GR551x Datasheet

Version: 2.5

Release Date: 2023-01-19

Shenzhen Goodix Technology Co., Ltd.

Contents

1 GR551x Overview

The Goodix GR551x family is a single-mode, low-power Bluetooth 5.1 System-on-Chip (SoC). It can be configured as a Broadcaster, an Observer, a Central, or a Peripheral and supports the combination of all the above roles, making it an ideal choice for Internet of Things (IoT) and smart wearable devices.

Based on ARM® Cortex®-M4F CPU core, the GR551x integrates Bluetooth 5.1 Protocol Stack, a 2.4 GHz RF transceiver, on-chip programmable Flash memory, RAM, and multiple peripherals.

GR551x SoCs are available in multiple packages (see Table 1-1) that meet your diverse project demands.

Table 1-1 GR551x series

Features	GR5515IGND	GR5515IENDU	GR5515I0NDA	GR5515RGBD	GR5515GGBD	GR5513BEND	GR5513BENDU
CPU	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F
RAM	256 KB	256 KB	256 KB	256 KB	256 KB	128 KB	128 KB
Flash	1 MB	512 KB	N/A	1 MB	1 MB	512 KB	512 KB
Package (mm)	QFN56 (7 x 7 x 0.75)	QFN56 (7 x 7 x 0.75)	QFN56 (7 x 7 x 0.75)	BGA68 (5.3 x 5.3 x 0.88)	BGA55 (3.5 x 3.5 x 0.60)	QFN40 (5 x 5 x 0.75)	QFN40 (5 x 5 x 0.75)
I/O Number	39	39	39	39	29	22	22

Note:

- GR5515IENDU and GR5513BENDU are embedded with wide-voltage Flash, with Flash power supply from 1.65 V to 3.6 V.
- GR5515I0NDA supports external wide-voltage Flash (range: 1.65 V–3.6 V).

1.1 Features

- A Bluetooth Low Energy (Bluetooth LE) 5.1 transceiver integrates Controller and Host layers
  - Supported data rates: 1 Mbps, 2 Mbps, Long Range 500 kbps, Long Range 125 kbps
  - TX power: -20 dBm to +7 dBm
  - -97 dBm sensitivity (in 1 Mbps mode)
  - -93 dBm sensitivity (in 2 Mbps mode)
  - -99.5 dBm sensitivity (in Long Range 500 kbps mode)
  - -103 dBm sensitivity (in Long Range 125 kbps mode)
  - TX current: 3.05 mA @ 0 dBm, 1 Mbps
  - RX current: 3.9 mA @ 1 Mbps
- ARM® Cortex®-M4F 32-bit micro-processor with floating point support

- Maximum frequency: 64 MHz
- Power consumption: 30 µA/MHz

- Memory
  - 256 KB RAM with retention (four 8 KB RAM blocks and seven 32 KB RAM blocks) for GR5515 series SoCs, and 128 KB RAM with retention (four 8 KB RAM blocks and three 32 KB RAM blocks) for the GR5513 SoC
  - 1 MB Flash for GR5515 series SoCs and 512 KB Flash for the GR5513 SoC (exceptions: GR5515I0NDA requiring external QSPI Flash and GR5515IENDU requiring 512 KB embedded Flash)
- Power management
  - On-chip DC-DC Converter
  - On-chip I/O LDO to provide I/O voltage and supply external components
  - Supply voltage: 1.7 V to 3.8 V. The supply voltage of GR5515I0NDA (when the external Flash of GR5515I0NDA supplied by high voltage) shall equal the working voltage of external QSPI Flash
  - I/O voltage: 1.8 V to 3.3 V (Typical) (for GR5515I0NDA/GR5515IENDU/GR5513BENDU Flash using high voltage, the VIO_LDO_OUT shall be connected to VBATL in schematic circuit.)
  - OFF mode: 0.15 µA (Typical); nothing is on except VBAT, chip in reset mode
  - Ultra deep sleep mode: 0.65 µA (Typical); I/O LDO off, no memory retention; woken up by AON GPIO or AON Timer
  - Sleep mode: 1.3 µA (Typical) when I/O LDO and BOD/BOD2 turned off, lowest voltage level for memory retention, and 32 KB memory retention; woken up by AON_RTC, AON GPIO, or Bluetooth LE Event
- Peripherals
  - 2 x QSPI interfaces, up to 32 MHz
  - 2 x SPI interfaces (1 SPI Master Interface with 2 slave CS pins + 1 SPI Slave Interface), up to 32 MHz
  - 2 x I2C interfaces at 100 kHz, 400 kHz, 1 MHz, 2 MHz
  - 2 x I2S interfaces (1 I2S Master Interface + 1 I2S Slave Interface)
  - 2 x UART interfaces, one with DMA channel.
  - 13-bit ADC, up to 1 Msps, 8 channels (5 external test channels and 3 internal signal channels), supporting both single-ended and differential inputs
  - ISO 7816 interface
  - 6-channel PWM
  - Built-in temperature and voltage sensors
  - 4 x Hardware timers
  - 1 x AON hardware timer

  - 2 watchdog timers (one for the system and one is always-on)
  - Calendar timer
  - Wake-up comparator
  - Up to 39 multiplexed GPIO pins
- Security
  - Complete secure computing engine:
    - AES 128-bit/192-bit/256-bit symmetric encryption (ECB, CBC)
    - Keyed Hash Message Authentication Code (HMAC)
    - PKC
    - TRNG
  - Comprehensive security operation mechanism:
    - Secure boot
    - Encrypted firmware runs directly from Flash
    - eFuse for encrypted key storage
    - Differentiate application data key and firmware key, supporting one data key per device/product
- Packages
  - QFN56: 7 mm x 7 mm
  - BGA68: 5.3 mm x 5.3 mm
  - BGA55: 3.5 mm x 3.5 mm
  - QFN40: 5 mm x 5 mm
- Operating temperature range: -40°C to +85°C

1.2 Block Diagram

Figure 1-1 shows the block diagram of GR551x.

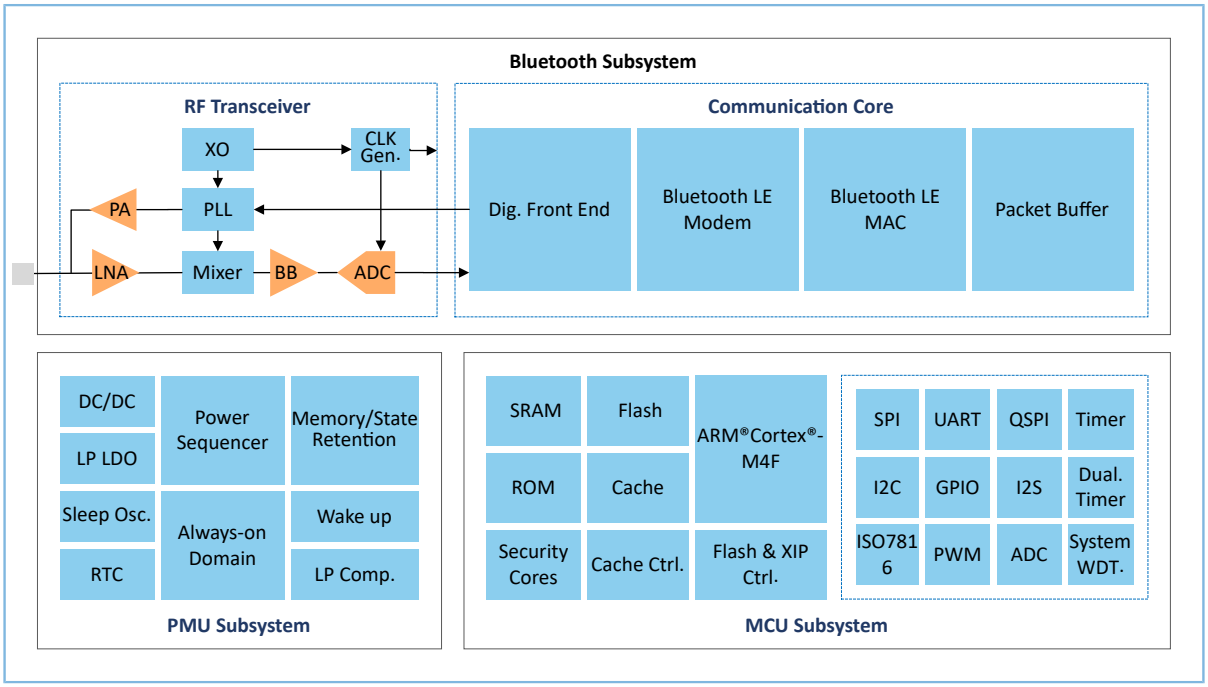

Figure 1-1 GR551x block diagram

- Bluetooth Subsystem
  - A 2.4 GHz transceiver and a digital communication core that supports Bluetooth LE 5.1.
- MCU Subsystem
  - An ARM® Cortex®-M4F with all the required memories and peripherals.
  - Security cores that can be used for application security and to secure boot implementation.
- Power Management Unit (PMU) Subsystem
  - All the required power management modules to supply sufficient power to both internal modules and external peripherals.
  - Modules required for ultra-low-power operation in standby mode. Sleep clock (Sleep Osc.), wakeup GPIOs (Wake up), low-power comparator (LP Comp.), and power state controller (Power Sequencer) are used to control the state of different modules.

1.3 Applications

GR551x can be used in rich sets of applications. Examples of these applications include:

- Wearables
  - Bluetooth LE health sensor applications
  - Bluetooth LE sports sensor applications
- Bluetooth HID devices
  - Voice remote control

- Bluetooth LE keyboard/mouse
- Bluetooth LE gamepad

- Smart home and industrial applications
  - Smart lock
  - Lighting and Smart home
  - E-shelf label
  - Beacon
  - Tire pressure monitoring system (TPMS)
  - Mesh applications

2 Pinout

This chapter introduces GR551x pinout available in multiple packages and provides detailed descriptions.

2.1 GR5515IGND/GR5515IENDU QFN56

Figure 2-1 shows the pin assignments of GR5515IGND/GR5515IENDU QFN56 package (top view).

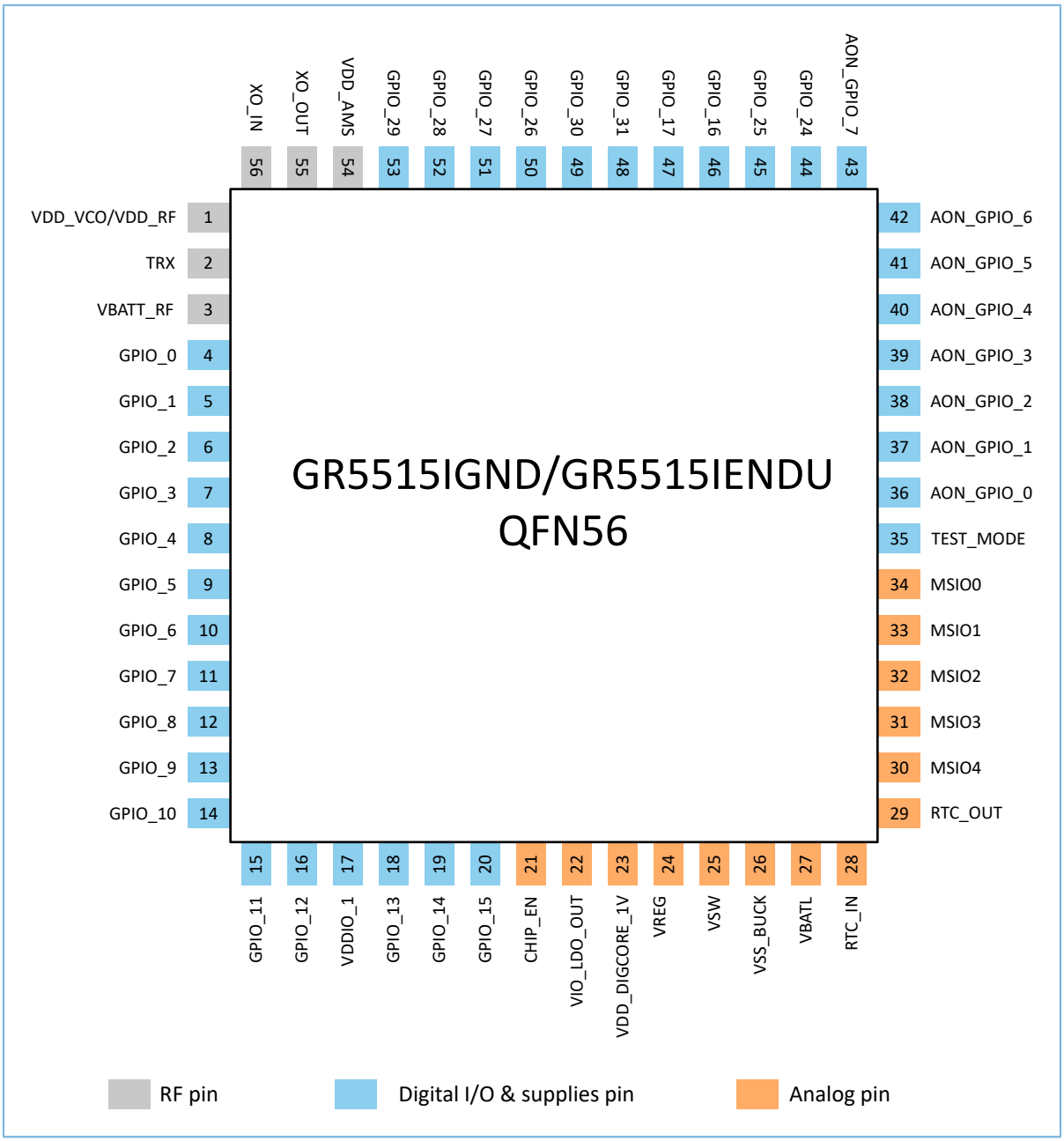

Figure 2-1 GR5515IGND/GR5515IENDU QFN56 package pinout

Table 2-1 shows pin descriptions of GR5515IGND/GR5515IENDU QFN56 package.

Table 2-1 GR5515IGND/GR5515IENDU QFN56 pin descriptions

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
1	VDD_VCO/VDD_RF	Analog/RF supply	Synthesizer VCO supply. RF supply. Connect to VREG.	
2	TRX	Analog/RF	RX input and TX output	
3	VBATT_RF	Analog/RF Supply	Connect to VBATL.	
4	GPIO_0	Digital I/O	SWDCLK	VDDIO1
5	GPIO_1	Digital I/O	SWDIO	VDDIO1
6	GPIO_2	Digital I/O	General purpose I/O	VDDIO1
7	GPIO_3	Digital I/O	General purpose I/O	VDDIO1
8	GPIO_4	Digital I/O	General purpose I/O	VDDIO1
9	GPIO_5	Digital I/O	General purpose I/O	VDDIO1
10	GPIO_6	Digital I/O	General purpose I/O	VDDIO1
11	GPIO_7	Digital I/O	General purpose I/O	VDDIO1
12	GPIO_8	Digital I/O	General purpose I/O	VDDIO1
13	GPIO_9	Digital I/O	General purpose I/O	VDDIO1
14	GPIO_10	Digital I/O	General purpose I/O	VDDIO1
15	GPIO_11	Digital I/O	General purpose I/O	VDDIO1
16	GPIO_12	Digital I/O	General purpose I/O	VDDIO1
17	VDDIO_1	Digital I/O supply	Digital I/O supply input	VDDIO1
18	GPIO_13	Digital I/O	General purpose I/O	VDDIO1
19	GPIO_14	Digital I/O	General purpose I/O	VDDIO1
20	GPIO_15	Digital I/O	General purpose I/O	VDDIO1
21	CHIP_EN	Mixed Signal IN	Master Enable for chip reset pin. Minimum value of high level for CHIP_EN is 1 V.	
22	VIO_LDO_OUT	PMU	Output of on-chip I/O supply regulator When GR5515IENDU is used and the Flash is supplied at a high voltage, the pin is used as the power input pin of VDDIO0 digital I/O domain by being connected to VBATL	Connected internally to VDDIO0
23	VDD_DIGCORE_1V	PMU	Output of on-chip LDO for digital core. Connect to a 1 µF capacitor.	
24	VREG	PMU	Feedback pin from switching regulator	
25	VSW	PMU	DC-DC converter switching node	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
26	VSS_BUCK	PMU	DC-DC converter supply and general battery GND	
27	VBATL	PMU	Power supply input	
28	RTC_IN	Analog/PMU	Input of inverting amplifier connected to 32.768 kHz crystal	
29	RTC_OUT	Analog/PMU	Output of inverting amplifier connected to 32.768 kHz crystal	
30	MSIO4	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
31	MSIO3	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
32	MSIO2	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
33	MSIO1	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
34	MSIO0	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
35	TEST_MODE	Digital I/O	Input pin, used to set test mode for FT or CP factory test. In the application phase, the value is set to 0 by default. • If TEST_MODE = 1, the chip is in test mode for factory test. • If TEST_MODE = 0, the chip is in normal operation mode.	VDDIO0
36	AON_GPIO_0	Digital I/O	Always-on GPIO	VDDIO0
37	AON_GPIO_1	Digital I/O	Always-on GPIO	VDDIO0
38	AON_GPIO_2	Digital I/O	Always-on GPIO	VDDIO0
39	AON_GPIO_3	Digital I/O	Always-on GPIO	VDDIO0
40	AON_GPIO_4	Digital I/O	Always-on GPIO	VDDIO0
41	AON_GPIO_5	Digital I/O	Always-on GPIO	VDDIO0
42	AON_GPIO_6	Digital I/O	Always-on GPIO	VDDIO0
43	AON_GPIO_7	Digital I/O	Always-on GPIO	VDDIO0
44	GPIO_24	Digital I/O	General purpose I/O	VDDIO0
45	GPIO_25	Digital I/O	General purpose I/O	VDDIO0
46	GPIO_16	Digital I/O	General purpose I/O	VDDIO0

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
47	GPIO_17	Digital I/O	General purpose I/O	VDDIO0
48	GPIO_31	Digital I/O	General purpose I/O	VDDIO0
49	GPIO_30	Digital I/O	General purpose I/O	VDDIO0
50	GPIO_26	Digital I/O	General purpose I/O	VDDIO0
51	GPIO_27	Digital I/O	General purpose I/O	VDDIO0
52	GPIO_28	Digital I/O	General purpose I/O	VDDIO0
53	GPIO_29	Digital I/O	General purpose I/O	VDDIO0
54	VDD_AMS	Analog/RF Supply	AMS supply. Connect to VREG.	
55	XO_OUT	Analog/RF	Output of inverting amplifier connected to 32 MHz crystal	
56	XO_IN	Analog/RF	Input of inverting amplifier connected to 32 MHz crystal	

2.2 GR5515I0NDA QFN56

Figure 2-2 shows the pin assignments of GR5515I0NDA QFN56 package (top view).

The pins (Pin 43 to Pin 53) of GR5515I0NDA QFN56 package are different from those of GR5515IGND/GR5515IENDU QFN56 package.

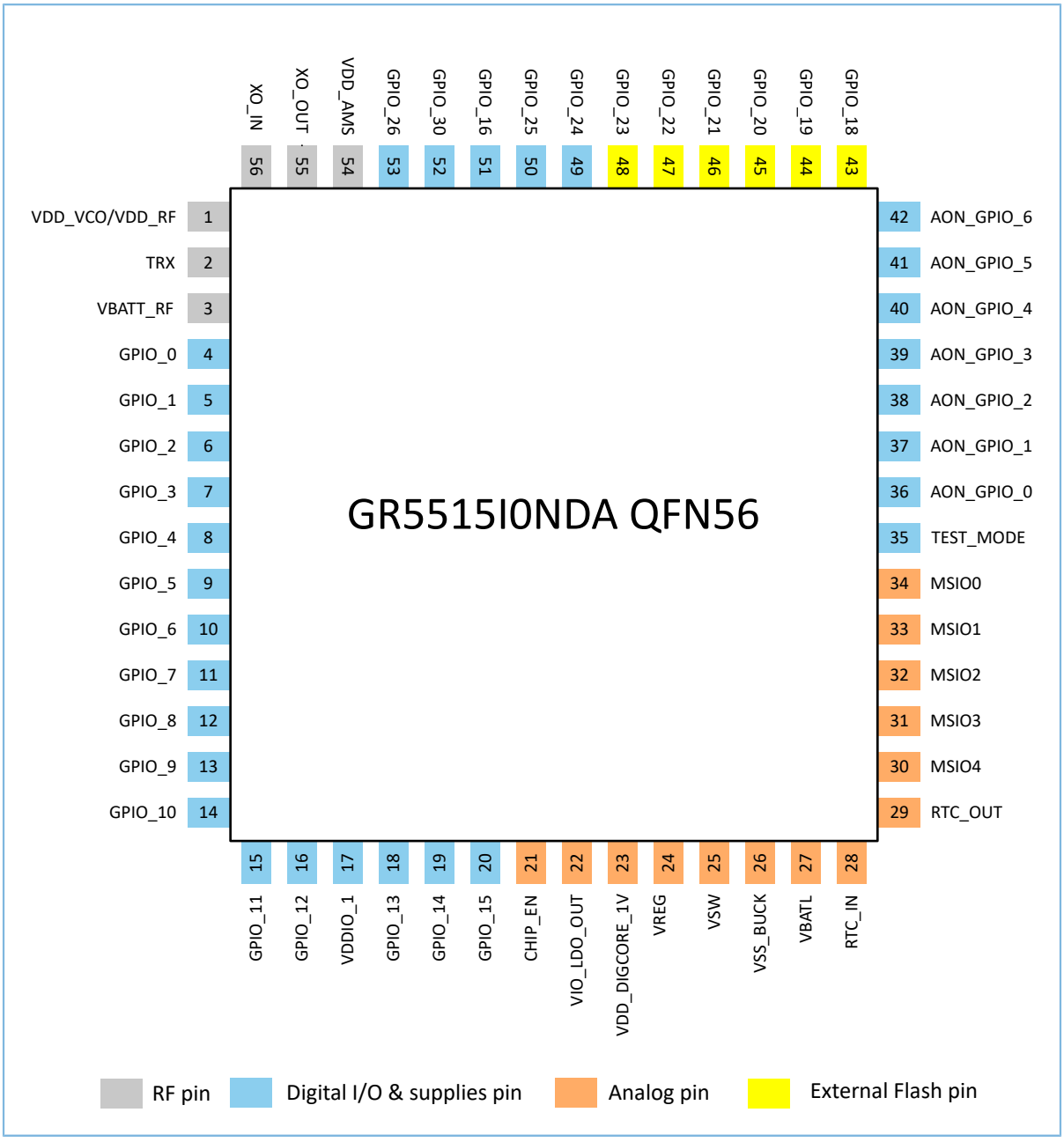

Figure 2-2 GR5515I0NDA QFN56 package pinout

Table 2-2 shows pin descriptions of GR5515I0NDA QFN56 package.

Table 2-2 GR5515I0NDA QFN56 pin descriptions

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
1	VDD_VCO/VDD_RF	Analog/RF supply	Synthesizer VCO supply. RF supply. Connect to VREG.	
2	TRX	Analog/RF	RX input and TX output	
3	VBATT_RF	Analog/RF Supply	Connect to VBATL.	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
4	GPIO_0	Digital I/O	SWDCLK	VDDIO1
5	GPIO_1	Digital I/O	SWDIO	VDDIO1
6	GPIO_2	Digital I/O	General purpose I/O	VDDIO1
7	GPIO_3	Digital I/O	General purpose I/O	VDDIO1
8	GPIO_4	Digital I/O	General purpose I/O	VDDIO1
9	GPIO_5	Digital I/O	General purpose I/O	VDDIO1
10	GPIO_6	Digital I/O	General purpose I/O	VDDIO1
11	GPIO_7	Digital I/O	General purpose I/O	VDDIO1
12	GPIO_8	Digital I/O	General purpose I/O	VDDIO1
13	GPIO_9	Digital I/O	General purpose I/O	VDDIO1
14	GPIO_10	Digital I/O	General purpose I/O	VDDIO1
15	GPIO_11	Digital I/O	General purpose I/O	VDDIO1
16	GPIO_12	Digital I/O	General purpose I/O	VDDIO1
17	VDDIO_1	Digital I/O supply	Digital I/O supply input	VDDIO1
18	GPIO_13	Digital I/O	General purpose I/O	VDDIO1
19	GPIO_14	Digital I/O	General purpose I/O	VDDIO1
20	GPIO_15	Digital I/O	General purpose I/O	VDDIO1
21	CHIP_EN	Mixed Signal IN	Master Enable for chip reset pin. Minimum value of high level for CHIP_EN is 1 V.	
22	VIO_LDO_OUT	PMU	Output of on-chip I/O supply regulator; when GR5515I0NDA is used (high Flash supply voltage for GR5515I0NDA), the pin is used as power input pin of VDDIO0 digital IO domain by being connected to VBATL	Connected internally to VDDIO0
23	VDD_DIGCORE_1V	PMU	Output of on-chip LDO for digital core. Connect to a 1 µF capacitor.	
24	VREG	PMU	Feedback pin from switching regulator	
25	VSW	PMU	DC-DC converter switching node	
26	VSS_BUCK	PMU	DC-DC converter supply and general battery GND	
27	VBATL	PMU	Power supply input	
28	RTC_IN	Analog/PMU	Input of inverting amplifier connected to 32.768 kHz crystal	
29	RTC_OUT	Analog/PMU	Output of inverting amplifier connected to 32.768 kHz crystal	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
30	MSIO4	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
31	MSIO3	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
32	MSIO2	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
33	MSIO1	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
34	MSIO0	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
35	TEST_MODE	Digital I/O	Input pin, used to set test mode for FT or CP factory test. In the application phase, the value is set to 0 by default. • If TEST_MODE = 1, the chip is in test mode for factory test. • If TEST_MODE = 0, the chip is in normal operation mode.	VDDIO0
36	AON_GPIO_0	Digital I/O	Always-on GPIO	VDDIO0
37	AON_GPIO_1	Digital I/O	Always-on GPIO	VDDIO0
38	AON_GPIO_2	Digital I/O	Always-on GPIO	VDDIO0
39	AON_GPIO_3	Digital I/O	Always-on GPIO	VDDIO0
40	AON_GPIO_4	Digital I/O	Always-on GPIO	VDDIO0
41	AON_GPIO_5	Digital I/O	Always-on GPIO	VDDIO0
42	AON_GPIO_6	Digital I/O	Always-on GPIO	VDDIO0
43	GPIO_18	Digital I/O	Connect to an external Flash	VDDIO0
44	GPIO_19	Digital I/O	Connect to an external Flash	VDDIO0
45	GPIO_20	Digital I/O	Connect to an external Flash	VDDIO0
46	GPIO_21	Digital I/O	Connect to an external Flash	VDDIO0
47	GPIO_22	Digital I/O	Connect to an external Flash	VDDIO0
48	GPIO_23	Digital I/O	Connect to an external Flash	VDDIO0
49	GPIO_24	Digital I/O	General purpose I/O	VDDIO0
50	GPIO_25	Digital I/O	General purpose I/O	VDDIO0
51	GPIO_16	Digital I/O	General purpose I/O	VDDIO0
52	GPIO_30	Digital I/O	General purpose I/O	VDDIO0
53	GPIO_26	Digital I/O	General purpose I/O	VDDIO0

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
54	VDD_AMS	Analog/RF Supply	AMS supply. Connect to VREG.	
55	XO_OUT	Analog/RF	Output of inverting amplifier connected to 32 MHz crystal	
56	XO_IN	Analog/RF	Input of inverting amplifier connected to 32 MHz crystal	

2.3 GR5515RGBD BGA68

Figure 2-3 shows the pin assignments of GR5515RGBD BGA68 package (top view).

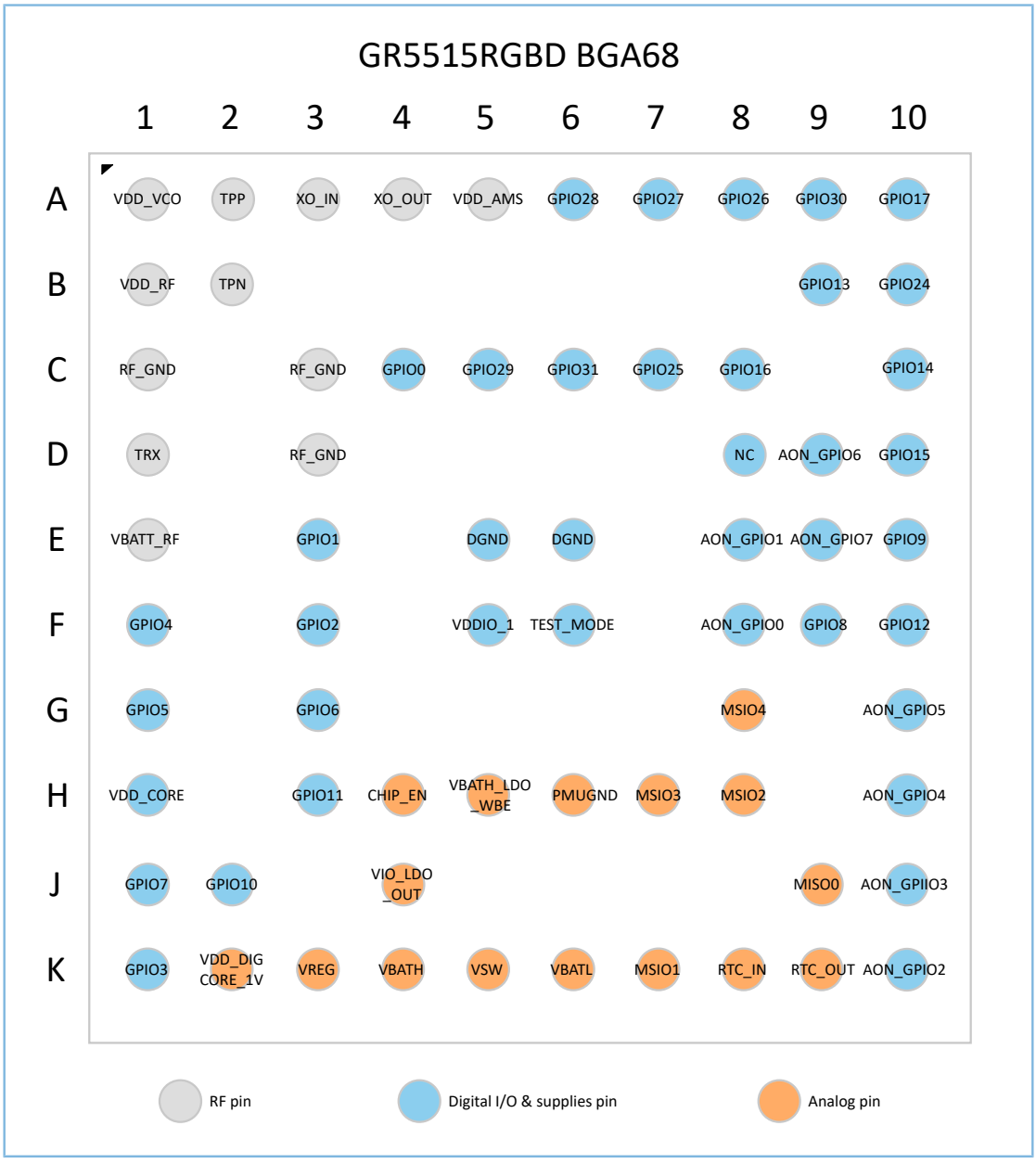

Figure 2-3 GR5515RGBD BGA68 package pinout

Table 2-3 shows pin descriptions of GR5515RGBD BGA68 package.

Table 2-3 GR5515RGBD BGA68 package pin descriptions

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
A1	VDD_VCO	Analog/RF supply	Synthesizer VCO supply: 1.1 V	
A2	TPP	Analog/RF	Test Mux +output	
A3	XO_IN	Analog/RF	Input of inverting amplifier connected to 32 MHz crystal	
A4	XO_OUT	Analog/RF	Output of inverting amplifier connected to 32 MHz crystal	
A5	VDD_AMS	Analog/RF	AMS supply 1.1 V	
A6	GPIO28	Digital I/O	General purpose I/O	VDDIO0
A7	GPIO27	Digital I/O	General purpose I/O	VDDIO0
A8	GPIO26	Digital I/O	General purpose I/O	VDDIO0
A9	GPIO30	Digital I/O	General purpose I/O	VDDIO0
A10	GPIO17	Digital I/O	General purpose I/O	VDDIO0
B1	VDD_RF	Analog/RF	RF supply 1.1 V	
B2	TPN	Analog/RF	Test Mux - output	
B9	GPIO13	Digital I/O	General purpose I/O	VDDIO1
B10	GPIO24	Digital I/O	General purpose I/O	VDDIO0
C1	RF_GND	Analog/RF	RF ground	
C3	RF_GND	Analog/RF	RF ground	
C4	GPIO0	Digital I/O	General purpose I/O, default SWDCLK	VDDIO1
C5	GPIO29	Digital I/O	General purpose I/O	VDDIO0
C6	GPIO31	Digital I/O	General purpose I/O	VDDIO0
C7	GPIO25	Digital I/O	General purpose I/O	VDDIO0
C8	GPIO16	Digital I/O	General purpose I/O	VDDIO0
C10	GPIO14	Digital I/O	General purpose I/O	VDDIO1
D1	TRX	Analog/RF	RX input and TX output	
D3	RF_GND	Analog/RF	RF ground	
D8	NC	-	-	
D9	AON_GPIO6	Digital I/O	Always-on General purpose I/O	VDDIO0
D10	GPIO15	Digital I/O	General purpose I/O	VDDIO1
E1	VBATT_RF	Analog/RF	Connect to VBATL	
E3	GPIO1	Digital I/O	General purpose I/O, default SWDIO	VDDIO1
E5	DGND	Digital GND	Digital Ground	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
E6	DGND	Digital GND	Digital Ground	
E8	AON_GPIO1	Digital I/O	Always-on general purpose I/O	VDDIO0
E9	AON_GPIO7	Digital I/O	Always-on general purpose I/O	VDDIO0
E10	GPIO9	Digital I/O	General purpose I/O	VDDIO1
F1	GPIO4	Digital I/O	General purpose I/O	VDDIO1
F3	GPIO2	Digital I/O	General purpose I/O	VDDIO1
F5	VDDIO_1	Digital Supply	I/O supply voltage input	VDDIO1
F6	TEST_MODE	Digital I/O	Input pin, used to set test mode for FT or CP factory test. In the application phase, the value is set to 0 by default. • If TEST_MODE = 1, the chip is in test mode for factory test. • If TEST_MODE = 0, the chip is in normal operation mode.	VDDIO0
F8	AON_GPIO0	Digital I/O	Always-on general purpose I/O	VDDIO0
F9	GPIO8	Digital I/O	General purpose I/O	VDDIO1
F10	GPIO12	Digital I/O	General purpose I/O	VDDIO1
G1	GPIO5	Digital I/O	General purpose I/O	VDDIO1
G3	GPIO6	Digital I/O	General purpose I/O	VDDIO1
G8	MSIO4	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
G10	AON_GPIO5	Digital I/O	Always-on general purpose I/O	VDDIO0
H1	VDD_CORE	Digital Supply	Digital core supply	
H3	GPIO11	Digital I/O	General purpose I/O	VDDIO1
H4	CHIP_EN	Analog/PMU	Master Enable for chip reset pin. Minimum value of high level for CHIP_EN is 1 V.	
H5	VBATH_LDO_WBE	Analog/PMU	Connect to GND.	
H6	PMUGND	Analog/PMU	DC-DC converter supply & general battery GND	
H7	MSIO3	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
H8	MSIO2	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
H10	AON_GPIO4	Digital I/O	Always-on general purpose I/O	VDDIO0
J1	GPIO7	Digital I/O	General purpose I/O	VDDIO1
J2	GPIO10	Digital I/O	General purpose I/O	VDDIO1

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
J4	VIO_LDO_OUT	Analog/PMU	Output of On-Chip I/O supply regulator.	Connected internally to VDDIO0
J9	MSIO0	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
J10	AON_GPIO3	Digital I/O	Always-on general purpose I/O	VDDIO0
K1	GPIO3	Digital I/O	General purpose I/O	VDDIO1
K2	VDD_DIGCORE_1V	Analog/PMU	LDO output of On-Chip of digital core, connected to 1 µF capacitor	
K3	VREG	Analog/PMU	Feedback pin of switch regulator	
K4	VBATH	Analog/PMU	Connect to VBATL	
K5	VSW	Analog/PMU	DC-DC Converter Switching Node	
K6	VBATL	Analog/PMU	Input from battery	
K7	MSIO1	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
K8	RTC_IN	Analog/PMU	Input of inverting amplifier connected to 32.768 kHz crystal	
K9	RTC_OUT	Analog/PMU	Output of inverting amplifier connected to 32.768 kHz crystal	
K10	AON_GPIO2	Digital I/O	Always-on general purpose I/O	VDDIO0

2.4 GR5515GGBD BGA55

Figure 2-4 shows the pin assignments of GR5515GGBD BGA55 package (top view).

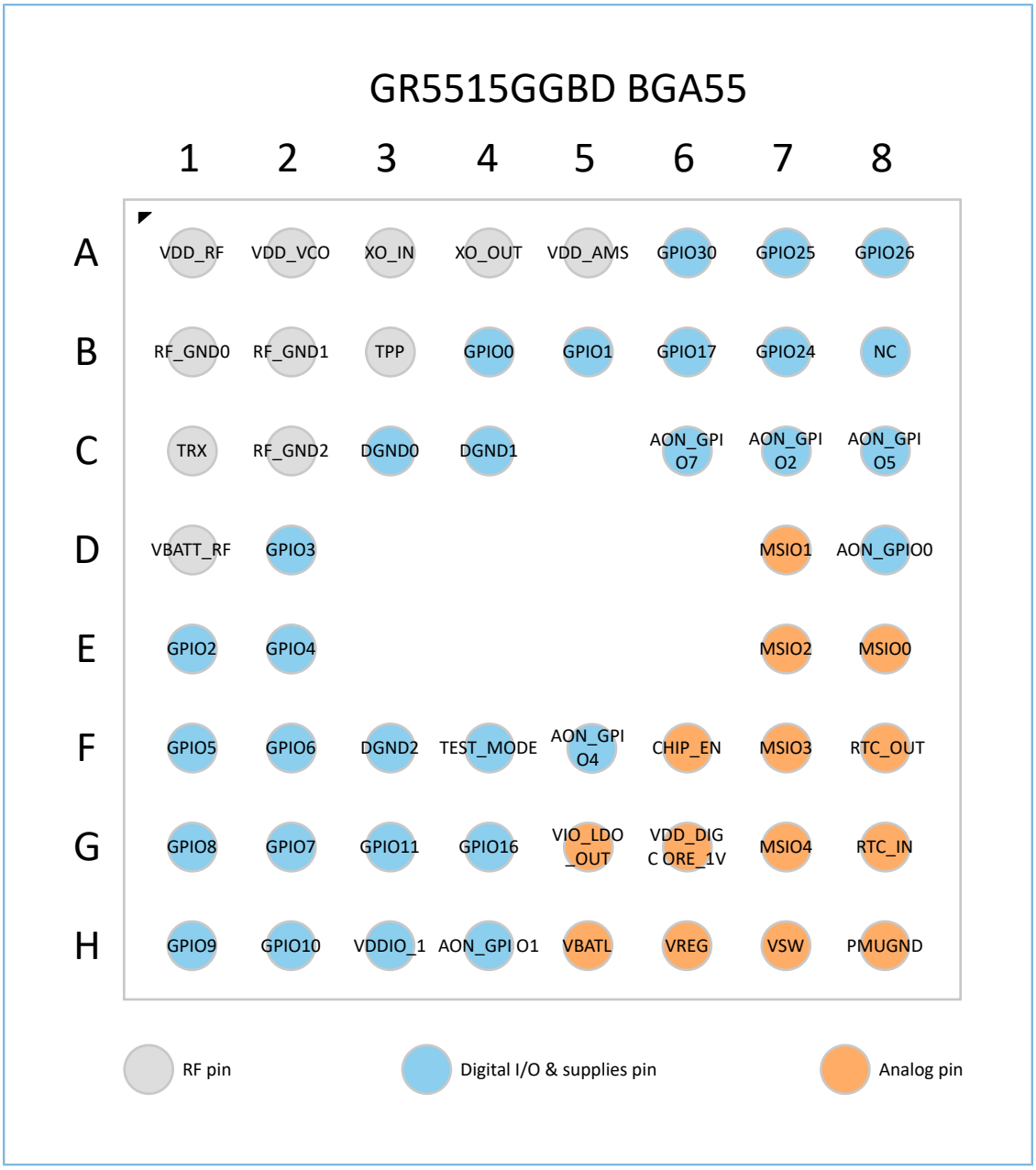

Figure 2-4 GR5515GGBD BGA55 package pinout

Table 2-4 shows pin descriptions of GR5515GGBD BGA55 package.

