R/W	LDO_CORE_RET_ENABLE_SLEEP Enables (1) or disables (0) LDO_CORE_RET in sleep mode	0x1
9	R/W	LDO_CORE_RET_ENABLE_ACTIVE Enables (1) or disables (0) LDO_CORE_RET in active mode	0x0
8	R/W	LDO_LOW_ENABLE_SLEEP Enable the LDO_LOW in sleep mode	0x1
7	R/W	LDO_LOW_HIGH_CURRENT Force high current mode for LDO_LOW	0x0
6	R/W	LDO_IO_BYPASS_SLEEP Enables the LDO_IO bypass in sleep mode. Note that, when this bit is asserted, it overrules the regulating operation of the LDO	0x0
5	R/W	LDO_IO_BYPASS_ACTIVE	0x0

Bit	Mode	Symbol/Description	Reset
		Enables the LDO_IO bypass in active mode. Note that, when this bit is asserted, it overrules the regulating operation of the LDO	
4	R/W	LDO_IO_RET_ENABLE_SLEEP Enables (1) or Disables (0) LDO_IO_RET in sleep mode	0x1
3	R/W	LDO_IO_RET_ENABLE_ACTIVE Enables (1) or Disables (0) LDO_IO_RET in active mode	0x0
2	R/W	LDO_CORE_ENABLE Enables the LDO_CORE (0) Disable (1) Enable	0x1
1	R/W	LDO_LOW_ENABLE_ACTIVE Enables the LDO_LOW (0) Disable (1) Enable	0x1
0	R/W	LDO_IO_ENABLE Enables the LDO_IO... (0) Disable (1) Enable	0x1

Table 566: POWER_LEVEL_REG (0x500000E4)

Bit	Mode	Symbol/Description	Reset
13:11	R/W	XTAL32M_LDO_LEVEL Set the Level of the XTAL LDO. 0x0 = 882 mV 0x1 = 888 mV 0x2 = 894 mV 0x3 = 900 mV 0x4 = 906 mV 0x5 = 912 mV 0x6 = 918 mV 0x7 = 924 mV	0x3
10:7	R/W	VDDIO_TRIM Set the VDDIO level (25 mV/LSB): 0x8: 1.6 V 0x9: 1.625 V ... 0xF: 1.775 V 0x0: 1.8 V 0x1: 1.825 V ... 0x7: 1.975 V	0x0
6:4	R/W	VDCDC_LEVEL Set the VDCDC level, 0x0 = 1.1 V, 0x7 = 1.45 V, step = 50 mV	0x0
3	R/W	VDD_LEVEL_SLEEP Set the VDD level in sleep mode. 0x0 = 0.75 V, 0x1 = 0.9 V	0x0
2:0	R/W	VDD_LEVEL_ACTIVE Level setting for VDD rail 0x0: 0.9 V, 0x7 = 1.25 V 50 mV steps	0x0

Table 567: HIBERN_CTRL_REG (0x500000F0)

Bit	Mode	Symbol/Description	Reset
5:4	R/W	HIBERN_WKUP_MASK Selects which pin to wake-up from: 00: No wake-up pin enabled x1: P0[14] enabled as wake-up pin 1x: P1[4] enabled as wake-up pin	0x0
3:2	R/W	HIBERN_WKUP_POLARITY Selects the polarity of the wake-up source: x0: P0[14] active high wake-up pin x1: P0[14] active low wake-up pin 0x: P1[4] active high wake-up pin 1x: P1[4] active low wake-up pin	0x0
1	R/W	- Reserved	0x0
0	R/W	HIBERNATION_ENABLE Enables the hibernation mode when sleeping 0: Deep sleep mode, PD_SLP remains on 1: Hibernation mode, PD_SLP goes off.	0x0

Table 568: PMU_SLEEP_REG (0x500000F4)

Bit	Mode	Symbol/Description	Reset
16	R/W	FAST_WAKEUP Speeds up the wake-up process by enabling all LDOs simultaneously instead of in staggered order. Only use if all voltages have been retained during sleep.	0x0
15	R/W	LDO_OK_BYPASS Do not wait for LDO_OK signals in the wake-up process from sleep mode but use a fixed timeout of 32 µs. The BOD comparator check will make sure the supplies are OK before going to active state.	0x0
14	R/W	FAST_WAKEUP_SKIP_BGR_OK Go to running mode immediately, do not wait for the BGR_OK signal after a fast_wakeup trigger	0x0
13	R/W	BOD_MASK_BGR_OK MASK BOD reset when BGR is NOK	0x1
12	R/W	BG_ENABLE_SLEEP Keep the bandgap enabled during deep sleep operation	0x0
11:0	R/W	BG_REFRESH_INTERVAL This is a value defining the interval every which the Bandgap will be activated for refresh. The value represents ticks of lp_clk/64, for example, 30,5 µs * 64 = 1,9 ms.	0x80

Table 569: STARTUP_STATUS_REG (0x500000FC)

Bit	Mode	Symbol/Description	Reset
19:10	R	- Reserved	0x0
9	R	VEFLASH_LVL_RD Read back the eFlash level setting.	0x0
8	R	BOD_VEFLASH_OK_SYNC Read back the synchronized signal from the veflash comparator. It will become 0x1 when the VDD >1.08 V. This bit does not inform of a BOD event	0x0
7	R	BOD_VDDIO_OK_SYNC_RD Read back the synchronized signal from the vdcdc bod comparator	0x0
6	R	BOD_VDDD_OK_SYNC_RD Read back the synchronized signal from the vddd bod comparator	0x0
5	R	BOD_VDCDC_OK_SYNC_RD Read back the synchronized signal from the vdcdc bod comparator	0x0
4:3	R	BOD_VDDD_LVL_RD Read back bod_vddd_lvl setting.	0x0
2	R	BOD_VDDIO_MASK_SYNC_RD Synchronized bod_vddio_mask bit. Can be used to check if mask bit has been set	0x0
1	R	BOD_VDDD_MASK_SYNC_RD Synchronized bod_vddd_mask bit. Can be used to check if mask bit has been set	0x0
0	R	BOD_VDCDC_MASK_SYNC_RD Synchronized bod_vdcdc_mask bit. Can be used to check if mask bit has been set	0x0

Table 570: XTAL32M_START_REG (0x50010000)

Bit	Mode	Symbol/Description	Reset
26:20	R/W	XTAL32M_TIMEOUT Timeout 0: disabled 1: 4 µs 2: 8 µs 63: 252 µs 64: 268 µs ... 127: 1260 µs	0x0
19:17	R/W	XTAL32M_CMP_BLANK Blanking time for comparator output 0: disabled	0x3

Bit	Mode	Symbol/Description	Reset
		1: 4 µs 2: 8 µs 3: 16 µs 4: 32 µs 5: 64 µs	
16:15	R/W	XTAL32M_CMP_LVL Comparator triplevel 0: 2 µA 1: 4 µA 2: 8 µA 3: 12 µA	0x2
14:12	R/W	XTAL32M_AMPL_SET Amplitude Regulator input level setting (peak-peak) in START phase of startup 0: 300 mV 1: 350 mV .. 7: 900 mV	0x0
11:8	R/W	XTAL32M_CUR_SET Current setting (µA) in START phase of startup 0: 32 1: 64 2: 128 3: 192 4: 256 5: 320 6: 384 7: 448 8: 512 9: 576 10: 640 11: 704 12: 768 13: 832 14: 960 15: 1088	0x5
7:0	R/W	XTAL32M_TRIM Capacitance bank setting in START phase of startup CL = 3.5 pF + 50 fF/LSB	0x50

Table 571: XTAL32M_SETTLE_REG (0x50010004)

Bit	Mode	Symbol/Description	Reset
26:20	R/W	XTAL32M_TIMEOUT Timeout 0: disabled 1: 4 µs	0x0

Bit	Mode	Symbol/Description	Reset
		2: 8 µs 63: 252 µs 64: 268 µs ... 127: 1260 µs	
19:17	R/W	XTAL32M_CMP_BLANK Blanking time for comparator output 0: disabled 1: 4 µs 2: 8 µs 3: 16 µs 4: 32 µs 5: 64 µs	0x3
16:15	R/W	XTAL32M_CMP_LVL Comparator triplevel 0: 2 µA 1: 4 µA 2: 8 µA 3: 12 µA	0x1
14:12	R/W	XTAL32M_AMPL_SET Amplitude Regulator input level setting (peak-peak) in SETTLE phase of startup 0: 300 mV 1: 350 mV .. 7: 900 mV	0x0
11:8	R/W	XTAL32M_CUR_SET Current setting (µA) in SETTLE phase of startup 0: 32 1: 64 2: 128 3: 192 4: 256 5: 320 6: 384 7: 448 8: 512 9: 576 10: 640 11: 704 12: 768 13: 832 14: 960 15: 1088	0x5
7:0	R/W	XTAL32M_TRIM Capacitance bank setting in SETLLE phase of startup	0xA0

Bit	Mode	Symbol/Description	Reset
		CL = 3.5 pF + 50 fF/LSB	

Table 572: XTAL32M_TRIM_REG (0x50010008)

Bit	Mode	Symbol/Description	Reset
16:15	R/W	XTAL32M_CMP_LVL Comparator triplevel 0: 2 µA 1: 4 µA 2: 8 µA 3: 12 µA	0x1
14:12	R/W	XTAL32M_AMPL_SET Amplitude Regulator input level setting (peak-peak) in running phase 0: 300 mV 1: 350 mV .. 7: 900 mV	0x0
11:8	R/W	XTAL32M_CUR_SET Current setting (µA) in running phase 0: 32 1: 64 2: 128 3: 192 4: 256 5: 320 6: 384 7: 448 8: 512 9: 576 10: 640 11: 704 12: 768 13: 832 14: 960 15: 1088	0x4
7:0	R/W	XTAL32M_TRIM Capacitance bank setting in running phase, use to trim the xtal32m output frequency CL = 3.5 pF + 50 fF/LSB	0xA0

Table 573: XTAL32M_CAP_MEAS_REG (0x5001000C)

Bit	Mode	Symbol/Description	Reset
8:6	R/W	XTAL32M_MEAS_TIME Select measurement time (in DIVN clock-cycles) 0: 32 1: 64	0x2