Table 2-4 GR5515GGBD BGA55 package pin descriptions

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
A1	VDD_RF	Analog/RF supply	RF supply: 1.1 V	
A2	VDD_VCO	Analog/RF supply	Synthesizer VCO supply: 1.1 V	
A3	XO_IN	Analog/RF	Input of inverting amplifier connected to 32 MHz crystal	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
A4	XO_OUT	Analog/RF	Output of inverting amplifier connected to 32 MHz crystal	
A5	VDD_AMS	Analog/RF supply	AMS supply 1.1 V	
A6	GPIO30	Digital I/O	General purpose I/O	VDDIO0
A7	GPIO25	Digital I/O	General purpose I/O	VDDIO0
A8	GPIO26	Digital I/O	General purpose I/O	VDDIO0
B1	RF_GND0	Analog/RF	RF ground	
B2	RF_GND1	Analog/RF	RF ground	
B3	TPP	Analog/RF	Test Mux + output	
B4	GPIO0	Digital I/O	General purpose I/O, default SWDCLK	VDDIO1
B5	GPIO1	Digital I/O	General purpose I/O, default SWDIO	VDDIO1
B6	GPIO17	Digital I/O	General purpose I/O	VDDIO0
B7	GPIO24	Digital I/O	General purpose I/O	VDDIO0
B8	NC	-	-	
C1	TRX	Analog/RF	RX input and TX output	
C2	RF_GND2	Analog/RF	RF ground	
C3	DGND0	Digital GND	Digital ground	
C4	DGND1	Digital GND	Digital ground	
C6	AON_GPIO7	Digital I/O	Always-on general purpose I/O	VDDIO0
C7	AON_GPIO2	Digital I/O	Always-on general purpose I/O	VDDIO0
C8	AON_GPIO5	Digital I/O	Always-on general purpose I/O	VDDIO0
D1	VBATT_RF	Analog/RF supply	Connect to VBATL	
D2	GPIO3	Digital I/O	General purpose I/O	VDDIO1
D7	MSIO1	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
D8	AON_GPIO0	Digital I/O	Always-on general purpose I/O	VDDIO0
E1	GPIO2	Digital I/O	General purpose I/O	VDDIO1
E2	GPIO4	Digital I/O	General purpose I/O	VDDIO1
E7	MSIO2	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
E8	MSIO0	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
F1	GPIO5	Digital I/O	General purpose I/O	VDDIO1
F2	GPIO6	Digital I/O	General purpose I/O	VDDIO1

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
F3	DGND2	Digital GND	Digital ground	
F4	TEST_MODE	Digital I/O	Input pin, used to set test mode for FT or CP factory test. In the application phase, the value is set to 0 by default. • If TEST_MODE = 1, the chip is in test mode for factory test. • If TEST_MODE = 0, the chip is in normal operation mode.	VDDIO0
F5	AON_GPIO4	Digital I/O	Always-on general purpose I/O	VDDIO0
F6	CHIP_EN	Analog/PMU	Master Enable for chip reset pin Minimum value of high level for CHIP_EN is 1 V	
F7	MSIO3	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
F8	RTC_OUT	Analog/PMU	Output of inverting amplifier connected to 32.768 kHz crystal	
G1	GPIO8	Digital I/O	General purpose I/O	VDDIO1
G2	GPIO7	Digital I/O	General purpose I/O	VDDIO1
G3	GPIO11	Digital I/O	General purpose I/O	VDDIO1
G4	GPIO16	Digital I/O	General purpose I/O	VDDIO0
G5	VIO_LDO_OUT	Analog/PMU	Output of On-Chip I/O supply regulator	Connected internally to VDDIO0
G6	VDD_DIGCORE_1V	Analog/PMU	LDO output of On-Chip of digital core, connected to 1 µF capacitor	
G7	MSIO4	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
G8	RTC_IN	Analog/PMU	Input of inverting amplifier connected to 32.768 kHz crystal	
H1	GPIO9	Digital I/O	General purpose I/O	VDDIO1
H2	GPIO10	Digital I/O	General purpose I/O	VDDIO1
H3	VDDIO_1	Digital I/O supply	I/O supply voltage input	VDDIO1
H4	AON_GPIO1	Digital I/O	Always-on general purpose I/O	VDDIO0
H5	VBATL	Analog/PMU	Input from battery	
H6	VREG	Analog/PMU	Feedback pin of switch regulator	
H7	VSW	Analog/PMU	DC-DC converter switching node	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
H8	PMUGND	Analog/PMU	DC-DC converter & general battery GND pin	

2.5 GR5513BEND (NRND)/GR5513BENDU QFN40

Figure 2-5 shows the pinout of GR5513BEND/GR5513BENDU QFN40 package (top view).

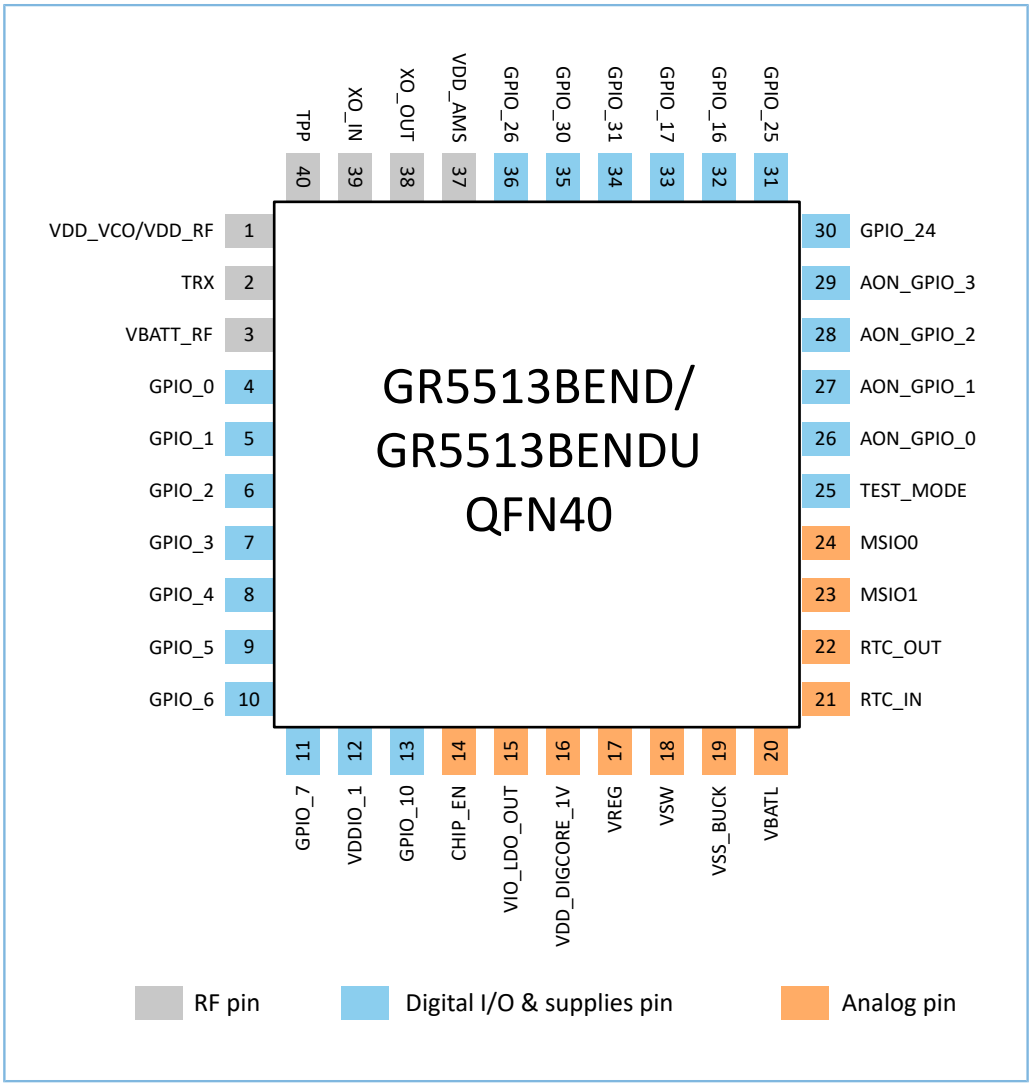

Figure 2-5 GR5513BEND/GR5513BENDU QFN40 package pinout

Table 2-5 shows pin descriptions of GR5513BEND/GR5513BENDU QFN40 package.

Table 2-5 GR5513BEND/GR5513BENDU QFN40 pin descriptions

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
1	VDD_VCO/VDD_RF	Analog/RF supply	Synthesizer VCO supply. RF supply. Connect to VREG.	
2	TRX	Analog/RF	RX input and TX output	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
3	VBATT_RF	Analog/RF	Connect to VBATL.	
4	GPIO_0	Digital I/O	General purpose I/O, default SWDCLK	VDDIO1
5	GPIO_1	Digital I/O	General purpose I/O, default SWDIO	VDDIO1
6	GPIO_2	Digital I/O	General purpose I/O	VDDIO1
7	GPIO_3	Digital I/O	General purpose I/O	VDDIO1
8	GPIO_4	Digital I/O	General purpose I/O	VDDIO1
9	GPIO_5	Digital I/O	General purpose I/O	VDDIO1
10	GPIO_6	Digital I/O	General purpose I/O	VDDIO1
11	GPIO_7	Digital I/O	General purpose I/O	VDDIO1
12	VDDIO_1	Digital I/O Supply	Digital I/O supply input	VDDIO1
13	GPIO_10	Digital I/O	General purpose I/O	VDDIO1
14	CHIP_EN	Analog/PMU	Master Enable for chip reset pin. Minimum value of high level for CHIP_EN is 1 V.	
15	VIO_LDO_OUT	Analog/PMU	Output of on-chip I/O supply regulator. For GR5513BENDU, power input pin of VDDIO0 digital IO domain. When VDDIO0 is set to 3.3 V/VBATL, VIO_LDO_OUT should be connected to 3.3 V/VBATL.	Connected internally to VDDIO0
16	VDD_DIGCORE_1V	Analog/PMU	Output of on-chip LDO for digital core. Connect to a 1 µF capacitor.	
17	VREG	Analog/PMU	Feedback pin from switching regulator	
18	VSW	Analog/PMU	DC/DC converter switching node	
19	VSS_BUCK	Analog/PMU	DC/DC converter supply and general battery GND	
20	VBATL	Analog/PMU	Input from Battery	
21	RTC_IN	Analog/PMU	Input of inverting amplifier connected to 32.768 kHz crystal	
22	RTC_OUT	Analog/PMU	Output of inverting amplifier connected to 32.768 kHz crystal	
23	MSIO1	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL
24	MSIO0	Mixed Signal I/O	Configurable to be a GPIO mixed signal (ADC interface)	VBATL

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
25	TEST_MODE	Digital I/O	Input pin, used to set test mode for FT or CP factory test. In the application phase, the value is set to 0 by default. • If TEST_MODE = 1, the chip is in test mode for factory test. • If TEST_MODE = 0, the chip is in normal operation mode.	VDDIO0
26	AON_GPIO_0	Digital I/O	Always-on general purpose I/O	VDDIO0
27	AON_GPIO_1	Digital I/O	Always-on general purpose I/O	VDDIO0
28	AON_GPIO_2	Digital I/O	Always-on general purpose I/O	VDDIO0
29	AON_GPIO_3	Digital I/O	Always-on general purpose I/O	VDDIO0
30	GPIO_24	Digital I/O	General purpose I/O	VDDIO0
31	GPIO_25	Digital I/O	General purpose I/O	VDDIO0
32	GPIO_16	Digital I/O	General purpose I/O	VDDIO0
33	GPIO_17	Digital I/O	General purpose I/O	VDDIO0
34	GPIO_31	Digital I/O	General purpose I/O	VDDIO0
35	GPIO_30	Digital I/O	General purpose I/O	VDDIO0
36	GPIO_26	Digital I/O	General purpose I/O	VDDIO0
37	VDD_AMS	Analog/RF	AMS supply. Connect to VREG.	
38	XO_OUT	Analog/RF	Output of inverting amplifier connected to 32 MHz crystal	
39	XO_IN	Analog/RF	Input of inverting amplifier connected to 32 MHz crystal	
40	TPP	Analog/RF	Test Mux + output	

3 MCU Subsystem

GR551x has an MCU subsystem that contains an ARM® Cortex®-M4F processor, its corresponding buses, and peripherals with all the multiplexing options for the GPIOs, as illustrated in Figure 3-1.

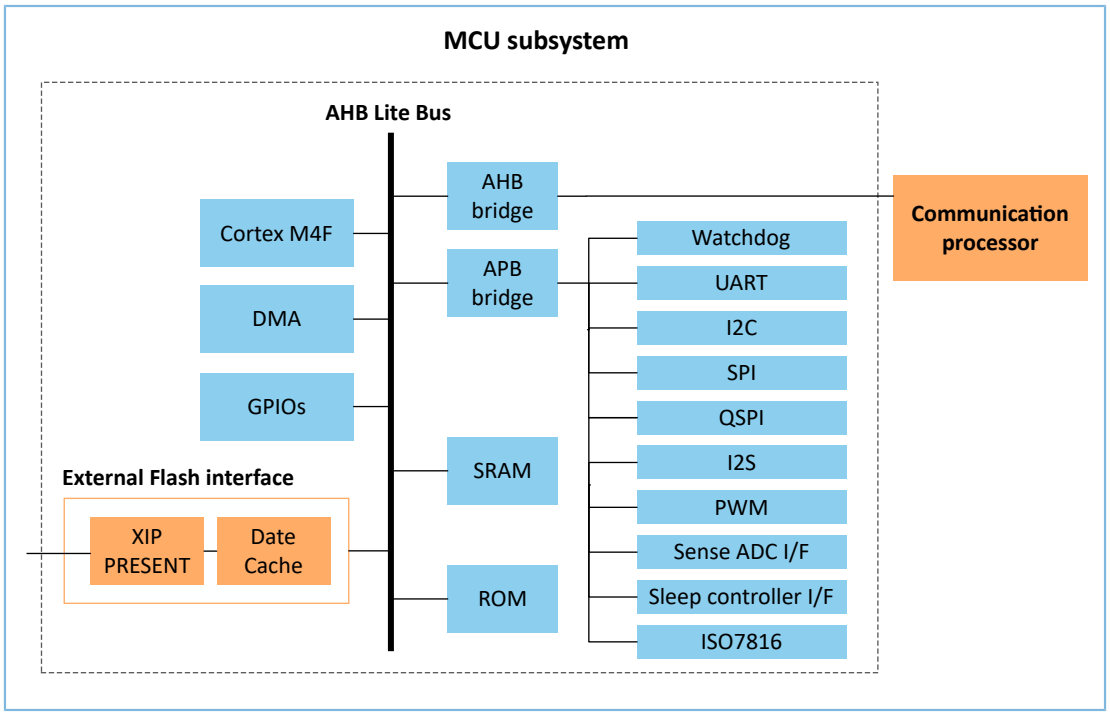

Figure 3-1 Microcontroller Subsystem

The processor has a floating-point unit and a 32-bit instruction set with Thumb-2 mode supported to use a hybrid of 16-bit and 32-bit instructions to maximize the code performance and density.

MCU memories are included in the same domain and have a special retention voltage, which controls to have the memories in different modes according to the application usage:

- OFF
- ON
- Retention

Digital pads are specially designed to retain their state even when the power is turned OFF in the MCU domain.

The following are the supported options for the Cortex®-M4F:

- Nested Vectored Interrupt Controller (NVIC) with 34 vectors
- Data Watchpoint and Trace (DWT)
- System Tick Timer (SysTick)
- Memory Protection Unit (MPU)
- Floating-point unit (FPU)

- Debug Access Port (DAP)
- Instrumentation Trace Macrocell (ITM)
- Trace Port Interface Unit (TPIU)
- Flash Patch and Breakpoint Unit (FPB)

3.1 MCU Debug

Serial Wire Debug (SWD) is used for debug. Serial Wire Output (SWO) pin is used to output trace information.

3.2 Interrupt Vector

Table 3-1 shows the NVIC interrupt vectors for GR551x.

Table 3-1 NVIC interrupt vector

Number	Acronym	Description	Address
-	-	-	0x00000000
-15	Reset	Reset	0x00000004
-14	NMI	Non-maskable interrupt	0x00000008
-13	HardFault	All classes of fault	0x0000000C
-12	MemManage	Memory Management fault	0x00000010
-11	BusFault	Error response received from the bus system	0x00000014
-10	UsageFault	Usage fault	0x00000018
-5	SVCall	System service call via SWI instruction	0x0000002C
-4	Debug	Debug monitor	0x00000030
-2	PendSV	Pendable request for System Service	0x00000038
-1	SysTick	System tick timer	0x0000003C
0	WDT	System watchdog	0x00000040
1	BLE_SDK	Bluetooth LE SDK schedule	0x00000044
2	BLE	Bluetooth LE event	0x00000048
3	DMA	DMA interrupt	0x0000004C
4	SPI_M	SPI Master interrupt	0x00000050
5	SPI_S	SPI Slave interrupt	0x00000054
6	EXT0	GPIO0 – GPIO15 interrupt	0x00000058
7	EXT1	GPIO16 – GPIO31 interrupt	0x0000005C
8	TIMER0	Timer0 global interrupt	0x00000060
9	TIMER1	Timer1 global interrupt	0x00000064
10	DUAL_TIMER	Dual Timer global interrupt	0x00000068

Number	Acronym	Description	Address
11	QSPI0	QSPI0 global interrupt	0x0000006C
12	UART0	UART0 global interrupt	0x00000070
13	UART1	UART1 global interrupt	0x00000074
14	I2C0	I2C0 global interrupt	0x00000078
15	I2C1	I2C1 global interrupt	0x0000007C
16	AES	AES global interrupt	0x00000080
17	HMAC	HMAC global interrupt	0x00000084
18	EXT2	AON_GPIO0 – AON_GPIO7 interrupt	0x00000088
19	RNG	RNG global interrupt	0x0000008C
20	PMU	Power management interrupt	0x00000090
21	PKC	PKC global interrupt	0x00000094
22	XQSPI	XIP QSPI global interrupt	0x00000098
23	QSPI1	QSPI1 global interrupt	0x0000009C
24	PWR_CMD	Power Command interrupt	0x000000A0
25	BLE_SLP	Bluetooth LE sleep interrupt	0x000000A4
26	SLEEP_TIMER	Sleep Timer interrupt	0x000000A8
27	COMP	Comparator interrupt	0x000000AC
28	AON_WDT	AON watchdog interrupt	0x000000B0
29	I2S_M	I2S Master global interrupt	0x000000B4
30	I2S_S	I2S Slave global interrupt	0x000000B8
31	ISO7816	ISO7816 global interrupt	0x000000BC
32	PRESENT	Present global interrupt	0x000000C0
33	CALENDAR	RTC global interrupt	0x000000C4

3.3 Electrical Specifications

Table 3-2 Electrical Specifications

Symbol	Description	Min.	Typ.	Max.	Unit
I_{chip}	CPU current at 16 MHz. Running CoreMark from Flash, Cache Enabled, Digital LDO, VBAT 3.3 V		1.50		mA
I_{chip}	CPU current at 16 MHz. Running CoreMark from Flash, Cache Disabled, Digital LDO, VBAT 3.3 V		2.45		mA
I_{chip}	CPU current at 16 MHz. Running CoreMark from SRAM, Digital LDO, VBAT 3.3 V		1.23		mA

Symbol	Description	Min.	Typ.	Max.	Unit
I_{chip}	CPU current at 64 MHz. Running CoreMark from Flash, Cache Enabled, Digital LDO, VBAT 3.3 V		2.88		mA
I_{chip}	CPU current at 64 MHz. Running CoreMark from Flash, Cache Disabled, Digital LDO, VBAT 3.3 V		5.25		mA
I_{chip}	CPU current at 64 MHz. Running CoreMark from SRAM, Digital LDO, VBAT 3.3 V		2.67		mA
$I_{MCU/MHz}$	CPU current. Running CoreMark from SRAM, Digital LDO, VBAT 3.3 V.		30		µA/MHz

4 Memory

4.1 Memory Introduction

GR551x SoC memory includes ROM, SRAM and stacked Flash for code and data storage. The CPU and peripheral can access the memory. The CPU can access the peripherals as well. The address mapping of the memories and devices are explained in the following sections.

4.2 Memory Map

The GR551x series SoCs memory map is shown in Figure 4-1.

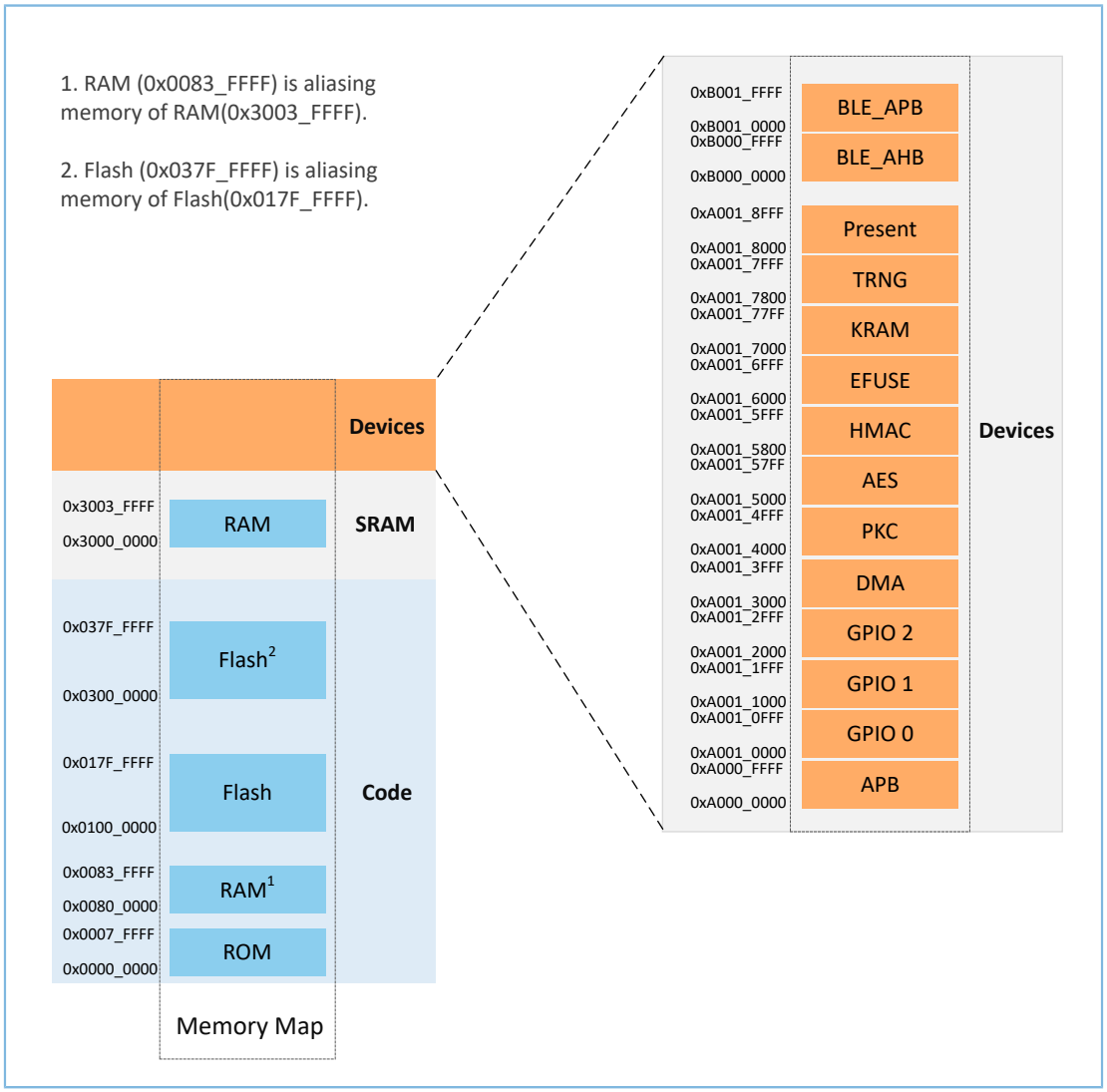

Figure 4-1 GR551x series SoCs memory map

Note:

- GR5515 series SoCs:

RAM space: 0x3000_0000to 0x3003_FFFF, 0x0080_0000 to 0x0083_FFFF(Aliase), 256 KB in total

Flash space: 0x0100_0000 to 0x010F_FFFF, 0x0300_0000 to 0x030F_FFFF (Aliase), 1 MB in total

- GR5513 SoC:

RAM space: 0x3000_0000 to 0x3001_FFFF, 0x0080_0000 to 0x0081_FFFF (Aliase), 128 KB in total

Flash space: 0x0100_0000 to 0x0107_FFFF, 0x0300_0000 to 0x0307_FFFF (Aliase), 512 KB in total

4.3 APB Address Space

Figure 4-2 shows the details of Advanced Peripheral Bus (APB) portion of the memory map.

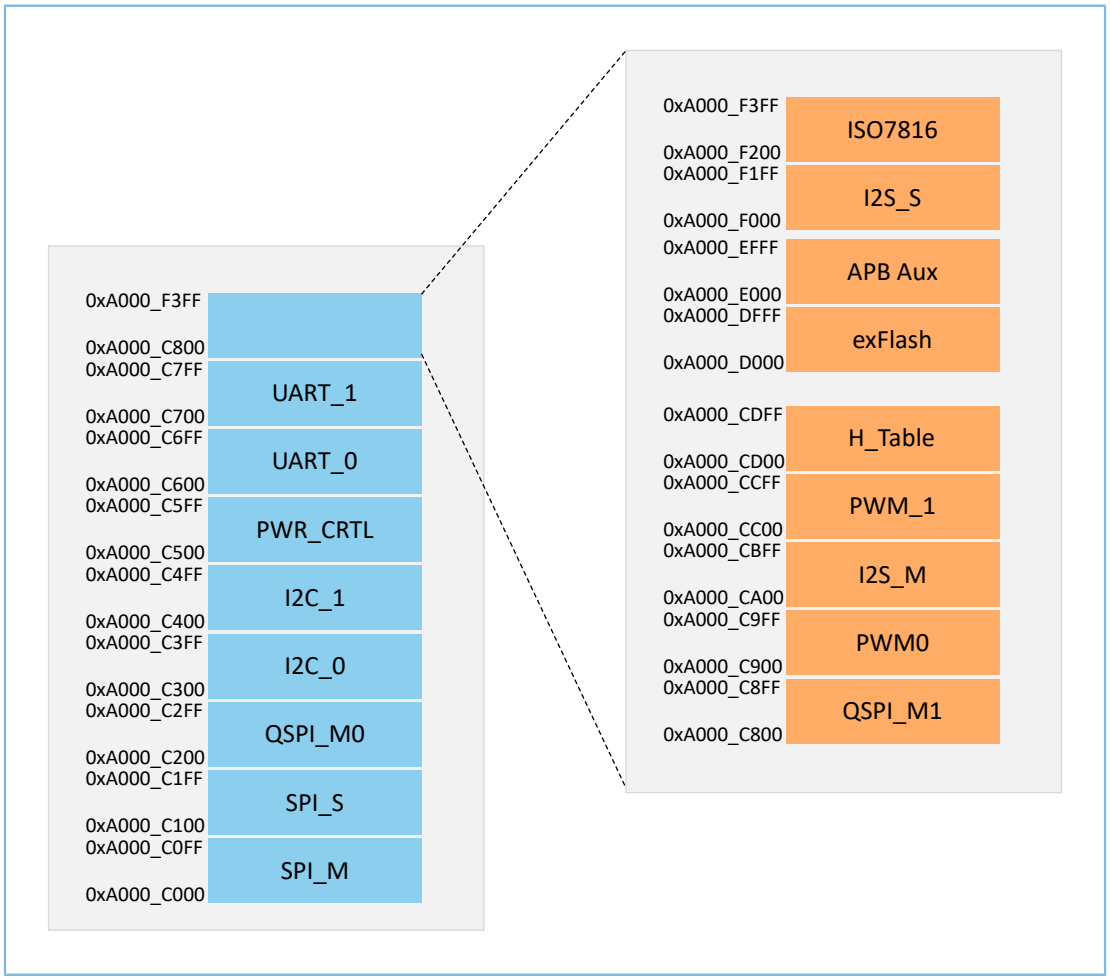

Figure 4-2 APB memory map

5 NVM Storage (eFuse)

GR551x eFuse (512 Bytes) stores the product ID, TRIM and security parameters. It consists of Main Region (240 Bytes), Backup Region (240 Bytes), and User Region (32 Bytes).

Figure 5-1 shows the structure of eFuse.

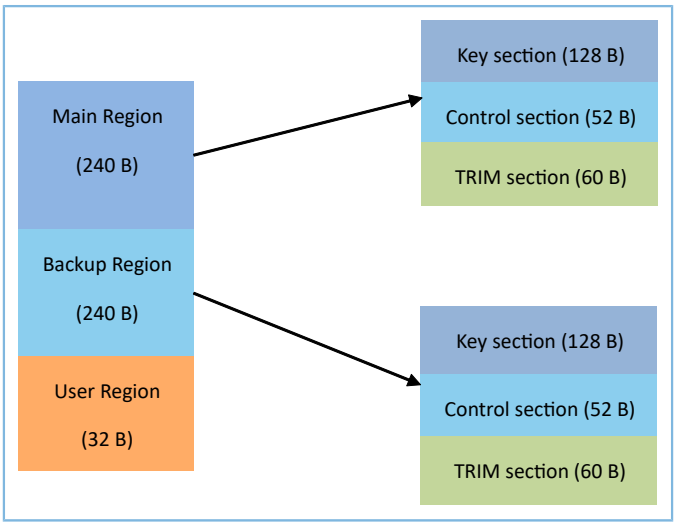

Figure 5-1 eFuse structure

- The Main Region has three sections: Key section, Control section and TRIM section.
  - Key section stores all the keys needed for Secure Boot and SPI Flash encryption.
  - Control section stores some parameters required by the boot loader for chip configuration.
  - TRIM section is used to save chip UID, package type,TRIM parameters and product configurations.

    The address mapping of chip uid and package type is shown in the table below:

Table 5-1 Address mapping of chip UID and package type

Name	Start Address (Absolute)	Size (Byte)	Description
chip UID	0xA001613C	0x10	The detailed definition of chip uid is as follows. - 0xA001613C to 0xA001613E: FAB - 0xA001613F: Year - 0xA0016140: Mon - 0xA0016141 to 0xA0016148: LOT_ID - 0xA0016149: Wafer_ID - 0xA001614A: Wafer_X - 0xA001614B: Wafer_Y
package type	0xA0016112	0x01	The package type indicates the type of chip used. - 00000: Unused, compatible with previous test firmware.

Name	Start Address (Absolute)	Size (Byte)	Description
			- 00001: GR5515RGBD is used. - 00010: GR5515GGBD is used. - 00011: GR5515IGND is used. - 00101: GR5513BEND is used. - 00111: GR5513BENDU is used. - Others: Reserved

- The Backup Region provides a redundant copy of Main Region to ensure the integrity of the eFuse content.
- The User Region is reserved for users to write and read by calling the eFuse driver. Every single byte of eFuse can only be written once.The absolute start address of User Region is 0xA0016000.

You can use GProgrammer to create and download eFuse files.

6 PMU

6.1 Power Management

Power Management Unit (PMU) is responsible for generating all required voltages for different blocks in GR551x. For active mode, a DC-DC converter generates the voltage for the transceiver and a Low Dropout (LDO) regulator generates the voltage for digital blocks. GR551x uses an LDO regulator to supply its Always-on (AON) modules that stay ON when both the MCU subsystem and the Bluetooth LE subsystem are OFF. LDO regulator also generates a lower voltage for content retention to the memories where their content is needed after wakeup. Both the retention voltage and the digital voltage are connected to all power islands through a control switch matrix on the chip.

Figure 6-1 shows the power management unit in GR551x.

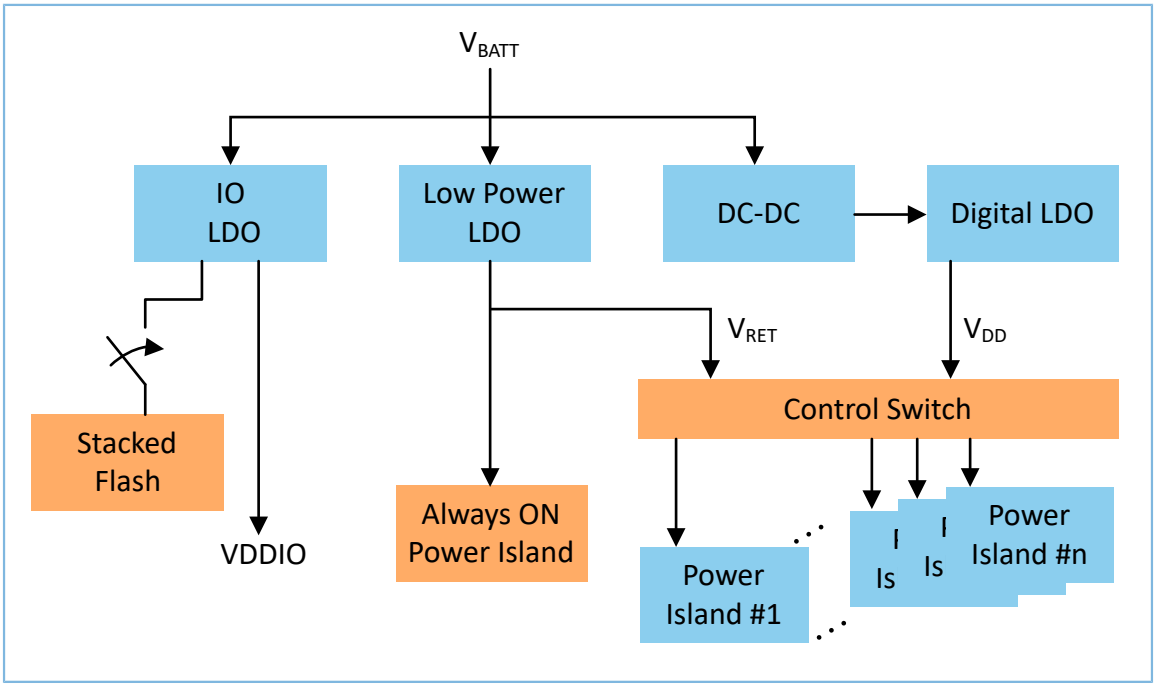

Figure 6-1 Power management block diagram

Memory on GR551x is divided into different banks. Each memory bank can be turned OFF to save power if not used. Moreover, used content in memory banks can be retained during sleep if needed.

Additionally, an IO-LDO is used to generate the IO voltage (VDDIO) to supply the pads of GR551x and the external devices connecting to GR551x. The IO-LDO is also used to supply power to the stacked Flash (except for GR5515I0NDA) through a switch that can be turned OFF to eliminate the power leakage of the Flash when not in use. Another LDO is used internally to provide programming voltage to eFuse when eFuse is written.

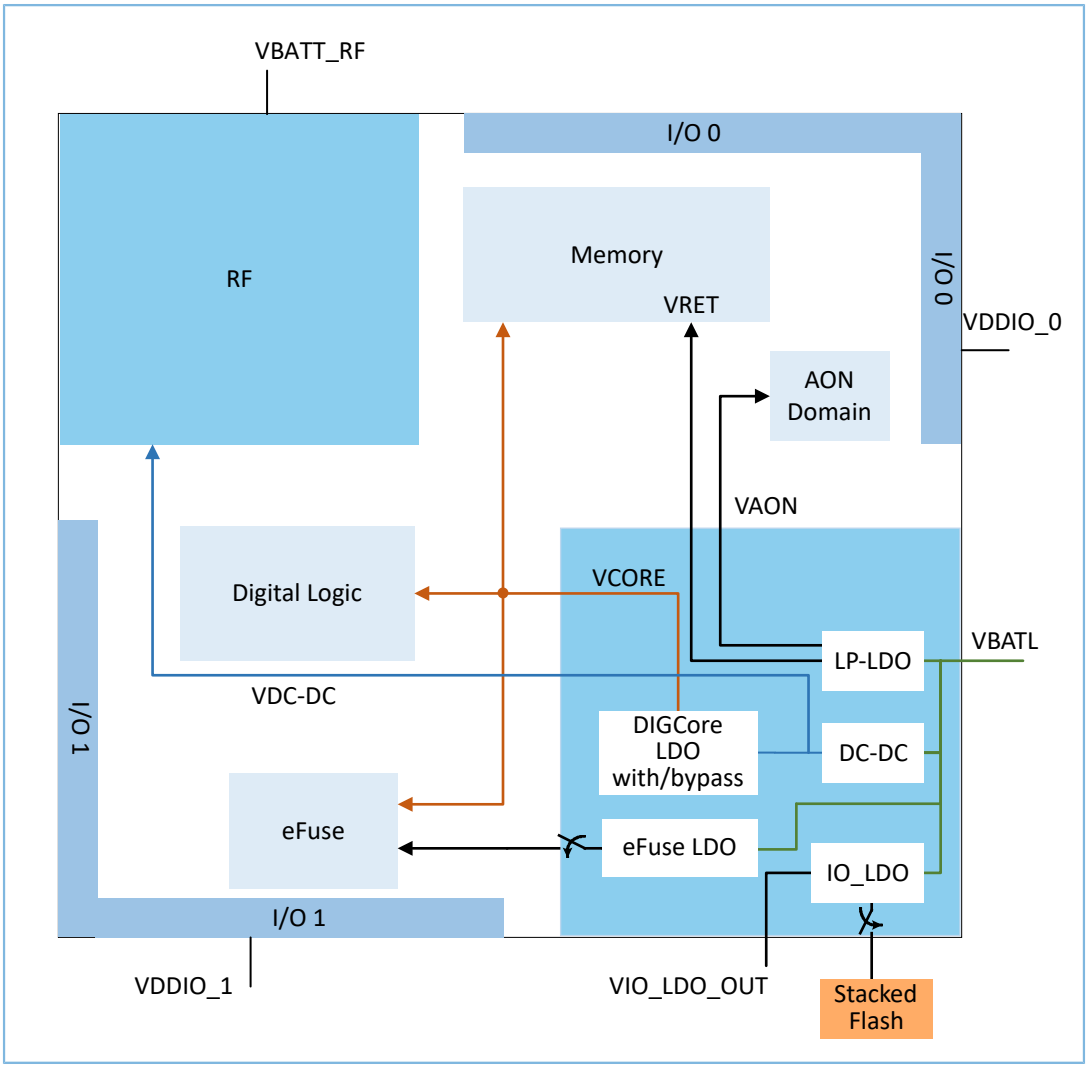

Figure 6-2 Power architecture

Three I/O voltage domains are provided for GR551x: two digital voltage domains (VDDIO0 and VDDIO1), as well as one MSIO digital mixed domain. The corresponding reference voltage levels are VDDIO0, VDDIO1, and VBATL respectively. Note that VDDIO0 is connected to VIO_LDO_OUT internally, and is not bonded to any package pins. Figure 6-3 is a circuit diagram showing the connections between VIO_LDO_OUT and the I/O voltage domains.

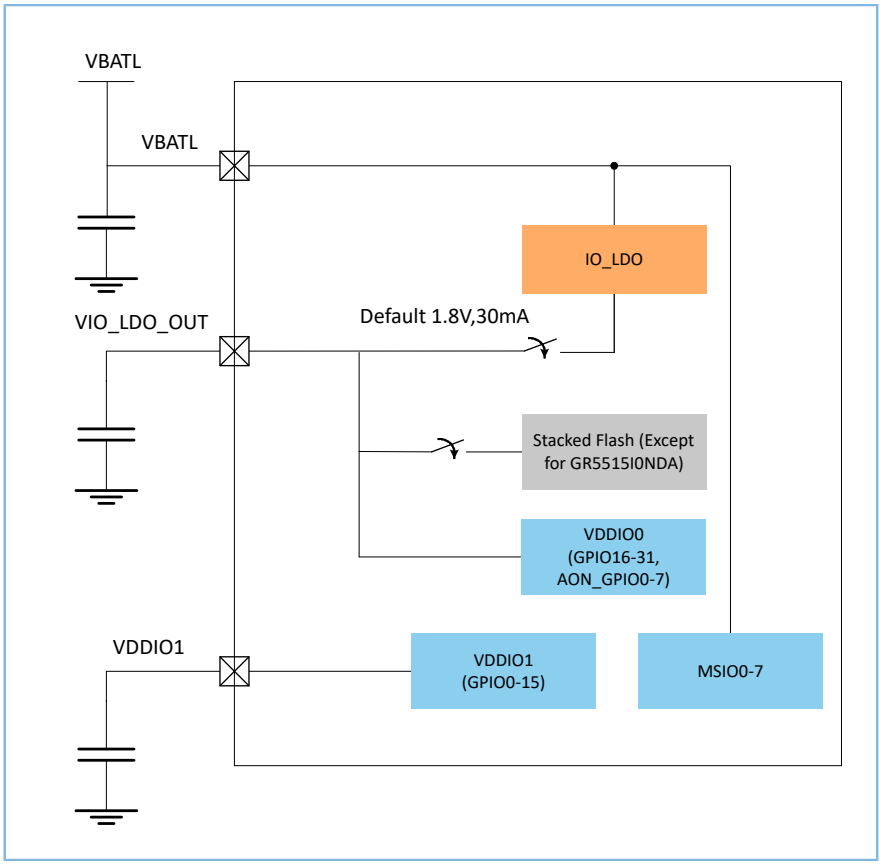

Figure 6-3 Connections between VIO_LDO_OUT and I/O voltage domains

6.2 DC-DC Converter

The DC-DC buck converter provides power to all the RF blocks, digital logic blocks and memories.

Table 6-1 DC-DC converter specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit	Comment
Output Voltage	Vdcdc	0.9	1.05	1.35	V	Programmable
Load Current	I_load			30	mA	
Startup Time	Tstartup		100		µs	
Voltage Ripples	Vripp		5		mV	

Note:

Voltage Ripples: Using $L_{ext} = 2.2\mu H + 9.1nH$ and $C_{ext} = 2.2\mu F$

6.3 Digital LDO

Digital LDO regulates the DC-DC converter to supply power to all the Digital Logic and Memory blocks.

Table 6-2 Digital Core LDO specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit.	Comment
Input Voltage	Vin	0.9	1.05	1.3	V	
Output Voltage	Vcore	0.85	0.95	1.3	V	Programmable
External Load Cap	Cload	0.1	1		µF	

6.4 I/O LDO

The I/O LDO creates the supply voltage to drive the IO pads.

Table 6-3 I/O LDO specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit	Comment
Supply Voltage	VBAT	1.7	3.3	3.8	V	
Output Voltage	VIO	1.6	1.8	3.6	V	Programmable, But VBAT needs to be greater than VIO output voltage + LDO voltage drop.
Quiescent Current	Iq		700		nA	
External Load Cap	CL	0.1	0.2		µF	
Maximum load current				30	mA	

6.5 POR/BOD

Power-on Reset (POR) circuit holds the system at reset while the supply reaches the required voltage level. Brown-out detector (BOD) circuit puts the system into reset state when the supply falls below the Brown-out Threshold.