Bit	Mode	Symbol/Description	Reset
		6: 2048 7: 4096	
5	R/W	XTAL32M_MEAS_START Starts capacitance measurement	0x0
4:3	R/W	XTAL32M_MEAS_CUR Select measurement current (minimum required capacitance) 0: 100 nA (0.44 pF) 1: 500 nA (2.22 pF) 2: 1 µA (4.44 pF) 3: 5 µA (22.2 pF)	0x3
2:0	R/W	XTAL32M_CAP_SELECT Select measured capacitance 0: disabled 1: hold capacitance 2: xtal_p 3: xtal_n 4: xtal_p + xtal_n 5: low reference on xtal_p 6: low reference on xtal_p	0x0

Table 574: XTAL32M_FSM_REG (0x50010010)

Bit	Mode	Symbol/Description	Reset
4	R/W	XTAL32M_FSM_APPLY_CONFIG CUR_SET, AMPL_SET, CMP_LVL and TRIM from XTAL32M_TRIM_REG are 0: applied at next startup 1: immediately applied	0x0
3	R/W	XTAL32M_FSM_FORCE_IDLE Forces FSM in IDLE state, allows for software control	0x0
2	R/W	XTAL32M_CMP_MODE Use following comparator trim settings in SETTLE state: 0: XTAL32M_TRIM_REG.CMP_LVL 1: XTAL32M_SETTLE_REG.CMP_LVL	0x0
1	R/W	XTAL32M_TRIM_MODE Use following trimsetting in the SETTLE state 0: XTAL32M_TRIM_REG.TRIM 1: XTAL32M_SETTLE_REG.TRIM	0x0
0	R/W	XTAL32M_CUR_MODE Use the following current setting in the SETTLE state 0: XTAL32M_START_REG.CUR_SET 1: XTAL32M_SETTLE_REG.CUR_SET	0x0

Table 575: XTAL32M_CTRL_REG (0x50010014)

Bit	Mode	Symbol/Description	Reset
8	R/W	XTAL32M_ENABLE Enables xtal32m (testing purposes)	0x0
7:6	R/W	XTAL32M_BIASPROT Bias startup circuit 0: enable during startup 1: always enabled 2: always disabled	0x0
5:4	R/W	XTAL32M_LDO_SAH Controls amplitude regulator sample-and-hold 2'b00: set to HOLD when IRQ fires 2'b01: always TRACK 2'b1x: always HOLD	0x1
3:2	R/W	XTAL32M_AMPREG_SAH Controls amplitude regulator sample-and-hold 2'b00: set to HOLD when IRQ fires 2'b01: always TRACK 2'b1x: always HOLD	0x1
1:0	R/W	XTAL32M_BIAS_SAH Controls bias sample-and-hold 2'b00: set to HOLD when IRQ fires 2'b01: always TRACK 2'b1x: always HOLD	0x1

Table 576: XTAL32M_IRQ_CTRL_REG (0x50010018)

Bit	Mode	Symbol/Description	Reset
11:10	R/W	XTAL32M_IRQ_CAP_CTRL The IRQ counter is captured in the XTAL32M_IRQ_STATUS_REG.IRQ_COUNT_CAP when leaving the following state 0: START 1: SETTLE 2: RUN	0x2
9	R/W	XTAL32M_IRQ_ENABLE Enable xtal interrupt generation.	0x0
8	R/W	XTAL32M_IRQ_CLK Clock divider for IRQ counter 0: 4 μs 1: 32 μs	0x1
7:0	R/W	XTAL32M_IRQ_CNT IRQ counter start value.	0xFF

Table 577: XTAL32M_STAT0_REG (0x50010024)

Bit	Mode	Symbol/Description	Reset
23:22	R	- Reserved	0x0
21:19	R	- Reserved	0x0
18:11	R	XTAL32M_TRIM_VAL Current value for oscillator trimming	0x0
10:7	R	XTAL32M_CUR_SET_STAT Current value for cur_set	0x0
6	R	XTAL32M_LDO_OK Indicates LDO voltage level is ok	0x0
5	R	XTAL32M_CMP_OUT Amplitude regulator comparator output state	0x0
4:1	R	XTAL32M_STATE State of XTAL startup FSM 0x0: IDLE 0x1: WAIT_LDO 0x2: WAIT_BIAS 0x3: START_BLANK 0x4: START 0x5: SETTLE_BLANK 0x6: SETTLE 0x7: RUN 0x8: CAP_TEST_IDLE 0x9: CAP_TEST_MEAS 0xA: CAP_TEST_END	0x0
0	R	XTAL32M_READY Indicates XTAL startup FSM has reached the RUNNING state and is ready for use (sysclk)	0x0

Table 578: XTAL32M_IRQ_STAT_REG (0x50010028)

Bit	Mode	Symbol/Description	Reset
15:8	R	XTAL32M_IRQ_COUNT_CAP Captured IRQ counter	0x0
7:0	R	XTAL32M_IRQ_COUNT_STAT Current IRQ counter value	0x0

Table 579: CLKDBLR_CTRL1_REG (0x50010060)

Bit	Mode	Symbol/Description	Reset
22:17	R/W	DELAY_TDC_OVR Overrule value for the delay_tdc output.	0x0

Bit	Mode	Symbol/Description	Reset
16:12	R/W	DELAY_64M_PRE_OVR Overrule value for the delay_64m_pre output.	0x0
11:8	R/W	DELAY_64M_OVR Overrule value for the delay_64m output.	0x0
7:3	R/W	DELAY_32M_OVR Overrule value for the delay_32m output.	0x0
2	R/W	OVERRIDE Bypass the Clock Doubler Control Logic. 0: Normal operation 1: Bypass the Control Logic and force its outputs to CLKDBLR_CTRL1_REG[DELAY*OVR] and CLKDBLR_CTRL2_REG[*OVR] values. Override mode enabling sequence: 1. Set CLKDBLR_CTRL*_REG[*OVR] bits except CLKDBLR_CTRL2_REG[OUTPUT_ENABLE_OVR] to desired values. 2. Release the reset via CLKDBLR_CTRL1_REG[RESET_N] 3. Set CLKDBLR_CTRL1_REG[OVERRIDE] to 1. 4. Set CLKDBLR_CTRL2_REG[OUTPUT_ENABLE_OVR] to 1. 5. Set CLKDBLR_CTRL1_REG[ENABLE] to 1.	0x0
1	R/W	ENABLE Enable the Control Logic. 0: Disabled 1: Enabled	0x0
0	R/W	RESET_N Reset the Control Logic. 0: Reset state 1: Normal operation.	0x1

Table 580: CLKDBLR_CTRL2_REG (0x50010064)

Bit	Mode	Symbol/Description	Reset
15:14	R/W	DELAY_64M_PRE_OFFSET Offset to be added to the delay_64m_pre output.	0x0
13	R/W	PRELOAD Preload the internal registers with CLKDBLR_CTRL1_REG[DELAY*OVR] inputs to give a good starting point to the FSM. 0: Normal mode. 1: Preload mode.	0x0
12	R/W	INV_CLK Invert the clock of the tdc_in register.	0x0
11:8	R/W	DUTY_CYCLE_CORR_COUNT Duty cycle correction interval. 0: Maintenance mode disabled. 1..15: Re-calculate delay outputs every 2^N clock cycles.	0x0
7	R/W	LOW_POWER_OVR	0x0

Bit	Mode	Symbol/Description	Reset
		Overrule value for the low_power output.	
6	R/W	EN_ADJ_32M_OVR Overrule value for the en_adj_32m.	0x0
5	R/W	EN_ADJ_64M_OVR Overrule value for the en_adj_64m.	0x0
4	R/W	EN_TDC_OVR Overrule value for the en_tdc.	0x0
3	R/W	PHASE_INV_32M_OVR Overrule value for the phase_inv_32m.	0x0
2	R/W	SEL_32M_64M_CLK_OVR Overrule value for the sel_32m_64m_clk	0x0
1	R/W	TDC_CLK_INV_OVR Overrule value for the tdc_clk_inv.	0x0
0	R/W	OUTPUT_ENABLE_OVR Overrule value for the output_enable.	0x0

Table 581: CLKDBLR_STATUS_REG (0x50010068)

Bit	Mode	Symbol/Description	Reset
31	R	PHASE_INV_32M Read-out of the phase_inv_32m output of the Clock Doubler Control Logic.	0x0
30:26	R	CLKDBLR_STATE Read-out of the state of the Clock Doubler Control Logic FSM state.	0x0
25	R	OUTPUT_READY Ready flag which indicates that 64 MHz clock is available 0: 64 MHz clock not available 1: 64 MHz clock available Note: This is an output of the calibration FSM, indicating calibration is finished and 64 MHz clock should be available. It is not a signal which shows the actual status of the 64-MHz clock.	0x0
24:20	R	TDC_BIN_OUT TDC binary output.	0x0
19:14	R	DELAY_TDC_OUT Delay value for the TDC block.	0x0
13:9	R	DELAY_64M_PRE_OUT Delay value for the 64 MHz DTC block.	0x0
8:5	R	DELAY_64M_OUT Delay value for the 64 MHz DTC block.	0x0
4:0	R	DELAY_32M_OUT Delay value for the 32 MHz DTC block.	0x0

Table 582: CLK_COM_REG (0x50020504)

Bit	Mode	Symbol/Description	Reset
9:8	R/W	- Reserved	0x0
7	R/W	I2C_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
6	R/W	I2C_ENABLE Enables the clock	0x0
5	R/W	SPI_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
4	R/W	SPI_ENABLE Enables the clock	0x0
3	R/W	UART2_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
2	R/W	UART2_ENABLE Enables the clock	0x0
1	R/W	UART_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
0	R/W	UART_ENABLE Enables the clock	0x0

Table 583: SET_CLK_COM_REG (0x50020508)

Bit	Mode	Symbol/Description	Reset
9:8	R0/W	- Reserved	0x0
7	R0/W	I2C_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
6	R0/W	I2C_ENABLE Enables the clock	0x0
5	R0/W	SPI_CLK_SEL Selects the clock source 1 = DIV1 clock	0x0

Bit	Mode	Symbol/Description	Reset
		0 = DIVN clock	
4	R0/W	SPI_ENABLE Enables the clock	0x0
3	R0/W	UART2_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
2	R0/W	UART2_ENABLE Enables the clock	0x0
1	R0/W	UART_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
0	R0/W	UART_ENABLE Enables the clock	0x0

Table 584: RESET_CLK_COM_REG (0x5002050C)

Bit	Mode	Symbol/Description	Reset
9:8	R0/W	- Reserved	0x0
7	R0/W	I2C_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
6	R0/W	I2C_ENABLE Disables the clock	0x0
5	R0/W	SPI_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
4	R0/W	SPI_ENABLE Disables the clock	0x0
3	R0/W	UART2_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0
2	R0/W	UART2_ENABLE Disables the clock	0x0
1	R0/W	UART_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0

Bit	Mode	Symbol/Description	Reset
0	R0/W	UART_ENABLE Disables the clock	0x0

Table 585: CLK_PER_REG (0x50040C04)

Bit	Mode	Symbol/Description	Reset
0	R/W	GPADC_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock/2	0x0

Table 586: SET_CLK_PER_REG (0x50040C08)

Bit	Mode	Symbol/Description	Reset
0	RWS	GPADC_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock/2	0x0

Table 587: RESET_CLK_PER_REG (0x50040C0C)

Bit	Mode	Symbol/Description	Reset
0	RW1C	GPADC_CLK_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock/2	0x0

Table 588: CLK_SYS_REG (0x50050500)

Bit	Mode	Symbol/Description	Reset
1	R/W	- Reserved	0x0
0	R/W	- Reserved	0x0

35.23 Cortex M33 cache controller registers

Table 589: Register map CACHE

Address	Register	Description
0x1A0C0020	CACHE_CTRL2_REG	Cache control register 2 (only Word (32-bits) access supported).
0x1A0C0028	CACHE_MRM_HITS_REG	Cache MRM (Miss Rate Monitor) HITS register (only Word (32-bits) access supported).
0x1A0C002C	CACHE_MRM_MISSES_REG	Cache MRM (Miss Rate Monitor) MISSES register (only Word (32-bits) access supported).

Address	Register	Description
0x1A0C0030	CACHE_MRM_CTRL_REG	Cache MRM (Miss Rate Monitor) CONTROL register (only Word (32-bits) access supported).
0x1A0C0034	CACHE_MRM_TINT_REG	Cache MRM (Miss Rate Monitor) TIME INTERVAL register (only Word (32-bits) access supported).
0x1A0C0038	CACHE_MRM_MISSES_THRES_REG	Cache MRM (Miss Rate Monitor) THRESHOLD register (only Word (32-bits) access supported).
0x1A0C003C	CACHE_MRM_HITS_THRES_REG	Cache MRM (Miss Rate Monitor) HITS THRESHOLD register (only Word (32-bits) access supported).
0x1A0C0040	CACHE_FLASH_REG	Cache QSPI Flash program size and base address register (only Word (32-bits) access supported). This register is NA for the CMAC Cache (no remapping done in the CMAC Cache).
0x1A0C0044	CACHE_EFLASH_REG	Cache eFlash program size and base address register (only Word (32-bits) access supported). This register is NA for the CMAC Cache (no remapping done in the CMAC Cache).
0x1A0C0048	CACHE_MRM_HITS1WS_REG	Cache MRM (Miss Rate Monitor) HITS with 1 Wait State register (only Word (32-bits) access supported).
0x1A0C0050	SWD_RESET_REG	SWD hardware reset control register (only Word (32-bits) access supported).

Table 590: CACHE_CTRL2_REG (0x1A0C0020)

Bit	Mode	Symbol/Description	Reset
31:29	-	- Reserved	0
28	R	CACHE_READY Cache Controller RO status bit. 0: Default. 1: Set to 1 when CACHE_CTRL is enabled, initialized and immediately ready for a cacheable access to service.	0
27	R	CACHE_RAM_INIT Cache Controller RO status bit. 0: Default. 1: Set to 1 when SRAM is being initialized (being flushed). Note: The flushing of the cache memory takes 256 HCLK cycles.	0
26	R/W	- Reserved	0
25:17	R/W	CACHE_EF_LEN Length of eFlash cacheable memory. N*64 kB. N = 0 to 512 (max. of 32 MB). Setting CACHE_EF_LEN = 0 disables the caching. Note 1: The max. relevant CACHE_EF_LEN setting depends on the chosen Flash region (program) size. Note 2: The first block (CACHE_EF_LEN = 1) includes the memory space specified by CACHE_EFLASH_REG[EFLASH_REGION_OFFSET].	0
16	R/W	CACHE_FLUSH_DISABLE 0: Default.	0

Bit	Mode	Symbol/Description	Reset
		1: Flushing of the Cache memory is disabled when SYS_CTRL_REG[CACHERAM_MUX] is switched from 1 to 0. Note: Setting this bit to 1 is only allowed for debugging purposes.	
15:14	R/W	CACHE_USE_FULL_DB_RANGE 00: CACHERAM (mirrored) read/write and NO use of the full 184 bits databus (for executing program code or extension of the SysRAM with the Cache RAM). In this mode 8 bits, 16 bits and 32 bits write access is supported. 01: CACHERAM (mirrored) read and use of the full 184 bits databus of "SRAM_1_0" (for testing and debugging purposes). In this mode only 32 bits write access is supported. 10: CACHERAM (mirrored) read and use of the full 184 bits databus of "SRAM_3_2" (for testing and debugging purposes). In this mode only 32 bits write access is supported. 11: Reserved. Note 1: SYS_CTRL_REG[CACHERAM_MUX] must be set to 0 before accessing the memory mapped (mirrored) Cache Data and TAG memory. Note 2: For all three settings, max. 8 kB is available from the memory map.	0
13	R/W	CACHE_MHCLKEN_DISABLE 0: Default. 1: The "m_HCLK_EN" input is ignored and the controller avoids inserting m_HTRANS = BUSY because of wait states. Note: This bit is only relevant for executing from QSPI Flash (when set to 1, it will improve performance). This bit should be kept 0 for executing from eFlash.	0
12	R/W	CACHE_CWF_DISABLE 0: Default. 1: The cache line refill is performed with INCR type burst and "Critical Word First" is disabled. Note: This bit is only relevant for executing from QSPI Flash (when set to 1, it will improve performance). This bit should be kept 0 for executing from eFlash.	0
11	R/W	- Reserved	0
10	R/W	CACHE_CGEN 0: Cache controller clock gating is not enabled. 1: Cache controller clock gating is enabled (enabling power saving).	0
9	R/W	CACHE_WEN 0: Cache Data and TAG memory read only. 1: Cache Data and TAG memory read/write. The Data and TAG memory are only updated by the cache controller. There is no hardware protection to prevent unauthorized access by the Arm. Note 1: When accessing the memory mapped Cache Data and TAG memory (which is only allowed for debugging purposes) only 32 bits access is supported. Note 2: SYS_CTRL_REG[CACHERAM_MUX] must be set to 0 before accessing the memory mapped Cache Data and TAG memory.	0

Bit	Mode	Symbol/Description	Reset
		See also the CACHE_CTRL2_REG[CACHE_USE_FULL_DB_RANGE] description.	
8:0	R/W	CACHE_LEN Length of QSPI Flash cacheable memory. N*64 kB. N = 0 to 512 (max. of 32 MB). Setting CACHE_LEN = 0 disables the caching. Note 1: The max. relevant CACHE_LEN setting depends on the chosen Flash region (program) size. Note 2: The first block (CACHE_LEN = 1) includes the memory space specified by CACHE_FLASH_REG[FLASH_REGION_OFFSET]	0

Table 591: CACHE_MRM_HITS_REG (0x1A0C0028)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_HITS Contains the amount of cache hits.	0x0

Table 592: CACHE_MRM_MISSES_REG (0x1A0C002C)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_MISSES Contains the amount of cache misses.	0x0

Table 593: CACHE_MRM_CTRL_REG (0x1A0C0030)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0
4	R/W	MRM_IRQ_HITS_THRES_STATUS 0: No interrupt is generated. 1: Interrupt (pulse-sensitive) is generated because the number of cache hits reached the programmed threshold (threshold != 0).	0
3	R/W	MRM_IRQ_MISSES_THRES_STATUS 0: No interrupt is generated. 1: Interrupt (pulse-sensitive) is generated because the number of cache misses reached the programmed threshold (threshold != 0).	0
2	R/W	MRM_IRQ_TINT_STATUS 0: No interrupt is generated. 1: Interrupt (pulse-sensitive) is generated because the time interval counter reached the end (time interval != 0).	0
1	R/W	MRM_IRQ_MASK 0: Disables interrupt generation. 1: Enables interrupt generation. Note: The Cache MRM generates a pulse-sensitive interrupt towards the ARM processor.	0
0	R/W	MRM_START	0

Bit	Mode	Symbol/Description	Reset
		0: Freeze the "misses/hits" counters and reset the time interval counter to the programmed value in CACHE_MRM_TINT_REG. 1: Enables the counters. Note: In case CACHE_MRM_CTRL_REG[MRM_START] is set to 1 and CACHE_MRM_TINT_REG (!=0) is used for the MRM interrupt generation, the time interval counter counts down (on a fixed reference clock of 32 MHz) until it is 0. At that time CACHE_MRM_CTRL_REG[MRM_START] will be reset automatically to 0 by the MRM hardware and the MRM interrupt will be generated.	

Table 594: CACHE_MRM_TINT_REG (0x1A0C0034)

Bit	Mode	Symbol/Description	Reset
31:19	-	- Reserved	0x0
18:0	R/W	MRM_TINT Defines the time interval for the monitoring in 32 MHz clock cycles. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_TINT_STATUS]. Note: When MRM_TINT = 0 (unrealistic value), no interrupt will be generated.	0x0

Table 595: CACHE_MRM_MISSES_THRES_REG (0x1A0C0038)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_MISSES_THRES Defines the misses threshold to trigger the interrupt generation. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_MISSES_THRES_STATUS]. Note: When MRM_MISSES_THRES = 0 (unrealistic value), no interrupt will be generated.	0x0

Table 596: CACHE_MRM_HITS_THRES_REG (0x1A0C003C)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_HITS_THRES Defines the hits threshold to trigger the interrupt generation. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_HITS_THRES_STATUS]. Note: When MRM_HITS_THRES = 0 (unrealistic value), no interrupt will be generated.	0x0

Table 597: CACHE_FLASH_REG (0x1A0C0040)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	FLASH_REGION_BASE These bits correspond with the Flash region base address bits [31:16]. The default value is 0x1600.	0x1600

Bit	Mode	Symbol/Description	Reset
		The Flash region base address bits [31:25] are fixed to 0x16 and bits [17:16] are fixed to 0x0. These register bits are retained. Note 1: The updated value takes effect only after a software reset. Note 2 The Flash region base address setting depends on the chosen Flash region size.	
15:4	R/W	FLASH_REGION_OFFSET Flash region offset address (in words). This value is added to the Flash (CPU) address bits [13:2]. These register bits are retained. Note 1: The updated value takes effect only after a software reset.	0x0
3	R	- Reserved	0x0
2:0	R/W	FLASH_REGION_SIZE Flash region size. The default value is 6 (0.5 MB). 0 = 32 MB 1 = 16 MB 2 = 8 MB 3 = 4 MB 4 = 2 MB 5 = 1 MB 6 = 0.5 MB 7 = 0.25 MB These register bits are retained. Note 1: The updated value takes effect only after a software reset. Note 2: See for the max. region (program) size the memory map.	0x6