Table 6-4 POR/BOD specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit.
Reset time	Trst		500		µs
Brown Out Threshold	TH_BOD		1.45		V

6.6 Wakeup Comparator

The wakeup comparator compares two analog signals, namely, v1 and v2, and generates an interrupt when v1 exceeds v2. The interrupt can be used to wake the system up from a sleep state. The two signals can be connected to any of the five MSIOs (MSIO0 - MSIO4). Additionally, the wakeup comparator can be used to compare an external signal to one of the following:

- An internal programmable reference voltage
- A programmable divided version of VBATL
- GND

The comparator accepts rail-to-rail inputs.

Table 6-5 Wakeup comparator specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit	Comment
Supply Voltage	VBAT	1.7	3.3	3.8	V	
Input Referred mismatch (3-Sigma)	Vos	-20		20	mV	
Internal Reference range	Vref	0.1		1.8	V	
VBATL-based reference	VREF_VBAT	10		80	%	Ratio of VBATL

7 Always-on Domain

The Always-on (AON) Domain contains the following:

- Retention registers

  A minimum set of registers that need to be retained through the sleep/wakeup cycles. For example, DC-DC startup parameters and XO configuration.

- Always-on GPIOs

  Up to 8 GPIOs that can be used as external wakeup sources. One example of their usage is to be connected to an external sensor. The sensor can stay ON while GR551x goes to sleep; based on a sensor event, the AON GPIO can be used to wake up GR551x.

7.1 WDT

7.1.1 Introduction

The Always-on Watchdog Timer (AON_WDT) is a 32-bit hardware timer that generates a system-level reset or interrupts if the firmware fails to set it in time. Since AON_WDT is in AON domain, the timer can keep running when the MCU is off.

7.1.2 Main Features

- 32-bit down counter running with a low power ring oscillator clock (30 kHz to 50 kHz).
- Generates an interrupt before reset and a status bit to notify MCU after chip-level reset.
- Generates an interrupt to warn MCU when the 32-bit down counter reaches alarm values.
- Work independently even when the MCU is off.

7.1.3 Functional Description

7.1.3.1 Block Diagram

The following figure shows the functional blocks of the Always-on watchdog module.

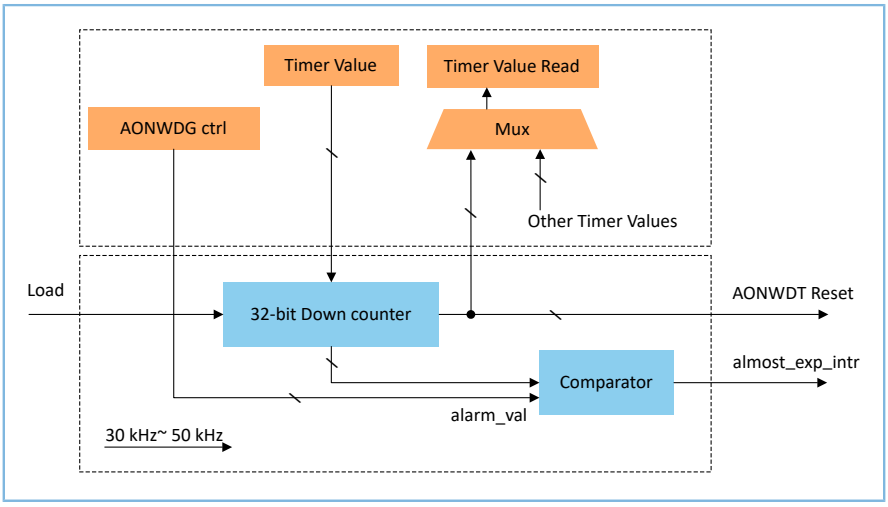

Figure 7-1 Always-on Watchdog Block Diagram

The watchdog starts running when AON_WDT_EN = 1 is synchronized to AON_WDT clock domain, which takes 1 – 3 RNG (Random Number Generators) clock cycles (20 – 100 µs).

If no reload operation is performed, the down counter default value is **0xFFFFFFFF**. Otherwise, the counter will be reloaded with the value of timer_value.

7.1.3.2 Clock

The AON_WDT works in RNG clock (30 kHz – 50 kHz), which is much slower than MCU system clock (16 MHz – 64 MHz). Therefore, synchronization time is needed to propagate the MCU operations. For example, in reload operation, MCU needs to check whether the **VAL_RD** value has changed as expected.

7.1.3.3 Enable/Disable and Timer Reload

- To enable the block, set AON_WDT_EN = 1 and wait until **AON_WDT_RUNNING** being high.
- To disable the block, set AON_WDT_EN = 0 and wait until **AON_WDT_RUNNING** being low.
- To reload down counter, program the desired value to timer_value, then set write-only register AON_WDT_RELOAD = 1. MCU can check read-only register **VAL_RD** to confirm.

7.1.3.4 Alarm

The AON_WDT provides an almost-expired alarm to warn MCU before system reset. By setting **ALARM_VAL= N(non-zero)**, the AON_WDT generates an interrupt N RNG clocks before the system reset.

7.1.4 Registers

7.1.4.1 AON_WDT_CTRL

- **Base Address:** 0xA000C500
- **Offset:** 0x58

- **Reset Value:**0xA0010001

Table 7-1 AON_WDT_CTRL

Bits	Field Name	RW	Reset	Description
31:27	ALARM_VAL	RW	0x14	**Alarm values:** • 0x0: No alarm interrupt. • 0x1 – 0x1F: Generate alarm interrupt N RNG clocks before it expires.
26	AON_WDT_RUNNING	R	0x0	**Value:** • 0x0: The AON_WDT is not running. • 0x1: The AON_WDT is running.
25	AON_WDT_RELOAD	W	0x0	Load the value in timer_value to the AON_WDT down counter.
24	AON_WDT_EN	RW	0x0	**Value:** • 0x0:Disable the AON_WDT. • 0x1:Enable the AON_WDT.
23:16	*EXTERNAL_WAKEUP_TYP	RW	0x1	**Value:** • 0x0: pulse • 0x1: level
15:8	*INVERT_EXTERNAL_WAKE	RW	0x0	**Value:** • 0x0: none • 0x1: invert external wakeup
7:0	*SRC_EN	RW	0x1	Enable external wakeup source. • [0]: AONGPIO0 • [1]: AONGPIO1 • [2]: AONGPIO2 • [3]: AONGPIO3 • [4]: AONGPIO4 • [5]: AONGPIO5 • [6]: AONGPIO6 • [7]: AONGPIO7 **Value:** • 0x0: Disable external wakeup source. • 0x1: Enable external wakeup source.

Note:

AON_WDT_CTRL does not support fields marked with an asterisk (*).

7.1.4.2 VAL_SET

- **Base Address:** 0xA000C500
- **Offset:** 0x90
- **Reset Value:** 0x00000000

Table 7-2 VAL_SET

Bits	Field Name	RW	Reset	Description
31:0	VAL_SET	RW	0x0	This register is shared with multiple timers. To apply the value, set the value then assert corresponding load registers.

7.1.4.3 VAL_RD

- **Base Address:** 0xA000C500
- **Offset:** 0x94
- **Reset Value:** 0x00000000

Table 7-3 VAL_RD

Bits	Field Name	RW	Reset	Description
31:0	VAL_RD	R	0x0	This register is to read the current value and shared with multiple timers.

7.2 RTC

7.2.1 Introduction

The Real-time Calendar (RTC) works as a generic 32-bit low power timer running on a low frequency clock. It provides an alarm signal to either wake up the chip or interrupt the MCU. It also generates an interrupt when it wraps around. Since it is in AON domain, the timer can keep running when the MCU is off.

7.2.2 Main Features

- 32-bit counting up counter
- Clock source: 32.768 kHz RTC clock.
- Clock Pre-divider: programmable by 1, 32, 64, 128, or 256.
- Alarm function: Compares to a 32-bit value and issue a wakeup interrupt when it matches.
- Generates an interrupt when it wraps around.

- Wraps around up to 16 times.

7.2.3 Functional Description

7.2.3.1 Block Diagram

The calendar timer includes:

- One alarm
- One 32-bit up counter
- One 32-bit alarm register
- One clock divider
- One wrap-around counter

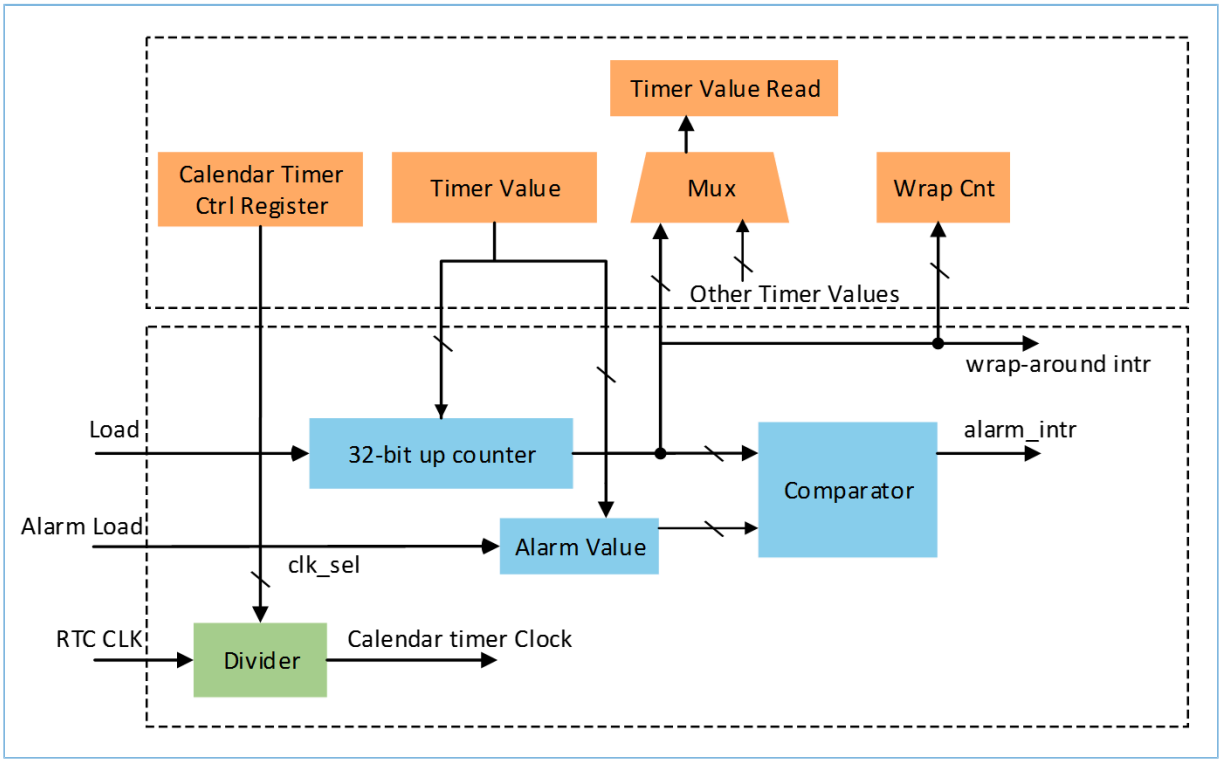

Figure 7-2 Calendar timer block diagram

7.2.3.2 Clocks

The calendar timer is working in 32.768 kHz RTC clock with 5 divide options controlled by register **CLK_SEL**.

The options are in the below table:

Table 7-4 Calendar timer clock options

Value	Option
0	Bypass divider. RTC 32.768 kHz clock

Value	Option
1	/32: 1.024 kHz clock
2	/64: 512 Hz clock
3	/128: 256 Hz clock
4	/256: 128 Hz clock
5 - 7	No clock

7.2.3.3 Counter Load

- To load value to up counter, program desired value to VAL then set write-only register VAL_LOAD = 1. It takes effect in 1 – 3 configured clock cycles as well. MCU can check read-only register **VAL_RD** to confirm (optional).
- To load value to alarm register, program desired value to VAL then set write-only register ALARM_VAL_LOAD = 1. It takes effect in 1 – 3 configured clock cycles as well. MCU can check read-only register **VAL_RD** to confirm (optional).

7.2.3.4 Timer Enable and Disable

- To enable the block, set EN = 1.
- To disable the block, set EN = 0.

Note:

Before enabling the block, make sure the value is loaded successfully.

7.2.3.5 Counter

The up counter starts to count up once the block is enabled. **WRAP_CNT** increments when the counter wraps around. An interrupt is generated as well.

7.2.3.6 Alarm

The calendar timer generates an alarm interrupt when the value of up counter equals to that of alarm register.

7.2.3.7 Reading Timer Value

Set **TIMER_READ_SEL** with its values in the table below, and read the corresponding values of the counter.

Table 7-5 TIMER_READ_SEL description

Value	Content
0	Calendar timer up counter value
1	Always-on watchdog down counter value
2	Sleep timer counter value

Value	Content
3	Calendar timer alarmvalue

7.2.4 Registers

7.2.4.1 RTC_CTRL

- **Base Address:** 0xA000C500
- **Offset:** 0x18
- **Reset Value:** 0x00003100

Table 7-6 RTC_CTRL

Bits	Field Name	RW	Reset	Description
31:14	RSVD	R		Reserved bits
13	WRAP_INT_EN	RW	0x1	Wrap-around interrupt enable. **Value:** • 0x0: Disable • 0x1: Enable
12	ALARM_INT_EN	RW	0x1	Timer alarm interrupt enable. **Value:** • 0x0: Disable • 0x1: Enable
11	RSVD	R		Reserved bits
10:8	CLK_SEL	RW	0x1	Timer clock select. Refer to Table 7-4.
7:4	WRAP_CNT	R	0x0	Wrap-around counter
3	RSVD	R		Reserved bits
2	ALARM_VAL_LOAD	W	0x0	Alarm value load. Write 1 to load the value.
1	VAL_LOAD	W	0x0	Up counter value load. Write 1 to load the value.
0	EN	RW	0x0	Enable calendar timer. **Value:** • 0x0: Disable • 0x1: Enable

7.2.4.2 VAL_SET

- **Base Address:** 0xA000C500
- **Offset:** 0x90

- **Reset Value:** 0x00000000

Table 7-7 VAL_SET

Bits	Field Name	RW	Reset	Description
31:0	VAL_SET	RW	0x0	This register is shared with multiple timers. To apply the value, set the value then assert corresponding load registers.

7.2.4.3 VAL_RD

- **Base Address:** 0xA000C500
- **Offset:** 0x94
- **Reset Value:** 0x00000000

Table 7-8 VAL_RD

Bits	Field Name	RW	Reset	Description
31:0	VAL_RD	R	0x0	This register is to read the current value and shared with multiple timers.

7.3 Sleep Timer

7.3.1 Introduction

The Sleep Timer is a 32-bit hardware timer that can wake up the chip from the Deep Sleep state. It can also be used while the chip is awake.

7.3.2 Main Features

- 32-bit down counter running with 4 different clock sources: external clock (XO 32 kHz), low power ring oscillator clock (30 kHz – 50 kHz), RTC clock (32.768 kHz), low power ring oscillator clock2 (better ppm).
- Three operation modes: Basic Sleep Timer, Single-load Sleep Timer, Auto-reload Sleep Timer.
- An interrupt and status bits are generated.

7.3.3 Functional Description

Sleep Timer has three operation modes.

- Mode 0 – Basic Sleep Timer: This is the original mode of the Sleep Timer. It only counts down in Deep Sleep state and is not activated when the chip is awake.
- Mode 1 – Single-load Sleep Timer: In this mode, the timer starts counting as soon as it is put in this mode and it will continue to count down even when the chip is awake. The timer will stop when the counter hits zero. The time-out signal will be saved as a time-out event.
- Mode 2 – Auto-reload Sleep Timer: Similar to Single-load Sleep Timer mode, the timer will not stop counting down even when the chip is awake. But the timer can auto reload to the same value until it is disabled by MCU.

To switch operation modes, software should issue separate power control commands. To initialize the timer, software needs to write to the sleep timer value register **VAL_SET** and issue a load command. To read the Sleep Timer value, software needs to set the timer select register **TIMER_READ_SEL** first, to choose from the three timers in AON power domain.

7.3.4 Registers

7.3.4.1 Timer Control

Sleep Timer is part of the Power State Controller design and therefore it is mainly controlled via Power State Controller commands.

7.3.4.2 VAL_SET

- **Base Address:** 0xA000C500
- **Offset:** 0x90
- **Reset Value:** 0x00000000

Table 7-9 VAL_SET

Bits	Field Name	RW	Reset	Description
31:0	VAL_SET	RW	0x0	This register is shared with multiple timers. To apply the value, set the value then assert corresponding load registers.

7.3.4.3 VAL_RD

- **Base Address:** 0xA000C500
- **Offset:** 0x94
- **Reset Value:** 0x00000000

Table 7-10 VAL_RD

Bits	Field Name	RW	Reset	Description
31:0	VAL_RD	R	0x0	This register is to read the current value and shared with multiple timers.

8 Reset

GR551x Reset includes system reset and power reset.

8.1 System Reset

System Reset functionality is illustrated in Figure 8-1.

- Power State Controller generates MCU Core Reset (**mcu_core_rst_n**), which goes directly to Cortex®-M4 core reset input.
- In order to support CPU-initiated system reset, the System Reset Request output of Cortex®-M4 is combined with **mcu_core_rst_n**, and the resulting System Reset (**sys_rst_n**) is connected to MCU peripherals as well as the XF Cache.

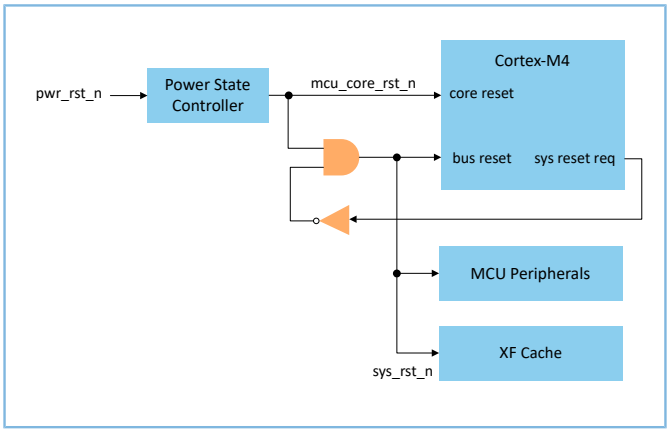

Figure 8-1 System reset

8.2 Power Reset

Power reset is used to reset the Power Sequencer that resides in the AON power domain, as illustrated in Figure 8-2. There are two sources for power reset:

1. **CHIP_EN** pad through the POR circuit, which works as a cold reset.
2. Watchdog reset from the MCU subsystem, which resides in the MCU power domain.

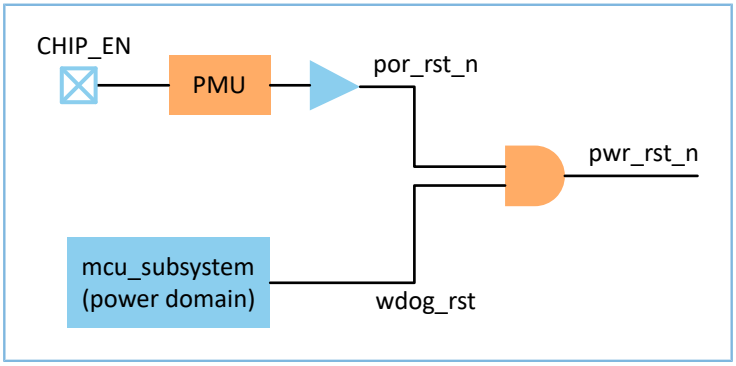

Figure 8-2 Power reset

9 Clock

GR551x clocks are generated using a crystal oscillator (XO) that requires a 32 MHz external crystal, which is needed to meet the offset requirements of the Bluetooth Core Spec v5.1. Figure 9-1 shows the clocking scheme of GR551x.

A divided down version of the XO clock can be muxed out through one of the AON GPIOs to provide an accurate 2 MHz clock output to an external peripheral.

To reduce power consumption, an integer clock supporting Phase-Locked Loop (PLL) is used to generate the main system clock to both RF and digital sections of a GR551x SoC. Frequency dividers are used to provide slower frequencies for different blocks of GR551x. PLL is also used to be tuned to the desired frequency hop for each packet. The settling time of the PLL is designed to be within the time defined by the Bluetooth Core Spec v5.1 to maintain the IFS timing.

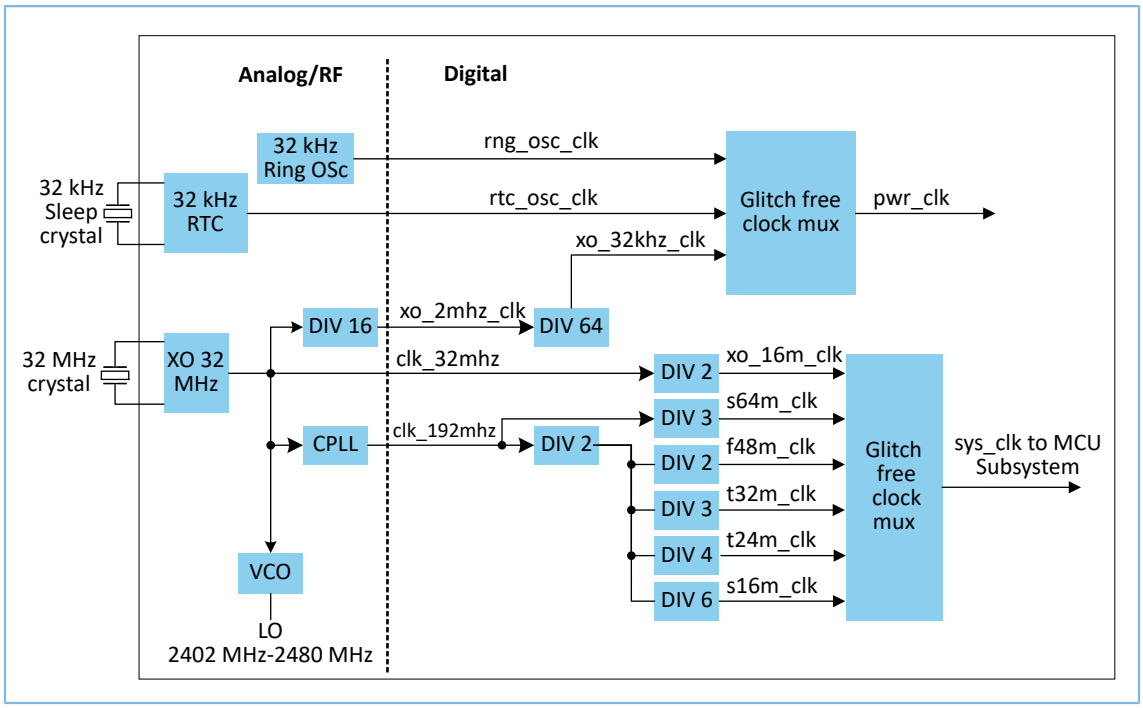

Figure 9-1 GR551x clocking scheme

9.1 XO

XO clock is a crystal-based oscillator that connects to a 32 MHz external crystal. The startup time of the XO clock depends on the crystal type as well as the internal bias current. The bias current is programmable to reduce the current consumption when the XO clock settles.

Table 9-1 XO clock specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
XO frequency	XOfreq		32		MHz
Accuracy	XOacc			100	ppm

9.2 Ring Oscillator

The ring oscillator clock is a low-frequency (32 kHz), ultra-low power clock that is ON as long as the chip is enabled. This clock is used to clock the power sequencers at both the cold and warm boots of the chip.

Table 9-2 Ring oscillator specification

Parameter	Symbol	Min.	Typ.	Max.	Unit
Output Frequency	Ringfreq	24	32	40	kHz
Deviation across Temp	RTCAcc	-500		500	ppm/°C

9.3 Sleep RTC

Real-time clock is a crystal-based accurate clock (32 kHz) that is more stable than the ring oscillator across voltage and temperature. This clock is used for accurate sleep and wakeup.

Table 9-3 Sleep RTC specification

Parameter	Symbol	Min.	Typ.	Max.	Unit
Output Frequency	RTCfreq		32.768		kHz

10 Peripherals

10.1 Pin Mux

10.1.1 Introduction

GR551x has a configurable pin multiplexing module (Pin Mux) which can bring different peripherals on different GPIOs.

10.1.2 Main Features

Each of the GPIO, AON_GPIO, and MSIO pads can be configured to connect to 1 of 8 choices (MUX_0 – MUX_7) of internal signals by programming the DPAD_MUX_SEL_*, AON_PAD_MUX_SEL_*, and MSIO_MUX_SEL_* registers, respectively.

10.1.3 Functional Description

The pin multiplexing choices for all pads are shown in:

- Table 10-1 (GPIO_0 – GPIO_7)
- Table 10-2 (GPIO_8 – GPIO_15)
- Table 10-3 (GPIO_16 – GPIO_23)
- Table 10-4 (GPIO_24 – GPIO_31)
- Table 10-5 (MSIO_0 – MSIO_4)
- Table 10-6 (AON_GPIO_0 – AON_GPIO_7)

Basically,

- GPIO_0 - GPIO_15 are supplied by VDDIO_1
- GPIO_16 - GPIO_31 are supplied by VDDIO_0

There are 8 mux options (from MUX_0 to MUX_7) in the pin mux tables as follows.

Table 10-1 Pin multiplexing for GPIO_0 – GPIO_7

	GPIO_0	GPIO_1	GPIO_2	GPIO_3	GPIO_4	GPIO_5	GPIO_6	GPIO_7
MUX_0	SWD_CLK	SWD_IO	UART0_CTS	UART0_TX	UART0_RX	UART0_RTS	I2S_M_WS	I2S_M_SDO
MUX_1	I2C0_SCL	I2C0_SDA	SIM_PRESENCE	SIM_RST	SIM_IO	SIM_CLK	I2S_S_WS	I2S_S_SDO
MUX_2	I2C1_SCL	I2C1_SDA	SWO	SPI_M_CLK	SPI_M_MOSI	SPI_M_MISO	SPI_M_CS0	SPI_M_CS1
MUX_3	UART1_RTS	UART1_CTS	SPI_S_CS	SPI_S_CLK	SPI_S_MISO	SPI_S_MOSI	UART1_RX	UART1_TX
MUX_4	UART0_TX	UART0_RX	I2C0_SDA	SPI_M_CS1	SPI_M_CS0	SPI_M_MISO	SPI_M_MOSI	SPI_M_CLK
MUX_5	UART1_TX	UART1_RX	PWM0_A	PWM0_B	PWM0_C	I2C0_SCL	I2C0_SDA	PWM1_A
MUX_6	UART0_RTS	UART0_CTS	-	-	-	-	-	-
MUX_7	GPIO_0	GPIO_1	GPIO_2	GPIO_3	GPIO_4	GPIO_5	GPIO_6	GPIO_7

Table 10-2 Pin multiplexing for GPIO_8 – GPIO_15

	GPIO_8	GPIO_9	GPIO_10	GPIO_11	GPIO_12	GPIO_13	GPIO_14	GPIO_15
MUX_0	XQSPI_IO0	XQSPI_CLK	I2S_M_SDI	I2S_M_SCLK	XQSPI_IO3	XQSPI_IO2	XQSPI_IO1	XQSPI_CS
MUX_1	I2C1_SDA	I2C1_SCL	I2S_S_SDI	I2S_S_SCLK	SPI_M_CLK	SPI_M_MOSI	SPI_M_MISO	SPI_M_CS0
MUX_2	QSPI1_IO0	QSPI1_CLK	UART0_TX	UART0_RX	QSPI1_IO3	QSPI1_IO2	QSPI1_IO1	QSPI1_CS
MUX_3	UART1_RX	UART1_TX	-	-	SIM_PRESENCE	SIM_RST	SIM_IO	SIM_CLK
MUX_4	-	-	I2C0_SCL	I2C0_SDA	I2S_M_WS	I2S_M_SDO	I2S_M_SDI	I2S_M_SCLK
MUX_5	PWM1_B	PWM1_C	PWM1_B	PWM1_C	I2S_S_WS	I2S_S_SDO	I2S_S_SDI	I2S_S_SCLK
MUX_6	-	-	-	-	SPI_S_CS	SPI_S_CLK	SPI_S_MISO	SPI_S_MOSI
MUX_7	GPIO_8	GPIO_9	GPIO_10	GPIO_11	GPIO_12	GPIO_13	GPIO_14	GPIO_15

Table 10-3 Pin multiplexing for GPIO_16 – GPIO_23

	GPIO_16	GPIO17	GPIO_18	GPIO_19	GPIO_20	GPIO_21	GPIO_22	GPIO_23
MUX_0	SPI_M_MISO	SPI_M_CS0	QSPI0_CS	QSPI0_IO3	QSPI0_CLK	QSPI0_IO2	QSPI0_IO1	QSPI0_IO0
MUX_1	SPI_S_MOSI	SPI_S_CS	XQSPI_CS	XQSPI_IO3	XQSPI_CLK	XQSPI_IO2	XQSPI_IO1	XQSPI_IO0
MUX_2	SIM_IO	SIM_CLK	-	-	-	-	-	-
MUX_3	I2S_M_SDI	I2S_M_SCLK	-	-	-	-	-	-
MUX_4	I2S_S_SDI	I2S_S_SCLK	-	-	-	-	-	-
MUX_5	QSPI0_IO1	QSPI0_IO2	-	-	-	-	-	-
MUX_6	-	-	-	-	-	-	-	-
MUX_7	GPIO_16	GPIO_17	GPIO_18	GPIO_19	GPIO_20	GPIO_21	GPIO_22	GPIO_23

Table 10-4 Pin multiplexing for GPIO_24 – GPIO_31

	GPIO_24	GPIO_25	GPIO_26	GPIO_27	GPIO_28	GPIO_29	GPIO_30	GPIO_31
MUX_0	SPI_M_CLK	SPI_M_MOSI	I2C1_SDA	-	-	-	I2C1_SCL	SPI_M_CS1
MUX_1	SPI_S_CLK	SPI_S_MISO	UART1_RX	UART1_RTS	UART1_CTS	-	UART1_TX	-
MUX_2	SIM_PRESENCE	SIM_RST	I2C0_SDA	-	-	-	I2C0_SCL	-
MUX_3	I2S_M_WS	I2S_M_SDO	PWM0_C	-	-	-	PWM0_B	PWM0_A
MUX_4	I2S_S_WS	I2S_S_SDO	PWM1_C	-	-	-	PWM1_B	PWM1_A
MUX_5	QSPI0_CLK	QSPI0_IO0	UART0_RX	UART0_RTS	UART0_CTS	-	UART0_TX	QSPI0_IO3
MUX_6	-	-	-	-	-	-	-	-
MUX_7	GPIO_24	GPIO_25	GPIO_26	GPIO_27	GPIO_28	GPIO_29	GPIO_30	GPIO_31

Table 10-5 Pin multiplexing for MSIO_0 – MSIO_4

	MSIO_0	MSIO_1	MSIO_2	MSIO_3	MSIO_4
MUX_0	PWM0_A	PWM0_B	PWM0_C	PWM1_A	PWM1_B
MUX_1	UART0_TX	UART0_RX	-	UART0_RTS	UART0_CTS
MUX_2	UART1_TX	UART1_RX	-	UART1_RTS	UART1_CTS
MUX_3	I2C0_SCL	I2C0_SDA	-	I2C0_SCL	I2C0_SDA
MUX_4	I2C1_SCL	I2C1_SDA	-	I2C1_SCL	I2C1_SDA
MUX_5	-	-	-	-	-
MUX_6	-	-	-	-	-
MUX_7	MSIO_0	MSIO_1	MSIO_2	MSIO_3	MSIO_4

Table 10-6 Pin multiplexing for AON_GPIO_0 – AON_GPIO_7

	AON_GPIO_0	AON_GPIO_1	AON_GPIO_2	AON_GPIO_3	AON_GPIO_4	AON_GPIO_5	AON_GPIO_6	AON_GPIO_7
MUX_0	-	-	SIM_PRESENCE	SIM_RST	SIM_IO	SIM_CLK	-	-
MUX_1	-	-	QSPI1_CS	QSPI1_IO0	QSPI1_IO1	QSPI1_CLK	-	-
MUX_2	-	-	I2S_M_WS	I2S_M_SDO	I2S_M_SDI	I2S_M_SCLK	-	-
MUX_3	-	-	I2S_S_WS	I2S_S_SDO	I2S_S_SDI	I2S_S_SCLK	-	-
MUX_4	-	-	-	-	-	-	-	-
MUX_5	-	QSPI0_CS	PWM0_C	PWM1_A	PWM1_B	PWM1_C	-	-
MUX_6	-	-	-	-	-	-	-	-
MUX_7	AON_GPIO_0	AON_GPIO_1	AON_GPIO_2	AON_GPIO_3	AON_GPIO_4	AON_GPIO_5	AON_GPIO_6	AON_GPIO_7

Note:

Two PWM modules (PWM0 and PWM1) are provided, with each containing three separate output channels: PWMA, PWMB, and PWMC. Frequencies of the three PWM channels in one group are the same, and individual frequency control is not supported. Phase and duty cycle of each channel can be configured via registers.

10.1.4 Registers

10.1.4.1 DPAD_MUX_CTRL_00_07

- **Base Address:** 0xA000E000
- **Offset:** 0x240
- **Reset Value:** 0x77777700

Table 10-7 DPAD_MUX_CTRL_00_07

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_07	RW	0x7	The value of GPIO7 MUX
27:24	DPAD_MUX_SEL_06	RW	0x7	The value of GPIO6 MUX
23:20	DPAD_MUX_SEL_05	RW	0x7	The value of GPIO5 MUX
19:16	DPAD_MUX_SEL_04	RW	0x7	The value of GPIO4 MUX
15:12	DPAD_MUX_SEL_03	RW	0x7	The value of GPIO3 MUX
11:8	DPAD_MUX_SEL_02	RW	0x7	The value of GPIO2 MUX
7:4	DPAD_MUX_SEL_01	RW	0x0	The value of GPIO1 MUX
3:0	DPAD_MUX_SEL_00	RW	0x0	The value of GPIO0 MUX

10.1.4.2 DPAD_MUX_CTRL_08_15

- **Base Address:** 0xA000E000
- **Offset:** 0x244
- **Reset Value:** 0x77777777

Table 10-8 DPAD_MUX_CTRL_08_15

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_15	RW	0x7	Mode for DPAD 15
27:24	DPAD_MUX_SEL_14	RW	0x7	Mode for DPAD 14
23:20	DPAD_MUX_SEL_13	RW	0x7	Mode for DPAD 13
19:16	DPAD_MUX_SEL_12	RW	0x7	Mode for DPAD 12
15:12	DPAD_MUX_SEL_11	RW	0x7	Mode for DPAD 11
11:8	DPAD_MUX_SEL_10	RW	0x7	Mode for DPAD 10
7:4	DPAD_MUX_SEL_09	RW	0x7	Mode for DPAD 9
3:0	DPAD_MUX_SEL_08	RW	0x7	Mode for DPAD 8

10.1.4.3 DPAD_MUX_CTRL_16_23

- **Base Address:** 0xA000E000
- **Offset:** 0x248
- **Reset Value:** 0x77777777

Table 10-9 DPAD_MUX_CTRL_16_23

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_23	RW	0x7	Mode for DPAD 23

Bits	Field Name	RW	Reset	Description
27:24	DPAD_MUX_SEL_22	RW	0x7	Mode for DPAD 22
23:20	DPAD_MUX_SEL_21	RW	0x7	Mode for DPAD 21
19:16	DPAD_MUX_SEL_20	RW	0x7	Mode for DPAD 20
15:12	DPAD_MUX_SEL_19	RW	0x7	Mode for DPAD 19
11:8	DPAD_MUX_SEL_18	RW	0x7	Mode for DPAD 18
7:4	DPAD_MUX_SEL_17	RW	0x7	Mode for DPAD 17
3:0	DPAD_MUX_SEL_16	RW	0x7	Mode for DPAD 16

10.1.4.4 DPAD_MUX_CTRL_24_31

- **Base Address:** 0xA000E000
- **Offset:** 0x24C
- **Reset Value:** 0x77777777

Table 10-10 DPAD_MUX_CTRL_24_31

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_31	RW	0x7	Mode for DPAD 31
27:24	DPAD_MUX_SEL_30	RW	0x7	Mode for DPAD 30
23:20	DPAD_MUX_SEL_29	RW	0x7	Mode for DPAD 29
19:16	DPAD_MUX_SEL_28	RW	0x7	Mode for DPAD 28
15:12	DPAD_MUX_SEL_27	RW	0x7	Mode for DPAD 27
11:8	DPAD_MUX_SEL_26	RW	0x7	Mode for DPAD 26
7:4	DPAD_MUX_SEL_25	RW	0x7	Mode for DPAD 25
3:0	DPAD_MUX_SEL_24	RW	0x7	Mode for DPAD 24

10.1.4.5 AON_PAD_MUX_CTRL

- **Base Address:** 0xA000E000
- **Offset:** 0x290
- **Reset value:** 0x00777770

Table 10-11 AON_PAD_MUX_CTRL

Bits	Field Name	RW	Reset	Description
31:23	RSVD	R		Reserved bits
22:20	AON_PAD_MUX_SEL_05	RW	0x7	Mode for AON PAD [5]
19	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
18:16	AON_PAD_MUX_SEL_04	RW	0x7	Mode for AON PAD [4]
15	RSVD	R		Reserved bits
14:12	AON_PAD_MUX_SEL_03	RW	0x7	Mode for AON PAD [3]
11	RSVD	R		Reserved bits
10:8	AON_PAD_MUX_SEL_02	RW	0x7	Mode for AON PAD [2]
7	RSVD	R		Reserved bits
6:4	AON_PAD_MUX_SEL_01	RW	0x7	Mode for AON PAD [1]
3:0	RSVD	R		Reserved bits

10.1.4.6 MSIO_PAD_MUX_CTRL

- **Base Address:** 0xA000E000
- **Offset:** 0x294
- **Reset Value:** 0x00077777

Table 10-12 MSIO_PAD_MUX_CTRL

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R		Reserved bits
18:16	MSIO_MUX_SEL_04	RW	0x7	Mode for MSIO PAD [4]
15	RSVD	R		Reserved bits
14:12	MSIO_MUX_SEL_03	RW	0x7	Mode for MSIO PAD [3]
11	RSVD	R		Reserved bits
10:8	MSIO_MUX_SEL_02	RW	0x7	Mode for MSIO PAD [2]
7	RSVD	R		Reserved bits
6:4	MSIO_MUX_SEL_01	RW	0x7	Mode for MSIO PAD [1]
3	RSVD	R		Reserved bits
2:0	MSIO_MUX_SEL_00	RW	0x7	Mode for MSIO PAD [0]

10.2 GPIO

10.2.1 Introduction

GR551x has GPIOs which can connect to various digital and mixed-signal interfaces.

10.2.2 Main Features

- 32 digital GPIOs for 2 groups, each group has 16 GPIOs.

- 8 digital GPIOs in the AON domain
- 5 mixed-signal GPIOs
- All IOs remaining at the last state before the system enters the sleep mode
- Default IO state: Input mode with pull-down resistor

10.2.3 Functional Description

32 digital GPIOs can be used:

- For general purpose input/output GPIOs.
- To multiplex different peripherals on different output pins of the package.
- To multiplex debug signals from different blocks of the GR551x SoC.

8 digital GPIOs in the AON domain can be configured:

- As wakeup sources from deep sleep.
- Output 2 MHz clocks.
- As digital GPIOs for input/output.
- To multiplex some different peripherals.

5 mixed-signal GPIOs can be configured:

- As input to the sense ADC or Wakeup Comparator.
- As digital GPIOs for input /output.
- To multiplex some different peripherals.

A functional drawing of the GPIO pad is shown in Figure 10-1. The digital GPIO pads can be configured to set direction, enable pull-up/pull-down resistors and enable output retention.

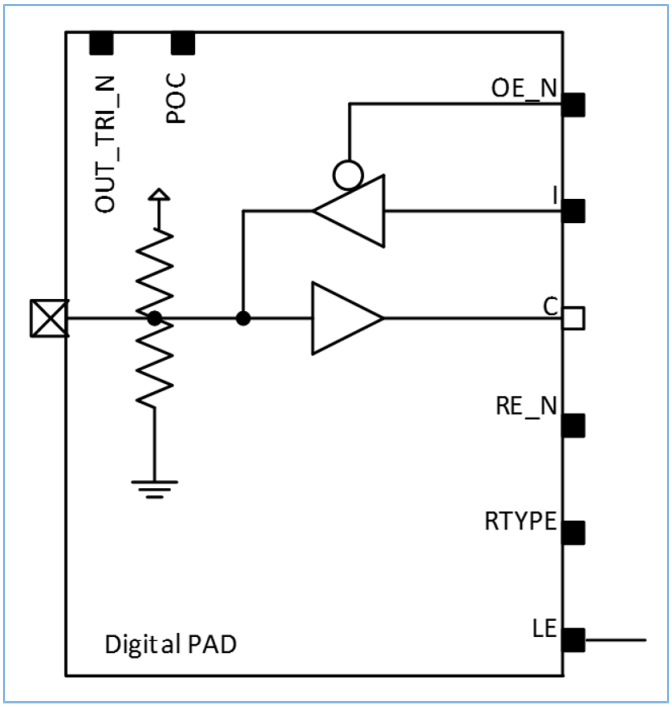

Figure 10-1 GPIO pad diagram

10.2.4 Registers

10.2.4.1 DPAD_RE_N

- **Name:** GPIO pull up/down resistor enable register
- **Description:** This register decides whether to use a pull up/down resistor.
- **Base Address:** 0xA000E000
- **Offset:** 0x208
- **Reset Value:** 0x00000000

Table 10-13 DPAD_RE_N

Bits	Field Name	RW	Reset	Description
31:0	DPAD_RE_N	RW	0x0	Enable a pull up/down resistor to the RETENTION PAD. **Value:** • 0x0: Enable resistor inputs to pad (GPIO_0 - GPIO_31) • 0x1: Disable resistor inputs to pad (GPIO_0 - GPIO_31)

10.2.4.2 DPAD_RTYPE

- **Name:** GPIO pull resistor type control register
- **Description:** This register chooses the pull resistor type.

- **Base Address:** 0xA000E000
- **Offset:** 0x210
- **Reset Value:** 0x00000000

Table 10-14 DPAD_RTYPE

Bits	Field Name	RW	Reset	Description
31:0	DPAD_RTYPE	RW	0x0	Resistor type inputs to the RETENTION PAD. **Value:** • 0x0: Pull down (when DPAD_RE_N = 0) • 0x1: Pull up (when DPAD_RE_N = 0)

10.2.4.3 GPIOx_DATA

- **Name:** GPIO data value register
- **Description:** This register contains the input data and can write data output register.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0000
- **Reset Value:**0x00000000

Table 10-15 GPIO data value register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	DATA	RW	0x0	Data value [15:0] **Read:** The input level state(high or low) at pin **Write:** To data output register

10.2.4.4 GPIOx_DATA_OUT

- **Name:** GPIO data output register
- **Description:** This register contains the output data.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0004
- **Reset Value:**0x00000000

Table 10-16 GPIO data output register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
15:0	DATA_OUT	RW	0x0	Data output register value [15:0] **Read:** Current value of data output register **Write:** To data output register

10.2.4.5 GPIOx_OUT_EN

- **Name:** GPIO output enable register
- **Description:** This register is used to enable output.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0010
- **Reset Value:** 0x00000000

Table 10-17 GPIO output enable register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0x0	Output enable set [15:0] **Read:** • 0x0: As input • 0x1: As output **Write:** • 0x0:No effect. • 0x1:Set the output enable bit

10.2.4.6 GPIOx_OUT_CLR

- **Name:** GPIO output clear register
- **Description:** This register is used to disable output.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0014
- **Reset Value:** 0x00000000

Table 10-18 GPIO output clear register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0x0	Output enable clear [15:0]

Bits	Field Name	RW	Reset	Description
				Read: • 0x0: As input • 0x1:As output **Write:** • 0x0:No effect • 0x1:Clear the output enable bit

10.2.4.7 GPIOx_INT_EN

- **Name:** GPIO interrupt enable register
- **Description:** This register is used to enable GPIO interrupt.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0020
- **Reset Value:** 0x00000000

Table 10-19 GPIO interrupt enable register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0x0	Interrupt enable set [15:0] **Read:** • 0x0: Interrupt disabled • 0x1: Interrupt enabled **Write:** • 0x0: No effect • 0x1: Set the interrupt enable bit

10.2.4.8 GPIOx_INT_CLR

- **Name:** GPIO interrupt clear register
- **Description:** This register is used to disable GPIO interrupt.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0024
- **Reset Value:** 0x00000000

Table 10-20 GPIO interrupt clear register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0x0	Interrupt enable clear [15:0] **Read:** • 0x0: Interrupt disabled • 0x1: Interrupt enabled **Write:** • 0x0: No effect • 0x1: Clear the interrupt enable bit

10.2.4.9 GPIOx_INT_TYPE_EN

- **Name:** GPIO interrupt type enable register
- **Description:** This register is used to set GPIO interrupt type. If interrupt type is 0 and interrupt polarity is 0, Low-level trigger interrupt. If interrupt type is 0 and interrupt polarity is 1, High-level trigger interrupt. If interrupt type is 1 and interrupt polarity is 0, Falling edge trigger interrupt. If interrupt type is 1 and interrupt polarity is 1, Rising edge trigger interrupt.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0028
- **Reset Value:** 0x00000000

Table 10-21 GPIO interrupt type enable register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0x0	Interrupt type set [15:0] **Read:** • 0x0: LOW or HIGH level • 0x1: Falling edge or rising edge **Write:** • 0x0: No effect • 0x1: Set the interrupt type bit

10.2.4.10 GPIOx_INT_TYPE_CLR

- **Name:** GPIO interrupt type clear register

- **Description:** This register is used to disable GPIO interrupt type.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x002C
- **Reset Value:** 0x00000000

Table 10-22 GPIO interrupt type clear register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0h 0	Interrupt type clear [15:0] **Read:** • 0x0: LOW or HIGH level • 0x1: Falling edge or rising edge **Write:** • 0x0: No effect • 0x1: Clear the interrupt type bit

10.2.4.11 GPIOx_INT_POL_EN

- **Name:** GPIO interrupt polarity enable register
- **Description:** This register is used to enable GPIO interrupt polarity.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0030
- **Reset Value:** 0x00000000

Table 10-23 GPIO interrupt polarity enable register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0h 0	Polarity-level, edge IRQ configuration [15:0] **Read:** • 0x0: LOW level or falling edge • 0x1: HIGH level or rising edge **Write:** • 0x0: No effect • 0x1: Set the interrupt polarity bit

10.2.4.12 GPIOx_INT_POL_CLR

- **Name:** GPIO interrupt polarity disable register
- **Description:** This register is used to disable GPIO interrupt polarity.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0034
- **Reset Value:** 0x00000000

Table 10-24 GPIO interrupt polarity disable register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0x0	Polarity-level, edge IRQ configuration [15:0] **Read:** • 0x0: LOW level or falling edge • 0x1:HIGH level or rising edge **Write:** • 0x0: No effect • 0x1: Clear the interrupt polarity bit

10.2.4.13 GPIOx_INT_STAT

- **Name:** GPIO IRQ status register
- **Description:** This register contains GPIO interrupt status, and used to clear the interrupt request.
- **Base Address:** 0xA0010000 + x*0x1000
- **Offset:** 0x0038
- **Reset Value:** 0x00000000

Table 10-25 GPIO IRQ status register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	STAT_CLR	RW	0x0	Write one to clear interrupt request **Read:** [15:0] IRQ status Register **Write:** • 0x0: No effect • 0x1: To clear the interrupt request.

10.2.4.14 MSIO_VAL

- **Name:** MSIO digital input value register
- **Description:** This register contains MSIO digital input value.
- **Base Address:** 0xA000E000
- **Offset:** 0x220
- **Reset Value:**0x00000000

Table 10-26 MSIO_VAL

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:0	MSIO_VAL	RO	N/A	MSIO digital value (when MSIO_PAD_CFG1.MSIO_A_EN = 1)

10.2.4.15 MSIO_PAD_CFG0

- **Name:** MSIO pad control register 0
- **Description:** This register controls MSIO pad.
- **Base Address:** 0xA000C500
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 10-27 MSIO_PAD_CFG0

Bits	Field Name	RW	Reset	Description
31:29	RSVD	R		Reserved bits
28:24	MSIO_OUT_EN	RW	0x0	MSIO output enable (active low) **Value:** • 0x0: Enable MSIOx output • 0x1: Disable MSIOx output
23:21	RSVD	R		Reserved bits
20:16	MSIO_IN_EN	RW	0x0	MSIO input enable (active low) **Value:** • 0x0: Enable MSIOx input • 0x1: Disable MSIOx input
15:13	RSVD	R		Reserved bits
12:8	MSIO_OUT	RW	0x0	MSIO Drive level value (valid in output mode) **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Drive MSIOx low • 0x1: Drive MSIOx high
7:5	RSVD	R		Reserved bits
4:0	MSIO_R_EN	RW	0x0	MSIO resistor enable (active low) **Value:** • 0x0: Enable MSIOx resistor • 0x1: Disable MSIOx resistor

10.2.4.16 MSIO_PAD_CFG1

- **Name:** MSIO pad control register 1
- **Description:** This register controls MSIO pad.
- **Base Address:** 0xA000C500
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 10-28 MSIO_PAD_CFG1

Bits	Field Name	RW	Reset	Description
31	*SADC_CLK_EN	RW	0x0	ADC clock enable **Value:** • 0x0: Disable ADC clock • 0x1: Enable ADC clock
30:28	*SADC_CLK_SEL	RW	0x0	ADC clock select **Value:** • 0x0: 16 M • 0x1: 8 M • 0x2: 4 M • 0x3: 2 M • 0x4, 0x6: 1.6 M • 0x5, 0x7: 1 M
27	*RSVD	R		Reserved bits
26:22	*MSIO_MCU_OVE	RW	0x0	Use the setting from MCU domain, only valid when MCU domain is ON. **Value:** • 0x0: Disable MSIOx digital setting • 0x1: Enable MSIOx digital setting

Bits	Field Name	RW	Reset	Description
21:16	*DEP_CTRL_WR	RW	0x0	Comm timer register Deep sleep control comm timer register write bit
15	*DEP_CTRL_RD	RO	0x0	Comm timer register Deep sleep control comm timer register read bit
14:13	RSVD	R		Reserved bits
12:8	MSIO_R_TYPE	RW	0x0	MSIO resistor type **Value:** • 0x0: Pull down • 0x1: Pull up
7:5	RSVD	R		Reserved bits
4:0	MSIO_A_EN	RW	0x0	Analog enable control for MSIO Pad **Value:** • 0x0: Analog mode • 0x1: Digital mode

Note:

MSIO_PAD_CFG1 does not support fields marked with an asterisk (*).

10.2.4.17 AON_PAD_CTRL0

- **Name:** AON GPIO pad control register 0
- **Description:** This register controls AON GPIO pad.
- **Base Address:** 0xA000C500
- **Offset:** 0x50
- **Reset value:** 0x00000000

Table 10-29 AON_PAD_CTRL_0

Bits	Field Name	RW	Reset	Description
31:30	*RSVD	R		Reserved bits
29:28	*TIMER_CLK_SEL	RW	0x0	Comm timer clock select **Value:** • 0x0: rng_osc_clk • 0x1: rtc_osc_clk • 0x2, 0x3: rng_2_osc_clk (an RNG clock with better ppm)

Bits	Field Name	RW	Reset	Description
27:24	*RSVD	R		Reserved bits
23:16	*AON_MCU_OVE	RW	0x0	Use the setting from MCU domain, only valid when MCU domain is ON **Value:** • 0x0: Disable AONx digital setting • 0x1: Enable AONx digital setting
15:8	AON_R_TYPE	RW	0x0	Always on PAD resistor type **Value:** • 0x0: Pull down • 0x1: Pull up
7:0	AON_R_EN	RW	0x0	Always on PAD resister enable **Value:** • 0x0: Enable AONx resistor • 0x1: Disable AONx resistor

Note:

AON_PAD_CTRL0 does not support fields marked with an asterisk (*).

10.2.4.18 AON_PAD_CTRL1

- **Name:** AON GPIO pad control register 1
- **Description**: This register controls AON GPIO pad.
- **Base Address:** 0xA000C500
- **Offset:** 0x5C
- **Reset value:** 0x000000FF

Table 10-30 AON_PAD_CTRL_1

Bits	Field Name	RW	Reset	Description
31:30	*TIMER_RD_SEL	RW	0x0	Select which timer value to read **Value:** • 0x0: calendar timer • 0x1: always on watchdog timer • 0x2: sleep timer • 0x3: Calendar timer alarm value
29:26	*RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
25	*OVR_EN	RW	0x0	Enable override for all stdby_n and vdd_iso_n values
24:16	*RSVD	R		Reserved bits
15:8	AON_OUT_VAL	RW	0x0	AON PAD output value (valid when oe_n = 0): • 0x0: Drive AONx low • 0x1: Drive AONx high
7:0	AON_OUT_EN	RW	0xFF	Always on PAD output enable (active low) **Value:** • 0x0: Enable AONx output • 0x1: Disable AONx output

Note:

AON_PAD_CTRL1 does not support fields marked with an asterisk (*).

10.2.5 Electrical Specifications

10.2.5.1 GPIO Electrical Specifications

The electrical parameters for the GPIO_0 – GPIO_31 and AON_GPIO_0 – AON_GPIO_7 are as follows:

Table 10-31 GPIO electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage	VDDIO x 0.7		VDDIO	V
V_{IL}	Input low voltage	VSSIO		VDDIO x 0.3	V
$V_{OH,L}$	Output high voltage, 2 mA, VDD ≥1.7 V	VDD - 0.4		VDD	V
$V_{OH,M}$	Output high voltage, 2.5 mA, VDD ≥ 2.5 V	VDD - 0.4		VDD	V
$V_{OH,H}$	Output high voltage, 2.5 mA, VDD ≥ 3 V	VDD - 0.4		VDD	V
$V_{OL,L}$	Output low voltage, 2 mA, VDD ≥1.7 V	VSS		VSS + 0.4	V
$V_{OL,M}$	Output low voltage, 2.5 mA, VDD ≥ 2.5 V	VSS		VSS + 0.4	V
$V_{OL,H}$	Output low voltage, 2.5 mA, VDD ≥ 3 V	VSS		VSS + 0.4	V
$I_{OL,L}$	Current at VSS+0.4 V, output set low, VDD ≥ 1.7 V	2	2.5		mA
$I_{OL,M}$	Current at VSS+0.4 V, output set low, VDD ≥ 2.5 V	2.5	3		mA
$I_{OH,L}$	Current at VDD-0.4 V, output set high, VDD ≥ 1.7 V	2	2.5		mA
$I_{OH,M}$	Current at VDD-0.4 V, output set high, VDD ≥ 2.5 V	2.5	2.8		mA
$t_{RF,15pF}$	Rise/Fall time, 10% - 90%, 15 pF load			8	ns
$t_{RF,25pF}$	Rise/Fall time, 10% - 90%, 25 pF load			13	ns

Parameter	Description	Min.	Typ.	Max.	Unit
R_{PU}	Pull-up resistance	70	120	150	kΩ
R_{PD}	Pull-down resistance	70	120	150	kΩ
C_{PAD}	Pad capacitance		5		pF

10.3 Timer

10.3.1 Introduction

Timer is a general-purpose timer module, which provides a 32-bit timer. Since Timer is in MCU Subsystem domain, Timer interrupt is not available during sleep.

10.3.2 Main Features

Generate an interrupt when the 32-bit down counter reaches 0.

10.3.3 Functional Description

The Timer can generate an interrupt request signal when the 32-bit down counter reaches **0**. The interrupt request is held until it is cleared by writing to the **INT_STAT** register.

If the Timer count reaches 0 and, at the same time, the software clears a previous interrupt status, the interrupt status is set to 1.

10.3.4 Registers

10.3.4.1 CTRL

- **Name:** Timer Control Register
- **Description:** The CTRL register enables the software to control the Timer unit.
- **Base Address:** 0xA0000000 + x*0x1000
- **Offset:** 0x0000
- **Reset Value:** 0x00

Table 10-32 Timer Control Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	INT_EN	RW	0x0	Timer interrupt enable **Value:** • 0x0: Disable the timer interrupt • 0x1: Enable the timer interrupt

Bits	Field Name	RW	Reset	Description
2:1	RSVD	R		Reserved bits
0	EN	RW	0x0	Disable the timer interrupt **Value:** • 0x0: Disable the timer • 0x1: Enable the timer

10.3.4.2 VAL

- **Name:** Timer Value Register
- **Description:** The VAL register indicates the current value of the decrementing counter.
- **Base Address:** 0xA0000000 + x*0x1000
- **Offset:** 0x04
- **Reset Value:** 0x00

Table 10-33 Timer Value Register

Bits	Field Name	RW	Reset	Description
31:0	COUNT	RW	0x0	Current value

10.3.4.3 RELOAD

- **Name:**Timer Reload Register
- **Description:** When this register is written to, the count is immediately restarted from the new value.
- **Base Address:** 0xA0000000 + x*0x1000
- **Offset:** 0x08
- **Reset Value:** 0x00

Table 10-34 Timer Reload Register

Bits	Field Name	RW	Reset	Description
31:0	COUNT	RW	0x0	Reload value. Write the current value to this register.

10.3.4.4 INT_STAT

- **Name:** Timer Interrupt Status Register
- **Description:** The INT_STAT register indicates the interrupt status from the counter. Write 1 to this register to clear the timer interrupt flag.
- **Base Address:** 0xA0000000 + x*0x1000

- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 10-35 Timer Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	RW	0x0	Timer interrupt. Write one to clear.

10.4 Dual-Timer

10.4.1 Introduction

The Dual-Timer module consists of two programmable 32-bit or 16-bit down counters that can generate interrupts when they reach 0.

10.4.2 Main Features

- A 32-bit or 16-bit counter.
- One of the following timer modes:
  - Free-running
  - Periodic
  - One-shot
- Dual-timer has a prescaler that can divide down the enabled timer clock rate by 1, 16, or 256.

10.4.3 Functional Description

Two timers are defined by default. For each timer, the following modes of operation are available:

- Free-running mode

  The counter wraps after reaching its zero value, and continues to count down from the maximum value. This is the default mode.
- Periodic mode

  The counter generates an interrupt at a constant interval, reloading the original value after wrapping past zero.
- One-shot mode

  The counter generates an interrupt once. When the counter reaches 0, it halts until you reprogram it.

10.4.4 Registers

10.4.4.1 LOADx

- **Name:** Dual-Timer Load Register

- **Description:** This register contains the value from which the counter is to decrease. This is the value used to reload the counter when periodic mode is enabled, and the current count reaches 0.
- **Base Address:** 0xA0002000
- **Offset:** 0x00 + x*0x20
- **Reset Value:** 0x00

Table 10-36 Dual-Timer Load Register

Bits	Field Name	RW	Reset	Description
31:0	COUNT	RW	0x0	Reload value. Write to this register sets the current value.

10.4.4.2 VALx

- **Name:** Dual-Timer Current Value Register
- **Description:** This register provides the current value of the decrementing counter.
- **Base Address:** 0xA0002000
- **Offset:** 0x04 + x*0x20
- **Reset Value:** 0xFFFFFFFF

Table 10-37 Dual-Timer Current Value Register

Bits	Field Name	RW	Reset	Description
31:0	COUNT	RO	0xFFFFFFFF	Current value

10.4.4.3 CTRLx

- **Name:** Dual-Timer Control Register
- **Description:** The CTRL register enables the software to control the Dual-Timer unit.
- **Base Address:** 0xA0002000
- **Offset:** 0x08 + x*0x20
- **Reset Value:** 0x20

Table 10-38 Dual-Timer Control Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	EN	RW	0x0	Timer enable **Value:** • 0x0: Timer disabled, default • 0x1: Timer enabled

Bits	Field Name	RW	Reset	Description
6	MODE	RW	0x0	Timer mode **Value:** • 0x0: Timer is in free-running mode, default • 0x1: Timer is in periodic mode.
5	INT_EN	RW	0x1	Interrupt enable **Value:** • 0x0: Timer interrupt disabled • 0x1: Timer interrupt enabled, default
4	RSVD	R		Reserved bits
3:2	PRE	RW	0x0	Prescale bits **Value:** • 0x00: 0 stage of prescale, clock is divided by 1, default. • 0x01: 4 stages of prescale, clock is divided by 16. • 0x10: 8 stages of prescale, clock is divided by 256. • 0x11: Undefined, do not use.
1	SIZE	RW	0x0	Select 16-bit or 32-bit counter operation **Value:** • 0x0: 16-bit counter, default • 0x1: 32-bit counter
0	COUNT_MODE	RW	0x0	Select Free-running or One-shot mode. **Value:** • 0x0: Free-running mode, default • 0x1: One-shot mode

10.4.4.4 INT_CLRx

- **Name:** Dual-Timer Interrupt Clear Register
- **Description:** Any write to the INT_CLRx register clears the interrupt output from the counter.
- **Base Address:** 0xA0002000
- **Offset:** 0x0C + x*0x20
- **Reset Value:**0x00

Table 10-39 Dual-Timer Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:0	STAT	WO	0x0	Clear the Dual-Timer interrupt event.

10.4.4.5 INT_RSTATx

- **Name:** Dual-Timer Raw Interrupt Status Register
- **Description:** The INT_RSTATx register indicates the raw interrupt status from the counter.
- **Base Address:** 0xA0002000
- **Offset:** 0x10 + x*0x20
- **Reset Value:**0x00

Table 10-40 Dual-Timer Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	RW	0x0	Raw interrupt status from the counter

10.4.4.6 INT_STATx

- **Name:** Dual-Timer Interrupt Status Register
- **Description:** The INT_STATx register indicates the masked interrupt status from the counter.
- **Base Address:** 0xA0002000
- **Offset:** 0x14 + x*0x20
- **Reset Value:** 0x00

Table 10-41 Dual-Timer Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	RW	0x0	Enable interrupt status from the counter.

10.5 System Watchdog

10.5.1 Introduction

The System Watchdog Timer (WDT) is used to regain control if the system fails due to a software error after an external device fails to respond as expected. The System WDT can generate an interrupt or a reset when the counter reaches 0. Since System watchdog is in MCU Subsystem domain, watchdog reset is not available during sleep.

10.5.2 Main Features

- 32-bit down counter running with a system clock.
- Generate an interrupt to warn MCU or reset signal to restart MCU when the 32-bit down counter reaches 0.

10.5.3 Functional Description

The System watchdog monitors the interrupt and asserts a reset request signal when the counter reaches 0, and the counter is stopped. On the next enabled clock edge, the counter is reloaded from the **LOAD** register and the countdown sequence continues. If the interrupt is not cleared when the time the counter reaches 0 next time, the watchdog module reasserts the reset signal.

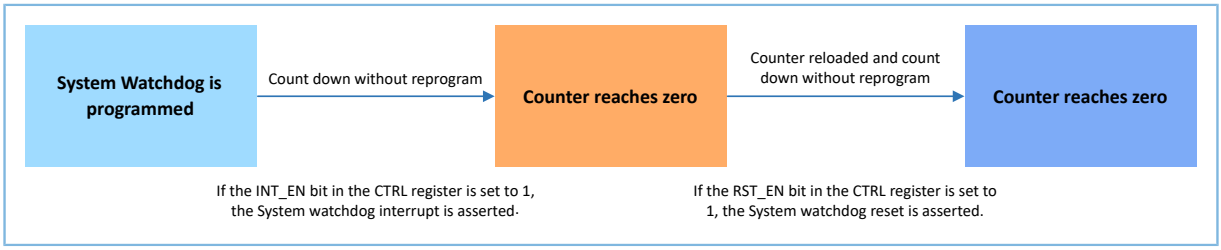

Figure 10-2 System watchdog operation flow diagram

10.5.4 Registers

10.5.4.1 LOAD

- **Name:** System Watchdog Load Register
- **Description:** The LOAD register contains the value from which the counter is to decrease. When this register is written to, the count is immediately restarted from the new value. The minimum valid value for LOAD register is 1.
- **Base Address:** 0xA0008000
- **Offset:** 0x0000
- **Reset Value:** 0xFFFFFFFF

Table 10-42 System Watchdog Load Register

Bits	Field Name	RW	Reset	Description
31:0	COUNT	RW	0xFFFFFFFF	Reload value. A write to this register sets the current value.

10.5.4.2 VAL

- **Name:** System Watchdog Value Register
- **Description:** This is a read-only register used to indicate the current value of the decrementing counter.
- **Base Address:** 0xA0008000

- **Offset:** 0x04
- **Reset Value:** 0xFFFFFFFF

Table 10-43 System Watchdog Value Register

Bits	Field Name	RW	Reset	Description
31:0	COUNT	RO	0xFFFFFFFF	Current value of the watchdog counter

10.5.4.3 CTRL

- **Name:** System Watchdog Control Register
- **Description:** The CTRL register enables the software to control the watchdog unit.
- **Base Address:** 0xA0008000
- **Offset:** 0x0008
- **Reset Value:** 0x00

Table 10-44 System Watchdog Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	RST_EN	RW	0x0	Enable watchdog reset output. **Value:** • 0x0: Disable the reset. • 0x1: Enable the reset.
0	INT_EN	RW	0x0	Enable the interrupt event. Reload the counter from the value in LOAD register when the interrupt is enabled, after previously being disabled. **Value:** • 0x0: Disable the counter and the interrupt. • 0x1: Enable the counter and the interrupt.

10.5.4.4 INT_CLR

- **Name:** System Watchdog Clear Interrupt Register
- **Description:** A write of any value to the INT_CLR register clears the watchdog interrupt, and reloads the counter from the value in LOAD register.
- **Base Address:** 0xA0008000
- **Offset:** 0x0C
- **Reset Value:**0x00000000

Table 10-45 System Watchdog Clear Interrupt Register

Bits	Field Name	RW	Reset	Description
31:0	CLR	WO	0x0	Clear the watchdog interrupt event.

10.5.4.5 INT_RAW_STAT

- **Name:** System Watchdog Raw Interrupt Status Register
- **Description:** The INT_RAW_STAT register indicates the raw interrupt status from the counter.
- **Base Address:** 0xA0008000
- **Offset:** 0x10
- **Reset Value:**0x00000000

Table 10-46 System Watchdog Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RAW_STAT	RO	0x0	Raw interrupt status from the counter

10.5.4.6 INT_STAT

- **Name:** System Watchdog Interrupt Status Register
- **Description:** The INT_STAT register indicates the masked interrupt status from the counter.
- **Base Address:** 0xA0008000
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 10-47 System Watchdog Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	RO	0x0	Enable interrupt status from the counter.

10.5.4.7 LOCK

- **Name:** System Watchdog Lock Register
- **Description:** The LOCK register disables write access to all other registers. This is to prevent rogue software from disabling the watchdog functionality. Writing a value of 0x1ACCE551 enables write access to all other registers. Write any other value disables write access.
- **Base Address:** 0xA0008000

- **Offset:** 0x0C00
- **Reset Value:** 0x00000000

Table 10-48 System Watchdog Lock Register

Bits	Field Name	RW	Reset	Description
31:1	WR_EN	RW	0x0	Enable write access to all other registers by writing 0x1ACCE551. Disable write access by writing any other value.
0	WR_EN_STAT	RW	0x0	Register write enable status **Value:** • 0x0: Write access to all other registers is enabled. This is the default. • 0x1: Write access to all other registers is disabled.

10.6 I2C

10.6.1 Introduction

The Inter-Integrated Circuit (I2C) bus is a two-wire serial interface that is widely used for low-speed communication between chipsets in a system. Using I2C, a microcontroller can communicate with peripherals such as sensors, data converters, and I/O interfaces.

GR551x has two I2C module instances: I2C0 and I2C1, which can be configured as either Master or Slave.

10.6.2 Main Features

The two-wire I2C serial interface consists of a serial data line (SDA) and a serial clock line (SCL) that support:

- Standard Mode (up to 100 Kbit/s)
- Fast Mode (up to 400 Kbit/s)
- Fast Plus Mode (up to 1000 Kbit/s)
- High-speed Mode (up to 2.0 Mbit/s)
- Clock synchronization
- Master or Slave I2C operation
- 7-bit or 10-bit addressing
- Mixed 7-bit and mixed 10-bit format transfer
- Transmit and receive buffers
- Interrupt or polled-mode operation
- Handle Bit and Byte waiting at all bus speeds
- Simple software interface

- DMA handshaking interface

10.6.3 Functional Description

The I2C Controller is made up of an APB slave interface, an I2C interface, and FIFO logic to maintain coherency between the two interfaces as well as DMA and interrupt logics. A simplified block diagram is illustrated in Figure 10-3.

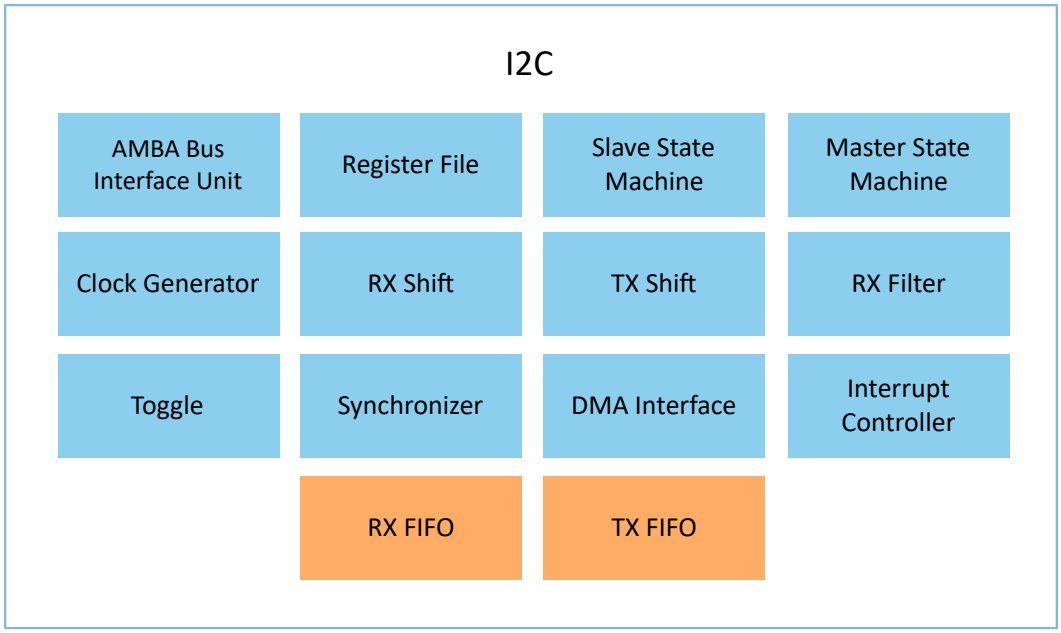

Figure 10-3 I2C block diagram

10.6.3.1 I2C Operation

The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only when the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wired-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.

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

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/$\overline{W}$) to determine if the master wants to transmit data to or receive data from the slave. The slave then sends an acknowledge (ACK) pulse after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behavior is illustrated in Figure 10-4.

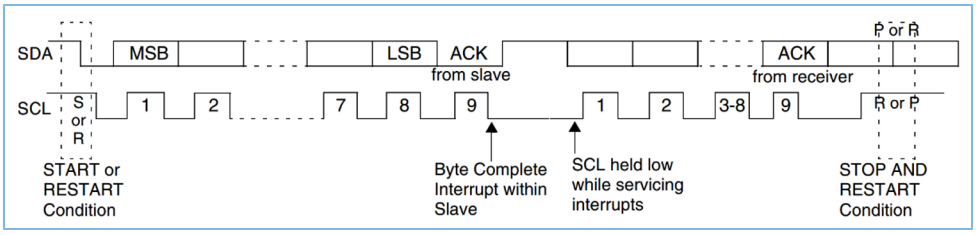

Figure 10-4 Data transfer on the I2C Bus

10.6.3.2 START and STOP Conditions

When I2C operates as a Master, putting data into the transmit FIFO causes the I2C block to generate a START condition on the I2C bus. Writing **1** to **IC_DATA_CMD[8]** causes the I2C block 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 the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When a master wants to start a transmission on the bus, the master issues a START condition. This is defined as a high-to-low transition of the SDA signal while SCL is high. When a master wants to terminate the transmission, the master issues a STOP condition. This is defined as a low-to-high transition of the SDA line while SCL is high. Figure 10-5 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when the SCL is high.

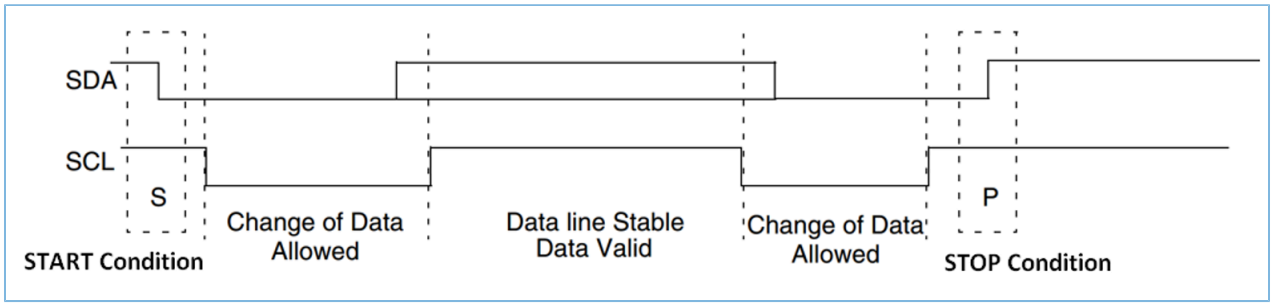

Figure 10-5 Timing of START and STOP conditions

10.6.3.3 Addressing Slave

There are two address formats: the 7-bit address format and the 10-bit address format. The I2C block supports mixed read and write combined format transactions in 7-bit or 10-bit addressing modes. The I2C block 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).

7-bit Address Format

Using the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the least significant bit (bit 0) (LSB) is the R/W bit as shown in Figure 10-6. When bit 0 (R/$\overline{W}$) is set to **0**, the master writes to the slave. When bit 0 (R/$\overline{W}$) is set to **1**, the master reads from the slave.

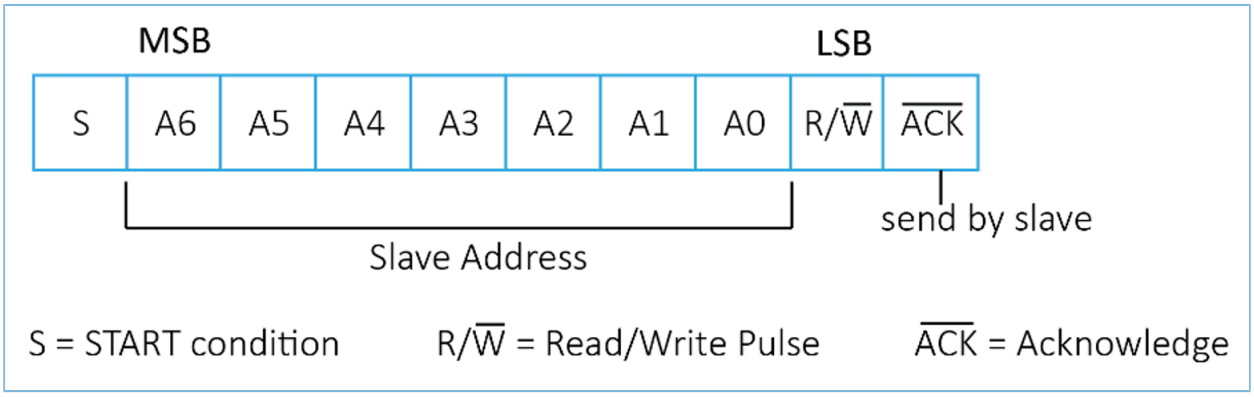

Figure 10-6 7-bit Address Format

10-bit Address Format

Using the 10-bit address format, 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 slave that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slave address (bits 9:8), and the LSB bit (bit 0) is the R/$\overline{W}$ bit. The second byte transferred sets bits 7:0 of the slave address. Figure 10-7 shows the 10-bit address format.

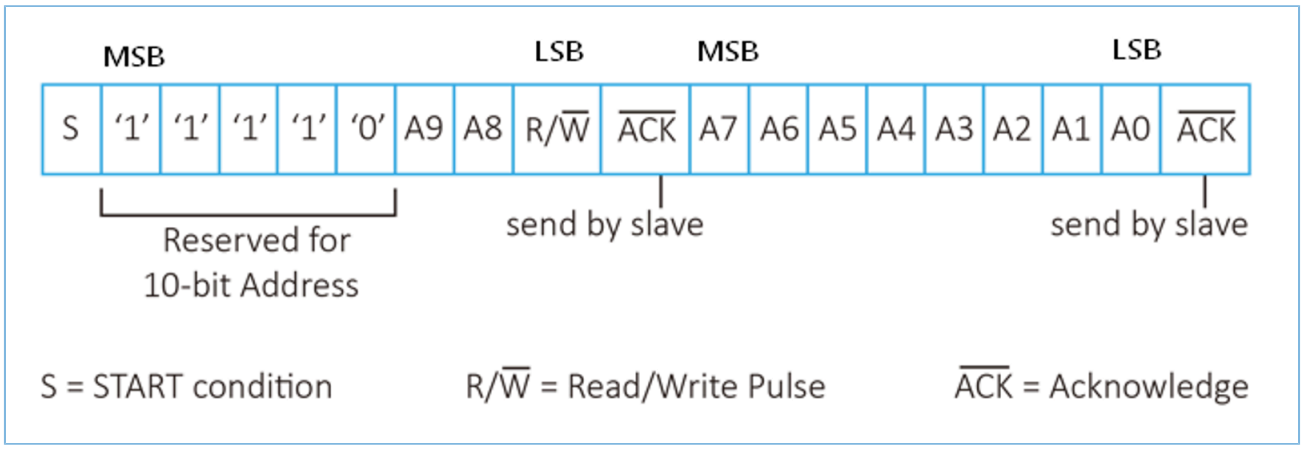

Figure 10-7 10-bit Address Format

Table 10-49 defines the special purpose and reserved first byte addresses.

Table 10-49 I2C Definition of bits in first byte

Slave Address	R/W Bit	Description
0000 000	0	General Call Address. DW_apb_i2c places the data in the receive buffer and issues a General Call interrupt.
0000 000	1	START byte
0000 001	X	CBUS address. I2C ignores these accesses.
0000 010	X	Reserved
0000 010	X	Reserved
0000 011	X	High-speed master code

Slave Address	R/W Bit	Description
1111 1XX	X	Reserved
1111 0XX	X	10-bit slave addressing
0001 000	X	SMBus Host
0001 100	X	SMBus Alert Response Address
1100 001	X	SMBus Device Default Address

10.6.3.4 Transmitting and Receiving

A 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/$\overline{W}$ bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the ACK signal. 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 10-8, the slave-receiver responds to the master-transmitter with an ACK pulse after every data byte is received.

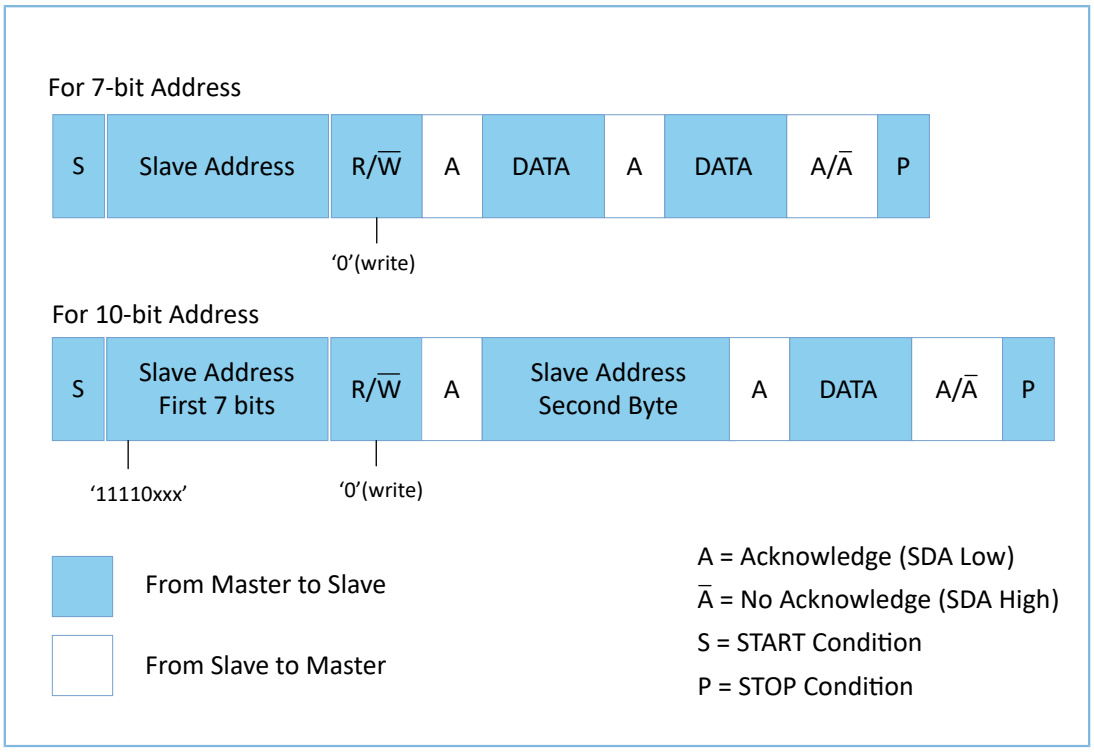

Figure 10-8 Master-transmitter protocol

Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 10-9, then the master responds to the slave-transmitter with an ACK 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.

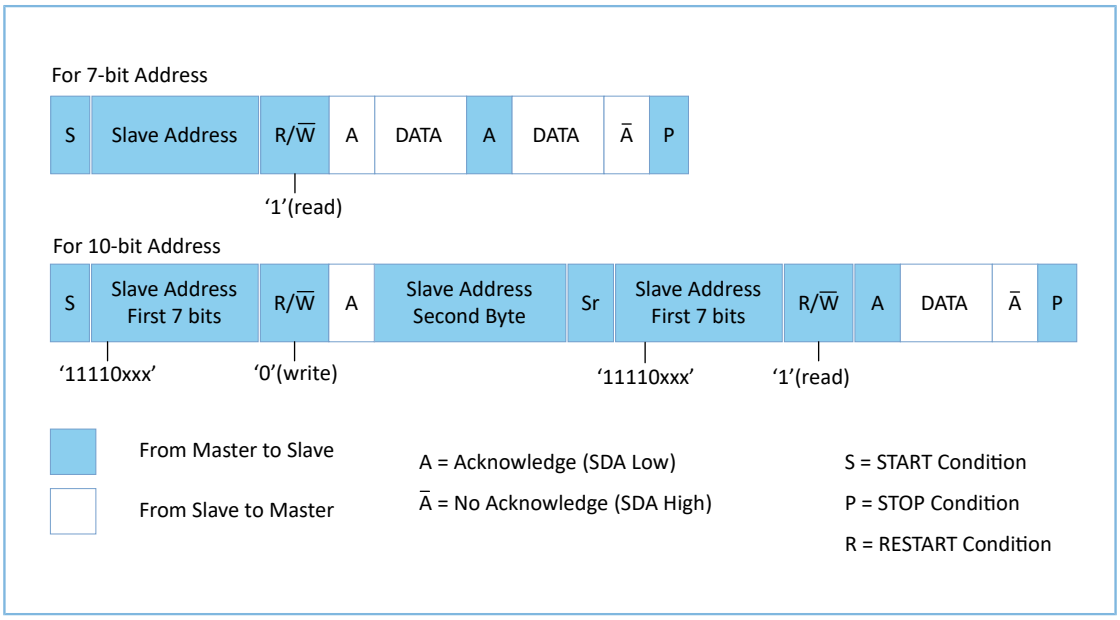

Figure 10-9 Master-Receiver Protocol

10.6.3.5 START BYTE Transfer

The START BYTE transfer protocol is set up for systems that do not have an on-board dedicated I2C hardware module. When the I2C block 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 the I2C block is a master, it supports the generation of START BYTE transfers that a slave device might require at the beginning of every transfer.