Table 598: CACHE_EFLASH_REG (0x1A0C0044)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	EFLASH_REGION_BASE These bits correspond with the eFlash region base address bits [31:16]. The default value is 0x00A0. The eFlash region base address bits [31:24] are fixed to 0x00. These register bits are retained. Note 1: The updated value takes effect only after a software reset. Note 2 The eFlash region base address setting depends on the chosen eFlash region size.	0xA0
15:4	R/W	EFLASH_REGION_OFFSET The eFlash region offset address (in words). This value is added to the eFlash (CPU) address bits [13:2]. These register bits are retained. Note 1: The updated value takes effect only after a software reset.	0x0
3	R	- Reserved	0x0
2:0	R/W	EFLASH_REGION_SIZE	0x6

Bit	Mode	Symbol/Description	Reset
		The eFlash region size. The default value is 6 (128 kB). 0 = 512 kB (reserved for bigger eFlash) 1 = 512 kB (reserved for bigger eFlash) 2 = 512 kB (reserved for bigger eFlash) 3 = 512 kB (reserved for bigger eFlash) 4 = 512 kB 5 = 256 kB 6 = 128 kB 7 = 64 kB These register bits are retained. Note 1: The updated value takes effect only after a software reset. Note 2: See for the max. region (program) size the memory map.	

Table 599: CACHE_MRM_HITS1WS_REG (0x1A0C0048)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_HITS1WS Contains the amount of cache hits.	0x0

Table 600: SWD_RESET_REG (0x1A0C0050)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0
0	R0/W	SWD_HW_RESET_REQ 0: default. 1: hardware reset request (from the debugger tool). The register is automatically reset with a HW_RESET. This bit can only be accessed by the debugger software and not by the application.	0

35.24 Cortex M0+ cache controller registers

Table 601: Register map CMAC_CACHE

Address	Register	Description
0x1A1C0020	CM_CACHE_CTRL2_REG	Cache control register 2 (only Word (32-bits) access supported).
0x1A1C0028	CM_CACHE_MRM_HITS_REG	Cache MRM (Miss Rate Monitor) HITS register (only Word (32-bits) access supported).
0x1A1C002C	CM_CACHE_MRM_MISSES_REG	Cache MRM (Miss Rate Monitor) MISSES register (only Word (32-bits) access supported).
0x1A1C0030	CM_CACHE_MRM_CTRL_REG	Cache MRM (Miss Rate Monitor) CONTROL register (only Word (32-bits) access supported).
0x1A1C0034	CM_CACHE_MRM_TINT_REG	Cache MRM (Miss Rate Monitor) TIME INTERVAL register (only Word (32-bits) access supported).

Address	Register	Description
0x1A1C0038	CM_CACHE_MRM_MISSES_THRES_REG	Cache MRM (Miss Rate Monitor) THRESHOLD register (only Word (32-bits) access supported).
0x1A1C003C	CM_CACHE_MRM_HITS_THRES_REG	Cache MRM (Miss Rate Monitor) HITS THRESHOLD register (only Word (32-bits) access supported).
0x1A1C0040	CM_CACHE_FLASH_REG	Cache QSPI Flash program size and base address register (only Word (32-bits) access supported). This register is NA for the CMAC Cache (no remapping done in the CMAC Cache).
0x1A1C0044	CM_CACHE_EFLASH_REG	Cache eFlash program size and base address register (only Word (32-bits) access supported). This register is NA for the CMAC Cache (no remapping done in the CMAC Cache).
0x1A1C0048	CM_CACHE_MRM_HITS1WS_REG	Cache MRM (Miss Rate Monitor) HITS with 1 Wait State register (only Word (32-bits) access supported).
0x1A1C0050	CM_CACHE_RESET_REG	SWD hardware reset control register (only Word (32-bits) access supported).

Table 602: CM_CACHE_CTRL2_REG (0x1A1C0020)

Bit	Mode	Symbol/Description	Reset
31:29	-	- Reserved	0
28	R	CACHE_READY Cache Controller RO status bit. 0: Default. 1: Set to 1 when CACHE_CTRL is enabled, initialized and immediately ready for a cacheable access to service.	0
27	R	CACHE_RAM_INIT Cache Controller RO status bit. 0: Default. 1: Set to 1 when SRAM is being initialized (being flushed). Note: The flushing of the cache memory takes 256 HCLK cycles.	0
26	R/W	- Reserved	0
25:17	R/W	CACHE_EF_LEN Length of eFlash cacheable memory. N*64 kB. N = 0 to 512 (max. of 32 MB). Setting CACHE_EF_LEN = 0 disables the caching. Note 1: The max. relevant CACHE_EF_LEN setting depends on the chosen Flash region (program) size. Note 2: The first block (CACHE_EF_LEN = 1) includes the memory space specified by CACHE_EFLASH_REG[EFLASH_REGION_OFFSET].	0
16	R/W	CACHE_FLUSH_DISABLE 0: Default. 1: Flushing of the Cache memory is disabled when SYS_CTRL_REG[CACHERAM_MUX] is switched from 1 to 0. Note: Setting this bit to 1 is only allowed for debugging purposes.	0
15:14	R/W	CACHE_USE_FULL_DB_RANGE	0

Bit	Mode	Symbol/Description	Reset
		00: CACHERAM (mirrored) read/write and NO use of the full 184 bits databus (for executing program code or extension of the SysRAM with the Cache RAM). In this mode 8 bits, 16 bits and 32 bits write access is supported. 01: CACHERAM (mirrored) read and use of the full 184 bits databus of "SRAM_1_0" (for testing and debugging purposes). In this mode only 32 bits write access is supported. 10: CACHERAM (mirrored) read and use of the full 184 bits databus of "SRAM_3_2" (for testing and debugging purposes). In this mode only 32 bits write access is supported. 11: Reserved. Note 1: SYS_CTRL_REG[CACHERAM_MUX] must be set to 0 before accessing the memory mapped (mirrored) Cache Data and TAG memory. Note 2: For all three settings, max. 8 kB is available from the memory map.	
13	R/W	CACHE_MHCLKEN_DISABLE 0: Default. 1: The "m_HCLK_EN" input is ignored and the controller avoids inserting m_HTRANS = BUSY because of wait states. Note: This bit is only relevant for executing from QSPI Flash (when set to 1, it will improve performance). This bit should be kept 0 for executing from eFlash.	0
12	R/W	CACHE_CWF_DISABLE 0: Default. 1: The cache line refill is performed with INCR type burst and "Critical Word First" is disabled. Note: This bit is only relevant for executing from QSPI Flash (when set to 1, it will improve performance). This bit should be kept 0 for executing from eFlash.	0
11	R/W	- Reserved	0
10	R/W	CACHE_CGEN 0: Cache controller clock gating is not enabled. 1: Cache controller clock gating is enabled (enabling power saving).	0
9	R/W	CACHE_WEN 0: Cache Data and TAG memory read only. 1: Cache Data and TAG memory read/write. The Data and TAG memory are only updated by the cache controller. There is no hardware protection to prevent unauthorized access by the ARM. Note 1: When accessing the memory mapped Cache Data and TAG memory (which is only allowed for debugging purposes) only 32 bits access is supported. Note 2: SYS_CTRL_REG[CACHERAM_MUX] must be set to 0 before accessing the memory mapped Cache Data and TAG memory. See also the CACHE_CTRL2_REG[CACHE_USE_FULL_DB_RANGE] description.	0
8:0	R/W	CACHE_LEN Length of QSPI Flash cacheable memory.	0

Bit	Mode	Symbol/Description	Reset
		N*64 kB. N = 0 to 512 (max. of 32 MB). Setting CACHE_LEN = 0 disables the caching. Note 1: The max. relevant CACHE_LEN setting depends on the chosen Flash region (program) size. Note 2: The first block (CACHE_LEN = 1) includes the memory space specified by CACHE_FLASH_REG[FLASH_REGION_OFFSET]	

Table 603: CM_CACHE_MRM_HITS_REG (0x1A1C0028)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_HITS Contains the amount of cache hits.	0x0

Table 604: CM_CACHE_MRM_MISSES_REG (0x1A1C002C)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_MISSES Contains the amount of cache misses.	0x0

Table 605: CM_CACHE_MRM_CTRL_REG (0x1A1C0030)

Bit	Mode	Symbol/Description	Reset
31:5	-	- Reserved	0
4	R/W	MRM_IRQ_HITS_THRES_STATUS 0: No interrupt is generated. 1: Interrupt (pulse-sensitive) is generated because the number of cache hits reached the programmed threshold (threshold != 0).	0
3	R/W	MRM_IRQ_MISSES_THRES_STATUS 0: No interrupt is generated. 1: Interrupt (pulse-sensitive) is generated because the number of cache misses reached the programmed threshold (threshold != 0).	0
2	R/W	MRM_IRQ_TINT_STATUS 0: No interrupt is generated. 1: Interrupt (pulse-sensitive) is generated because the time interval counter reached the end (time interval != 0).	0
1	R/W	MRM_IRQ_MASK 0: Disables interrupt generation. 1: Enables interrupt generation. Note: The Cache MRM generates a pulse-sensitive interrupt towards the ARM processor.	0
0	R/W	MRM_START 0: Freeze the "misses/hits" counters and reset the time interval counter to the programmed value in CACHE_MRM_TINT_REG. 1: Enables the counters.	0

Bit	Mode	Symbol/Description	Reset
		Note: In case CACHE_MRM_CTRL_REG[MRM_START] is set to 1 and CACHE_MRM_TINT_REG (!=0) is used for the MRM interrupt generation, the time interval counter counts down (on a fixed reference clock of 32 MHz) until it is 0. At that time CACHE_MRM_CTRL_REG[MRM_START] will be reset automatically to 0 by the MRM hardware and the MRM interrupt will be generated.	