This protocol consists of seven zeros being transmitted followed by a one, as illustrated in Figure 10-10. 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.

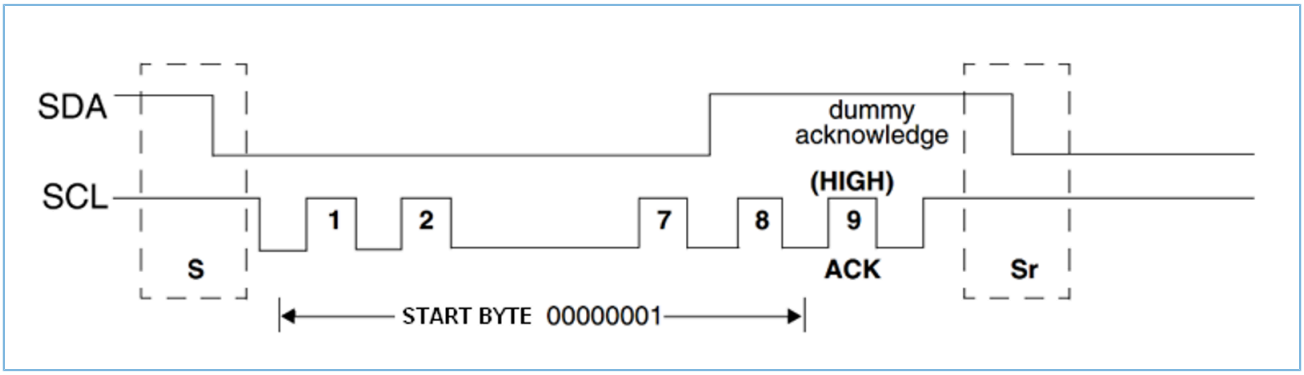

Figure 10-10 START BYTE Transfer

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 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.

10.6.4 Registers

10.6.4.1 CTRL

- **Name:** I2C Control Register
- **Description:** This register can be written only when the I2C is disabled, which corresponds to the EN[0] register being set to 0. Writes at other times have no effect.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x00
- **Reset Value:** 0x0000007F

Table 10-50 I2C Control Register

Bits	Field Name	RW	Reset	Description
31:11	RSVD	R		Reserved bits
10	STOP_DET_M_ACTIVE	RW	0x0	In Master mode **Value:** • 0x1: issues the STOP_DET interrupt only when master is active. • 0x0: issues the STOP_DET irrespective of whether master is active or not.
9	RSVD	R		
8	TX_EMPTY_CTRL	RW	0x0	This bit controls the generation of the TX_EMPTY interrupt, as described in the RAW_INT_STAT register. **Value:** • 0x0 (DISABLED): Default behavior of TX_EMPTY interrupt • 0x1 (ENABLED): Controlled generation of TX_EMPTY interrupt
7	STOP_DET_INT	RW	0x0	In slave mode **Value:** • 0x0: issues the STOP_DET irrespective of whether it's addressed or not

Bits	Field Name	RW	Reset	Description
				• 0x1: issues the STOP_DET interrupt only When it is addressed • 0x0 (DISABLED): Slave issues STOP_DET interrupt always. • 0x1 (ENABLED): Slave issues STOP_DET interrupt only if addressed. **Note:** During a general call address, the slave does not issue the STOP_DET interrupt if STOP_DET_INT = 1, 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 (S_ADDR).
6	S_DIS	RW	0x1	This bit controls whether I2C has its slave disabled. By default, the slave is always disabled (in reset state as well). If you need to enable it after reset, set this bit to 0. If this bit is set (slave is disabled), I2C functions only as a master and does not perform any action that requires a slave. **Note:** Software should ensure that if this bit is written with 0, then bit 0 should also be written with 0. **Value:** • 0x0 (S_EN): Slave mode is enabled. • 0x1 (S_DIS): Slave mode is disabled.
5	RESTART_EN	RW	0x1	Determine whether RESTART conditions may be sent when I2C acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several I2C operations. When RESTART is disabled, the master is prohibited from performing the following functions: • Sending a START BYTE • Performing any high-speed mode operation • Performing direction changes in combined format mode • Performing a read operation with a 10-bit address By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple I2C transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABORT) of the RAW_INT_STAT register. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (DISABLED): Master restart disabled • 0x1 (ENABLED): Master restart enabled
4	ADDR_BIT_M	RW	0x1	This bit controls whether the I2C starts its transfers in 7-bit or 10-bit addressing mode when I2C acting as a master. **Value:** • 0x0 (ADDR_7BITS): Master 7-bit addressing mode • 0x1 (ADDR_10BITS): Master 10-bit addressing mode
3	ADDR_BIT_S	RW	0x1	When acting as a slave, this bit controls whether the I2C responds to 7- or 10-bit addresses. **Value:** • 0x0 (ADDR_7BITS): Slave 7-Bit addressing • 0x1 (ADDR_10BITS): Slave 10-Bit addressing
2:1	SPEED	RW	0x3	These bits control at which speed the I2C operates; its setting is relevant only if one is operating the I2C in master mode. Hardware protects against illegal values being programmed by software. These bits must be programmed appropriately for slave mode also, as it is used to capture correct value of spike filter as per the speed mode. This register should be programmed only with a value in the range of 1 to 3. • 0x1: standard mode (100 kbit/s) • 0x2: fast mode (≤400 kbit/s) or fast plus mode (≤1000 kbit/s) • 0x3: high-speed mode (2 Mbit/s) **Value:** • 0x1 (STANDARD): Standard Speed mode of operation • 0x2 (FAST): Fast or Fast Plus mode of operation • 0x3 (HIGH): High-speed mode of operation
0	M_MODE	RW	0x1	This bit controls whether the I2C master is enabled. **Note:** Software should ensure that if this bit is written with '1', then bit 6 should also be written with '1'. **Value:** • 0x0 (DISABLED): Master mode is disabled • 0x1 (ENABLED): Master mode is enabled

10.6.4.2 TARGET_ADDR

- **Name:** I2C Target Address Register
- **Description:** This register can be written to only when EN[0] is set to 0. You cannot change the TARGET_ADDR address dynamically.

Note:

- If the software or application is aware that the I2C is not using the TARGET_ADDR address for the pending commands in the TX FIFO, then it is possible to update the TARGET_ADDR address even while the TX FIFO has entries (STAT[2]= 0).
- It is not necessary to perform any write to this register if the I2C is enabled as an I2C slave only.

- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x04
- **Reset Value:** 0x00000055

Table 10-51 I2C Target Address Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	SPECIAL	RW	0x0	This bit indicates whether software performs a Device-ID or General Call or START BYTE command. **Value:** • 0x0 (DISABLED): Disable programming of GENERAL_CALL or START_BYTE transmission • 0x1 (ENABLED): Enable programming of GENERAL_CALL or START_BYTE transmission
10	TX_CTRL	RW	0x0	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 I2C. **Value:** • 0x0 (GENERAL_CALL): GENERAL_CALL byte transmission • 0x1 (START_BYTE): START BYTE transmission
9:0	TARGET	RW	0x55	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. If the TARGET_ADDR and S_ADDR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one

Bits	Field Name	RW	Reset	Description
				direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.

10.6.4.3 S_ADDR

- **Name:** I2C Slave Address Register
- **Description:** I2C Slave Address Register
- **Base Address:**0xA000C300 + x*0x100
- **Offset:** 0x08
- **Reset Value:** 0x00000055

Table 10-52 I2C Slave Address Register

Bits	Field Name	RW	Reset	Description
31:10	RSVD	R		Reserved bits
9:0	S_ADDR	RW	0x55	The S_ADDR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only S_ADDR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the EN[0] register being set to 0. Writes at other times have no effect. **Note:** The default values cannot be any of the reserved address locations: 0x00 to 0x07, or 0x78 to 0x7F. The correct operation of the device is not guaranteed if you program the S_ADDR or TARGET_ADDR to a reserved value. Refer to Table 10-49 for a complete list of these reserved values.

10.6.4.4 M_HS_ADDR

- **Name:** I2C High-Speed Master Mode Code Address Register
- **Description:** I2C High-Speed Master Mode Code Address Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x0C
- **Reset Value:** 0x00000001

Table 10-53 I2C High-Speed Master Mode Code Address Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
2:0	M_HS_ADDR	RW	0x1	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 EN[0] register being set to 0. Writes at other times have no effect.

10.6.4.5 DATA_CMD

- **Name:** I2C RX/TX Data Buffer and Command Register
- **Description:** This is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

Note:

In order to continue acknowledging reads, a read command should be written for every byte that is to be received; otherwise the I2C will stop acknowledging.

- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 10-54 I2C RX/TX Data Buffer and Command Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	CMD	W	0x0	This bit controls whether a read or a write is performed. This bit does not control the direction when the I2C acts as a slave. It controls only the direction when it acts as a master. **Value:** • 0x0 (WRITE): Master Write command • 0x1 (READ): Master Read command
7:0	DATA	RW	0x0	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 (DATA) are ignored by the I2C. However, when you read this register, these bits return the value of data received on the I2C interface.

10.6.4.6 SS_CLK_HCOUNT

- **Name:** Standard Speed I2C Clock SCL High Count Register
- **Description:** Standard Speed I2C Clock SCL High Count Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x14
- **Reset Value:**0x00000190

Table 10-55 Standard Speed I2C Clock SCL High Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x0190	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register can be written only when the I2C interface is disabled which corresponds to the EN[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, the value will be set to 6. **Note:** This register must not be programmed to a value higher than 65525, because I2C uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of COUNT + 10.

10.6.4.7 SS_CLK_LCOUNT

- **Name:** Standard Speed I2C Clock SCL Low Count Register
- **Description:** Standard Speed I2C Clock SCL Low Count Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x18
- **Reset Value:** 0x000001D6

Table 10-56 Standard Speed I2C Clock SCL Low Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x01D6	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing.

Bits	Field Name	RW	Reset	Description
				This register can be written only when the I2C interface is disabled which corresponds to the EN[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, the value will be set to 8.

10.6.4.8 FS_CLK_HCOUNT

- **Name:** Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- **Description:**Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x1C
- **Reset Value:**0x0000003C

Table 10-57 Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x003C	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register can be written only when the I2C interface is disabled, which corresponds to the EN[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, the value will be set to 6.

10.6.4.9 FS_CLK_LCOUNT

- **Name:** Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- **Description:**Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x20
- **Reset Value:**0x00000082

Table 10-58 Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
15:0	COUNT	RW	0x0082	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register can be written only when the I2C interface is disabled, which corresponds to the EN[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, the value will be set to 8.

10.6.4.10 HS_CLK_HCOUNT

- **Name:** High Speed I2C Clock SCL High Count Register
- **Description:** High Speed I2C Clock SCL High Count Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x24
- **Reset Value:** 0x00000006

Table 10-59 High Speed I2C Clock SCL High Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x0006	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. 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 can be written only when the I2C interface is disabled, which corresponds to the EN[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, the value will be set to 6.

10.6.4.11 HS_CLK_LCOUNT

- **Name:** High Speed I2C Clock SCL Low Count Register
- **Description:** High Speed I2C Clock SCL Low Count Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x28
- **Reset Value:** 0x00000010

Table 10-60 High Speed I2C Clock SCL Low Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x0010	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. The SCL low time depends on the loading of the bus. For 100pF loading, the SCL low time is 160 ns; For 400 pF loading, the SCL low time is 320 ns. This register can be written only when the I2C interface is disabled, which corresponds to the EN[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, the value will be set to 8.

10.6.4.12 INT_STAT

- **Name:** I2C Interrupt Status Register
- **Description:** Each bit in this register has a corresponding mask bit in the INT_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the RAW_INT_STAT register.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x2C
- **Reset Value:** 0x00000000

Table 10-61 I2C Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:14	RSVD	R		Reserved bits
13	RAW_M_HOLD	R	0x0	See RAW_INT_STAT for a detailed description of RAW_M_HOLD bit. **Value:** • 0x0 (INACTIVE): RAW_M_HOLD interrupt is inactive • 0x1 (ACTIVE): RAW_M_HOLD interrupt is active
12	RAW_RESTART_DET	R	0x0	See RAW_INT_STAT for a detailed description of RAW_RESTART_DET bit. **Value:** • 0x0 (INACTIVE): RAW_RESTART_DET interrupt is inactive • 0x1 (ACTIVE): RAW_RESTART_DET interrupt is active
11	RAW_GEN_CALL	R	0x0	See RAW_INT_STAT for a detailed description of RAW_GEN_CALL bit. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (INACTIVE): RAW_GEN_CALL interrupt is inactive • 0x1 (ACTIVE): RAW_GEN_CALL interrupt is active
10	RAW_START_DET	R	0x0	See RAW_INT_STAT for a detailed description of RAW_START_DET bit. **Value:** • 0x0 (INACTIVE): RAW_START_DET interrupt is inactive • 0x1 (ACTIVE): RAW_START_DET interrupt is active
9	RAW_STOP_DET	R	0x0	See RAW_INT_STAT for a detailed description of RAW_STOP_DET bit. **Value:** • 0x0 (INACTIVE): RAW_STOP_DET interrupt is inactive • 0x1 (ACTIVE): RAW_STOP_DET interrupt is active
8	RAW_ACTIVITY	R	0x0	See RAW_INT_STAT for a detailed description of RAW_ACTIVITY bit. **Value:** • 0x0 (INACTIVE): RAW_ACTIVITY interrupt is inactive • 0x1 (ACTIVE): RAW_ACTIVITY interrupt is active
7	RAW_RX_DONE	R	0x0	See RAW_INT_STAT for a detailed description of RAW_RX_DONE bit. **Value:** • 0x0 (INACTIVE): RAW_RX_DONE interrupt is inactive • 0x1 (ACTIVE): RAW_RX_DONE interrupt is active
6	RAW_TX_ABORT	R	0x0	See RAW_INT_STAT for a detailed description of RAW_TX_ABORT bit. **Value:** • 0x0 (INACTIVE): RAW_TX_ABORT interrupt is inactive • 0x1 (ACTIVE): RAW_TX_ABORT interrupt is active
5	RAW_RD_REQ	R	0x0	See RAW_INT_STAT for a detailed description of RAW_RD_REQ bit. **Value:** • 0x0 (INACTIVE): RAW_RD_REQ interrupt is inactive • 0x1 (ACTIVE): RAW_RD_REQ interrupt is active
4	RAW_TX_EMPTY	R	0x0	See RAW_INT_STAT for a detailed description of RAW_TX_EMPTY bit. **Value:** • 0x0 (INACTIVE): RAW_TX_EMPTY interrupt is inactive • 0x1 (ACTIVE): RAW_TX_EMPTY interrupt is active
3	RAW_TX_OVER	R	0x0	See RAW_INT_STAT for a detailed description of RAW_TX_OVER bit. **Value:** • 0x0 (INACTIVE): RAW_TX_OVER interrupt is inactive

Bits	Field Name	RW	Reset	Description
				• 0x1 (ACTIVE): RAW_TX_OVER interrupt is active
2	RAW_RX_FULL	R	0x0	See RAW_INT_STAT for a detailed description of RAW_RX_FULL bit. **Value:** • 0x0 (INACTIVE): RAW_RX_FULL interrupt is inactive • 0x1 (ACTIVE): RAW_RX_FULL interrupt is active
1	RAW_RX_OVER	R	0x0	See RAW_INT_STAT for a detailed description of RAW_RX_OVER bit. **Value:** • 0x0 (INACTIVE): RAW_RX_OVER interrupt is inactive • 0x1 (ACTIVE): RAW_RX_OVER interrupt is active
0	RAW_RX_UNDER	R	0x0	See RAW_INT_STAT for a detailed description of RAW_RX_UNDER bit. **Value:** • 0x0 (INACTIVE): RX_UNDER interrupt is inactive • 0x1 (ACTIVE): RX_UNDER interrupt is active

10.6.4.13 INT_MASK

- **Name:** I2C Interrupt Mask Register
- **Description:** These bits mask their corresponding interrupt status bits. This register is active low; a value of 0 masks the interrupt, whereas a value of 1 unmasks the interrupt.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x30
- **Reset Value:** 0x000008FF

Table 10-62 I2C Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	MASK_GEN_CALL	RW	0x1	This bit masks the RAW_GEN_CALL interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): GEN_CALL interrupt is masked • 0x1 (DISABLED): GEN_CALL interrupt is unmasked
10	MASK_START_DET	RW	0x0	This bit masks the RAW_START_DET interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): START_DET interrupt is masked • 0x1 (DISABLED): START_DET interrupt is unmasked
9	MASK_STOP_DET	RW	0x0	This bit masks the RAW_STOP_DET interrupt in INT_STAT register.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (ENABLED): STOP_DET interrupt is masked • 0x1 (DISABLED): STOP_DET interrupt is unmasked
8	MASK_ACTIVITY	RW	0x0	This bit masks the RAW_ACTIVITY interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): ACTIVITY interrupt is masked • 0x1 (DISABLED): ACTIVITY interrupt is unmasked
7	MASK_RX_DONE	RW	0x1	This bit masks the RAW_RX_DONE interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): RX_DONE interrupt is masked • 0x1 (DISABLED): RX_DONE interrupt is unmasked
6	MASK_TX_ABORT	RW	0x1	This bit masks the RAW_TX_ABORT interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): TX_ABORT interrupt is masked • 0x1 (DISABLED): TX_ABORT interrupt is unmasked
5	MASK_RD_REQ	RW	0x1	This bit masks the RAW_RD_REQ interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): RD_REQ interrupt is masked • 0x1 (DISABLED): RD_REQ interrupt is unmasked
4	MASK_TX_EMPTY	RW	0x1	This bit masks the RAW_TX_EMPTY interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): TX_EMPTY interrupt is masked • 0x1 (DISABLED): TX_EMPTY interrupt is unmasked
3	MASK_TX_OVER	RW	0x1	This bit masks the RAW_TX_OVER interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): TX_OVER interrupt is masked • 0x1 (DISABLED): TX_OVER interrupt is unmasked
2	MASK_RX_FULL	RW	0x1	This bit masks the RAW_RX_FULL interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): RX_FULL interrupt is masked • 0x1 (DISABLED): RX_FULL interrupt is unmasked
1	MASK_RX_OVER	RW	0x1	This bit masks the RAW_RX_OVER interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): RX_OVER interrupt is masked

Bits	Field Name	RW	Reset	Description
				• 0x1 (DISABLED): RX_OVER interrupt is unmasked
0	MASK_RX_UNDER	RW	0x1	This bit masks the RAW_RX_UNDER interrupt in INT_STAT register. **Value:** • 0x0 (ENABLED): RX_UNDER interrupt is masked • 0x1 (DISABLED): RX_UNDER interrupt is unmasked

10.6.4.14 RAW_INT_STAT

- **Name:** I2C Raw Interrupt Status Register
- **Description:** Unlike the INT_STAT register, these bits are not masked so they always show the true status of the I2C.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 10-63 I2C Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:14	RSVD	R		Reserved bits
13	M_HOLD	R	0x0	Indicate whether the master is holding the bus and TX FIFO is empty. **Value:** • 0x0 (INACTIVE): M_HOLD interrupt is inactive • 0x1 (ACTIVE): M_HOLD interrupt is active
12	RESTART_DET	R	0x0	Indicate whether a RESTART condition has occurred on the I2C interface when I2C is operating in Slave mode and the slave is being addressed. **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 I2C does not generate the RESTART_DET interrupt. **Value:** • 0x0 (INACTIVE): RESTART_DET interrupt is inactive • 0x1 (ACTIVE): RESTART_DET interrupt is active
11	GEN_CALL	R	0x0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the CLR_GEN_CALL register. I2C stores the received data in the RX buffer.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (INACTIVE): GEN_CALL interrupt is inactive • 0x1 (ACTIVE): GEN_CALL interrupt is active
10	START_DET	R	0x0	Indicate whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode. **Value:** • 0x0 (INACTIVE): START_DET interrupt is inactive • 0x1 (ACTIVE): START_DET interrupt is active
9	STOP_DET	R	0x0	Indicate whether a STOP condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode. **In Slave Mode:** • If CTRL[7] = 0x1 (STOP_DET_INT), the STOP_DET interrupt will be issued only if slave is addressed. **Note:** During a general call address, this slave does not issue a STOP_DET interrupt if STOP_DET_INT = 0x1, 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 (S_ADDR). • If CTRL[7] = 0x0 (STOP_DET_INT), the STOP_DET interrupt is issued irrespective of whether it is being addressed. **In Master Mode:** • If CTRL[10] = 0x1 (STOP_DET_IF_MASTER_ACTIVE), the STOP_DET interrupt will be issued only if Master is active. • If CTRL[10] = 0x0 (STOP_DET_INT), the STOP_DET interrupt will be issued irrespective of whether master is active or not. **Value:** • 0x0 (INACTIVE): STOP_DET interrupt is inactive • 0x1 (ACTIVE): STOP_DET interrupt is active
8	ACTIVITY	R	0x0	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: • Disabling the I2C • Reading the CLR_ACTIVITY register • Reading the CLR_INT register • System reset

Bits	Field Name	RW	Reset	Description
				Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus. **Value:** • 0x0 (INACTIVE): ACTIVITY interrupt is inactive • 0x1 (ACTIVE): ACTIVITY interrupt is active
7	RX_DONE	R	0x0	When the I2C 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. **Value:** • 0x0 (INACTIVE): RX_DONE interrupt is inactive • 0x1 (ACTIVE): RX_DONE interrupt is active
6	TX_ABORT	R	0x0	This bit indicates if I2C, 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 TX_ABORT_SRC register indicates the reason why the transmit abort takes place. **Note:** The I2C flushes/resets/empties only the TX_FIFO whenever there is a transmit abort caused by any of the events tracked by the TX_ABORT_SRC register. The TX FIFO remains in this flushed state until the register CLR_TX_ABORT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface. **Value:** • 0x0 (INACTIVE): TX_ABORT interrupt is inactive • 0x1 (ACTIVE): TX_ABORT interrupt is active
5	RD_REQ	R	0x0	This bit is set to 1 when I2C is acting as a slave and another I2C master is attempting to read data from I2C. The I2C 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 DATA_CMD register. This bit is set to 0 just after the processor reads the CLR_RD_REQ register. **Value:** • 0x0 (INACTIVE): RD_REQ interrupt is inactive

Bits	Field Name	RW	Reset	Description
				• 0x1 (ACTIVE): RD_REQ interrupt is active
4	TX_EMPTY	R	0x0	The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the CTRL register. • When TX_EMPTY_CTRL = 0: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_FIFO_THD register. • When TX_EMPTY_CTRL = 1: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_FIFO_THD register and the transmission of the address/data from the internal shift register for the most recently popped command is completed. It is automatically cleared by hardware when the buffer level goes above the threshold. When EN[0] is set to 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 any activity, then with EN[0] = 0, this bit is set to 0. **Value:** • 0x0 (INACTIVE): TX_EMPTY interrupt is inactive • 0x1 (ACTIVE): TX_EMPTY interrupt is active
3	TX_OVER	R	0x0	Set during transmit if the transmit buffer is filled to 8 and the processor attempts to issue another I2C command by writing to the DATA_CMD register. When I2C module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when EN[0] goes to 0, this interrupt is cleared. **Value:** • 0x0 (INACTIVE): TX_OVER interrupt is inactive • 0x1 (ACTIVE): TX_OVER interrupt is active
2	RX_FULL	R	0x0	Set when the receive buffer reaches or goes above the RX_FIFO_THD threshold in the RX_FIFO_THD register. It is automatically cleared by hardware when buffer level goes below the threshold. If I2C module is disabled (EN[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 EN bit 0 is programmed with a 0, regardless of the activity that continues. **Value:** • 0x0 (INACTIVE): RX_FULL interrupt is inactive • 0x1 (ACTIVE): RX_FULL interrupt is active

Bits	Field Name	RW	Reset	Description
1	RX_OVER	R	0x0	Set if the receive buffer is completely filled to 8 and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If I2C module is disabled (EN[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when EN[0] goes to 0, this interrupt is cleared. **Value:** • 0x0 (INACTIVE): RX_OVER interrupt is inactive • 0x1 (ACTIVE): RX_OVER interrupt is active
0	RX_UNDER	R	0x0	Set if the processor attempts to read the receive buffer when it is empty by reading from the DATA_CMD register. If the module is disabled (EN[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when EN[0] goes to 0, this interrupt is cleared. **Value:** • 0x0 (INACTIVE): RX_UNDER interrupt is inactive • 0x1 (ACTIVE): RX_UNDER interrupt is active

10.6.4.15 RX_FIFO_THD

- **Name:**I2C Receive FIFO Threshold Register
- **Description:** I2C Receive FIFO Threshold Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 10-64 I2C Receive FIFO Threshold Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	THD	RW	0x00	Receive FIFO Threshold Level. Control the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in RAW_INT_STAT register). The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.

10.6.4.16 TX_FIFO_THD

- **Name:** I2C Transmit FIFO Threshold Register
- **Description:** I2C Transmit FIFO Threshold Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 10-65 I2C Transmit FIFO Threshold Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	THD	RW	0x00	Transmit FIFO Threshold Level. Control the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in RAW_INT_STAT register). The valid range is 0 – 255, 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 255 sets the threshold for 255 entries.

10.6.4.17 CLR_INT

- **Name:** Clear Combined and Individual Interrupt Register
- **Description:** Clear Combined and Individual Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 10-66 Clear Combined and Individual Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_INT	R	0x0	Read this register to clear the combined interrupt, all individual interrupts, and the TX_ABORT_SRC register. This bit does not clear hardware clearable interrupts but software clearable interrupts.

10.6.4.18 CLR_RX_UNDER

- **Name:** Clear RX_UNDER Interrupt Register

- **Description:** Clear RX_UNDER Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x44
- **Reset Value:**0x00000000

Table 10-67 Clear RX_UNDER Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_RX_UNDER	R	0x0	Read this register to clear the RX_UNDER interrupt (bit 0) of the RAW_INT_STAT register.

10.6.4.19 CLR_RX_OVER

- **Name:** Clear RX_OVER Interrupt Register
- **Description:** Clear RX_OVER Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 10-68 Clear RX_OVER Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_RX_OVER	R	0x0	Read this register to clear the RX_OVER interrupt (bit 1) of the RAW_INT_STAT register.

10.6.4.20 CLR_TX_OVER

- **Name:** Clear TX_OVER Interrupt Register
- **Description:** Clear TX_OVER Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x4C
- **Reset Value:**0x00000000

Table 10-69 Clear TX_OVER Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
0	CLR_TX_OVER	R	0x0	Read this register to clear the TX_OVER interrupt (bit 3) of the RAW_INT_STAT register.

10.6.4.21 CLR_RD_REQ

- **Name:** Clear RD_REQ Interrupt Register
- **Description:** Clear RD_REQ Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 10-70 Clear RD_REQ Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_RD_REQ	R	0x0	Read this register to clear the RD_REQ interrupt (bit 5) of the RAW_INT_STAT register.

10.6.4.22 CLR_TX_ABORT

- **Name:** Clear TX_ABORT Interrupt Register
- **Description:** Clear TX_ABORT Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 10-71 Clear TX_ABORT Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_TX_ABORT	R	0x0	Read this register to clear the TX_ABORT interrupt (bit 6) of the RAW_INT_STAT register, and the TX_ABORT_SRC register.

10.6.4.23 CLR_RX_DONE

- **Name:** Clear RX_DONE Interrupt Register
- **Description:** Clear RX_DONE Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100

- **Offset:** 0x58
- **Reset Value:** 0x00000000

Table 10-72 Clear RX_DONE Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_RX_DONE	R	0x0	Read this register to clear the RX_DONE interrupt (bit 7) of the RAW_INT_STAT register.

10.6.4.24 CLR_ACTIVITY

- **Name:** Clear ACTIVITY Interrupt Register
- **Description:** Clear ACTIVITY Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x5C
- **Reset Value:** 0x00000000

Table 10-73 Clear ACTIVITY Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_ACTIVITY	R	0x0	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 RAW_INT_STAT register.

10.6.4.25 CLR_STOP_DET

- **Name:** Clear STOP_DET Interrupt Register
- **Description:** Clear STOP_DET Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 10-74 Clear STOP_DET Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_STOP_DET	R	0x0	Read this register to clear the STOP_DET interrupt (bit 9) of the RAW_INT_STAT register.

10.6.4.26 CLR_START_DET

- **Name:** Clear START_DET Interrupt Register
- **Description:** Clear START_DET Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x64
- **Reset Value:** 0x00000000

Table 10-75 Clear START_DET Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_START_DET	R	0x0	Read this register to clear the START_DET interrupt (bit 10) of the RAW_INT_STAT register.

10.6.4.27 CLR_GEN_CALL

- **Name:** Clear GEN_CALL Interrupt Register
- **Description:** Clear GEN_CALL Interrupt Register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x68
- **Reset Value:** 0x00000000

Table 10-76 Clear GEN_CALL Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_GEN_CALL	R	0x0	Read this register to clear the GEN_CALL interrupt (bit 11) of RAW_INT_STAT register.

10.6.4.28 EN

- **Name:** I2C ENABLE Register

- **Description:** I2C enable register
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x6C
- **Reset Value:** 0x00000000

Table 10-77 I2C ENABLE Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	TX_CMD_BLOCK	RW	0x0	In Master mode: **Value:** • 0x0 (NOT_BLOCKED): TX Command execution not blocked • 0x1 (BLOCKED): TX Command execution blocked
1	ABORT	RW	0x0	When set, the controller initiates the transfer abort. The software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when EN 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. **Value:** • 0x0 (DISABLED): ABORT operation not in progress • 0x1 (ENABLED): ABORT operation in progress
0	ACTIVITY	RW	0x0	Control whether the I2C is enabled. Software can disable I2C while it is active. However, it is important that care should be taken to ensure that I2C is disabled properly. When I2C is disabled, the following occurs: The TX FIFO and RX FIFO get flushed. Status bits in the INT_STAT register are still active until I2C goes into IDLE state. If the I2C module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is completed. If the module is receiving, the I2C stops the current transfer at the end of the current byte and does not acknowledge the transfer. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (DISABLED): I2C is disabled • 0x1 (ENABLED): I2C is enabled

10.6.4.29 STAT

- **Name:** I2C STATUS Register
- **Description:** This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register requests an interrupt.

  When the I2C is disabled by writing 0 in bit 0 of the EN register:

  ◦ Bits 1 and 2 are set to 1

  ◦ Bits 3 and 6 are set to 0

  When the master or slave state machine goes to idle and EN[0]=0:

  ◦ Bits 5 and 6 are set to 0
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x70
- **Reset Value:** 0x00000006

Table 10-78 I2C STATUS Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6	S_ACTIVITY	R	0x0	Slave Activity Status. When the Slave is not in the IDLE state, this bit is set. **Value:** • 0x0 (INACTIVE): Slave is idle • 0x1 (ACTIVE): Slave is not idle
5	M_ACTIVITY	R	0x0	Master Activity Status. When the Master is not in the IDLE state, this bit is set. **Note:** STAT[0], ACTIVITY bit, is the OR of S_ACTIVITY and M_ACTIVITY Y bits. **Value:** • 0x0 (INACTIVE): Master is idle • 0x1 (ACTIVE): Master not idle
4	RX_FIFO_CF	R	0x0	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. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (NOT_FULL): RX FIFO is not full • 0x1 (FULL): RX FIFO is full
3	RX_FIFO_NE	R	0x0	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. **Value:** • 0x0 (EMPTY): RX FIFO is empty • 0x1 (NOT_EMPTY): RX FIFO not empty
2	TX_FIFO_CE	R	0x1	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. **Value:** • 0x0 (NOT_EMPTY): TX FIFO not empty • 0x1 (EMPTY): TX FIFO is empty
1	TX_FIFO_NF	R	0x1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. **Value:** • 0x0 (FULL): TX FIFO is full • 0x1 (NOT_FULL): TX FIFO not full
0	ACTIVITY	R	0x0	I2C Activity Status **Value:** • 0x0 (INACTIVE): I2C is idle • 0x1 (ACTIVE): I2C is active

10.6.4.30 TX_FIFO_LEVEL

- **Name:** I2C Transmit FIFO Level Register
- **Description:** This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:
  - The I2C is disabled.
  - If TX_ABORT bit is set in the RAW_INT_STAT register, the slave bulk transmit mode is aborted.
  - The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x74

- **Reset Value:** 0x00000000

Table 10-79 I2C Transmit FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	TLEVEL	R	0x0	Transmit FIFO Level. It contains the number of valid data entries in the transmit FIFO.

10.6.4.31 RX_FIFO_LEVEL

- **Name:** I2C Receive FIFO Level Register
- **Description:** This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:
  - The I2C is disabled.
  - There is a transmit abort caused by any of the events tracked in TX_ABORT_SRC.
  - The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x78
- **Reset Value:** 0x00000000

Table 10-80 I2C Receive FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	LEVEL	R	0x0	Receive FIFO Level. It contains the number of valid data entries in the receive FIFO.

10.6.4.32 SDA_HOLD

- **Name:** I2C SDA Hold Time Length Register
- **Description:** I2C SDA Hold Time Length Register
  - The bits [15:0] of this register are used to control the hold time of SDA during transmit in both slave and master mode (after SCL goes from HIGH to LOW).
  - The bits [23:16] of this register are used to extend the SDA transition (if any) whenever SCL is HIGH in the receiver in either master or slave mode.
  - Writes to this register succeed only when EN[0]=0.

- The values in this register are in units of clock period. The value programmed in TX_HOLD must be greater than the minimum hold time in each mode one cycle in master mode, seven cycles in slave mode for the value to be implemented.
- The programmed SDA hold time during transmit (TX_HOLD) cannot exceed at any time the duration of the low part of SCL. Therefore the programmed value cannot be larger than N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the SCL period measured in clock cycles.

- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x7C
- **Reset Value:** 0x00000001

Table 10-81 I2C SDA Hold Time Length Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits
23:16	RX_HOLD	RW	0x0	Set the required SDA hold time in units of clock period, when I2C acts as a receiver.
15:0	TX_HOLD	RW	0x1	Set the required SDA hold time in units of clock period, when I2C acts as a transmitter.

10.6.4.33 TX_ABORT_SRC

- **Name:** I2C Transmit Abort Source Register
- **Description:** I2C Transmit Abort Source Register
  - This register has 32 bits that indicates the source of the TX_ABORT bit. Except for Bit 9, this register is cleared whenever the CLR_TX_ABORT register or the CLR_INT register is read. To clear Bit 9, the source of the ABORT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (CTRL[5] = 1), the SPECIAL bit must be cleared (TARGET_ADDR[11]), or the TX_CTRL bit must be cleared (TARGET_ADDR[10]).
  - Once the source of the ABORT_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 ABORT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x80
- **Reset Value:** 0x00000000

Table 10-82 I2C Transmit Abort Source Register

Bits	Field Name	RW	Reset	Description
31:23	TX_FLUSH_CNT	R	0x0	This field indicates the number of TX FIFO Data Commands which are flushed due to TX_ABORT interrupt. It is cleared whenever I2C is disabled.

Bits	Field Name	RW	Reset	Description
				Role of I2C: Master-Transmitter or Slave-Transmitter
22:17	RSVD	R		Reserved bits
16	ABORT_USER_ABORT	R	0x0	This is a master-mode-only bit. Master has detected the transfer abort (EN[1]) **Role of I2C:** Master-Transmitter **Value:** • 0x0 (ABORT_USER_ABORT_VOID): Transfer abort detected by master-scenario not present • 0x1 (ABORT_USER_ABORT_GENERATED): Transfer abort detected by master
15	ABORT_SLVRD_INTX	R	0x0	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 DATA_CMD register. **Role of I2C:** Slave-Transmitter **Value:** • 0x0 (ABORT_SLVRD_INTX_VOID): Slave trying to transmit to remote master in read mode- scenario not present • 0x1 (ABORT_SLVRD_INTX_GENERATED): Slave trying to transmit to remote master in read mode
14	ABORT_S_ARBLOST	R	0x0	This field indicates that a Slave has lost the bus while transmitting data to a remote master. TX_ABORT_SRC[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 I2C no longer owns the bus. **Role of I2C:** Slave-Transmitter **Value:** • 0x0 (ABORT_S_ARBLOST_VOID): Slave lost arbitration to remote master-scenario not present • 0x1 (ABORT_S_ARBLOST_GENERATED): Slave lost arbitration to remote master
13	ABORT_SLVFLUSH_TXFIFO	R	0x0	This field specifies that the Slave has received a read command and some data exists in the TX FIFO, so the slave issues a TX_ABORT interrupt to flush old data in TX FIFO. **Role of I2C:** Slave-Transmitter

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (ABORT_SLVFLUSH_TXFIFO_VOID): Slave flushes existing data in TX-FIFO upon getting read command- scenario not present • 0x1 (ABORT_SLVFLUSH_TXFIFO_GENERATED): Slave flushes existing data in TX-FIFO upon getting read command
12	ABORT_LOST	R	0x0	This field specifies that the Master has lost arbitration, or if TX_ABORT_SRC[14] is also set, then the slave transmitter has lost arbitration. **Role of I2C:** Master-Transmitter or Slave-Transmitter **Value:** • 0x0 (ABORT_LOST_VOID): Master or Slave-Transmitter lost arbitration- scenario not present • 0x1 (ABORT_LOST_GENERATED): Master or Slave-Transmitter lost arbitration
11	ABORT_MASTER_DIS	R	0x0	This field indicates that the User tries to initiate a Master operation with the Master mode disabled. **Role of I2C:** Master-Transmitter or Master-Receiver **Value:** • 0x0 (ABORT_MASTER_DIS_VOID): User initiating master operation when Master disabled- scenario not present • 0x1 (ABORT_MASTER_DIS_GENERATED): User initiating master operation when Master disabled
10	ABORT_10B_RD_NORSTR	R	0x0	This field indicates that the restart is disabled (RESTART_EN bit (CTRL[5]) =0) and the master sends a read command in 10-bit addressing mode. **Role of I2C:** Master-Receiver **Value:** • 0x0 (ABRT_10B_RD_VOID): Master not trying to read in 10-bit addressing mode when RESTART disabled • 0x1 (ABRT_10B_RD_GENERATED): Master trying to read in 10-bit addressing mode when RESTART disabled
9	ABORT_SBYTE_NORSTRT	R	0x0	To clear Bit 9, the source of the ABORT_SBYTE_NORSTRT must be fixed first; restart must be enabled (CTRL[5]=1), the SPECIAL bit must be cleared (TARGET_ADDR[11]), or the TX_CTRL bit must be cleared (TARGET_ADDR[10]). Once the source of the ABORT_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 ABORT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets

Bits	Field Name	RW	Reset	Description
				reasserted. When this field is set to 1, the restart is disabled (RESTART_EN bit (CTRL[5]) =0) and the user is trying to send a START BYTE. **Role of I2C:** Master **Value:** • 0x0 (ABORT_SBYTE_NORSTRT_VOID): User trying to send START BYTE when RESTART disabled- scenario not present • 0x1 (ABORT_SBYTE_NORSTRT_GENERATED): User trying to send START BYTE when RESTART disabled
8	ABORT_HS_NORSTRT	R	0x0	This field indicates that the restart is disabled (RESTART_EN bit (CTRL[5]) =0) and the user is trying to use the master to transfer data in High Speed mode. **Role of I2C:** Master-Transmitter or Master-Receiver **Value:** • 0x0 (ABORT_HS_NORSTRT_VOID): User trying to switch Master to HS mode when RESTART disabled- scenario not present • 0x1 (ABORT_HS_NORSTRT_GENERATED): User trying to switch Master to HS mode when RESTART disabled
7	ABORT_SBYTE_ACKDET	R	0x0	This field indicates that the Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). **Role of I2C:** Master **Value:** • 0x0 (ABORT_SBYTE_ACKDET_VOID): ACK detected for START byte- scenario not present • 0x1 (ABORT_SBYTE_ACKDET_GENERATED): ACK detected for START byte
6	ABORT_HS_ACKDET	R	0x0	This field indicates that the Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). **Role of I2C:** Master **Value:** • 0x0 (ABRT_HS_ACK_VOID): HS Master code ACKed in HS Mode- scenario not present • 0x1 (ABRT_HS_ACK_GENERATED): HS Master code ACKed in HS Mode
5	ABORT_GCALL_RD	R	0x0	This field indicates that I2C in the master mode has sent a General Call but the user programmed the byte following the General Call to be a read from the bus (DATA_CMD[9] is set to 1). **Role of I2C:** Master-Transmitter **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (ABORT_GCALL_RD_VOID): GCALL is followed by read from bus-scenario not present • 0x1 (ABORT_GCALL_RD_GENERATED): GCALL is followed by read from bus
4	ABORT_GCALL_NOACK	R	0x0	This field indicates that I2C in master mode has sent a General Call and no slave on the bus acknowledged the General Call. **Role of I2C:** Master-Transmitter **Value:** • 0x0 (ABORT_GCALL_NOACK_VOID): GCALL not ACKed by any slave-scenario not present • 0x1 (ABORT_GCALL_NOACK_GENERATED): GCALL not ACKed by any slave
3	ABORT_TX_NOACK	R	0x0	This field indicates the master-mode only bit. When the master receives an acknowledgement for the address, but when it sends data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). **Role of I2C:**Master-Transmitter **Value:** • 0x0 (ABORT_TX_NOACK_VOID): Transmitted data non-ACKed by addressed slave-scenario not present • 0x1 (ABORT_TX_NOACK_GENERATED): Transmitted data not ACKed by addressed slave
2	ABORT_10B2_NOACK	R	0x0	This field indicates that the Master is in 10-bit address mode and that the second address byte of the 10-bit address was not acknowledged by any slave. **Role of I2C:** Master-Transmitter or Master-Receiver **Value:** • 0x0 (INACTIVE): This abort is not generated • 0x1 (ACTIVE): Byte 2 of 10-bit Address not ACKed by any slave
1	ABORT_10B1_NOACK	R	0x0	This field indicates that the Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. **Role of I2C:** Master-Transmitter or Master-Receiver **Value:** • 0x0 (INACTIVE): This abort is not generated • 0x1 (ACTIVE): Byte 1 of 10-bit Address not ACKed by any slave

Bits	Field Name	RW	Reset	Description
0	ABORT_7B_NOACK	R	0x0	This field indicates that the Master is in 7-bit address mode and the address sent was not acknowledged by any slave. **Role of I2C:** Master-Transmitter or Master-Receiver **Value:** • 0x0 (INACTIVE): This abort is not generated • 0x1 (ACTIVE): This abort is generated because of NACK for 7-bit address

10.6.4.34 DMA_CTRL

- **Name:** DMA Control Register
- **Description:** This register is only valid when I2C is configured with a set of DMA Controller interface signals. When I2C is not configured for DMA operation, this register does not exist and writing to the register's address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmitting and receiving. This can be programmed regardless of the state of EN.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x88
- **Reset Value:** 0x00000000

Table 10-83 DMA Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	TX_EN	RW	0x0	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. **Value:** • 0x0 (DISABLED): Transmit FIFO DMA channel disabled • 0x1 (ENABLED): Transmit FIFO DMA channel enabled
0	RX_EN	RW	0x0	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. **Value:** • 0x0 (DISABLED): Receive FIFO DMA channel disabled • 0x1 (ENABLED): Receive FIFO DMA channel enabled

10.6.4.35 DMA_TX_LEVEL

- **Name:** DMA Transmit Data Level Register

- **Description:** This register is only valid when the I2C is configured with a set of DMA interface signals. When I2C is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x8C
- **Reset Value:** 0x00000000

Table 10-84 DMA Transmit Data Level Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	LEVEL	RW	0x0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic.