Table 606: CM_CACHE_MRM_TINT_REG (0x1A1C0034)

Bit	Mode	Symbol/Description	Reset
31:19	-	- Reserved	0x0
18:0	R/W	MRM_TINT Defines the time interval for the monitoring in 32 MHz clock cycles. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_TINT_STATUS]. Note: When MRM_TINT = 0 (unrealistic value), no interrupt will be generated.	0x0

Table 607: CM_CACHE_MRM_MISSES_THRES_REG (0x1A1C0038)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_MISSES_THRES Defines the misses threshold to trigger the interrupt generation. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_MISSES_THRES_STATUS]. Note: When MRM_MISSES_THRES = 0 (unrealistic value), no interrupt will be generated.	0x0

Table 608: CM_CACHE_MRM_HITS_THRES_REG (0x1A1C003C)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_HITS_THRES Defines the hits threshold to trigger the interrupt generation. See also the description of CACHE_MRM_CTRL_REG[MRM_IRQ_HITS_THRES_STATUS]. Note: When MRM_HITS_THRES = 0 (unrealistic value), no interrupt will be generated.	0x0

Table 609: CM_CACHE_FLASH_REG (0x1A1C0040)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	FLASH_REGION_BASE These bits correspond with the Flash region base address bits [31:16]. The default value is 0x1600. The Flash region base address bits [31:25] are fixed to 0x16 and bits [17:16] are fixed to 0x0. These register bits are retained. Note 1: The updated value takes effect only after a software reset.	0x1600

Bit	Mode	Symbol/Description	Reset
		Note 2 The Flash region base address setting depends on the chosen Flash region size.	
15:4	R/W	FLASH_REGION_OFFSET Flash region offset address (in words). This value is added to the Flash (CPU) address bits [13:2]. These register bits are retained. Note 1: The updated value takes effect only after a software reset.	0x0
3	R	- Reserved	0x0
2:0	R/W	FLASH_REGION_SIZE Flash region size. The default value is 6 (0.5 MB). 0 = 32 MB 1 = 16 MB 2 = 8 MB 3 = 4 MB 4 = 2 MB 5 = 1 MB 6 = 0.5 MB 7 = 0.25 MB These register bits are retained. Note 1: The updated value takes effect only after a software reset. Note 2: See for the max. region (program) size the memory map.	0x6

Table 610: CM_CACHE_EFLASH_REG (0x1A1C0044)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	EFLASH_REGION_BASE These bits correspond with the eFlash region base address bits [31:16]. The default value is 0x00A0. The eFlash region base address bits [31:24] are fixed to 0x00. These register bits are retained. Note 1: The updated value takes effect only after a software reset. Note 2 The eFlash region base address setting depends on the chosen eFlash region size.	0xA0
15:4	R/W	EFLASH_REGION_OFFSET The eFlash region offset address (in words). This value is added to the eFlash (CPU) address bits [13:2]. These register bits are retained. Note 1: The updated value takes effect only after a software reset.	0x0
3	R	- Reserved	0x0
2:0	R/W	EFLASH_REGION_SIZE The eFlash region size. The default value is 6 (128 kB). 0 = 512 kB (reserved for bigger eFlash)	0x6

Bit	Mode	Symbol/Description	Reset
		1 = 512 kB (reserved for bigger eFlash) 2 = 512 kB (reserved for bigger eFlash) 3 = 512 kB (reserved for bigger eFlash) 4 = 512 kB 5 = 256 kB 6 = 128 kB 7 = 64 kB These register bits are retained. Note 1: The updated value takes effect only after a software reset. Note 2: See for the max. region (program) size the memory map.	

Table 611: CM_CACHE_MRM_HITS1WS_REG (0x1A1C0048)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	MRM_HITS1WS Contains the amount of cache hits.	0x0

Table 612: CM_CACHE_RESET_REG (0x1A1C0050)

Bit	Mode	Symbol/Description	Reset
31:1	-	- Reserved	0
0	R0/W	SWD_HW_RESET_REQ 0: default. 1: hardware reset request (from the debugger tool). The register is automatically reset with a HW_RESET. This bit can only be accessed by the debugger software and not by the application.	0

35.25 Audio unit registers

Table 613: Register map CRG_AUD

Address	Register	Description
0x50030040	PCM_DIV_REG	PCM divider and enables
0x50030044	PCM_FDIV_REG	PCM fractional division register
0x50030048	PDM_DIV_REG	PDM divider and enables
0x5003004C	SRC_DIV_REG	SRC divider and enables

Table 614: PCM_DIV_REG (0x50030040)

Bit	Mode	Symbol/Description	Reset
13	R/W	PCM_SRC_SEL Selects the clock source 1 = DIV1 clock 0 = DIVN clock	0x0

Bit	Mode	Symbol/Description	Reset
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. Minimum value is 0x2.	0x0

Table 615: PCM_FDIV_REG (0x50030044)

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. For example, 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 616: PDM_DIV_REG (0x50030048)

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 617: SRC_DIV_REG (0x5003004C)

Bit	Mode	Symbol/Description	Reset
17	R/W	CLK_SRC2_EN Enable for the internally generated SRC2 clock The SRC2_DIV must be set before or together with CLK_SRC2_EN.	0x0
16	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
15:8	R/W	SRC2_DIV SRC2 clock divider	0x0
7:0	R/W	SRC_DIV SRC clock divider	0x0

Table 618: Register map PCM1

Address	Register	Description
0x50030300	PCM1_CTRL_REG	PCM1 Control register
0x50030304	PCM1_IN1_REG	PCM1 data in 1
0x50030308	PCM1_IN2_REG	PCM1 data in 2
0x5003030C	PCM1_OUT1_REG	PCM1 data out 1
0x50030310	PCM1_OUT2_REG	PCM1 data out 2

Table 619: PCM1_CTRL_REG (0x50030300)

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	-	- Reserved	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 clock 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

Bit	Mode	Symbol/Description	Reset
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 620: PCM1_IN1_REG (0x50030304)

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

Table 621: PCM1_IN2_REG (0x50030308)

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

Table 622: PCM1_OUT1_REG (0x5003030C)

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

Table 623: PCM1_OUT2_REG (0x50030310)

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

Table 624: Register map SRC

Address	Register	Description
0x50030100	SRC1_CTRL_REG	SRC control register
0x50030104	SRC1_IN_FS_REG	SRC Sample input rate
0x50030108	SRC1_OUT_FS_REG	SRC Sample output rate
0x5003010C	SRC1_IN1_REG	SRC data in 1
0x50030110	SRC1_IN2_REG	SRC data in 2
0x50030114	SRC1_OUT1_REG	SRC data out 1
0x50030118	SRC1_OUT2_REG	SRC data out 2
0x5003011C	SRC1_MUX_REG	SRC mux register
0x50030120	SRC1_COEF10_SET1_REG	SRC coefficient 1,0 set 1
0x50030124	SRC1_COEF32_SET1_REG	SRC coefficient 3,2 set 1
0x50030128	SRC1_COEF54_SET1_REG	SRC coefficient 5,4 set 1
0x5003012C	SRC1_COEF76_SET1_REG	SRC coefficient 7,6 set 1
0x50030130	SRC1_COEF98_SET1_REG	SRC coefficient 9,8 set 1
0x50030134	SRC1_COEF0A_SET1_REG	SRC coefficient 10 set 1
0x50030200	SRC2_CTRL_REG	SRC control register
0x50030204	SRC2_IN_FS_REG	SRC Sample input rate
0x50030208	SRC2_OUT_FS_REG	SRC Sample output rate
0x5003020C	SRC2_IN1_REG	SRC data in 1
0x50030210	SRC2_IN2_REG	SRC data in 2
0x50030214	SRC2_OUT1_REG	SRC data out 1
0x50030218	SRC2_OUT2_REG	SRC data out 2
0x5003021C	SRC2_MUX_REG	SRC mux register
0x50030220	SRC2_COEF10_SET1_REG	SRC coefficient 1,0 set 1
0x50030224	SRC2_COEF32_SET1_REG	SRC coefficient 3,2 set 1
0x50030228	SRC2_COEF54_SET1_REG	SRC coefficient 5,4 set 1
0x5003022C	SRC2_COEF76_SET1_REG	SRC coefficient 7,6 set 1
0x50030230	SRC2_COEF98_SET1_REG	SRC coefficient 9,8 set 1
0x50030234	SRC2_COEF0A_SET1_REG	SRC coefficient 10 set 1

Table 625: SRC1_CTRL_REG (0x50030100)

Bit	Mode	Symbol/Description	Reset
31:30	R/W	SRC_PDM_DO_DEL PDM_DO output delay line (typical) 0: No delay 1: 8 ns 2: 12 ns 3: 16 ns	0
29:28	R/W	SRC_PDM_MODE PDM Output mode selection on PDM_DO1 00: No output 01: Right channel (data from SRC1_IN_REG) 10: Left channel (data from SRC2_IN_REG) 11: Left and Right channel	0
27:26	R/W	SRC_PDM_DI_DEL PDM_DI input delay line (typical) 0: No delay 1: 4 ns 2: 8 ns 3: 12 ns	0
25	R0/W	SRC_OUT_FLOWCLR Writing 1 clears the SRC1_OUT Overflow/underflow bits 23-22. No more over/underflow indications while bit is 1. Keep 1 until the over/underflow bit is cleared	0
24	W	SRC_IN_FLOWCLR Writing 1 clears the SRC1_IN Overflow/underflow bits 21-20. No more over/underflow indications while bit is 1. Keep 1 until the over/underflow 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 synchronization	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 48 kHz	0

Bit	Mode	Symbol/Description	Reset
		01: For sample rates of 96 kHz 10: Reserved 11: For sample rates of 192 kHz	
15	-	- Reserved	0
14	R/W	SRC_OUT_CAL_BYPASS SRC1_OUT1 UpSampling 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	R/W	SRC_PDM_OUT_INV Swap the left and the right output PDM channel	0
11	R/W	SRC_FIFO_DIRECTION 0 = SRC FIFO is used to store samples from memory to SRC 1 = SRC FIFO is used to store sample from SRC to memory	0x0
10	R/W	SRC_FIFO_ENABLE 0 = FIFO disable. On each src request, one sample is serviced 1 = FIFO enable. FIFO is used to store samples from/to src SRC supports only DMA burst size 4 when FIFO is enabled else no burst	0x0
9	R/W	SRC_OUT_DSD_MODE 0 = SRC1 OUT PDM mode 1 = SRC1 OUT DSD mode	0x0
8	R/W	SRC_IN_DSD_MODE 0: SRC1 IN PDM mode 1: SRC1 IN DSD mode	0x0
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 48 kHz 01: For sample rates of 96 kHz 10: Reserved 11: For sample rates of 192 kHz	0
3	R/W	SRC_PDM_IN_INV	0