10.6.4.36 DMA_RX_LEVEL

- **Name:** DMA Receive Data Level Register
- **Description:** This register is only valid when I2C is configured with a set of DMA interface signals (HAS_DMA = 1). When I2C is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x90
- **Reset Value:** 0x00000000

Table 10-85 DMA Receive Data Level Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	LEVEL	RW	0x0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic.

10.6.4.37 SDA_SETUP

- **Name:** I2C SDA Setup Register
- **Description:** This register controls the amount of time delay (in terms of number of clock periods) introduced in the rising edge of SCL - relative to SDA changing - when I2C services a read request in a slave-transmitter operation. The relevant I2C requirement is tSU_STO as detailed in the Figure 10-11. This register must be programmed with a value equal to or greater than 2.

  Writes to this register succeed only when EN[0] = 0.

Note:

The length of setup time is calculated using [(SDA_SETUP - 1) * (clk_period)], so if the user requires 10 clock periods of setup time, they should program a value of 11. The SDA_SETUP register is only used by the I2C when operating as a slave transmitter.

- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x94
- **Reset Value:** 0x00000064

Table 10-86 I2C SDA Setup Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	SETUP	RW	0x64	SDA Setup. It is recommended that if the required delay is 1000 ns, then for a clock frequency of 10 MHz, SETUP should be programmed to a value of 11. SETUP must be programmed with a minimum value of 2.

10.6.4.38 ACK_GEN_CALL

- **Name:** I2C ACK General Call Register
- **Description:** The register controls whether I2C responds with an ACK or NACK when it receives an I2C General Call address. This register is applicable only when the I2C is in slave mode.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x98
- **Reset Value:** 0x00000001

Table 10-87 I2C ACK General Call Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	ACK_GEN_CALL	RW	0x1	ACK General Call. When set to 1, I2C responds with an ACK (by asserting data_oe) when it receives a General Call. Otherwise, I2C responds with a NACK (by negating data_oe). **Value:** • 0x0 (DISABLED): Generate NACK for General Call • 0x1 (ENABLED): Generate ACK for a General Call

10.6.4.39 EN_STAT

- **Name:** I2C Enable Status Register

- **Description:** The register is used to report the I2C hardware status when the EN[0] register is set from 1 to 0; that is, when I2C is disabled.
  - If EN[0] has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to '1'.
  - If EN[0] has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Note:

When EN[0] has been set to 0, a delay occurs for bit 0 to be read as 0 because disabling the I2C depends on I2C bus activities.

- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0x9C
- **Reset Value:** 0x00000000

Table 10-88 I2C Enable Status Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	S_RX_DATA_LOST	R	0x0	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 bit 0 of EN from 1 to 0. When read as 1, I2C 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 I2C has a chance to NACK transfer, and EN[0] has been set to 0, then this bit is also set to 1. • When read as 0, I2C is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. • The CPU can safely read this bit when EN (bit 0) is read as 0. **Value:** • 0x0 (INACTIVE): Slave RX Data is not lost • 0x1 (ACTIVE): Slave RX Data is lost
1	S_DIS_BUSY	R	0x0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting bit 0 of the ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the ENABLE register while: • I2C is receiving the address byte of the Slave-Transmitter operation from a remote master.

Bits	Field Name	RW	Reset	Description
				• Address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, I2C 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 (S_ADDR register) or if the transfer is completed before EN is set to 0 but has not taken effect. **Note:** • If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK a transfer, and EN[0] has been set to 0, then this bit will also be set to 1. • When read as 0, I2C is deemed to have been disabled when there is master activity, or when the I2C bus is idle. • The CPU can safely read this bit when EN (bit 0) is read as 0. **Value:** • 0x0 (INACTIVE): Slave is disabled when it is idle • 0x1 (ACTIVE): Slave is disabled when it is active
0	EN	R	0x0	EN Status. This bit always reflects the value driven on the output port EN. When read as 1, I2C is deemed to be in an enabled state. When read as 0, I2C is deemed completely inactive. **Note:** The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read S_RX_DATA_LOST (bit 2) and S_DIS_BUSY (bit 1). **Value:** • 0x0 (DISABLED): I2C disabled • 0x1 (ENABLED): I2C enabled

10.6.4.40 FS_SPKLEN

- **Name:** I2C SS, FS or FM+ spike suppression limit
- **Description:** This register is used to store the duration, measured in clock cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in SS, FS or FM+ mode. The relevant I2C requirement is tSP as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0xA0
- **Reset Value:** 0x00000005

Table 10-89 I2C SS, FS or FM+ spike suppression limit

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	FS_SPKLEN	RW	0x5	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in clock 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 EN[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.

10.6.4.41 HS_SPKLEN

- **Name:** I2C HS spike suppression limit register
- **Description:** This register is used to store the duration, measured in clock cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in HS mode. The relevant I2C requirement is tSP as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1 and is implemented only if the component is configured to support HS mode.
- **Base Address:** 0xA000C300 + x*0x100
- **Offset:** 0xA4
- **Reset Value:** 0x00000001

Table 10-90 I2C HS spike suppression limit register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	HS_SPKLEN	RW	0x1	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in clock 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 EN[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.

10.6.5 Electrical Specifications

Table 10-91 Electrical Specifications

Symbol	Parameter	Standard Mode(SS)		Fast Mode (FS)		Fast-plus Mode (FM+)		High-speed Mode (HS)		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
f_{SCL}	SCL clock frequency	0	100	0	400	0	1000	0	2000	kHz
t_{LOW}	SCL clock LOW time	4.7	-	1.3	-	0.5	-	0.16	-	µs
t_{HIGH}	SCL clock HIGH time	4.0	-	0.6	-	0.26	-	0.06	-	µs
t_{RSDA}	SDA rise time	-	1000	20	300	-	120	10	80	ns
t_{FSDA}	SDA fall time	-	300		300		120	10	80	ns
t_{RSCL}	SCL rise time	-	1000	20	300	-	120	10	40	ns
t_{FSCL}	SCL fall time	-	300		300		120	10	40	ns
t_{HD_STA}	START hold time	4.0	-	0.6	-	0.26	-	0.16	-	µs
t_{SU_DAT}	SDA setup time	250	-	100	-	50	-	10	-	ns
t_{HD_DAT}	SDA hold time	0	-	-	-	-	-	0	70	µs
t_{SU_STO}	STOP setup time	4.0	-	0.6	-	0.26	-	0.16	-	µs
t_{BUF}	BUS FREE time from STOP to START	4.7	-	1.3	-	0.5	-	-	-	µs

Note:

The bus load capacitance C_b = 100 pF.

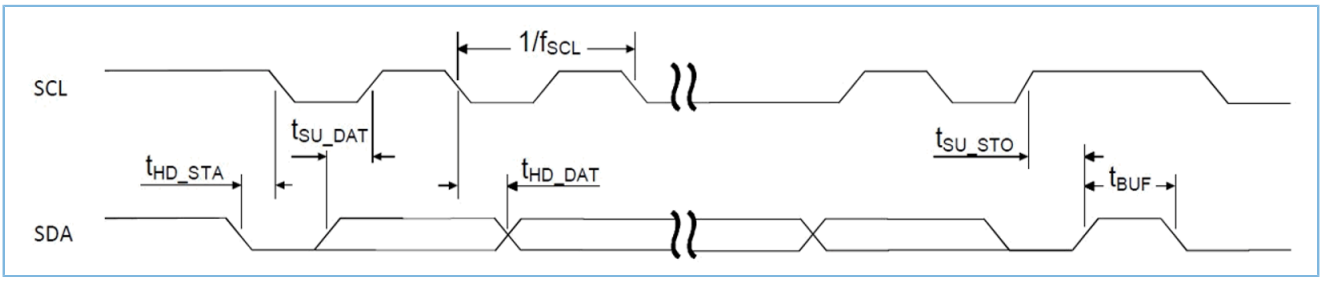

Figure 10-11 I2C timing diagram

10.7 UART

10.7.1 Introduction

GR551x UART follows the industry-standard 16550 Universal Asynchronous Receiver and Transmitter and the asynchronous part of the RS-232 Serial Protocol. UART is used for serial communication with peripherals, modems (data carrier equipment, DCE), and the like. Data is written from a master (CPU) over the APB bus to the UART, and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

GR551x has two UART instances: UART0 and UART1.

10.7.2 Main Features

Both UART instances support the following:

- Full-duplex asynchronous communications
- Programmable data size (5 – 8 bits)
- Programmable parity bit (none, odd, even, 0, and 1)
- Programmable stop bits (1, 1.5, or 2 bits)
- Hardware flow control
- 128-byte buffer
- DMA (only for UART0)
- Up to 4 M baud rates

10.7.3 Functional Description

UART functional block diagram is illustrated in Figure 10-12.

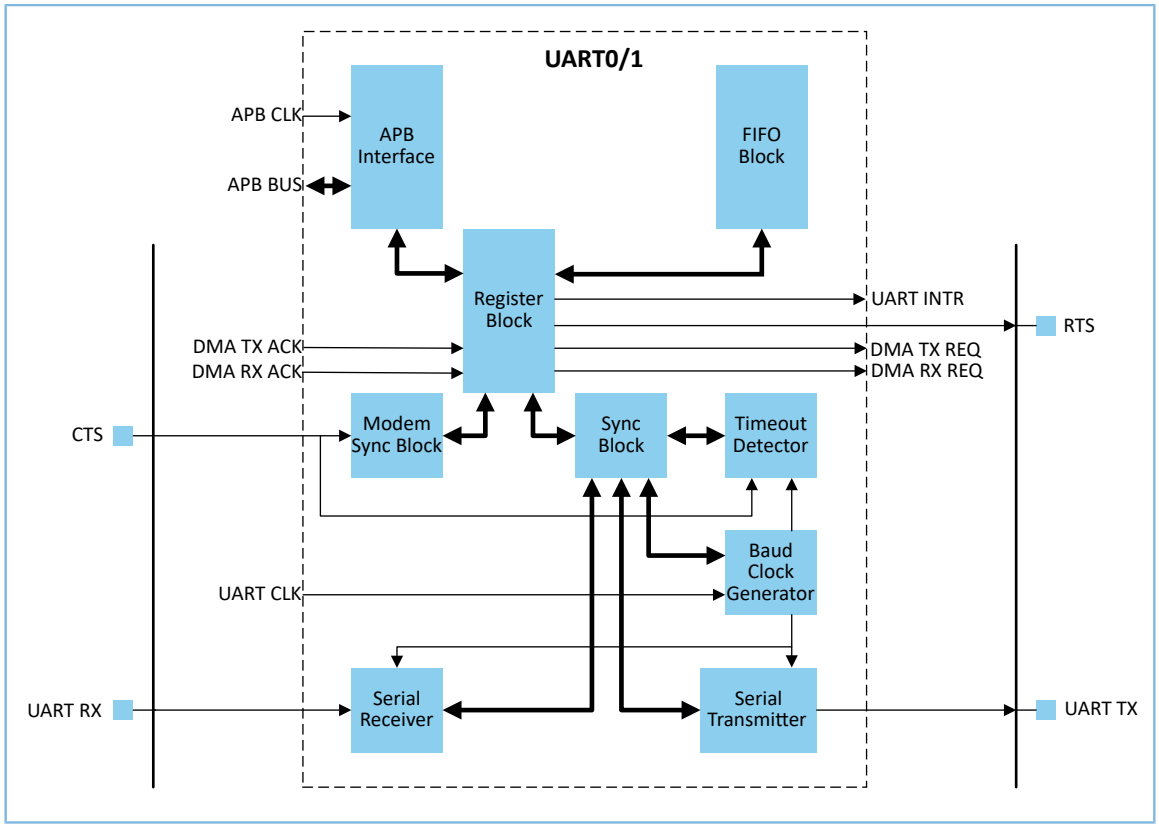

Figure 10-12 UART Functional Block Diagram

10.7.3.1 UART (RS232) Serial Protocol

Because the serial communication between the UART and a 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 10-13.

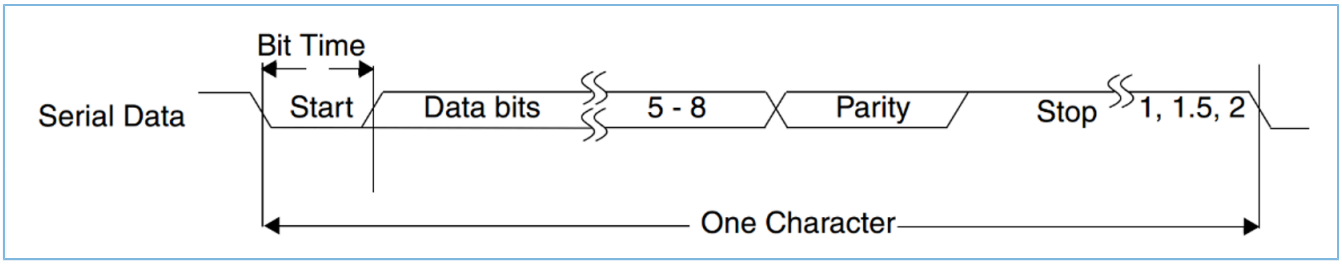

Figure 10-13 Serial Data Format

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

The UART Line Control Register (**LINE_CTRL** register) 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 are transmitted for exactly the same time duration except when **1.5** stop bits are used. This duration is referred to as a Bit Period or Bit Time; one Bit Time equals sixteen baud clocks.

To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time when the start bit has been detected. The exact number of baud clocks is known for which each bit was transmitted, so calculating the midpoint for sampling is not difficult; every sixteen baud clocks sample from the Start bit to the midpoint.

Together with serial input debouncing, this sampling helps to avoid the detection of false start bits. Short glitches are filtered out by debouncing, and no transition is detected on the line. If a glitch is wide enough to avoid filtering by debouncing, a falling edge is detected. However, a start bit is detected only if the line is again sampled low after half a bit time has elapsed.

Figure 10-14 shows the sampling points of the first two bits in a serial character.

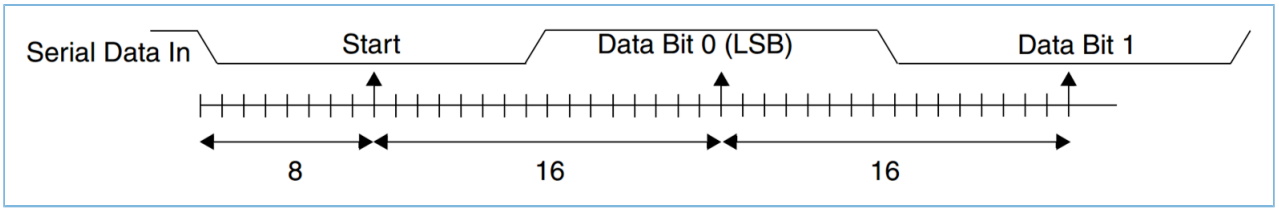

Figure 10-14 Receiver Serial Data Sample Points

As part of the 16550 standard, an optional baud clock reference output signal (**baudout_n**) provides timing information to receive 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 (**DIV_LATCH_HIGH** and **DIV_LATCH_LOW**).

Serial data baud rate can be calculated as follows: `baud rate = (system clock frequency) / (16 * divisor)`. Divisor is composed of integer part (**DIV_LATCH_HIGH/DIV_LATCH_LOW** register) and fractional part (4-bit **DIV_LATCH_FRACTION** register).

10.7.3.2 Interrupts

Assertion of the UART interrupt output signal (INTR)—a positive-level interrupt—occurs whenever one of the several prioritized interrupt types are enabled and active. When an interrupt occurs, the master accesses the **INT_ID** register.

The following interrupt types can be enabled with the **INT_EN** register:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below the threshold (in Programmable TX_HDG_EMPTY interrupt mode)
- Modem Status
- Busy Detect Indication

These interrupt types are explained in detail in Table 10-92 .

Table 10-92 Interrupt Control Functions

Interrupt ID				Interrupt Set and Reset Functions			
Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	–	None	None	–
0	1	1	0	Highest	Receiver line status	Overrun/parity/ framing errors, break interrupt, or address received interrupt	For overrun/parity/framing/break interrupt reset control, the behavior is as follows: • If LSR_STATUS_CLEAR = 0 (RX_BUF Read or LINE_STAT Read), then the status is cleared on: Reading the LINE_STAT register or in addition to an LINE_STAT read, the Receiver line status is also cleared when RX_FIFO is read. • If LSR_STATUS_CLEAR=1 (LINE_STAT Read), the status is cleared only on: Reading the LINE_STAT register. • For address received interrupt, the status is cleared on: Reading the LINE_STAT register.
0	1	0	0	Second	Received data available	Receiver data available (non- FIFO mode or FIFOs disabled) or RX FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO add above the trigger level (FIFO mode and FIFOs enabled)
1	1	0	0	Second	Character timeout indication	No characters in or out of the RX FIFO during the last 4 character times and there is at least 1 character in RX FIFO during this time	Reading the receiver buffer register
0	0	1	0	Third	Transmit holding register empty	Transmitter holding register empty (Prog. TX_HDG_EMPTY Mode disabled) or TX FIFO at or below threshold (Prog. TX_HDG_EMPTY Mode enabled)	Reading the INT_ID register (if source of interrupt); or, writing into TX_HDG (FIFOs or TX_HDG_EMPTY Mode not selected or disabled) or TX FIFO below threshold (FIFOs and TX_HDG_EMPTY Mode selected and enabled).
0	0	0	0	Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode	Reading the Modem status register

Interrupt ID				Interrupt Set and Reset Functions			
						is enabled, a change in CTS (DCTS set) does not cause an interrupt.	

10.7.3.3 Programmable TX Holding Empty Interrupt

The UART can be configured for a Programmable TX_HDG_EMPTY Interrupt mode in order to increase system performance; if FIFOs are not implemented, then this mode cannot be selected.

- When Programmable TX_HDG_EMPTY Interrupt mode is not selected, none of the logic is implemented and the mode cannot be enabled, reducing the overall gate counts.
- When Programmable TX_HDG_EMPTY Interrupt mode is selected, it can be enabled using the Interrupt Enable Register (**INT_EN[7]**).

When FIFOs and TX_HDG_EMPTY mode are implemented and enabled, the TX_HDG_EMPTY Interrupts and dma_tx_req_n are active at, and below, a programmed transmitter FIFO empty threshold level, as opposed to empty, as shown in the flowchart in Figure 10-15.

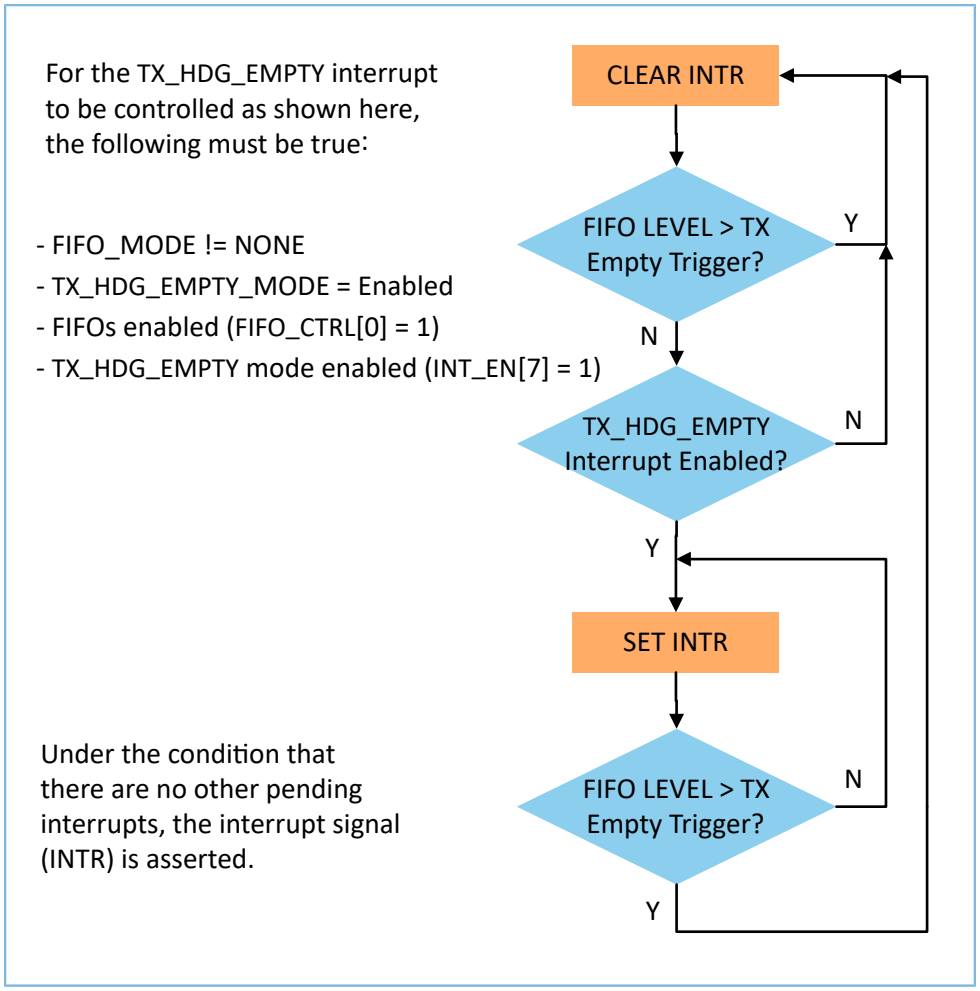

Figure 10-15 Interrupt Generation for Programmable TX_HDG_EMPTY Interrupt Mode

The threshold level is programmed into **FIFO_CTRL[5:4]**. Available empty thresholds are: **empty**, **2**, **¼**, **½**. Selection of the best threshold value depends on the system's ability to start a new transmission sequence in a timely manner. However, one of these thresholds should be optimal for increasing system performance by preventing the transmitter FIFO from running empty. For threshold setting details, refer to "Section 10.7.4.6 FIFO_CTRL".

In addition to the interrupt change, the Line Status Register (**LINE_STAT[5]**) also switches from indicating that the transmitter FIFO is empty to the FIFO being full. This allows software to fill the FIFO for each transmit sequence by polling **LINE_STAT[5]** before writing another character. The flow then allows the transmitter FIFO to be filled whenever an interrupt occurs and there is data to transmit, rather than waiting until the FIFO is completely empty. Waiting until the FIFO is empty causes a reduction in performance whenever the system is too busy to respond immediately. Further system efficiency is achieved when this mode is enabled in combination with Auto Flow Control.

Even if everything else is selected and enabled, if the FIFOs are disabled using the **FIFO_CTRL[0]** bit, the Programmable TX_HDG_EMPTY Interrupt mode is also disabled. When not selected or disabled, TX_HDG_EMPTY interrupts and the **LINE_STAT[5]** bit function normally, signifying an empty TX_HDG or FIFO. Figure 10-16 illustrates the flowchart of TX_HDG_EMPTY interrupt generation when not in programmable TX_HDG_EMPTY interrupt mode.

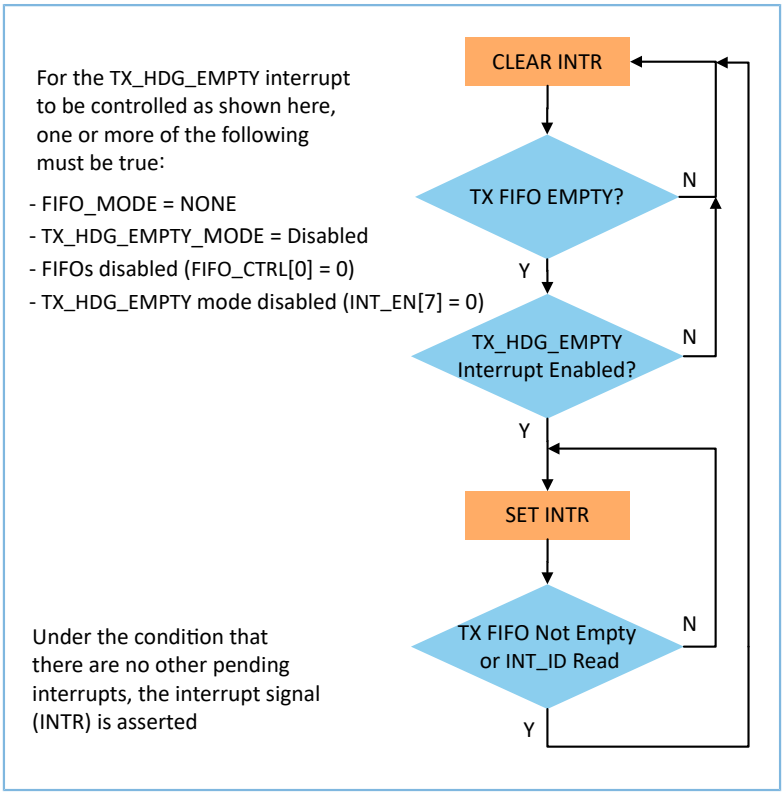

Figure 10-16 Interrupt generation when not in Programmable TX_HDG_EMPTY Interrupt Mode

10.7.3.4 Auto Flow Control

The UART have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode; if FIFOs are not implemented, this mode cannot be selected. When Auto Flow Control is not selected, none of the corresponding logic is implemented and the mode cannot be enabled, reducing overall gate counts. When Auto Flow Control mode is selected, it can be enabled with the Modem Control Register.

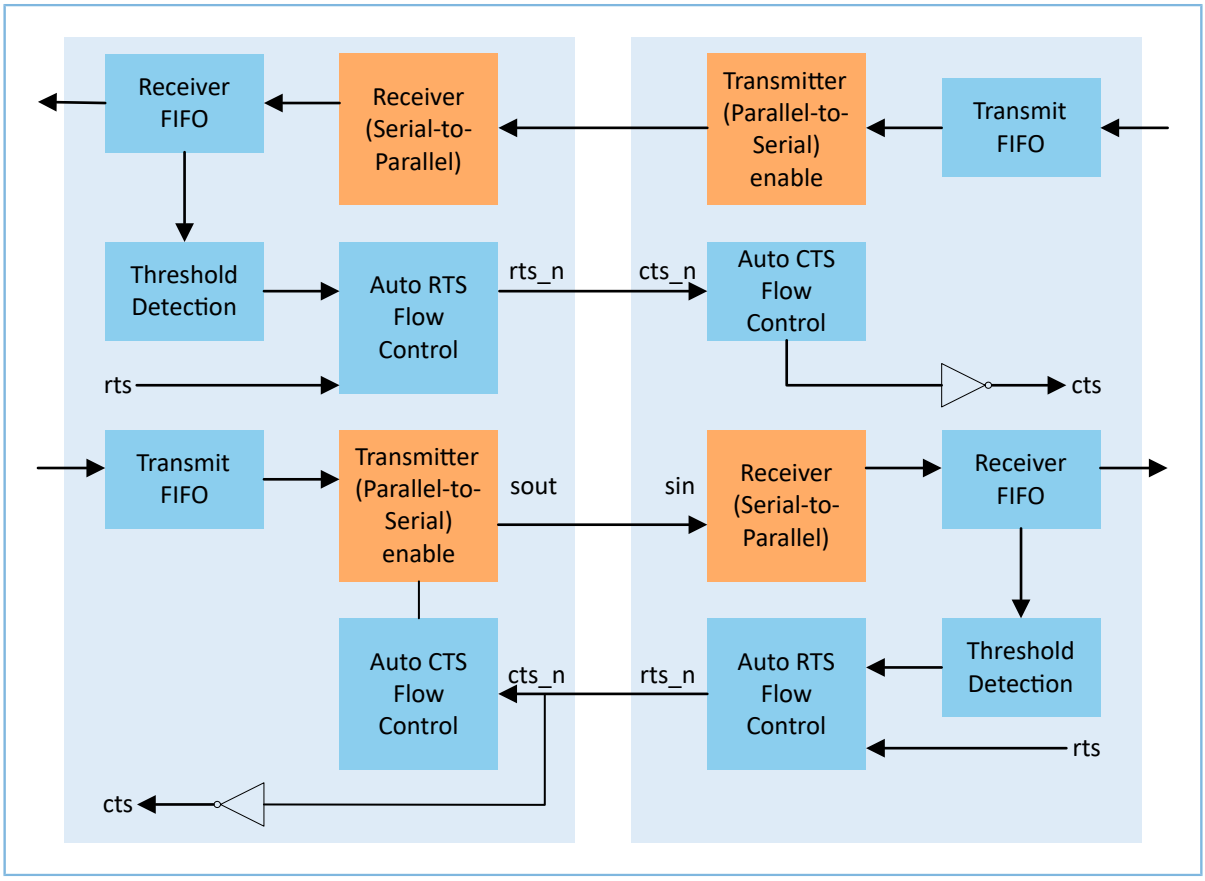

Figure 10-17 Block diagram of the Auto Flow Control functionality

When Auto RTS is enabled, the rts_n output is forced inactive (high) when the receiver FIFO level reaches the threshold. When rts_n is connected to the cts_n input of another UART device, the other UART stops sending serial data until the receiver FIFO has available space; that is, until it is completely empty. The selectable threshold values of Receiver FIFO are:

- 1
- 1/4
- 1/2
- 2 less than full

Since one additional character can be transmitted to the UART, after rts_n has become inactive due to data already having entered the transmitter block in the other UART—setting the threshold to "2 less than full" allows maximum use of the FIFO with a safety zone of one character. Once the receiver FIFO becomes completely empty by reading the Receiver Buffer Register, rts_n again becomes active (low), signaling the other UART to continue sending data.

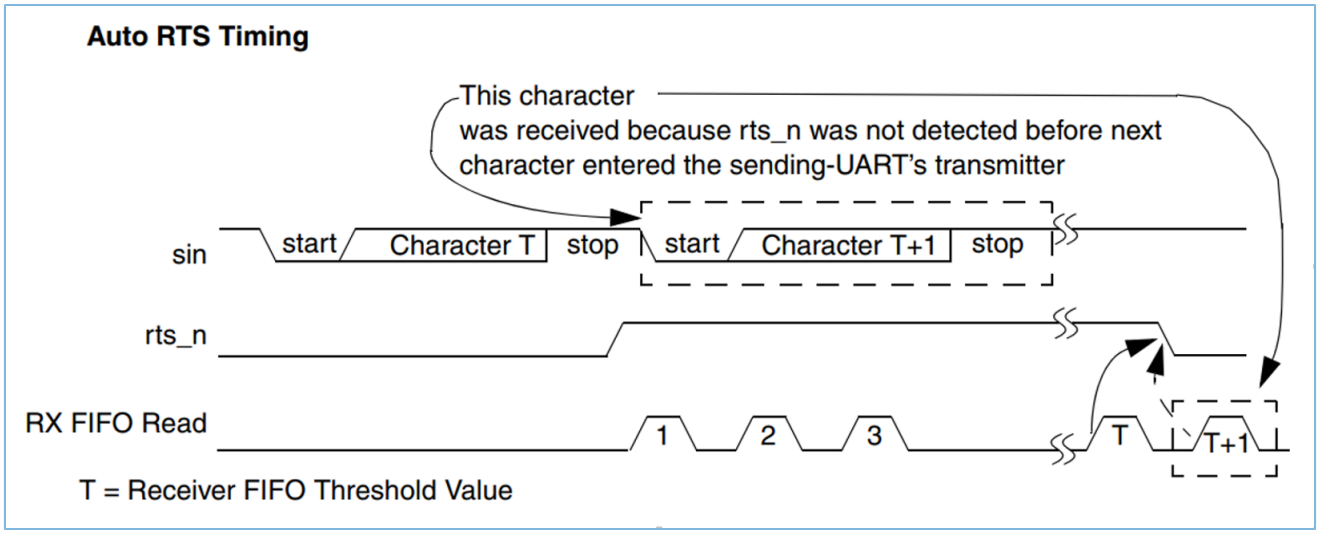

Figure 10-18 Auto RTS timing

When Auto CTS is enabled (active), the UART transmitter becomes disabled whenever the cts_n input becomes inactive (high); this prevents overflowing the FIFO of the receiving UART. If the cts_n input is not inactivated before the middle of the last stop bit, another character is transmitted before the transmitter is disabled. While the transmitter is disabled, the transmitter FIFO can still be written to, and even overflowed.

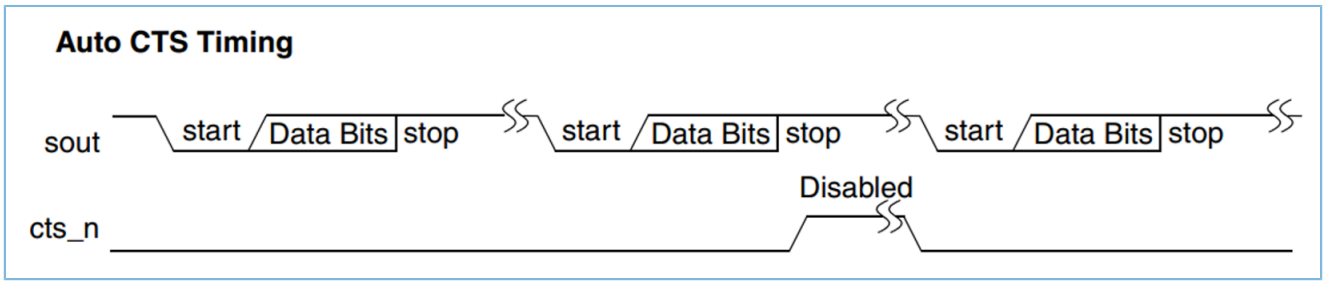

Figure 10-19 Auto CTS timing

10.7.3.5 FIFO and DMA

The UART module provide TWO 128-byte FIFO, one (UART TX FIFO) for transmit and one (UART RX FIFO) for receive. By configuring the Shadow RX Trigger Register (**SHADOW_RX_TRG**), you can program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. By configuring the Shadow TX Trigger Register (**SHADOW_TX_TRG**), you can program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. The UART module supported TWO DMA channel, one for UART transmit, and one for UART receive.

10.7.4 Registers

10.7.4.1 RX_BUF

- **Name:**Receive Buffer Register
- **Description:** This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL[7]) is cleared.

- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 10-93 Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	RX_BUF	R	0x0	Receive Buffer Register. This register contains the data byte received on the serial input port (sin) in UART mode. The data in this register is valid only if the Data Ready (DATA_READY) bit in the Line Status Register (LINE_STAT)) is set.

10.7.4.2 DIV_LATCH_LOW

- **Name:** Divisor Latch (Low)
- **Description:** This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL[7]) is set.
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 10-94 Divisor Latch (Low)

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	DIV_LATCH_LOW	RW	0x0	Divisor Latch (Low). This register makes up the lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. Note that with the Divisor Latch Registers (DIV_LATCH_LOW and DIV_LATCH_HIGH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DIV_LATCH_LOW is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

10.7.4.3 TX_HDG

- **Name:** Transmit Holding Register
- **Description:** This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL[7]) is cleared.
- **Base Address:**0xA000C600 + x*0x100

- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 10-95 Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	TX_HDG	W	0x0	Transmit Holding Register. This register contains data to be transmitted on the serial output port (sout) in UART mode. Data should only be written to the TX_HDG when the TX_HDG Empty (TX_HDG_EMPTY) bit (LINE_STAT [5]) is set.