Bit	Mode	Symbol/Description	Reset
		Swap the left and the right input PDM channel	
2	R/W	SRC_IN_CAL_BYPASS SRC1_IN UpSampling 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 626: SRC1_IN_FS_REG (0x50030104)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0

23:0	R/W	SRC_IN_FS
		SRC_IN Sample rate
		SRC_IN_FS = SRC_DIV*4096*Sample_rate/100
		Sample_rate upper limit is 192 kHz. For 96 kHz and 192 kHz SRC_CTRLx_REG[SRC_IN_DS] must be set as shown below:
		(for SRC_DIV = 1)
		`Sample_rate SRC_IN_FS SRC_IN_DS Audio bandwidth`
		`8000 Hz 0x050000 0 4000 Hz`
		`11025 Hz 0x06E400 0 5512 Hz`
		`16000 Hz 0x0A0000 0 8000 Hz`
		`22050 Hz 0x0DC800 0 11025 Hz`
		`32000 Hz 0x140000 0 16000 Hz`
		`44100 Hz 0x1B9000 0 22050 Hz`
		`48000 Hz 0x1E0000 0 24000 Hz`
		`96000 Hz 0x1E0000 1 24000 Hz`
		`192000 Hz 0x1E0000 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 192 kHz.
			0

Table 627: SRC1_OUT_FS_REG (0x50030108)

Bit	Mode	Symbol/Description	Reset
31:24	-	-	0

Bit	Mode	Symbol/Description	Reset
		Reserved	
23:0	R/W	SRC_OUT_FS	0

		SRC_OUT Sample rate
		SRC_OUT_FS = SRC_DIV*4096*Sample_rate/100
		Sample_rate upper limit is 192 kHz. For 96 kHz and 192 kHz SRC_CTRLx_REG[SRC_DS] must be set as shown below:
		(for SRC_DIV = 1)
		`Sample_rate SRC_OUT_FS  SRC_OUT_DS Audio bandwidth`
		`8000 Hz     0x050000    0          4000 Hz`
		`11025 Hz    0x06E400    0          5512 Hz`
		`16000 Hz    0x0A0000    0          8000 Hz`
		`22050 Hz    0x0DC800    0          11025 Hz`
		`32000 Hz    0x140000    0          16000 Hz`
		`44100 Hz    0x1B9000    0          22050 Hz`
		`48000 Hz    0x1E0000    0          24000 Hz`
		`96000 Hz    0x1E0000    1          24000 Hz`
		`192000 Hz   0x1E0000    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 192 kHz.	

Table 628: SRC1_IN1_REG (0x5003010C)

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

Table 629: SRC1_IN2_REG (0x50030110)

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

Table 630: SRC1_OUT1_REG (0x50030114)

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

Table 631: SRC1_OUT2_REG (0x50030118)

Bit	Mode	Symbol/Description	Reset
31:0	R	SRC_OUT	0

Bit	Mode	Symbol/Description	Reset
		SRC1_OUT2	

Table 632: SRC1_MUX_REG (0x5003011C)

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 3 = SRC2 output	0x0
2:0	R/W	SRC1_MUX_IN SRC1 input mux 0 = off 1 = PCM output 2 = SRC1 input registers 3 = SDADC output	0x0

Table 633: SRC1_COEF10_SET1_REG (0x50030120)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF1 Coefficient 1	0x7A20
15:0	R/W	SRC_COEF0 Coefficient 0	0x8EC4

Table 634: SRC1_COEF32_SET1_REG (0x50030124)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF3 Coefficient 3	0x70FD
15:0	R/W	SRC_COEF2 Coefficient 2	0x8936

Table 635: SRC1_COEF54_SET1_REG (0x50030128)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF5 Coefficient 5	0x9758
15:0	R/W	SRC_COEF4	0xB686

Bit	Mode	Symbol/Description	Reset
		Coefficient 4	

Table 636: SRC1_COEF76_SET1_REG (0x5003012C)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF7 Coefficient 7	0x89C4
15:0	R/W	SRC_COEF6 Coefficient 6	0x7DF5

Table 637: SRC1_COEF98_SET1_REG (0x50030130)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF9 Coefficient 9	0x8F18
15:0	R/W	SRC_COEF8 Coefficient 8	0x7771

Table 638: SRC1_COEF0A_SET1_REG (0x50030134)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	SRC_COEF10 Coefficient 10	0x497D

Table 639: SRC2_CTRL_REG (0x50030200)

Bit	Mode	Symbol/Description	Reset
31:30	R/W	- Reserved	0
29:28	R/W	SRC_PDM_MODE PDM Output mode selection on PDM_DO1 00: No output 01: Right channel (data from SRC1_IN_REG) 10: Left channel (data from SRC2_IN_REG) 11: Left and Right channel	0
27:26	R/W	- Reserved	0
25	R0/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/underflow bit is cleared	0
24	W	SRC_IN_FLOWCLR	0

Bit	Mode	Symbol/Description	Reset
		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/underflow bit is cleared	
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 synchronization	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 48 kHz 01: For sample rates of 96 kHz 10: Reserved 11: For sample rates of 192 kHz	0
15	-	- Reserved	0
14	R/W	SRC_OUT_CAL_BYPASS SRC1_OUT1 UpSampling 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	R/W	SRC_PDM_OUT_INV Swap the left and the right output PDM channel	0
11	R/W	SRC_FIFO_DIRECTION 0 = SRC FIFO is used to store samples from memory to SRC 1 = SRC FIFO is used to store sample from SRC to memory	0x0
10	R/W	SRC_FIFO_ENABLE 0 = FIFO disable. On each src request, one sample is serviced 1 = FIFO enable. FIFO is used to store samples from/to src SRC supports only DMA burst size 4 when FIFO is enabled, else no burst	0x0
9	R/W	SRC_OUT_DSD_MODE	0x0

Bit	Mode	Symbol/Description	Reset
		0 = SRC1 OUT PDM mode 1 = SRC1 OUT DSD mode	
8	R/W	SRC_IN_DSD_MODE 0: SRC1 IN PDM mode 1: SRC1 IN DSD mode	0x0
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 48 kHz 01: For sample rates of 96 kHz 10: Reserved 11: For sample rates of 192 kHz	0
3	R/W	SRC_PDM_IN_INV Swap the left and the right input PDM channel	0
2	R/W	SRC_IN_CAL_BYPASS SRC1_IN UpSampling 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 640: SRC2_IN_FS_REG (0x50030204)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0
23:0	R/W	SRC_IN_FS SRC_IN Sample rate SRC_IN_FS = SRC_DIV*4096*Sample_rate/100 Sample_rate upper limit is 192 kHz. For 96 kHz and 192 kHz SRC_CTRLx_REG[SRC_IN_DS] must be set as shown below:	0

Bit	Mode	Symbol/Description	Reset

		(for SRC_DIV = 1)
		`Sample_rate SRC_IN_FS SRC_IN_DS Audio bandwidth`
		`8000 Hz 0x050000 0 4000 Hz`
		`11025 Hz 0x06E400 0 5512 Hz`
		`16000 Hz 0x0A0000 0 8000 Hz`
		`22050 Hz 0x0DC800 0 11025 Hz`
		`32000 Hz 0x140000 0 16000 Hz`
		`44100 Hz 0x1B9000 0 22050 Hz`
		`48000 Hz 0x1E0000 0 24000 Hz`
		`96000 Hz 0x1E0000 1 24000 Hz`
		`192000 Hz 0x1E0000 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 192 kHz.	

Table 641: SRC2_OUT_FS_REG (0x50030208)

Bit	Mode	Symbol/Description	Reset
31:24	-	- Reserved	0

23:0	R/W	SRC_OUT_FS
		SRC_OUT Sample rate
		SRC_OUT_FS = SRC_DIV*4096*Sample_rate/100
		Sample_rate upper limit is 192 kHz. For 96 kHz and 192 kHz SRC_CTRLx_REG[SRC_DS] must be set as shown below:
		(for SRC_DIV = 1)
		`Sample_rate SRC_OUT_FS SRC_OUT_DS Audio bandwidth`
		`8000 Hz 0x050000 0 4000 Hz`
		`11025 Hz 0x06E400 0 5512 Hz`
		`16000 Hz 0x0A0000 0 8000 Hz`
		`22050 Hz 0x0DC800 0 11025 Hz`
		`32000 Hz 0x140000 0 16000 Hz`
		`44100 Hz 0x1B9000 0 22050 Hz`
		`48000 Hz 0x1E0000 0 24000 Hz`
		`96000 Hz 0x1E0000 1 24000 Hz`
		`192000 Hz 0x1E0000 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 192 kHz.
			0

Table 642: SRC2_IN1_REG (0x5003020C)

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

Table 643: SRC2_IN2_REG (0x50030210)

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

Table 644: SRC2_OUT1_REG (0x50030214)

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

Table 645: SRC2_OUT2_REG (0x50030218)

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

Table 646: SRC2_MUX_REG (0x5003021C)

Bit	Mode	Symbol/Description	Reset
6	R/W	PDM1_MUX_IN PDM1 input mux 0 = SRC2_MUX_IN 1 = PDM input	0x0
5:3	R/W	PDM_MUX_OUT PDM output mux 0 = SRC1 PDM output 1 = SRC2 PDM output 2..7 = Reserved	0x0
2:0	R/W	SRC2_MUX_IN SRC1 input mux 0 = off 1 = PCM output 2 = SRC2 input registers 3 = SDADC output	0x0

Table 647: SRC2_COEF10_SET1_REG (0x50030220)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF1 Coefficient 1	0x7A20
15:0	R/W	SRC_COEF0 Coefficient 0	0x8EC4

Table 648: SRC2_COEF32_SET1_REG (0x50030224)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF3 Coefficient 3	0x70FD
15:0	R/W	SRC_COEF2 Coefficient 2	0x8936

Table 649: SRC2_COEF54_SET1_REG (0x50030228)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF5 Coefficient 5	0x9758
15:0	R/W	SRC_COEF4 Coefficient 4	0xB686

Table 650: SRC2_COEF76_SET1_REG (0x5003022C)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF7 Coefficient 7	0x89C4
15:0	R/W	SRC_COEF6 Coefficient 6	0x7DF5

Table 651: SRC2_COEF98_SET1_REG (0x50030230)

Bit	Mode	Symbol/Description	Reset
31:16	R/W	SRC_COEF9 Coefficient 9	0x8F18
15:0	R/W	SRC_COEF8 Coefficient 8	0x7771

Table 652: SRC2_COEF0A_SET1_REG (0x50030234)

Bit	Mode	Symbol/Description	Reset
15:0	R/W	SRC_COEF10	0x497D

Bit	Mode	Symbol/Description	Reset
		Coefficient 10	

35.26 Analog miscellaneous registers

Table 653: Register map ANAMISC

Address	Register	Description
0x50040B10	CLK_REF_SEL_REG	Select clock for oscillator calibration
0x50040B14	CLK_REF_CNT_REG	Count value for oscillator calibration
0x50040B18	CLK_REF_VAL_REG	DIVN reference cycles, lower 16 bits
0x50040B1C	CLK_CAL_IRQ_REG	Select clock for oscillator calibration

Table 654: CLK_REF_SEL_REG (0x50040B10)

Bit	Mode	Symbol/Description	Reset
7:5	R/W	CAL_CLK_SEL Select calibration clock input to be used in calibration: 0x0: DIVN clock 0x1: RCLP 0x2: RC32M 0x3: XTAL32K 0x4: XTAL32M_VARICAP_TEST 0x5, 0x6 and 0x7: Reserved	0x0
4	R/W	EXT_CNT_EN_SEL 0: Enable XTAL_CNT counter by the REF_CLK selected by REF_CLK_SEL. 1: Enable XTAL_CNT counter from an external input.	0x0
3	R/W	REF_CAL_START Writing 1 starts calibration. This bit is cleared when calibration is finished, and CLK_REF_VAL is ready.	0x0
2:0	R/W	REF_CLK_SEL Select reference clock input for calibration: 0x0: RCLP 0x1: RC32M 0x2: XTAL32K 0x3: RCX 0x4: XTAL32M_VARICAP_TEST 0x5: DIVN clock	0x0

Table 655: CLK_REF_CNT_REG (0x50040B14)

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

Table 656: CLK_REF_VAL_REG (0x50040B18)

Bit	Mode	Symbol/Description	Reset
31:0	R	XTAL_CNT_VAL Returns the number of DIVN clock cycles counted during the calibration time, defined with REF_CNT_VAL	0x0

Table 657: CLK_CAL_IRQ_REG (0x50040B1C)

Bit	Mode	Symbol/Description	Reset
3	R/W	- Reserved	0x0
2	R0/WC	CLK_CAL_IRQ_CLR Clear the IRQ. 1: Clear the IRQ 0: No effect. Read out 0 always.	0x0
1	R	CLK_CAL_IRQ_STATUS Shows the IRQ bit status.	0x0
0	R/W	CLK_CAL_IRQ_EN Enable clk calibration IRQ. 0: Disabled. 1*: Enabled. *Note: If IRQ feature is enabled, every calibration should be started by setting CLK_REF_SEL_REG[REF_CAL_START] to 1.	0x0

35.27 Crypto-engine registers

Table 658: Register map AES_HASH

Address	Register	Description
0x30040000	CRYPTO_CTRL_REG	Crypto Control register
0x30040004	CRYPTO_START_REG	Crypto Start calculation
0x30040008	CRYPTO_FETCH_ADDR_REG	Crypto DMA fetch register
0x3004000C	CRYPTO_LEN_REG	Crypto Length of the input block in bytes
0x30040010	CRYPTO_DEST_ADDR_REG	Crypto DMA destination memory
0x30040014	CRYPTO_STATUS_REG	Crypto Status register
0x30040018	CRYPTO_CLRIRQ_REG	Crypto Clear interrupt request
0x3004001C	CRYPTO_MREG0_REG	Crypto Mode depended register 0
0x30040020	CRYPTO_MREG1_REG	Crypto Mode depended register 1

Address	Register	Description
0x30040024	CRYPTO_MREG2_REG	Crypto Mode depended register 2
0x30040028	CRYPTO_MREG3_REG	Crypto Mode depended register 3
0x30040100	CRYPTO_KEYS_START	Crypto First position of the AES keys storage memory

Table 659: CRYPTO_CTRL_REG (0x30040000)

Bit	Mode	Symbol/Description	Reset
16	R/W	CRYPTO_AES_KEXP It forces (active high) the execution of the key expansion process with the starting of the AES encryption/decryption process. The bit will be cleared automatically by the hardware, after the completion of the AES key expansion process.	0x0
15	R/W	CRYPTO_MORE_IN 0: Define that this is the last input block. When the current input is consumed by the crypto engine and the output data is written to the memory, the calculation ends (CRYPTO_INACTIVE goes to one). 1: The current input data block is not the last. More input data will follow. When the current input is consumed, the engine stops and waits for more data (CRYPTO_WAIT_FOR_IN goes to one).	0x0
14:10	R/W	CRYPTO_HASH_OUT_LEN The number of bytes minus one of the hash results which will be saved at the memory by the DMA. In relation with the selected hash algorithm the accepted values are: SHA-256: 0…31 -> 1 - 32 bytes SHA-256/224: 0…27 -> 1- 28 bytes	0x0
9	R/W	CRYPTO_HASH_SEL Selects the type of the algorithm 0: The encryption algorithm (AES) 1: A hash algorithm. The exact algorithm is defined by the CRYPTO_ALG and CRYPTO_ALG_MD fields.	0x0
8	R/W	CRYPTO_IRQ_EN Interrupt Request Enable 0: The interrupt generation ability is disabled. 1: The interrupt generation ability is enabled. Generates an interrupt request at the end of operation.	0x0
7	R/W	CRYPTO_ENCDEC Encryption/Decryption 0: Decryption 1: Encryption	0x0
6:5	R/W	CRYPTO_AES_KEY_SZ The size of AES Key 0x0: 128 bits AES Key 0x1: 192 bits AES Key 0x2: 256 bits AES Key	0x0

Bit	Mode	Symbol/Description	Reset
		0x3: 256 bits AES Key	
4	R/W	CRYPTO_OUT_MD Output Mode. This field makes sense only when the AES algorithm is selected (CRYPTO_HASH_SEL = 0) 0: Write back to memory all the resulting data 1: Write back to memory only the final block of the resulting data	0x0
3:2	R/W	CRYPTO_ALG_MD It defines the mode of operation of the AES algorithm when the controller is configured for an encryption/decryption processing (CRYPTO_HASH_SEL = 0). 0x0: ECB 0x1: ECB 0x2: CTR 0x3: CBC	0x0
1:0	R/W	CRYPTO_ALG Algorithm selection. When CRYPTO_HASH_SEL = 0, the only available choice is the AES algorithm. 0x0: AES 0x1: Reserved 0x2: Reserved 0x3: Reserved When CRYPTO_HASH_SEL = 1, this field selects the desired hash algorithm. 0x0 or 0x2: SHA-256/224 0x1 or 0x3: SHA-256	0x0

Table 660: CRYPTO_START_REG (0x30040004)

Bit	Mode	Symbol/Description	Reset
0	R0/W	CRYPTO_START Write 1 to initiate the processing of the input data. This register is auto-cleared.	0x0

Table 661: CRYPTO_FETCH_ADDR_REG (0x30040008)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	CRYPTO_FETCH_ADDR The memory address from where will be retrieved the data that will be processed. The value of this register is updated as the calculation proceeds, and the output data are written to the memory.	0x0

Table 662: CRYPTO_LEN_REG (0x3004000C)

Bit	Mode	Symbol/Description	Reset
23:0	R/W	CRYPTO_LEN It contains the number of bytes of input data. If this number is not a multiple of a block size, the data is automatically extended with zeros. The	0x0

Bit	Mode	Symbol/Description	Reset
		value of this register is updated as the calculation proceeds, and the output data are written to the memory.	

Table 663: CRYPTO_DEST_ADDR_REG (0x30040010)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	CRYPTO_DEST_ADDR Destination address at where the result of the processing is stored. The value of this register is updated as the calculation proceeds, and the output data are written to the memory.	0x0

Table 664: CRYPTO_STATUS_REG (0x30040014)

Bit	Mode	Symbol/Description	Reset
2	R	CRYPTO_IRQ_ST The status of the interrupt request line of the CRYPTO block. 0: There is no active interrupt request. 1: An interrupt request is pending.	0x0
1	R	CRYPTO_WAIT_FOR_IN Indicates the situation where the engine waits for more input data. This is applicable when the CRYPTO_MORE_IN = 1, so the input data are fragmented in the memory. 0: The crypto is not waiting for more input data. 1: The crypto waits for more input data. The CRYPTO_INACTIVE flag remains zero to indicate that the calculation is not finished. The supervisor of the CRYPTO must program a new input data fragment to CRYPTO_FETCH_ADDR and CRYPTO_LEN. The calculation is continued as soon as the CRYPTO_START register is written with 1. This action clears the CRYPTO_WAIT_FOR_IN flag.	0x0
0	R	CRYPTO_INACTIVE 0: The CRYPTO is active. The processing is in progress. 1: The CRYPTO is inactive. The processing has finished.	0x1

Table 665: CRYPTO_CLRIRQ_REG (0x30040018)

Bit	Mode	Symbol/Description	Reset
0	R0/W	CRYPTO_CLRIRQ Write 1 to clear a pending interrupt request.	0x0

Table 666: CRYPTO_MREG0_REG (0x3004001C)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	CRYPTO_MREG0 It contains information that are depended by the mode of operation, when is used the AES algorithm: CBC - IV[31:0] CTR - CTRBLK[31:0]. It is the initial value of the 32 bits counter.	0x0

Bit	Mode	Symbol/Description	Reset
		At any other mode, the contents of this register have no meaning.	