10.7.4.4 DIV_LATCH_HIGH

- **Name:** Divisor Latch High
- **Description:** This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL [7]) is set.
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x04
- **Reset Value:** 0x0000000

Table 10-96 Divisor Latch High

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	DIV_LATCH_HIGH	RW	0x0	Upper 8-bit of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. Note that with the Divisor Latch Registers (DIV_LATCH_LOW and DIV_LATCH_HIGH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DIV_LATCH_HIGH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

10.7.4.5 INT_EN

- **Name:** Interrupt Enable Register
- **Description:** This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL [7]) is cleared.
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x04
- **Reset Value:**0x00000000

Table 10-97 Interrupt Enable Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	INT_GTHEE	RW	0x0	This is used to enable/disable the generation of TX_HDG_EMPTY Interrupt. **Value:** • 0x0 (DISABLED): Disable Programmable TX_HDG_EMPTY Interrupt Mode • 0x1 (ENABLED): Enable Programmable TX_HDG_EMPTY Interrupt Mode
6:4	RSVD	R		Reserved bits
3	INT_MSE	RW	0x0	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. **Value:** • 0x0 (DISABLED): Disable Modem Status Interrupt • 0x1 (ENABLED): Enable Modem Status Interrupt
2	INT_RX_LSE	RW	0x0	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. **Value:** • 0x0 (DISABLED): Disable Receiver Line Status Interrupt • 0x1 (ENABLED): Enable Receiver Line Status Interrupt
1	INT_TX_HEE	RW	0x0	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. **Value:** • 0x0 (DISABLED): Disable Transmit empty interrupt • 0x1 (ENABLED): Enable Transmit empty interrupt
0	INT_RX_EN	RW	0x0	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. **Value:** • 0x0 (DISABLED): Disable Receive data Interrupt • 0x1 (ENABLED): Enable Receive data Interrupt

10.7.4.6 FIFO_CTRL

- **Name:** FIFO Control Register
- **Description:**FIFO Control Register

- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x08
- **Reset Value:**0x00000000

Table 10-98 FIFO Control Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:6	RX_FIFO_TRG	W	0x0	RX Trigger (or RX_FIFO_TRG). 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 only when RTC_FCT is disabled. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. **Value:** • 0x0 (FIFO_CHAR_1): 1 character in FIFO • 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 full • 0x2 (FIFO_HALF_FULL): FIFO 1/2 full • 0x3 (FIFO_FULL_2): FIFO 2 less than full
5:4	TX_EMPTY_TRG	W	0x0	TX Empty Trigger (or TX_EMPTY_TRG). This is used to select the empty threshold level at which the TX_HDG_EMPTY 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. **Value:** • 0x0 (FIFO_EMPTY): FIFO Empty • 0x1 (FIFO_CHAR_2): 2 characters in FIFO • 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 full • 0x3 (FIFO_HALF_FULL): FIFO 1/2 full
3	DMA_MODE	W	0x0	DMA Mode (or DMA_MODE). This determines the DMA mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected. **Value:** • 0x0 (MODE0): Mode 0 • 0x1 (MODE1): Mode 1
2	TX_FIFO_RST	W	0x0	TX FIFO Reset (or TX_FIFO_RST). This resets the control portion of the transmit FIFO and treats the FIFO as empty. This will also de-assert the

Bits	Field Name	RW	Reset	Description
				DMA TX request and single signals when additional DMA handshaking signals are selected. **Value:** 0x1 (RESET): Transmit FIFO reset
1	RX_FIFO_RST	W	0x0	RX FIFO Reset (or RX_FIFO_RST). This resets the control portion of the receive FIFO and treats the FIFO as empty. This will also de-assert the DMA RX request and single signals when additional DMA handshaking signals are selected. **Value:** 0x1 (RESET): Receive FIFO reset
0	FIFO_EN	W	0x0	FIFO Enable (or FIFO_EN). This enables/disables the transmit (TX) and receive (RX) FIFOs. Whenever the value of this bit is changed both the TX and RX controller portion of FIFOs is reset. **Value:** • 0x0 (DISABLED): FIFO disabled • 0x1 (ENABLED): FIFO enabled

10.7.4.7 INT_ID

- **Name:** Interrupt Identification Register
- **Description:** Interrupt Identification Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x08
- **Reset Value:** 0x00000001

Table 10-99 Interrupt Identification Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:6	FIFO_EN	R	0x0	FIFOs Enabled (or FIFO_EN). This is used to indicate whether the FIFOs are enabled or disabled. **Value:** • 0x0 (DISABLED): FIFOs are disabled • 0x3 (ENABLED): FIFOs are enabled
5:4	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
3:0	INT_PRIOR	R	0x1	Interrupt ID (or priority index). This indicates the highest priority pending interrupt which can be one of the following types specified in Values. **Value:** • 0x0 (MODEM_STATUS): modem status • 0x1 (NO_INTERRUPT_PENDING): no interrupt pending • 0x2 (TX_HDG _EMPTY): TX_HDG empty • 0x4 (RECEIVED_DATA_AVAILABLE): received data available • 0x6 (RECEIVER_LINE_STATUS): receiver line status • 0xC (CHARACTER_TIMEOUT): character timeout

10.7.4.8 LINE_CTRL

- **Name:** Line Control Register
- **Description:** Line Control Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x0C
- **Reset Value:**0x00000000

Table 10-100 Line Control Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	DIV_LATCH_AB	RW	0x0	Divisor Latch Access Bit. This bit is used to enable reading and writing of the Divisor Latch Register (DIV_LATCH_LOW and DIV_LATCH_HIGH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. **Value:** • 0x0 (DISABLED): Divisor Latch register is writable only when UART is not busy. • 0x1 (ENABLED): Divisor Latch register is always readable and writable
6	BREAK_CTRL_BIT	RW	0x0	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 MODEM_CTRL[4], the sout line is forced low until the Break bit is cleared. When in

Bits	Field Name	RW	Reset	Description
				Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. **Value:** • 0x0 (DISABLED): Serial output is released for data transmission • 0x1 (ENABLED): Serial output is forced to spacing state
5	STICK_PARITY	RW	0x0	Stick Parity. This bit is used to force parity value. When PARITY_EN, EVEN_PARITY_SEL and STICK_PARITY are set to 1, the parity bit is transmitted and checked as logic 0. If PARITY_EN and Stick Parity are set to 1 and EVEN_PARITY_SEL 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. **Value:** • 0x0 (DISABLED): STICK_PARITY disabled • 0x1 (ENABLED): STICK_PARITY enabled
4	EVEN_PARITY_SEL	RW	0x0	Even Parity Select. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If this bit is set to one, an even number of logic '1's is transmitted or checked. If set to zero, an odd number of logic '1's is transmitted or checked. **Value:** • 0x0 (ODD_PARITY): an odd parity is transmitted or checked • 0x1 (EVEN_PARITY): an even parity is transmitted or checked
3	PARITY_EN	RW	0x0	Parity Enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. **Value:** • 0x0 (DISABLED): disable parity • 0x1 (ENABLED): enable parity
2	STOP_BITS	RW	0x0	Number of Stop bits. This is used to select the number of Stop bits per character that the peripheral will transmit and receive. If set to zero, one stop bit is transmitted in the serial data. If this bit is set to one and the Data bits are set to 5 (LINE_CTRL [1:0] set to zero), one and a half Stop bits are transmitted. Otherwise, two Stop

Bits	Field Name	RW	Reset	Description
				bits are transmitted. Note that regardless of the number of stop bits selected, the receiver will only check the first Stop bit. **Note:** The Stop bit duration implemented by UART may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction. **Value:** • 0x0 (STOP_1BIT): 1 stop bit • 0x1 (STOP_1_5BIT_OR_2BIT): 1.5 stop bits when DATA_LEN_SEL (LINE_CTRL[1:0]) is zero, else 2 stop bit
1:0	DATA_LEN_SEL	RW	0x0	Data Length Select (or CLS as used in legacy). When DATA_LEN_SEL_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral will transmit and receive. **Value:** • 0x0 (CHAR_5BITS): 5 data bits per character • 0x1 (CHAR_6BITS): 6 data bits per character • 0x2 (CHAR_7BITS): 7 data bits per character • 0x3 (CHAR_8BITS): 8 data bits per character

10.7.4.9 MODEM_CTRL

- **Name:** Modem Control Register
- **Description:** Modem Control Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x10
- **Reset Value:**0x00000000

Table 10-101 Modem Control Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4	LOOP_BACK	RW	0x0	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. **Value:** • 0x0 (DISABLED): Loopback mode disabled

Bits	Field Name	RW	Reset	Description
				• 0x1 (ENABLED): Loopback mode enabled
3:2	RSVD	R		Reserved bits
1	REQ_TO_SEND	RW	0x0	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. **Value:** • 0x0 (INACTIVE): Request to Send rts_n de-asserted (logic 1) • 0x1 (ACTIVE): Request to Send rts_n asserted (logic 0)
0	RSVD	R		Reserved bits

10.7.4.10 LINE_STAT

- **Name:** Line Status Register
- **Description:** Line Status Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x14
- **Reset Value:** 0x00000060

Table 10-102 Line Status Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	RX_FIFO_ERR	R	0x0	Receiver FIFO Error bit. This is used to indicate if there is at least one parity error, framing error, or break indication in the 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. **Value:** • 0x0 (NO_RX_FIFO_ERROR): No error in RX FIFO • 0x1 (RX_FIFO_ERROR): Error in RX FIFO
6	TX_EMPTY	R	0x1	Transmitter Empty bit. If FIFO is enabled (FIFO_CTRL[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. **Value:** • 0x0 (DISABLED): Transmitter not empty

Bits	Field Name	RW	Reset	Description
				• 0x1 (ENABLED): Transmitter empty
5	TX_HDG_EMPTY	R	0x1	Transmit Holding Register Empty bit. If TX_HDG_EMPTY mode is disabled (INT_EN[7] set to zero) and regardless of FIFO's being implemented/ enabled or not, this bit indicates that the TX_HDG or TX FIFO is empty. **Value:** • 0x0 (DISABLED): TX_HDG_EMPTY interrupt control is disabled • 0x1 (ENABLED): TX_HDG_EMPTY interrupt control is enabled
4	BREAK_INT	R	0x0	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. **Value:** • 0x0 (NO_BREAK): No break sequence detected • 0x1 (BREAK): Break sequence detected
3	FRAMING_ERR	R	0x0	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_BITS bit in the received data. **Value:** • 0x0 (NO_FRAMING_ERROR): no framing error • 0x1 (FRAMING_ERROR): framing error
2	PARITY_ERR	R	0x0	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PARITY_EN) bit (LINE_CTRL[3]) is set. **Value:** • 0x0 (NO_PARITY_ERROR): no parity error • 0x1 (PARITY_ERROR): parity error
1	OVER_ERR	R	0x0	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. **Value:** • 0x0 (NO_OVER_RUN_ERROR): no overrun error • 0x1 (OVER_RUN_ERROR): overrun error
0	DATA_READY	R	0x0	Data Ready bit.

Bits	Field Name	RW	Reset	Description
				This is used to indicate that the receiver contains at least one character in the RX_BUF or the receiver FIFO. This bit is cleared when the receiver FIFO is empty, in the FIFO mode. **Value:** • 0x0 (NOT_READY): data not ready • 0x1 (READY): data ready

10.7.4.11 MODEM_STAT

- **Name:** Modem Status Register
- **Description:** Whenever bit 0 is set to logic 1 to indicate a change on the modem control inputs, a modem status interrupt will be generated if enabled via the INT_EN regardless of when the change occurred. The bits (bit 0) can be set after a reset, even though their respective modem signals are inactive, because the synchronized version of the modem signals have a reset value of 0 and change to value 1 after reset. To prevent unwanted interrupts due to this change, a read of the MODEM_STAT register can be performed after reset.
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 10-103 Modem Status Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4	CLR_SEND	R	0x0	Clear to Send. This is used to indicate the current state of the modem control line cts_n. That is, this bit is the complement 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. In Loopback Mode (MODEM_CTRL[4] set to one), CLR_SEND is the same as MODEM_CTRL[1] (REQ_TO_SEND). **Value:** • 0x0 (DEASSERTED): cts_n input is de-asserted (logic 1) • 0x1 (ASSERTED): cts_n input is asserted (logic 0)
3:1	RSVD	R		Reserved bits
0	DELTA_CLR_SEND	R	0x0	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MODEM_STAT was read.

Bits	Field Name	RW	Reset	Description
				Reading the MODEM_STAT clears the DELTA_CLR_SEND bit. In Loopback Mode (MODEM_CTRL[4] set to one), DELTA_CLR_SEND reflects changes on MODEM_CTRL[1] (REQ_TO_SEND). **Note:** If the DELTA_CLR_SEND bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DELTA_CLR_SEND bit will get set when the reset is removed if the cts_n signal remains asserted. **Value:** • 0x0 (NO_CHANGE): no change on cts_n since last read of MODEM_STAT • 0x1 (CHANGE): change on cts_n since last read of MODEM_STAT

10.7.4.12 SCRATCHPAD

- **Name:** Scratchpad Register
- **Description:** Scratchpad Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 10-104 Scratchpad Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	SCRATCHPAD	RW		This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.

10.7.4.13 SHADOW_RX_BUFx

- **Name:** Shadow Receive Buffer Register
- **Description:** This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL[7]) is cleared.
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 10-105 Shadow Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	SHADOW_RX_BUFn	R		Shadow Receive Buffer Register n. This is a shadow register for the RX_BUF 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. The data in this register is valid only if the Data Ready (DATA_READY) bit in the Line Status Register (LINE_STAT) is set. • If FIFOs are disabled (FIFO_CTRL[0] set to zero), the data in the RX_BUF must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. • If FIFOs are enabled (FIFO_CTRL[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.

10.7.4.14 SHADOW_TX_HDGx

- **Name:** Shadow Transmit Holding Register
- **Description:** This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL[7]) is cleared.
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 10-106 Shadow Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD_STHRn	R		SHADOW_TX_HDG0 31 to SHADOW_TX_HDGN_REG_SIZE Reserved bits read as 0.
7:0	SHADOW_TX_HDGn	W	0x0	Shadow Transmit Holding Register n. This is a shadow register for the TX_HDG 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. Data should only be written to the TX_HDG when the TX_HDG Empty (TX_HDG_EMPTY) bit (LINE_STAT[5]) is set. • If FIFOs are disabled (FIFO_CTRL[0] set to zero) and TX_HDG_EMPTY is set, writing a single character to the TX_HDG clears the

Bits	Field Name	RW	Reset	Description
				TX_HDG_EMPTY. Any additional writes to the TX_HDG before the TX_HDG_EMPTY is set again causes the TX_HDG data to be overwritten. • If FIFOs are enabled (FIFO_CTRL[0] set to one) and TX_HDG_EMPTY is set, x number of characters of data may be written to the TX_HDG before the FIFO is full. The number x (default=128) 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.

10.7.4.15 FIFO_ACCESS

- **Name:** FIFO Access Register
- **Description:** FIFO Access Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x70
- **Reset Value:** 0x00000000

Table 10-107 FIFO Access Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	FIFO_ACCESS	R	0x0	This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFO's are implemented and enabled. When FIFOs are not implemented or not enabled, it allows the RX_BUF to be written by the master and the TX_HDG to be read by the master. **Note:** When the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFO's are treated as empty. **Value:** • 0x0 (DISABLED): FIFO access mode disabled • 0x1 (ENABLED): FIFO access mode enabled

10.7.4.16 STAT

- **Name:** UART Status Register
- **Description:** UART Status register
- **Base Address:** 0xA000C600 + x*0x100

- **Offset:** 0x7C
- **Reset Value:** 0x00000006

Table 10-108 UART Status Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4	RX_FIFO_FULL	R	0x0	Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. This bit is cleared when the RX FIFO is no longer full. **Value:** • 0x0 (NOT_FULL): Receive FIFO not full • 0x1 (FULL): Receive FIFO full
3	RX_FIFO_EMPTY	R	0x0	Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. This bit is cleared when the RX FIFO is empty. **Value:** • 0x0 (EMPTY): Receive FIFO is empty • 0x1 (NOT_EMPTY): Receive FIFO is not empty
2	TX_FIFO_EMPTY	R	0x1	Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. This bit is cleared when the TX FIFO is no longer empty. **Value:** • 0x0 (NOT_EMPTY): Transmit FIFO is not empty • 0x1 (EMPTY): Transmit FIFO is empty
1	TX_FIFO_FULL	R	0x1	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. This bit is cleared when the TX FIFO is full. **Value:** • 0x0 (FULL): Transmit FIFO is full • 0x1 (NOT_FULL): Transmit FIFO is not full
0	RSVD	R		Reserved bit

10.7.4.17 TX_FIFO_LEVEL

- **Name:** Transmit FIFO Level
- **Description:** Transmit FIFO Level

- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x80
- **Reset Value:** 0x00000000

Table 10-109 Transmit FIFO Level

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	TX_FIFO_LEVEL	R	0x0	Transmit FIFO Level This indicates the number of data entries in the transmit FIFO.

10.7.4.18 RX_FIFO_LEVEL

- **Name:** Receive FIFO Level
- **Description:** Receive FIFO Level
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x84
- **Reset Value:** 0x00000000

Table 10-110 Receive FIFO Level

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	RX_FIFO_LEVEL	R	0x0	Receive FIFO Level This is indicates the number of data entries in the receive FIFO.

10.7.4.19 SW_RST

- **Name:** Software Reset Register
- **Description:** Software Reset Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x88
- **Reset Value:** 0x00000000

Table 10-111 Software Reset Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	TX_FIFO_RST	W	0x0	TX FIFO Reset.

Bits	Field Name	RW	Reset	Description
				This is a shadow register for the TX FIFO Reset bit (FIFO_CTRL[2]). This can be used to remove the burden on software having to store previously written FIFO_CTRL values (which are pretty static) just to reset the transmit FIFO.
1	RX_FIFO_RST	W	0x0	RX FIFO Reset. This is a shadow register for the RX FIFO Reset bit (FIFO_CTRL[1]). This can be used to remove the burden on software having to store previously written FIFO_CTRL values (which are pretty static) just to reset the receive FIFO.
0	UART_RST	W	0x0	UART Reset. This asynchronously resets the UART and synchronously removes the reset assertion. For a two-clock implementation both PCLK and SCLK domains will be reset. **Value:** • 0x0 (NO_RESET): No UART Reset • 0x1 (RESET): UART reset

10.7.4.20 SHADOW_REQ_SEND

- **Name:** Shadow Request to Send
- **Description:** Shadow Request to Send
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x8C
- **Reset Value:** 0x00000000

Table 10-112 Shadow Request to Send

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SHADOW_REQ_SEND	RW	0x0	Shadow Request to Send. This is a shadow register for the REQ_TO_SEND bit (MODEM_CTRL [1]), this can be used to remove the burden of having to performing a read modify write on the MODEM_CTRL. 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. **Value:** • 0x0 (DEASSERTED): Shadow Request to Send uart_rts_n logic 1

Bits	Field Name	RW	Reset	Description
				• 0x1 (ASSERTED): Shadow Request to Send uart_rts_n logic 0

10.7.4.21 SHADOW_BREAK_CTRL

- **Name:** Shadow Break Control Register
- **Description:** Shadow Break Control Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x90
- **Reset Value:** 0x00000000

Table 10-113 Shadow Break Control Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SHADOW_BREAK_CB	RW	0x0	Shadow Break Control Bit. This is a shadow register for the Break bit (LINE_CTRL[6]). This can be used to remove the burden of having to performing a read modify write on the LINE_CTRL. 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 MODEM_CTRL [4], the sout line is forced to be low until the Break bit is cleared. **Value:** • 0x0 (NO_BREAK): No spacing on serial output • 0x1 (BREAK): Serial output forced to the spacing

10.7.4.22 SHADOW_DMA_MODE

- **Name:** Shadow DMA Mode Register
- **Description:** Shadow DMA Mode Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x94
- **Reset Value:** 0x00000000

Table 10-114 Shadow DMA Mode Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
0	SHADOW_DMA_MODE	RW	0x0	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FIFO_CTRL[3]). This can be used to remove the burden of having to store the previously written value to the FIFO_CTRL in memory and having to mask this value so that only the DMA Mode bit gets updated. **Value:** • 0x0 (MODE_0): Mode 0 • 0x1 (MODE_1): Mode 1

10.7.4.23 SHADOW_FIFO_EN

- **Name:** Shadow FIFO Enable Register
- **Description:** Shadow FIFO Enable Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x98
- **Reset Value:** 0x00000000

Table 10-115 Shadow FIFO Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SHADOW_FIFO_EN	RW	0x0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FIFO_CTRL[0]). This can be used to remove the burden of having to store the previously written value to the FIFO_CTRL in memory and having to mask this value so that only the FIFO enable bit gets updated. This enables/disables the transmit (TX) and receive (RX) FIFO's. If this bit is set to zero (disabled) after being enabled then both the TX and RX controller portions of FIFO's will be reset. **Value:** • 0x0 (DISABLED): FIFOs are disabled • 0x1 (ENABLED): FIFOs are enabled

10.7.4.24 SHADOW_RX_TRG

- **Name:** Shadow RX Trigger Register
- **Description:** Shadow RX Trigger Register

- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0x9C
- **Reset Value:** 0x00000000

Table 10-116 Shadow RCVR Trigger Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	SHADOW_RX_TRG	RW	0x0	Shadow RX Trigger. This is a shadow register for the RX trigger bits (FIFO_CTRL[7:6]). This can be used to remove the burden of having to store the previously written value to the FIFO_CTRL in memory and having to mask this value so that only the RX trigger bit gets updated. **Value:** • 0x0 (FIFO_CHAR_1): 1 character in FIFO • 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 full • 0x2 (FIFO_HALF_FULL): FIFO 1/2 full • 0x3 (FIFO_FULL_2): FIFO 2 less than full

10.7.4.25 SHADOW_TX_TRG

- **Name:** Shadow TX Empty Trigger Register
- **Description:** Shadow TX Empty Trigger Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0xA0
- **Reset Value:** 0x00000000

Table 10-117 Shadow TX Empty Trigger Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	SHADOW_TX_TRG	RW	0x0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FIFO_CTRL[5:4]). This can be used to remove the burden of having to store the previously written value to the FIFO_CTRL in memory and having to mask this value so that only the TX empty trigger bit gets updated. **Value:** 0x0 (FIFO_EMPTY): FIFO empty

Bits	Field Name	RW	Reset	Description
				0x1 (FIFO_CHAR_2): 2 characters in FIFO 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 full 0x3 (FIFO_HALF_FULL): FIFO 1/2 full

10.7.4.26 HALT_TX

- **Name:** Halt TX
- **Description:** Halt TX
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0xA4
- **Reset Value:** 0x00000000

Table 10-118 Halt TX

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	HALT_TX	RW	0x0	Halt TX. This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. **Value:** • 0x0 (DISABLED): Halt Transmission disabled • 0x1 (ENABLED): Halt Transmission enabled

10.7.4.27 DMA_SW_ACK

- **Name:** DMA Software Acknowledge Register
- **Description:** DMA Software Acknowledge Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0xA8
- **Reset Value:** 0x00000000

Table 10-119 DMA Software Acknowledge Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	DMA_SW_ACK	W		DMA Software Acknowledge. This register is use to perform DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables

Bits	Field Name	RW	Reset	Description
				the channel, then the 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. **Value:** 0x1 (SOFT_ACK): DMA software acknowledge

10.7.4.28 DIV_LATCH_FRACTION

- **Name:** Divisor Latch Fraction Register
- **Description:** Divisor Latch Fraction Register
- **Base Address:** 0xA000C600 + x*0x100
- **Offset:** 0xC0
- **Reset Value:** 0x00000000

Table 10-120 Divisor Latch Fraction Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	DIV_LATCH_FRACTION	RW	0x0	Fractional part of divisor. The fractional value is added to integer value set by DIV_LATCH_HIGH, and DIV_LATCH_LOW. Fractional value is determined by (Divisor Fraction value)/(2^DIV_LATCH_FRACTION_SIZE).

10.7.5 Electrical Specifications

10.7.5.1 UART Timing Specification

GR551x UART timing complies with the asynchronous mode of RS-232 protocol. Signal timing adjustments such as choosing a different stop bit period can be achieved by programming pertinent UART registers.

10.8 SPI

10.8.1 Introduction

GR551x has two Serial Peripheral Interface (SPI) instances.

- 1 SPI Master (SPIM) with two slave select lines that can interface to up to two slaves that share the same lines with separate selects.
- 1 SPI Slave (SPIS) that can connect to a single master.

10.8.2 Main Features

- Motorola support
- 2 slave select lines for master mode to support up to 2 slaves
- Built-in 8-word RX/TX FIFOs for continuous SPI bursts
- Transfer data size up to 32 bits
- Programmable output interface frequency up to 32 MHz for SPIM and 10.67 MHz for SPIS
- Supports SPI four modes for different clock edge and phase configurations.
- DMA support
- Maskable interrupt generation

Note:

Refer to Section 10.8.5.1 SPIM Electrical Specifications for details.

10.8.3 Functional Description

The SPI Controller can be used with one of the following interfaces:

- Motorola Serial Peripheral Interface (SPI)

The FRAME_FORMAT (frame format) bit field in the Control Register 0 (**CTRL0**) can be programmed to select which protocol is used.

Figure 10-20 depicts the SPI Controller with the following functions and interfaces:

- APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt logic

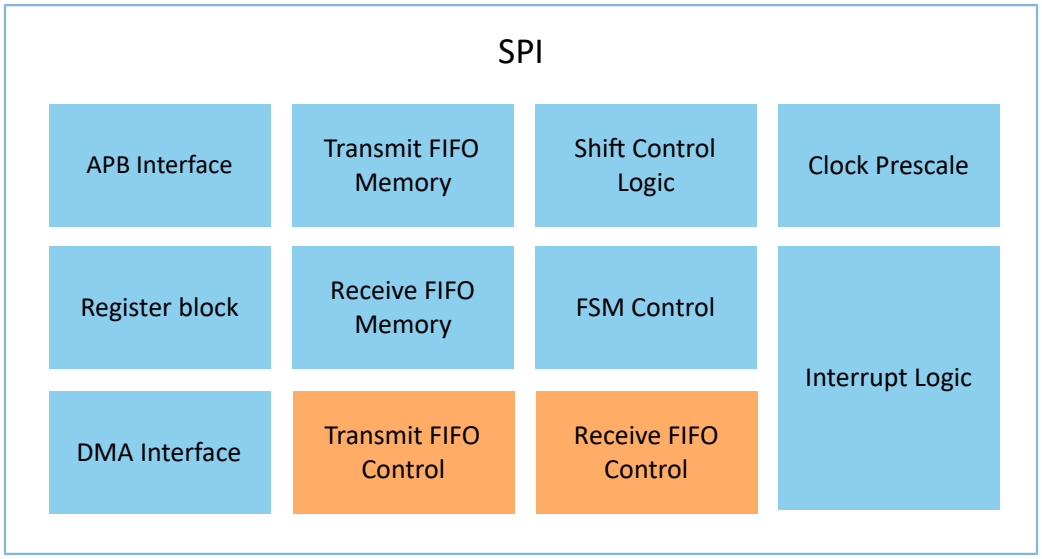

Figure 10-20 SPI block diagram

10.8.3.1 Transmit and Receive FIFO Buffers

The width of both transmit and receive FIFO buffers is fixed at 32 bits. Data frames that are less than 32 bits must be right-justified when written into the transmit FIFO buffer. The shift control logic automatically right-justifies receive data in the receive FIFO buffer.

Each data entry in the FIFO buffers contains a single data frame. It is not possible to store multiple data frames in a single FIFO location; for example, you may not store two 8-bit data frames in a single FIFO location. If an 8-bit data frame is required, the upper 8-bit of the FIFO entry are ignored or unused when the serial shifter transmits the data.

The transmit FIFO is loaded by writing to the SPI Controller data register (**DATA**). Data is popped from the transmit FIFO by the shift control logic into the transmit shift register. The transmit FIFO generates a FIFO empty interrupt request (**ssi_txe_intr**) when the number of entries in the FIFO is less than or equal to the FIFO threshold value. The threshold value, set through the programmable register **TX_FIFO_TL**, determines the level of FIFO entries at which an interrupt is generated. The threshold value allows you to provide early indication to the processor that the transmit FIFO is nearly empty. A transmit FIFO overflow interrupt (**ssi_txo_intr**) is generated if you attempt to write data into an already full transmit FIFO.

Data is popped from the receive FIFO by reading the SPI Controller data register (**DATA**). The receive FIFO is loaded from the receive shift register by the shift control logic. The receive FIFO generates a FIFO-full interrupt request (**ssi_rxf_intr**) when the number of entries in the FIFO is greater than or equal to the FIFO threshold value plus 1. The threshold value, set through programmable register **RX_FIFO_TL**, determines the level of FIFO entries at which an interrupt is generated.

The threshold value allows you to provide early indication to the processor that the receive FIFO is nearly full. A receive FIFO overrun interrupt (**ssi_rxo_intr**) is generated when the receive shift logic attempts to load data into a completely full receive FIFO. However, this newly received data is lost. A receive FIFO underflow interrupt (**ssi_rxu_intr**) is generated if you attempt to read from an empty receive FIFO. This alerts the processor that the read data is invalid.

10.8.3.2 SPI Interrupts

The SPI Controller supports individual interrupt requests, each of which can be masked.

The SPI Controller interrupts are described as follows:

- Transmit FIFO Empty Interrupt (**ssi_txe_intr**) – Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of transmit FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is written into the transmit FIFO buffer, bringing it over the threshold level.
- Transmit FIFO Overflow Interrupt (**ssi_txo_intr**) – Set when an APB access attempts to write into the transmit FIFO after it has been completely filled. When set, data written from the APB is discarded. This interrupt remains set until you read the transmit FIFO overflow interrupt clear register (**TX_FIFO_OIC**).
- Receive FIFO Full Interrupt (**ssi_rxf_intr**) – Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of receive FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is read from the receive FIFO buffer, bringing it below the threshold level.
- Receive FIFO Overflow Interrupt (**ssi_rxo_intr**) – Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. When set, newly received data is discarded. This interrupt remains set until you read the receive FIFO overflow interrupt clear register (**RX_FIFO_OIC**).
- Receive FIFO Underflow Interrupt (**ssi_rxu_intr**) – Set when an APB access attempts to read from the receive FIFO when it is empty. When set, zeros are read back from the receive FIFO. This interrupt remains set until you read the receive FIFO underflow interrupt clear register (**RX_FIFO_UIC**).
- Multi-Master Contention Interrupt (**ssi_mst_intr**) – Present only when the SPI Controller component is configured as a serial-master device. The interrupt is set when another serial master on the serial bus selects the SPI Controller master as a serial-slave device and is actively transferring data. This informs the processor of possible contention on the serial bus. This interrupt remains set until you read the multi-master interrupt clear register (**MULTI_M_IC**).

10.8.3.3 Transfer Modes

When transferring data on the serial bus, the SPI Controller operates in the modes discussed in this section. The transfer mode (XFE_MODE) is set by writing to control register 0 (**CTRL0**).

Note:

The transfer mode setting does not affect the duplex of the serial transfer. XFE_MODE is ignored for Microwire transfers, which are controlled by the **MW_CTRL** register.

10.8.3.3.1 Transmit and Receive

When XFE_MODE = 2'b00, both transmit and receive logic are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data is popped from the transmit FIFO and sent through the

txd line to the target device, which replies with data on the rxd line. The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame.

10.8.3.3.2 Transmit Only

When XFE_MODE = 2'b01, the receive data is invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data is popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. You should mask interrupts originating from the receive logic when this mode is entered.

10.8.3.3.3 Receive Only

When XFE_MODE = 2'b10, the transmit data is invalid. When configured as a slave, the transmit FIFO is never popped in Receive Only mode. The txd output remains at a constant logic level during the transmission. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. You should mask interrupts originating from the transmit logic when this mode is entered.

10.8.3.3.4 EEPROM Read

Note:

This transfer mode is only valid for SPIM configurations.

When XFE_MODE = 2'b11, the transmit data is used to transmit an opcode and/or an address to the EEPROM device. Typically this takes three data frames (8-bit opcode followed by 8-bit upper address and 8-bit lower address). During the transmission of the opcode and address, no data is captured by the receive logic (as long as the SPI Controller master is transmitting data on its txd line, data on the rxd line is ignored). The SPI Controller master continues to transmit data until the transmit FIFO is empty. Therefore, you should only have enough data frames in the transmit FIFO to supply the opcode and address to the EEPROM. If more data frames are in the transmit FIFO than are needed, then read data is lost.

When the transmit FIFO becomes empty (all control information has been sent), data on the receive line (rxd) is valid and stored in the receive FIFO; the txd output is held at a constant logic level. The serial transfer continues until the number of data frames received by the SPI Controller master matches the value of the **NUM_DATA_FRAME** field in the **CTRL1** register + 1.

Note:

EEPROM read mode is not supported when the SPI Controller is configured to be in the SSP mode.

10.8.3.4 Serial Master Operation

10.8.3.4.1 Data Transfers

Data transfers are started by the serial-master device. When the SPI Controller is enabled (SSI_EN=1), at least one valid data entry is present in the transmit FIFO and a serial-slave device is selected. When actively transferring data, the busy flag (**SSI_BUSY**) in the status register (**STAT**) is set. You must wait until the busy flag is cleared before attempting a new serial transfer.

10.8.3.4.2 Master SPI and SSP Serial Transfers

When the transfer mode is “transmit and receive” or “transmit only” (XFE_MODE = 2'b00 or XFE_MODE = 2'b01, respectively), transfers are terminated by the shift control logic when the transmit FIFO is empty. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data has been transmitted. The transmit FIFO threshold level (**TX_FIFO_TL**) can be used to early interrupt (**ssi_txe_intr**) the processor indicating that the transmit FIFO buffer is nearly empty. When a DMA is used for APB accesses, the transmit data level (**DMA_TX_DL**) can be used to early request (**dma_tx_req**) the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. The user may also write a block of data (at least two FIFO entries) into the transmit FIFO before enabling a serial slave. This ensures that serial transmission does not begin until the number of data-frames that make up the continuous transfer are present in the transmit FIFO.

When the transfer mode is “receive only” (XFE_MODE = 2'b10), a serial transfer is started by writing one “dummy” data word into the transmit FIFO when a serial slave is selected. The TXD output from the SPI Controller is held at a constant logic level for the duration of the serial transfer. The transmit FIFO is popped only once at the beginning and may remain empty for the duration of the serial transfer. The end of the serial transfer is controlled by the “number of data frames” (**NUM_DATA_FRAME**) field in control register 1 (**CTRL1**).

If, for example, you want to receive 24 data frames from a serial-slave peripheral, you should program the **NUM_DATA_FRAME** field with the value **23**; the receive logic terminates the serial transfer when the number of frames received is equal to the **NUM_DATA_FRAME** value + **1**. This transfer mode increases the bandwidth of the APB bus as the transmit FIFO never needs to be serviced during the transfer. The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow.

When the transfer mode is “eeprom_read” (XFE_MODE = 2‘b11), a serial transfer is started by writing the opcode and/or address into the transmit FIFO when a serial slave (EEPROM) is selected. The opcode and address are transmitted to the EEPROM device, after which read data is received from the EEPROM device and stored in the receive FIFO. The end of the serial transfer is controlled by the **NUM_DATA_FRAME** field in the control register 1 (**CTRL1**).

The receive FIFO threshold level (**RX_FIFO_TL**) can be used to give early indication that the receive FIFO is nearly full. When a DMA is used for APB accesses, the receive data level (**DMA_RX_DL**) can be used to early request (**dma_rx_req**) the DMA Controller, indicating that the receive FIFO is nearly full.

10.8.3.5 Serial Slave Operation

10.8.3.5.1 Slave SPI and SSP Serial Transfers

If the SPI Controller slave is receive only (XFE_MODE =10), the transmit FIFO need not contain valid data because the data currently in the transmit shift register is resent each time the slave device is selected. The TX_ERR error flag

in the status register (**STAT**) is not set when XFE_MODE =01. You should mask the transmit FIFO empty interrupt when this mode is used.

If the SPI Controller slave transmits data to the master, you must ensure that data exists in the transmit FIFO before a transfer is initiated by the serial-master device. If the master initiates a transfer to the SPI Controller slave when no data exists in the transmit FIFO, an error flag (**TX_ERR**) is set in the SPI Controller status register, and the previously transmitted data frame is resent on txd. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data has been transmitted. The transmit FIFO threshold level register (**TX_FIFO_TL**) can be used to early interrupt (**ssi_txe_intr**) the processor, indicating that the transmit FIFO buffer is nearly empty. When a DMA Controller is used for APB accesses, the DMA transmit data level register (**DMA_TX_DL**) can be used to early request (**dma_tx_req**) the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow. The receive FIFO threshold level register (**RX_FIFO_TL**) can be used to give early indication that the receive FIFO is nearly full. When a DMA Controller is used for APB accesses, the DMA receive data level register (**DMA_RX_DL**) can be used to early request (**dma_rx_req**) the DMA controller, indicating that the receive FIFO is nearly full.

10.8.3.6 DMA Operation

SPI Controller has a handshaking interface to a DMA Controller to request and control transfers. To enable the DMA Controller interface, User must write the DMA Control Register (**DMA_CTRL**). Writing a 1 into the **TX_DMA_EN** bit field of **DMA_CTRL** register enables the transmit handshaking interface. Writing a 1 into the **RX_DMA_EN** bit field of the **DMA_CTRL** register enables the receive handshaking interface.

10.8.3.6.1 Transmit Watermark Level

During SPI serial transfers, transmit FIFO requests are made to the DMA Controller whenever the number of entries in the transmit FIFO is less than or equal to the DMA Transmit Data Level Register (**DMA_TX_DL**) value; this is known as the watermark level. Then the DMA responds by writing a burst of data to the transmit FIFO buffer, and the length of data is determined by **DEST_MSIZE** of DMA.

Data should be fetched from the DMA often enough for the transmit FIFO to perform serial transfers continuously; Otherwise the FIFO could run out of data (underflow). To avoid this situation, the user must set the watermark level correctly.

It is recommended to adhere to the following equation to configure the DMA transmit Operation for SPI, which help to reduce the number of DMA bursts needed for a block transfer, and to improve system utilization.

$$DMA.CTLx.DEST\_MSIZE = SPI.FIFO\_DEPTH - SPI.DMA\_TX\_DL \quad (1)$$

- **DMA.CTLx.DEST_MSIZE**, please refer to Section 10.8.4.20 DMA_CTRL.
- **SPI.FIFO_DEPTH** equals 8, means the FIFO Depth for SPI.
- **SPI.DMA_TX_DL**, please refer to Section 10.8.4.21 DMA_TX_DL.

10.8.3.6.2 Receive Watermark Level

During SPI serial transfers, receive FIFO requests are made to the DMA Controller whenever the number of entries in the receive FIFO is at or above the DMA Receive Data Level Register (**DMA_RX_DL**+1). This is known as the watermark level. The DMA Controller responds by fetching a burst of data from the receive FIFO buffer, and the length of data is determined by **SRC_MSIZE** of DMA.

Data should be fetched by the DMA often enough for the receive FIFO to accept serial transfers continuously. Otherwise, the FIFO will fill with data (overflow). To avoid this situation, the user must set the watermark level correctly.

It is recommended to adhere to the following equation to configure the DMA receive Operation for SPI, which help to improve system utilization.

$$DMA.CTLx.SRC\_MSIZE = SPI.DMA\_RX\_DL + 1 \quad (2)$$

- **DMA.CTLx.SRC_MSIZE**, please refer to Section 10.8.4.20 DMA_CTRL.
- **SPI.DMA_RX_DL**, please refer to Section 10.8.4.22 DMA_RX_DL.

10.8.4 Registers

10.8.4.1 CTRL0

- **Name:** Control Register 0
- **Description:** This register controls the serial data transfer. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x00
- **Reset Value:** 0x01070000

Table 10-121 Control Register 0

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24	S_ST_EN	RW	0x1	Slave Select Toggle Enable. When operating in SPI mode with clock phase (SERIAL_CLK_PHASE) set to 0, this register controls the behavior of the slave select line (CS) between data frames. If this register field is set to 1, the CS line will toggle between consecutive data frames, with the serial clock (SCLK) being held to its default value while CS is high; if this register field is set to 0, the CS will stay low and SCLK will run continuously for the duration of the transfer. **Note:** When the SPI is configured as a slave, this register serves no purpose.
23:21	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
20:16	DATA_FRAME_SIZE	RW	0x7	**Data Frame Size** in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. When the data frame size is programmed to be less than 32 bits, the receive data is automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. **Value:** • 0x3 (FRAME_04BITS): 4-bit serial data transfer • 0x4 (FRAME_05BITS): 5-bit serial data transfer • 0x5 (FRAME_06BITS): 6-bit serial data transfer • 0x6 (FRAME_07BITS): 7-bit serial data transfer • 0x7 (FRAME_08BITS): 8-bit serial data transfer • 0x8 (FRAME_09BITS): 9-bit serial data transfer • 0x9 (FRAME_10BITS): 10-bit serial data transfer • 0xa (FRAME_11BITS): 11-bit serial data transfer • 0xb (FRAME_12BITS): 12-bit serial data transfer • 0xc (FRAME_13BITS): 13-bit serial data transfer • 0xd (FRAME_14BITS): 14-bit serial data transfer • 0xe (FRAME_15BITS): 15-bit serial data transfer • 0xf (FRAME_16BITS): 16-bit serial data transfer • 0x10 (FRAME_17BITS): 17-bit serial data transfer • 0x11 (FRAME_18BITS): 18-bit serial data transfer • 0x12 (FRAME_19BITS): 19-bit serial data transfer • 0x13 (FRAME_20BITS): 20-bit serial data transfer • 0x14 (FRAME_21BITS): 21-bit serial data transfer • 0x15 (FRAME_22BITS): 22-bit serial data transfer • 0x16 (FRAME_23BITS): 23-bit serial data transfer • 0x17 (FRAME_24BITS): 24-bit serial data transfer • 0x18 (FRAME_25BITS): 25-bit serial data transfer • 0x19 (FRAME_26BITS): 26-bit serial data transfer • 0x1a (FRAME_27BITS): 27-bit serial data transfer • 0x1b (FRAME_28BITS): 28-bit serial data transfer

Bits	Field Name	RW	Reset	Description
				• 0x1c (FRAME_29BITS): 29-bit serial data transfer • 0x1d (FRAME_30BITS): 30-bit serial data transfer • 0x1e (FRAME_31BITS): 31-bit serial data transfer • 0x1f (FRAME_32BITS): 32-bit serial data transfer
15:12	CTRL_FRAME_SIZE	RW	0x0	**Control Frame Size**. Selects the length of the control word for the Microwire frame format. **Value:** • 0x0 (SIZE_01_BIT): 1-bit Control Word • 0x1 (SIZE_02_BIT): 2-bit Control Word • 0x2 (SIZE_03_BIT): 3-bit Co ntrol Word • 0x3 (SIZE_04_BIT): 4-bit Control Word • 0x4 (SIZE_05_BIT): 5-bit Control Word • 0x5 (SIZE_06_BIT): 6-bit Control Word • 0x6 (SIZE_07_BIT): 7-bit Control Word • 0x7 (SIZE_08_BIT): 8-bit Control Word • 0x8 (SIZE_09_BIT): 9-bit Control Word • 0x9 (SIZE_10_BIT): 10-bit Control Word • 0xa (SIZE_11_BIT): 11-bit Control Word • 0xb (SIZE_12_BIT): 12-bit Control Word • 0xc (SIZE_13_BIT): 13-bit Control Word • 0xd (SIZE_14_BIT): 14-bit Control Word • 0xe (SIZE_15_BIT): 15-bit Control Word • 0xf (SIZE_16_BIT): 16-bit Control Word
11	SHIFT_REG_LOOP	RW	0x0	**Shift Register Loop**. Used for testing purposes only. When internally active, connect the transmit shift register output to the receive shift register input. **Value:** • 0x0 (NORMAL_MODE): Normal mode operation • 0x1 (TESTING_MODE): Test mode: TX & RX shift register connected
10	S_OUT_EN	RW	0x0	**Slave Output Enable**. Relevant only when the SPI is configured as a serial-slave device. When configured as a serial-master, this bit field has no functionality. **Value:** • 0x0 (ENABLED): Slave Output is enabled

Bits	Field Name	RW	Reset	Description
				• 0x1 (DISABLED): Slave Output is disabled **Note:** When the SPI is configured as a master, this register serves no purpose.
9:8	XFE_MODE	RW	0x0	**Transfer Mode**. This transfer mode is only valid when the SPI is configured as master device. • 0x0 - Transmit & Receive • 0x1 - Transmit Only • 0x2 - Receive Only • 0x3 - EEPROM Read
7	SERIAL_CLK_POL	RW	0x0	**Serial Clock Polarity**. Valid when the frame format (FRAME_FORMAT) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. **Value:** • 0x0 (SCLK_LOW): Inactive state of serial clock is low • 0x1 (SCLK_HIGH): Inactive state of serial clock is high
6	SERIAL_CLK_PHASE	RW	0x0	**Serial Clock Phase**. Valid when the frame format (FRAME_FORMAT) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SERIAL_CLK_PHASE = 0, data is captured on the first edge of the serial clock. When SERIAL_CLK_ PHASE = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data is captured on the second edge of the serial clock. **Value:** • 0x0 (SERIAL_CLK_PHASE_MIDDLE): Serial clock toggles in middle of first data bit • 0x1 (SERIAL_CLK_PHASE_START): Serial clock toggles at start of first data bit
5:4	FRAME_FORMAT	RW	0x0	**Frame Format**. Select which serial protocol transfers the data. **Value:** • 0x0 (MOTOROLA_SPI): Motorola SPI Frame Format • 0x1 (TEXAS_SSP): Texas Instruments SSP Frame Format

Bits	Field Name	RW	Reset	Description
				• 0x2 (NS_MICROWIRE): National Microwire Frame Format • 0x3 (RESERVED): Reserved value
3:0	RSVD	R		Reserved bits

10.8.4.2 CTRL1

- **Name:** Control Register 1
- **Description:** This register exists only when the SPI is configured as a master device. Control register 1 controls the end of serial transfers when in receive-only mode. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x04
- **Reset Value:** 0x00000000

Table 10-122 Control Register 1

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	NUM_DATA_FRAM	RW	0x0	Number of Data Frames. When XFE_MODE = 10 or XFE_MODE = 11, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer. **Note:** When the SPI is configured as a slave, this register serves no purpose.