Table 667: CRYPTO_MREG1_REG (0x30040020)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	CRYPTO_MREG1 It contains information that are depended by the mode of operation, when is used the AES algorithm: CBC - IV[63:32] CTR - CTRBLK[63:32] At any other mode, the contents of this register have no meaning.	0x0

Table 668: CRYPTO_MREG2_REG (0x30040024)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	CRYPTO_MREG2 It contains information that are depended by the mode of operation, when is used the AES algorithm: CBC - IV[95:64] CTR - CTRBLK[95:64] At any other mode, the contents of this register have no meaning.	0x0

Table 669: CRYPTO_MREG3_REG (0x30040028)

Bit	Mode	Symbol/Description	Reset
31:0	R/W	CRYPTO_MREG3 It contains information that are depended by the mode of operation, when is used the AES algorithm: CBC - IV[127:96] CTR - CTRBLK[127:96] At any other mode, the contents of this register have no meaning.	0x0

Table 670: CRYPTO_KEYS_START (0x30040100)

Bit	Mode	Symbol/Description	Reset
31:0	W	CRYPTO_KEY_X CRYPTO_KEY_(0-63) This is the AES keys storage memory. This memory is accessible via AHB slave interface, only when the CRYPTO is inactive (CRYPTO_INACTIVE = 1).	N/A

36 Ordering Information

Table 671: Ordering Information (Samples)

Part Number	Package	Size (mm)	Shipment Form	Pack Quantity
DA14592-01000O92	WLCSP39	3.32 x 2.48 x 0.37	Reel	100/1000
DA14592-010006F2	FCQFN52	5.1 x 4.3 x 0.78	Reel	100/1000
DA14592-02000O92	WLCSP39	3.32 x 2.48 x 0.37	Contact sales for availability	
DA14592-020006F2	FCQFN52	5.1 x 4.3 x 0.78	Contact sales for availability	

Table 672: Ordering Information (Production)

Part Number	Package	Size (mm)	Shipment Form	Pack Quantity
DA14592-01000O92	WLCSP39	3.32 x 2.48 x 0.37	Reel	8000
DA14592-010006F2	FCQFN52	5.1 x 4.3 x 0.78	Reel	5500
DA14592-02000O92	WLCSP39	3.32 x 2.48 x 0.37	Contact sales for availability	
DA14592-020006F2	FCQFN52	5.1 x 4.3 x 0.78	Contact sales for availability	

Part number legend:

DA14592-RRXXXYYZ

RR: chip revision number

XXX: variant (000: Standard)

YY: package code (O9: WLCSP39, 6F: FCQFN52)

Z: packing method (1: Tray, 2: Reel, A: Mini-Reel)

37 Package Information

37.1 Moisture sensitivity level

The moisture sensitivity level (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.

FCQFN 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

37.2 WLCSP handling

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

Removal will cause damage to the solder balls, and therefore you cannot reuse a removed sample.

WLCSP is sensitive to visible and infrared light. Take precautions to properly shield the chip in the final product.

37.3 Soldering information

See the JEDEC standard J-STD-020 for relevant soldering information. This document can be downloaded from http://www.jedec.org.

37.4 Package outline drawings

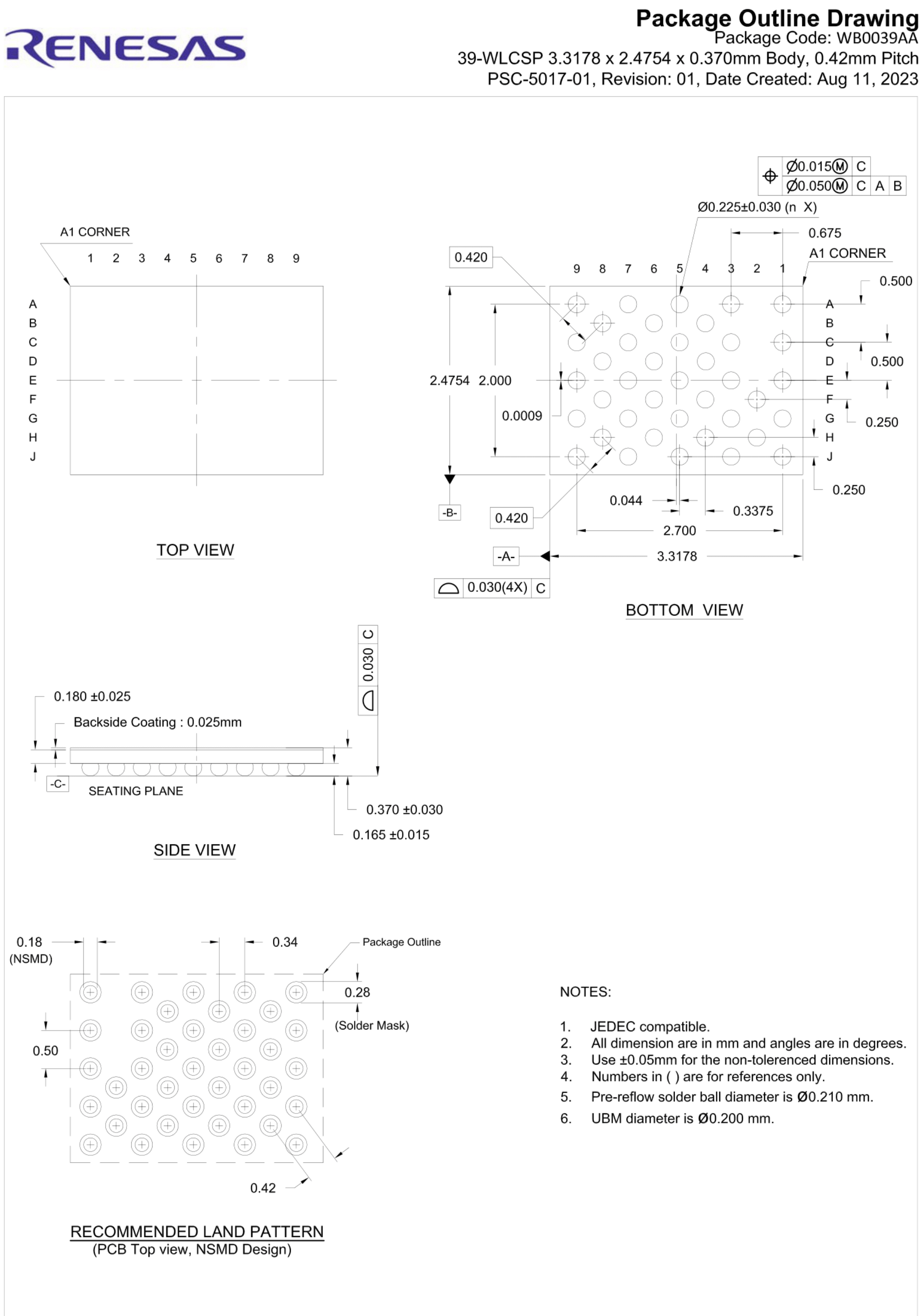

Figure 86. WLCSP39 package outline drawing

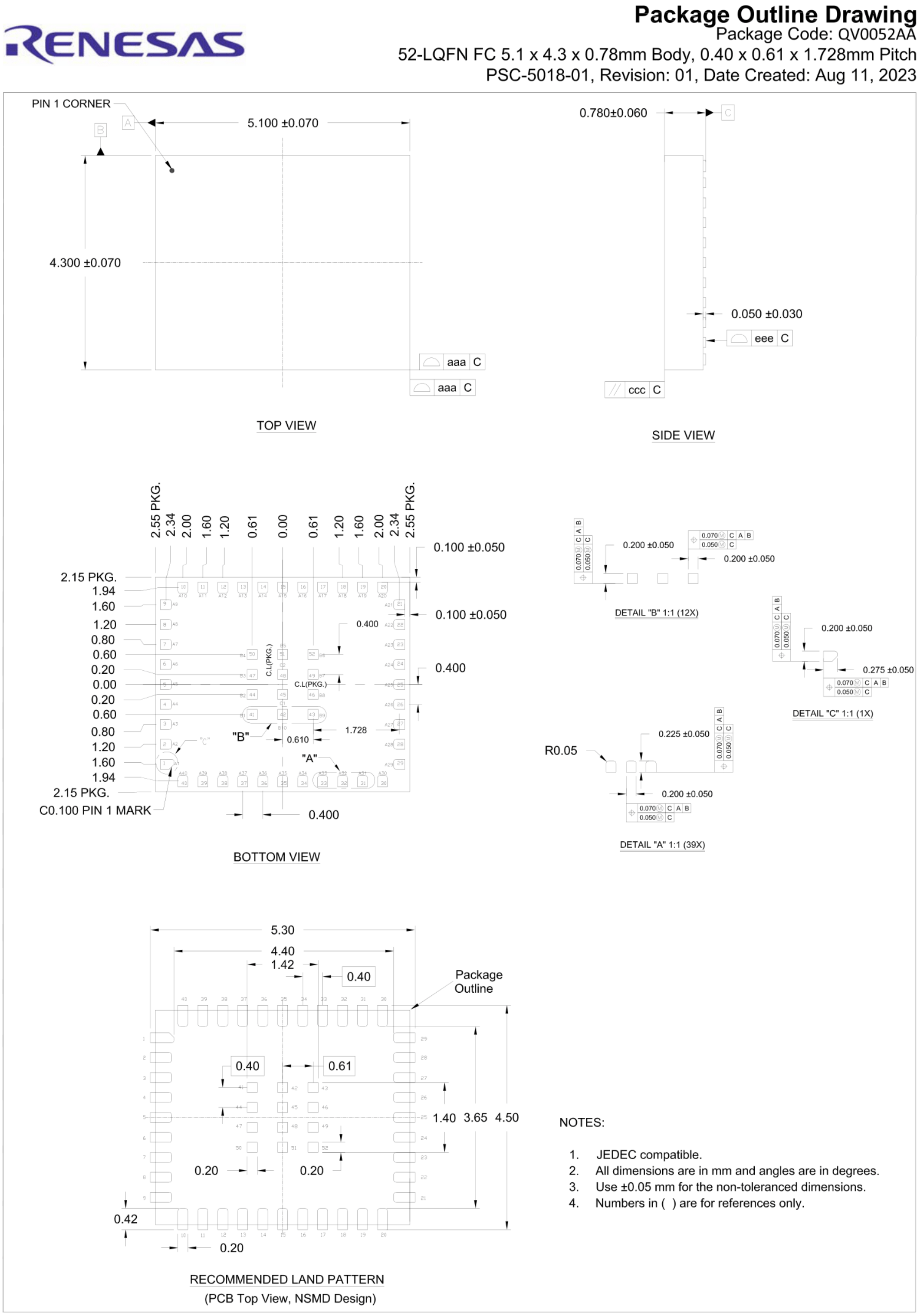

Figure 87. FCQFN52 package outline drawing

Revision History

Revision	Date	Description
3.0	Dec 28, 2023	Datasheet Status: Final, Product Status: Production
2.0	Nov 22, 2022	Datasheet Status: Preliminary, Product Status: Qualification
1.2	Oct 06, 2020	Datasheet Status: Target, Product Status: Development
1.1	July 02, 2020	Datasheet Status: Target, Product Status: Development
1.0	Sep 27, 2019	Initial version. Product Status: Development

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.renesas.com.
4.<n>	Obsolete	Archived	This datasheet contains the specifications for discontinued products. The information is provided for reference only.

RoHS Compliance

Renesas 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.

Important Notice and Disclaimer

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu

Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

https://www.renesas.com/contact/

Trademarks