10.8.4.3 SSI_EN

- **Name:** SSI Enable Register
- **Description:** This register enables and disables the SPI.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x08
- **Reset Value:** 0x00000000

Table 10-123 SSI Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SSI_EN	RW	0x0	**SSI Enable**. Enable and disable all SPI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when all serial transfers are disabled. **Value:** • 0x0 (DISABLE): Disables Serial Transfer • 0x1 (ENABLED): Enables Serial Transfer

10.8.4.4 MW_CTRL

- **Name:** Microwire Control Register
- **Description:** This register controls the direction of the data word for the half-duplex Microwire serial protocol. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 10-124 Microwire Control Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	MW_HSG	RW	0x0	**Microwire Handshaking.** Used to enable and disable the busy/ready handshaking interface for the Microwire protocol. When enabled, the SPI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the SSI_BUSY status in the STAT register. **Value:** • 0x0 (DISABLE): Handshaking interface is disabled • 0x1 (ENABLED): Handshaking interface is enabled **Note:** When the SPI is configured as a slave, this register serves no purpose.
1	MW_DIR_DW	RW	0x0	**Microwire Control**. Define the direction of the data word when the Microwire serial protocol is used.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (RECEIVE): SPI receives data • 0x1 (TRANSMIT): SPI transmits data
0	MW_XFE_MODE	RW	0x0	**Microwire Transfer Mode**. Define whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. **Value:** • 0x0 (NON_SEQUENTIAL): Non-Sequential Microwire Transfer • 0x1 (SEQUENTIAL): Sequential Microwire Transfer

10.8.4.5 S_EN

- **Name:** Slave Enable Register
- **Description:** This register exists only when the SPI is configured as a master device. The register enables the individual slave select output lines from the SPI master.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 10-125 Slave Enable Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	S_EN	RW	0x0	**Slave Select Enable**. Each bit in this register corresponds to a slave select line (CSn) from the SPI master. **Value:** • 0x0 (NOT_SELECTED): No slave selected • 0x1 (SELECTED): Slave is selected **Note:** When the SPI is configured as a slave, this register serves no purpose.

10.8.4.6 BAUD

- **Name:** Baud Rate Register

- **Description:** This register exists only when the SPI is configured as a master device. The register derives the frequency of the serial clock that regulates the data transfer. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 10-126 Baud Rate Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	SSI_CLK_DIV	RW	0x0	**SSI Clock Divider**. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: $F_{sclk\_out} = F_{ssi\_clk}$/ SSI_CLK_DIV **Note:** When the SPI is configured as a slave, this register serves no purpose.

10.8.4.7 TX_FIFO_TL

- **Name:** Transmit FIFO Threshold Level
- **Description:** This register controls the threshold value for the transmit FIFO memory.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 10-127 Transmit FIFO Threshold Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	TX_FIFO_THD	RW	0x0	**Transmit FIFO Threshold**. Control the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of

Bits	Field Name	RW	Reset	Description
				transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

10.8.4.8 RX_FIFO_TL

- **Name:** Receive FIFO Threshold Level
- **Description:** This register controls the threshold value for the receive FIFO memory.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 10-128 Receive FIFO Threshold Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	RX_FIFO_THD	RW	0x0	**Receive FIFO Threshold**. Control the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2 – 256. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

10.8.4.9 TX_FIFO_LEVEL

- **Name:** Transmit FIFO Level Register
- **Description:** This register contains the number of valid data entries in the transmit FIFO memory.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 10-129 Transmit FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	TX_FIFO_LEVEL	R	0x0	**Transmit FIFO Level**.

Bits	Field Name	RW	Reset	Description
				Contain the number of valid data entries in the transmit FIFO.

10.8.4.10 RX_FIFO_LEVEL

- **Name:** Receive FIFO Level Register
- **Description:** This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 10-130 Receive FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	RX_FIFO_LEVEL	R	0x0	**Receive FIFO Level**. Contain the number of valid data entries in the receive FIFO.

10.8.4.11 STAT

- **Name:** Status Register
- **Description:** This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x28
- **Reset Value:** 0x00000006

Table 10-131 Status Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6	DATA_COLN_ERR	R	0x0	**Data Collision Error**. This bit will be set if MISO input is asserted by other master, when the SPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. **Note:** When the SPI is configured as a slave, this register serves no purpose.
5	TX_ERR	R	0x0	**Transmission Error**.

Bits	Field Name	RW	Reset	Description
				Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the txd line. This bit is cleared when read. **Value:** • 0x0 (NO_ERROR): No Error • 0x1 (TX_ERROR): Transmission Error **Note:** When the SPI is configured as a master, this register serves no purpose.
4	RX_FIFO_FULL	R	0x0	**Receive FIFO 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. **Value:** • 0x0 (NOT_FULL): Receive FIFO is not full • 0x1 (FULL): Receive FIFO is full
3	RX_FIFO_NE	R	0x0	**Receive FIFO Not Empty**. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. **Value:** • 0x0 (EMPTY): Receive FIFO is empty • 0x1 (NOT_EMPTY): Receive FIFO is not empty
2	TX_FIFO_EMPTY	R	0x1	**Transmit FIFO Empty**. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. **Value:** • 0x0 (NOT_EMPTY): Transmit FIFO is not empty • 0x1 (EMPTY): Transmit FIFO is empty
1	TX_FIFO_NF	R	0x1	**Transmit FIFO Not Full**. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. **Value:** • 0x0 (FULL): Transmit FIFO is full • 0x1 (NOT_FULL): Transmit FIFO is not Full
0	SSI_BUSY	R	0x0	**SSI Busy Flag**.

Bits	Field Name	RW	Reset	Description
				When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. **Value:** • 0x0 (INACTIVE): SPI is idle or disabled • 0x1 (ACTIVE): SPI is actively transferring data

10.8.4.12 INT_MASK

- **Name:** Interrupt Mask Register
- **Description:** This read/write register masks or enables all interrupts generated by the SPI. When the SPI is configured as a slave device, the MULTI_M_CIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x2C
- **Reset Value:** master ? 0x0000003F: 0x0000001F

Table 10-132 Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MULTI_M_CIM	RW	0x1	Multi-Master Contention Interrupt Mask. **Value:** • 0x0 (MASKED): Disable mst_intr interrupt • 0x1 (UNMASKED): Enable mst_intr interrupt **Note:** When the SPI is configured as a slave, this register serves no purpose.
4	RX_FIFO_FIS	RW	0x1	Receive FIFO Full Interrupt Mask **Value:** • 0x0 (MASKED): Disable rxf_intr interrupt • 0x1 (UNMASKED): Enable rxf_intr interrupt
3	RX_FIFO_OIS	RW	0x1	Receive FIFO Overflow Interrupt Mask **Value:** • 0x0 (MASKED): Disable rxo_intr interrupt • 0x1 (UNMASKED): Enable rxo_intr interrupt
2	RX_FIFO_UIS	RW	0x1	Receive FIFO Underflow Interrupt Mask **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (MASKED): Disable rxu_intr interrupt • 0x1 (UNMASKED): Enable rxu_intr interrupt
1	TX_FIFO_OIS	RW	0x1	Transmit FIFO Overflow Interrupt Mask **Value:** • 0x0 (MASKED): Disable txo_intr interrupt • 0x1 (UNMASKED): Enable txo_intr interrupt
0	TX_FIFO_EIS	RW	0x1	Transmit FIFO Empty Interrupt Mask **Value:** • 0x0 (MASKED): Disable txe_intr interrupt • 0x1 (UNMASKED): Enable txe_intr interrupt

10.8.4.13 INT_STAT

- **Name:** Interrupt Status Register
- **Description:** This register reports the status of the SPI interrupts after they have been enabled.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 10-133 Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MULTI_M_CIS	R	0x0	Multi-Master Contention Interrupt Status **Value:** • 0x0 (INACTIVE): mst_intr interrupt is not active after being enabled • 0x1 (ACTIVE): mst_intr interrupt is active after being enabled
4	RX_FIFO_FIS	R	0x0	Receive FIFO Full Interrupt Status **Value:** • 0x0 (INACTIVE): rxf_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxf_intr interrupt is active after being enabled
3	RX_FIFO_OIS	R	0x0	Receive FIFO Overflow Interrupt Status **Value:** • 0x0 (INACTIVE): rxo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxo_intr interrupt is active after being enabled
2	RX_FIFO_UIS	R	0x0	Receive FIFO Underflow Interrupt Status

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (INACTIVE): rxu_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxu_intr interrupt is active after being enabled
1	TX_FIFO_OIS	R	0x0	Transmit FIFO Overflow Interrupt Status **Value:** • 0x0 (INACTIVE): txo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txo_intr interrupt is active after being enabled
0	TX_FIFO_EIS	R	0x0	Transmit FIFO Empty Interrupt Status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txe_intr interrupt is active after being enabled

10.8.4.14 RAW_INT_STAT

- **Name:** Raw Interrupt Status Register
- **Description:** This read-only register reports the status of the SPI interrupts prior to be enabled.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 10-134 Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MULTI_M_CRIS	R	0x0	Multi-Master Contention Raw Interrupt Status **Value:** • 0x0 (INACTIVE): mst_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): mst_intr interrupt is active prior to be enabled
4	RX_FIFO_FRIS	R	0x0	Receive FIFO Full Raw Interrupt Status **Value:** • 0x0 (INACTIVE): rxf_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): rxf_intr interrupt is active prior to be enabled
3	RX_FIFO_ORIS	R	0x0	Receive FIFO Overflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): rxo_intr interrupt is not active prior be enabled

Bits	Field Name	RW	Reset	Description
				• 0x1 (ACTIVE): rxo_intr interrupt is active prior to be enabled
2	RX_FIFO_URIS	R	0x0	Receive FIFO Underflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): rxu_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): rxu_intr interrupt is active prior to be enabled
1	TX_FIFO_ORIS	R	0x0	Transmit FIFO Overflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): txo_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): txo_intr interrupt is active prior be enabled
0	TX_FIFO_ERIS	R	0x0	Transmit FIFO Empty Raw Interrupt Status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): txe_intr interrupt is active prior be enabled

10.8.4.15 TX_FIFO_OIC

- **Name:** Transmit FIFO Overflow Interrupt Clear Register
- **Description:** Transmit FIFO Overflow Interrupt Clear Register
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 10-135 Transmit FIFO Overflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_FIFO_OIC	R	0x0	**Clear Transmit FIFO Overflow Interrupt.** This register reflects the status of the interrupt. A read from this register clears the txo_intr interrupt; writing has no effect.

10.8.4.16 RX_FIFO_OIC

- **Name:** Receive FIFO Overflow Interrupt Clear Register
- **Description:** Receive FIFO Overflow Interrupt Clear Register
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x3C

- **Reset Value:** 0x00000000

Table 10-136 Receive FIFO Overflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_FIFO_OIC	R	0x0	**Clear Receive FIFO Overflow Interrupt.** This register reflects the status of the interrupt. A read from this register clears the rxo_intr interrupt; writing has no effect.

10.8.4.17 RX_FIFO_UIC

- **Name:** Receive FIFO Underflow Interrupt Clear Register
- **Description:** Receive FIFO Underflow Interrupt Clear Register
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 10-137 Receive FIFO Underflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	MULTI_M_IC	R	0x0	**Clear Receive FIFO Underflow Interrupt.** This register reflects the status of the interrupt. A read from this register clears the rxu_intr interrupt; writing has no effect.

10.8.4.18 MULTI_M_IC

- **Name:** Multi-Master Interrupt Clear Register
- **Description:** Multi-Master Interrupt Clear Register
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 10-138 Multi-Master Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	MULTI_M_IC	R	0x0	**Clear Multi-Master Contention Interrupt**.

Bits	Field Name	RW	Reset	Description
				This register reflects the status of the interrupt. A read from this register clears the mst_intr interrupt; writing has no effect.

10.8.4.19 INT_CLR

- **Name:** Interrupt Clear Register
- **Description:** Interrupt Clear Register
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 10-139 Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	INT_CLR	R	0x0	**Clear Interrupts.** This register is set if any of the interrupts below are active. A read clears the txo_intr, rxu_intr, rxo_intr, and the mst_intr interrupts. Writing to this register has no effect.

10.8.4.20 DMA_CTRL

- **Name:** DMA Control Register
- **Description:** The register is used to enable the DMA Controller interface operation.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 10-140 DMA Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	TX_DMA_EN	RW	0x0	**Transmit DMA Enable**. This bit enables/disables the transmit FIFO DMA channel. **Value:** • 0x0 (DISABLED): Transmit DMA disabled • 0x1 (ENABLED): Transmit DMA enabled
0	RX_DMA_EN	RW	0x0	**Receive DMA Enable**.

Bits	Field Name	RW	Reset	Description
				This bit enables/disables the receive FIFO DMA channel. **Value:** • 0x0 (DISABLED): Receive DMA disabled • 0x1 (ENABLED): Receive DMA enabled

10.8.4.21 DMA_TX_DL

- **Name:** DMA Transmit Data Level
- **Description:** This register controls the threshold value for the transmit FIFO memory.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 10-141 DMA Transmit Data Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	DMA_TX_DL	RW	0x0	**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 TX_EN = 1.

10.8.4.22 DMA_RX_DL

- **Name:** DMA Receive Data Level
- **Description:** This register controls the threshold value for the receive FIFO memory.
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 10-142 DMA Receive Data Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	DMA_RX_DL	RW	0x0	**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

Bits	Field Name	RW	Reset	Description
				when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RX_EN=1.

10.8.4.23 DATA

- **Name:** Data Register
- **Description:** The SPI data register is a 32-bit read/write buffer for the transmit/receive FIFOs. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data is moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0
- **Base Address:** 0xA000C000 + x*0x100
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 10-143 Data Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	RW	0x0	Data Register. When writing to this register, you must right-justify the data. Read data is automatically right-justified.

10.8.5 Electrical Specifications

10.8.5.1 SPIM Electrical Specifications

Table 10-144 SPIM Electrical Specifications

Parameter	Description	Min	Typ	Max	Unit
f_{SCK}	Clock frequency			32	MHz
t_{CSCK}	SCK clock period	31.25			ns
t_{RSCK}	SCK rise time, 15 pF loading			$t_{RF,15pF}$ [1]	ns
t_{FSCK}	SCK fall time, 15 pF loading			$t_{RF,15pF}$ [1]	ns
t_{WSCKH}	SCK clock high time	7.625 [2]			ns
t_{WSCKL}	SCK clock low time	7.625 [2]			ns
t_{SUMI}	Data input setup time (MISO to SCK edge)	5			ns
t_{HMI}	Data input hold time (SCK edge to MISO change)	5			ns
t_{VMO}	Data output valid time (SCK edge to MOSI valid)	0			ns
t_{HMO}	MOSI hold time after CLK edge			20	ns

[1] The SCK rise time and the SCK fall time are from Figure 10-21.
[2] The SCK clock high time and the SCK clock low time are calculated by (t_{CSCK}/2)–t_{RSCK} and (t_{CSCK}/2)–t_{FSCK}, respectively.

The SPIM Timing Diagram is as below:

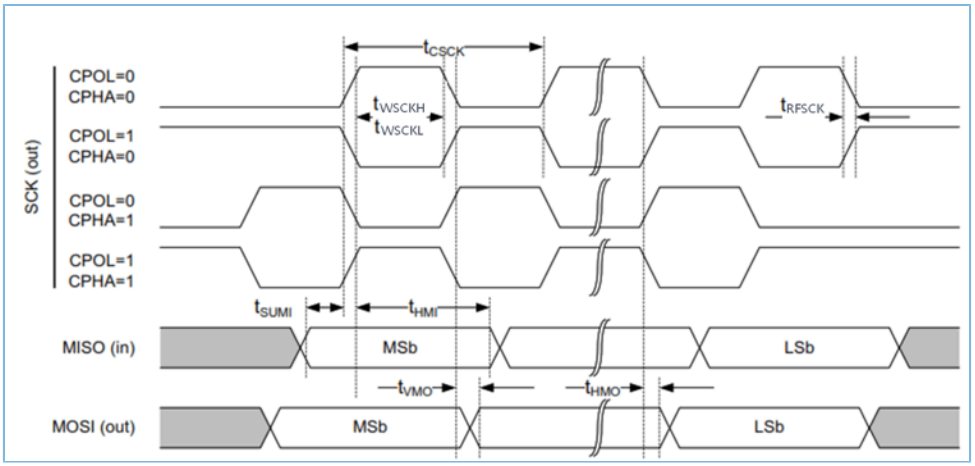

Figure 10-21 SPIM Timing Diagram

10.8.5.2 SPIS Electrical Specifications

Table 10-145 SPIS Electrical Specifications

Symbol	Description	Min.	Typ.	Max.	Unit
f_{SCK}	Clock frequency			10.67	MHz
t_{CSCK}	SCK input clock period	93.7			ns
t_{RFSCK}	SCK input clock rise/fall time			22	ns
t_{WSCKH}	SCK input clock high time	22			ns
t_{WSCKL}	SCK input clock low time	22			ns
t_{SUCSN}	CSN input setup time	62.5			ns
t_{HCSN}	CSN input hold time	31.25			ns
t_{ASO}	Data output access time (from CSN to MISO valid)			62.5	ns
t_{DISSO}	Data output disable time (from CSN to MISO disabled)			62.5	ns
t_{VSO}	Data output valid time (from SCK to MISO valid)			20	ns
t_{HSO}	Data output hold time (from SCK to MISO change)	8			ns
t_{SUSI}	Data input setup time (from MOSI to SCK)	18			ns
t_{HSI}	Data input hold time (from SCK to MOSI change)	0			ns

Note:

If SPIS performs RX and TX concurrently, Max bit rate will decrease to 5.33 Mbps.

SPIS Timing Diagram is shown below:

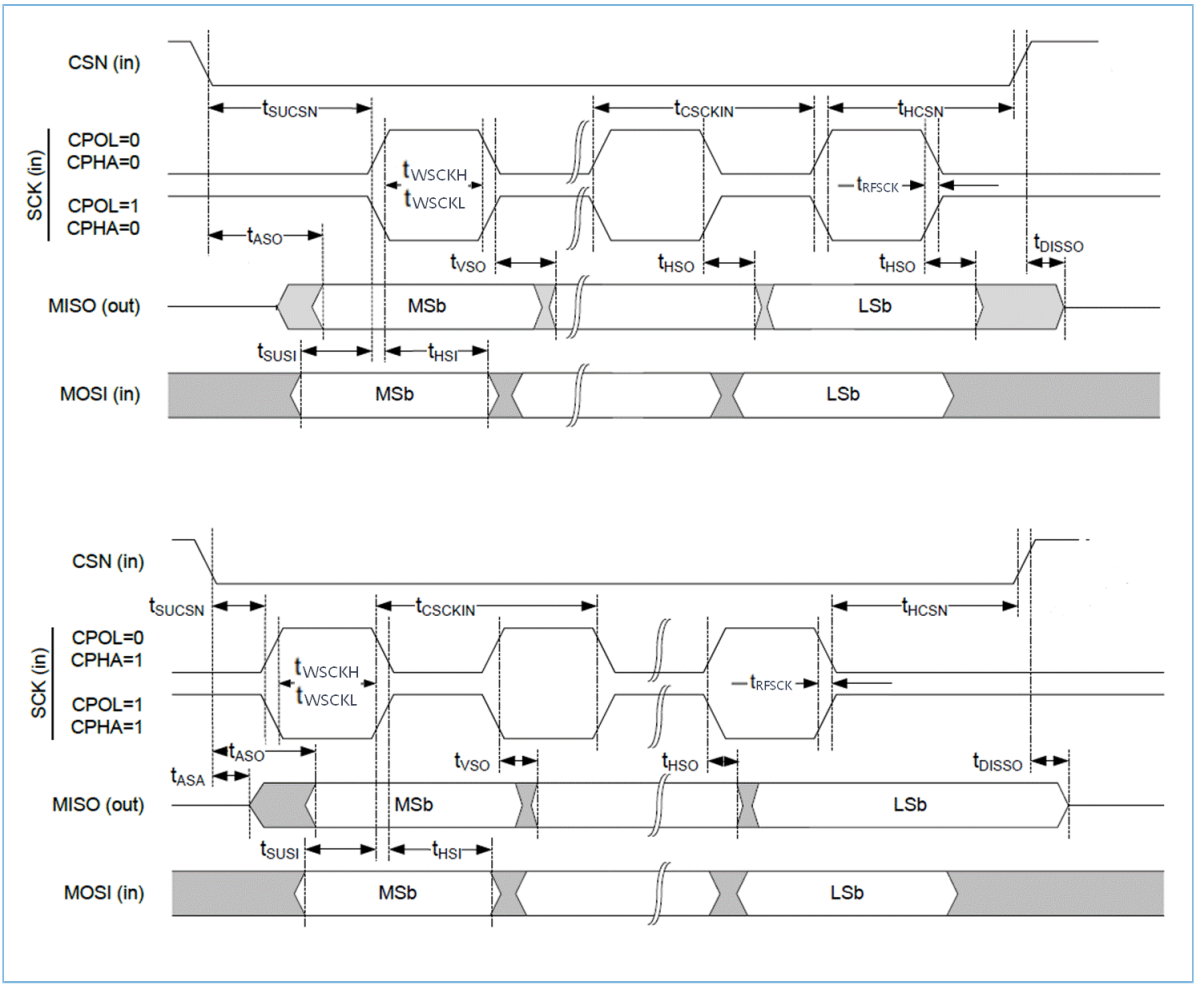

Figure 10-22 SPIS Timing Diagram

10.9 QSPI

10.9.1 Introduction

GR551x has two instances of Quad Serial Peripheral Interface (QSPI) Master, QSPI M0 and QSPI M1, which are used to interface to external Slave devices.

10.9.2 Main Features

- Motorola SPI support
- Single slave select line to support 1 slave.
- Built-in 8-byte RX/TX FIFOs for continuous SPI bursts.
- Transfer size up to 32 bits.
- Programmable output interface frequency up to 32 MHz.
- Supports SPI four modes (0, 1, 2, or 3) for different clock edge and phase configurations (Dual/Quad Frame Format only support mode0 and mode2).

- Maskable interrupt generation
- DMA Support

Note:

Refer to Section 10.9.5.1 QSPI Electrical Specifications for details.

10.9.3 Functional Description

Figure 10-23 depicts the QSPI Controller with the following functions and interfaces:

- APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt logic

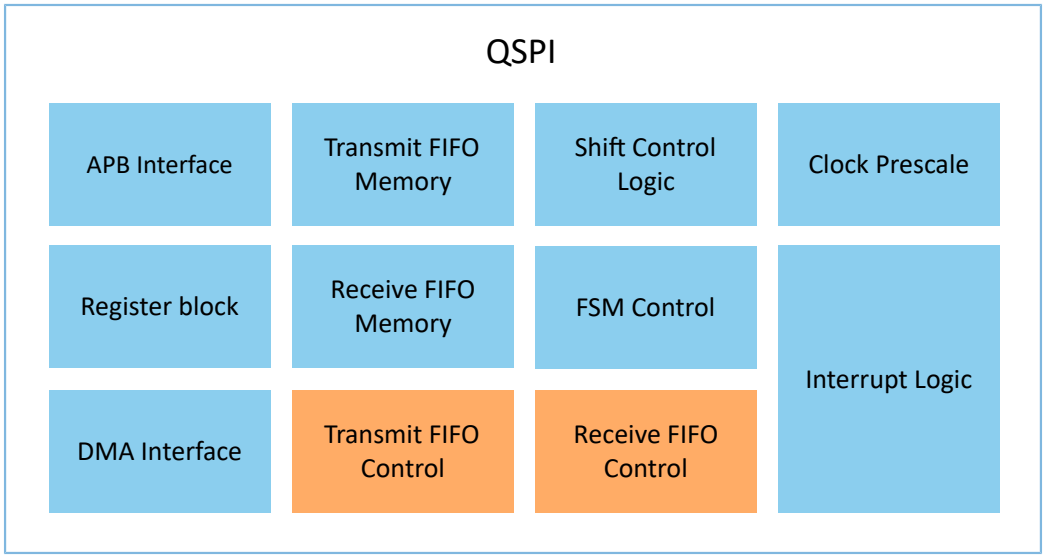

Figure 10-23 QSPI block diagram

10.9.3.1 Transmit and Receive FIFO Buffers

The width of both transmit and receive FIFO buffers is fixed at 32-bit. Data frames that are less than 32 bits must be right-justified when written into the transmit FIFO buffer. The shift control logic automatically right-justifies receive data in the receive FIFO buffer.

Each data entry in the FIFO buffers contains a single data frame. It is not possible to store multiple data frames in a single FIFO location; for example, you may not store two 8-bit data frames in a single FIFO location. If an 8-bit data frame is required, the upper 8-bit of the FIFO entry are ignored or unused when the serial shifter transmits the data.

The transmit FIFO is loaded by writing to the QSPI Controller data register (**DATA**). Data are popped from the transmit FIFO by the shift control logic into the transmit shift register. The transmit FIFO generates a FIFO empty interrupt request (**ssi_txe_intr**) when the number of entries in the FIFO is less than or equal to the FIFO threshold value. The threshold value, set through the programmable register **TX_FIFO_TL**, determines the level of FIFO entries at which an

interrupt is generated. The threshold value allows you to provide early indication to the processor that the transmit FIFO is nearly empty. A transmit FIFO overflow interrupt **(ssi_txo_intr**) is generated if you attempt to write data into an already full transmit FIFO.

Data are popped from the receive FIFO by reading the QSPI Controller data register (**DATA**). The receive FIFO is loaded from the receive shift register by the shift control logic. The receive FIFO generates a FIFO-full interrupt request (**ssi_rxf_intr**) when the number of entries in the FIFO is greater than or equal to the FIFO threshold value plus 1. The threshold value, set through programmable register **RX_FIFO_TL**, determines the level of FIFO entries at which an interrupt is generated.

The threshold value allows you to provide early indication to the processor that the receive FIFO is nearly full. A receive FIFO overrun interrupt (**ssi_rxo_intr**) is generated when the receive shift logic attempts to load data into a completely full receive FIFO. However, this newly received data is lost. A receive FIFO underflow interrupt (**ssi_rxu_intr**) is generated if you attempt to read from an empty receive FIFO. This alerts the processor that the read data is invalid.

10.9.3.2 QSPI Interrupts

The QSPI Controller supports individual interrupt requests, each of which can be masked.

The QSPI Controller interrupts are described as follows:

- Transmit FIFO Empty Interrupt (**ssi_txe_intr**) – Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of transmit FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is written into the transmit FIFO buffer, bringing it over the threshold level.
- Transmit FIFO Overflow Interrupt (**ssi_txo_intr**) – Set when an APB access attempts to write into the transmit FIFO after it has been completely filled. When set, data written from the APB is discarded. This interrupt remains set until you read the transmit FIFO overflow interrupt clear register (**TX_FIFO_OIC**).
- Receive FIFO Full Interrupt (**ssi_rxf_intr**) – Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of receive FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is read from the receive FIFO buffer, bringing it below the threshold level.
- Receive FIFO Overflow Interrupt (**ssi_rxo_intr**) – Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. When set, newly received data is discarded. This interrupt remains set until you read the receive FIFO overflow interrupt clear register (**RX_FIFO_OIC**).
- Receive FIFO Underflow Interrupt (**ssi_rxu_intr**) – Set when an APB access attempts to read from the receive FIFO when it is empty. When set, zeros are read back from the receive FIFO. This interrupt remains set until you read the receive FIFO underflow interrupt clear register (**RX_FIFO_UIC**).
- Multi-Master Contention Interrupt (**ssi_mst_intr**) –The interrupt is set when another serial master on the serial bus selects the QSPI Controller master as a serial-slave device and is actively transferring data. This informs the processor of possible contention on the serial bus. This interrupt remains set until you read the multi-master interrupt clear register (**MULTI_M_IC**).

10.9.3.3 Transfer Modes

When transferring data on the serial bus, the QSPI Controller operates in the modes discussed in this section. The transfer mode (**XFE_MODE**) is set by writing to control register 0 (**CTRL0**).

Note:

The transfer mode setting does not affect the duplex of the serial transfer. **XFE_MODE** is ignored for Microwire transfers, which are controlled by the **MW_CTRL** register.

10.9.3.3.1 Transmit

When XFE_MODE = 2‘b01, the receive data is invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data is popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. You should mask interrupts originating from the receive logic when this mode is entered.

10.9.3.3.2 Receive

When XFE_MODE = 2‘b10, the transmit data is invalid. When configured as a slave, the transmit FIFO is never popped in Receive Only mode. The txd output remains at a constant logic level during the transmission. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. You should mask interrupts originating from the transmit logic when this mode is entered.

10.9.3.4 Serial Master Operation

10.9.3.4.1 Data Transfers

Data transfers are started by the serial-master device. When the QSPI Controller is enabled (SSI_EN = 1), at least one valid data entry is present in the transmit FIFO and a serial-slave device is selected. When actively transferring data, the busy flag (**SSI_BUSY**) in the status register (**STAT**) is set. You must wait until the busy flag is cleared before attempting a new serial transfer.

10.9.3.4.2 Master SPI and SSP Serial Transfers

When the transfer mode is “transmit and receive” or “transmit only” (XFE_MODE = 2'b00 or XFE_MODE = 2'b01, respectively), transfers are terminated by the shift control logic when the transmit FIFO is empty. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data has been transmitted. The transmit FIFO threshold level (**TX_FIFO_TL**) can be used to early interrupt (**ssi_txe_intr**) the processor indicating that the transmit FIFO buffer is nearly empty. When a DMA is used for APB accesses, the transmit data level (**DMA_TX_DL**) can be used to early request (**dma_tx_req**) the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. The user may also write a block of data (at least two FIFO entries) into the transmit FIFO before enabling a serial slave. This ensures that serial

transmission does not begin until the number of data-frames that make up the continuous transfer are present in the transmit FIFO.

When the transfer mode is “receive only” (XFE_MODE = 2'b10), a serial transfer is started by writing one “dummy” data word into the transmit FIFO when a serial slave is selected. The txd output from the QSPI Controller is held at a constant logic level for the duration of the serial transfer. The transmit FIFO is popped only once at the beginning and may remain empty for the duration of the serial transfer. The end of the serial transfer is controlled by the “number of data frames” (**NUM_DATA_FRAME**) field in control register 1 (**CTRL1**).

If, for example, you want to receive 24 data frames from a serial-slave peripheral, you should program the **NUM_DATA_FRAME** field with the value 23; the receive logic terminates the serial transfer when the number of frames received is equal to the **NUM_DATA_FRAME** value + **1**. This transfer mode increases the bandwidth of the APB bus as the transmit FIFO never needs to be serviced during the transfer. The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow.

When the transfer mode is “eeprom_read” (XFE_MODE = 2‘b11), a serial transfer is started by writing the opcode and/or address into the transmit FIFO when a serial slave (EEPROM) is selected. The opcode and address are transmitted to the EEPROM device, after which read data is received from the EEPROM device and stored in the receive FIFO. The end of the serial transfer is controlled by the NUM_DATA_FRAME field in the control register 1 (**CTRL1**).

The receive FIFO threshold level (**RX_FIFO_TL**) can be used to give early indication that the receive FIFO is nearly full. When a DMA is used for APB accesses, the receive data level (**DMA_RX_DL**) can be used to early request (**dma_rx_req**) the DMA Controller, indicating that the receive FIFO is nearly full.

10.9.3.5 DMA Operation

QSPI Controller has a handshaking interface to a DMA Controller to request and control transfers. To enable the DMA Controller interface, User must write the DMA Control Register (DMA_CTRL). Writing a 1 into the TX_DMA_EN bit field of DMA_CTRL register enables the transmit handshaking interface. Writing a 1 into the RX_DMA_EN bit field of the DMA_CTRL register enables the receive handshaking interface.

10.9.3.5.1 Transmit Watermark Level

During QSPI serial transfers, transmit FIFO requests are made to the DMA Controller whenever the number of entries in the transmit FIFO is less than or equal to the DMA Transmit Data Level Register (**DMA_TX_DL**) value; this is known as the watermark level. Then the DMA responds by writing a burst of data to the transmit FIFO buffer, and the length of data is determined by **DEST_MSIZE** of DMA.

Data should be fetched from the DMA often enough for the transmit FIFO to perform serial transfers continuously; Otherwise the FIFO could run out of data (underflow). To avoid this situation, the user must set the watermark level correctly.

It is recommended to adhere to the following equation to configure the DMA transmit Operation for QSPI, which help to reduce the number of DMA bursts needed for a block transfer, and improve system utilization.

$$DMA.CTLx.DEST\_MSIZE = QSPI.FIFO\_DEPTH - QSPI.DMA\_TX\_DL \quad (3)$$

- **DMA.CTLx.DEST_MSIZE**, please refer to Section 10.9.4.20 DMA_CTRL.

- **QSPI.FIFO_DEPTH** equals 8, means the FIFO Depth for QSPI.
- **QSPI.DMA_TX_DL**, please refer to Section 10.9.4.21 DMA_TX_DL.

10.9.3.5.2 Receive Watermark Level

During QSPI serial transfers, receive FIFO requests are made to the DMA Controller whenever the number of entries in the receive FIFO is at or above the DMA Receive Data Level Register (**DMA_RX_DL**+1). This is known as the watermark level. The DMA Controller responds by fetching a burst of data from the receive FIFO buffer, and the length of data is determined by **SRC_MSIZE** of DMA.

Data should be fetched by the DMA often enough for the receive FIFO to accept serial transfers continuously. Otherwise, the FIFO will fill with data (overflow). To avoid this situation, the user must set the watermark level correctly.

It is recommended to adhere to the following equation to configure the DMA receive Operation for QSPI, which help to improve system utilization

$$DMA.CTLx.SRC\_MSIZE = QSPI.DMA\_RX\_DL + 1 \quad (4)$$

- **DMA.CTLx.SRC_MSIZE**, please refer to Section 10.9.4.20 DMA_CTRL.
- **QSPI.DMA_RX_DL**, please refer to Section 10.9.4.22 DMA_RX_DL.

10.9.4 Registers

10.9.4.1 CTRL0

- **Name:** Control Register 0
- **Description:** This register controls the serial data transfer. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x00
- **Reset Value:** 0x01070000

Table 10-146 Control Register 0

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24	S_ST_EN	RW	0x1	**Slave Select Toggle Enable**. When operating in QSPI mode with clock phase (SERIAL_CLK_PHASE) set to 0, this register controls the behavior of the slave select line (CS) between data frames. If this register field is set to 1, the CS line will toggle between consecutive data frames, with the serial clock (SCLK) being held to its default value while CS is high; if this register field is set to 0 the CS will stay low and SCLK will run continuously for the duration of the transfer.

Bits	Field Name	RW	Reset	Description
23	RSVD	R		Reserved bits
22:21	SPI_FRAME_FORMAT	RW	0x0	**SPI Frame Format:** Selects data frame format for Transmitting/Receiving the data bits. **Value:** • 0x0 (STD_SPI_FRAME_FORMAT): Standard SPI Frame Format • 0x1 (DUAL_SPI_FRAME_FORMAT): Dual SPI Frame Format • 0x2 (QUAD_SPI_FRAME_FORMAT): Quad SPI Frame Format • 0x3: Reserved
20:16	DATA_FRAME_SIZE	RW	0x7	**Data Frame Size** in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. When the data frame size is programmed to be less than 32 bits, the receive data is automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. **Note:** When SPI_FRAME_FORMAT is not set to 0x0, • DFS value should be multiple of 2 if SPI_FRAME_FORMAT = 0x01, • DFS value should be multiple of 4 if SPI_FRAME_FORMAT = 0x10. **Value:** • 0x3 (FRAME_04BITS): 4-bit serial data transfer • 0x4 (FRAME_05BITS): 5-bit serial data transfer • 0x5 (FRAME_06BITS): 6-bit serial data transfer • 0x6 (FRAME_07BITS): 7-bit serial data transfer • 0x7 (FRAME_08BITS): 8-bit serial data transfer • 0x8 (FRAME_09BITS): 9-bit serial data transfer • 0x9 (FRAME_10BITS): 10-bit serial data transfer • 0xa (FRAME_11BITS): 11-bit serial data transfer • 0xb (FRAME_12BITS): 12-bit serial data transfer • 0xc (FRAME_13BITS): 13-bit serial data transfer • 0xd (FRAME_14BITS): 14-bit serial data transfer • 0xe (FRAME_15BITS): 15-bit serial data transfer • 0xf (FRAME_16BITS): 16-bit serial data transfer • 0x10 (FRAME_17BITS): 17-bit serial data transfer

Bits	Field Name	RW	Reset	Description
				• 0x11 (FRAME_18BITS): 18-bit serial data transfer • 0x12 (FRAME_19BITS): 19-bit serial data transfer • 0x13 (FRAME_20BITS): 20-bit serial data transfer • 0x14 (FRAME_21BITS): 21-bit serial data transfer • 0x15 (FRAME_22BITS): 22-bit serial data transfer • 0x16 (FRAME_23BITS): 23-bit serial data transfer • 0x17 (FRAME_24BITS): 24-bit serial data transfer • 0x18 (FRAME_25BITS): 25-bit serial data transfer • 0x19 (FRAME_26BITS): 26-bit serial data transfer • 0x1a (FRAME_27BITS): 27-bit serial data transfer • 0x1b (FRAME_28BITS): 28-bit serial data transfer • 0x1c (FRAME_29BITS): 29-bit serial data transfer • 0x1d (FRAME_30BITS): 30-bit serial data transfer • 0x1e (FRAME_31BITS): 31-bit serial data transfer • 0x1f (FRAME_32BITS): 32-bit serial data transfer
15:12	CTRL_FRAME_SIZE	RW	0x0	Control Frame Size. Selects the length of the control word for the Microwire frame format. **Value:** • 0x0 (SIZE_01_BIT): 1-bit Control Word • 0x1 (SIZE_02_BIT): 2-bit Control Word • 0x2 (SIZE_03_BIT): 3-bit Control Word • 0x3 (SIZE_04_BIT): 4-bit Control Word • 0x4 (SIZE_05_BIT): 5-bit Control Word • 0x5 (SIZE_06_BIT): 6-bit Control Word • 0x6 (SIZE_07_BIT): 7-bit Control Word • 0x7 (SIZE_08_BIT): 8-bit Control Word • 0x8 (SIZE_09_BIT): 9-bit Control Word • 0x9 (SIZE_10_BIT): 10-bit Control Word • 0xa (SIZE_11_BIT): 11-bit Control Word • 0xb (SIZE_12_BIT): 12-bit Control Word • 0xc (SIZE_13_BIT): 13-bit Control Word • 0xd (SIZE_14_BIT): 14-bit Control Word • 0xe (SIZE_15_BIT): 15-bit Control Word

Bits	Field Name	RW	Reset	Description
				• 0xf (SIZE_16_BIT): 16-bit Control Word
11	SHIFT_REG_LOOP	RW	0x0	**Shift Register Loop**. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. **Value:** • 0x0 (NORMAL_MODE): Normal mode operation • 0x1 (TESTING_MODE): Test mode: TX & RX shift register connected
10	RSVD	R		Reserved bits
9:8	XFE_MODE	RW	0x0	**Transfer Mode**. This transfer mode is only valid when the SPI is configured as master device. • 0x0 - Transmit & Receive • 0x1 - Transmit Only • 0x2 - Receive Only • 0x3 - EEPROM Read When SPI_FRAME_FORMAT is not set to 2'b00. There are only two valid combinations: • 0x1 - Write • 0x2 - Read **Value:** • 0x0 (TX_AND_RX): Transmit & receive • 0x1 (TX_ONLY): Transmit only mode or Write (SPI_FRAME_FORMAT != 0x0) • 0x2 (RX_ONLY): Receive only mode or Read (SPI_FRAME_FORMAT != 0x0) • 0x3 (EEPROM_READ): EEPROM Read mode
7	SERIAL_CLK_POL	RW	0x0	**Serial Clock Polarity**. Valid when the frame format (FRAME_FORMAT) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. **Value:** • 0x0 (SCLK_LOW): Inactive state of serial clock is low • 0x1 (SCLK_HIGH): Inactive state of serial clock is high
6	SERIAL_CLK_PHASE	RW	0x0	**Serial Clock Phase**.

Bits	Field Name	RW	Reset	Description
				Valid when the frame format (FRAME_FORMAT) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data is captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data is captured on the second edge of the serial clock. **Value:** • 0x0 (SERIAL_CLK_PHASE_MIDDLE): Serial clock toggles in middle of first data bit • 0x1 (SERIAL_CLK_PHASE_START): Serial clock toggles at start of first data bit
5:4	FRAME_FORMAT	RW	0x0	**Frame Format.** Selects which serial protocol transfers the data. **Value:** • 0x0 (MOTOROLA_SPI): Motorola SPI Frame Format • 0x1 (TEXAS_SSP): Texas Instruments SSP Frame Format • 0x2 (NS_MICROWIRE): National Microwire Frame Format • 0x3 (RESERVED): Reserved value
3:0	RSVD	R		Reserved bits

10.9.4.2 CTRL1

- **Name:** Control Register 1
- **Description:** This register exists only when the SPI is configured as a master device. Control register 1 controls the end of serial transfers when in receive-only mode. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x04
- **Reset Value:** 0x00000000

Table 10-147 Control Register 1

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	NUM_DATA_FRAME	RW	0x0	**Number of Data Frames**. When XFE_MODE = 0x2 or XFE_MODE = 0x3, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is

Bits	Field Name	RW	Reset	Description
				equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer. **Note:** When the SPI is configured as a slave, this register serves no purpose.

10.9.4.3 SSI_EN

- **Name:** SSI Enable Register
- **Description:** This register enables and disables the SPI.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x08
- **Reset Value:** 0x00000000

Table 10-148 SSI Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SSI_EN	RW	0x0	**SSI Enable**. Enables and disables all QSPI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. **Value:** • 0x0 (DISABLED): Disables Serial Transfer • 0x1 (ENABLED): Enables Serial Transfer

10.9.4.4 MW_CTRL

- **Name:** Microwire Control Register
- **Description:** This register controls the direction of the data word for the half-duplex Microwire serial protocol. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 10-149 Microwire Control Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
2	MW_HSG	RW	0x0	**Microwire Handshaking**. Used to enable and disable the busy/ready handshaking interface for the Microwire protocol. When enabled, the SPI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the SSI_BUSY status in the STAT register. **Value:** • 0x0 (DISABLE): Handshaking interface is disabled • 0x1 (ENABLED): Handshaking interface is enabled **Note:** When the SPI is configured as a slave, this register serves no purpose.
1	MW_DIR_DW	RW	0x0	**Microwire Control**. Defines the direction of the data word when the Microwire serial protocol is used. **Value:** • 0x0 (RECEIVE): SPI receives data • 0x1 (TRANSMIT): SPI transmits data
0	MW_XFE_MODE	RW	0x0	**Microwire Transfer Mode**. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. **Value:** • 0x0 (NON_SEQUENTIAL): Non-Sequential Microwire Transfer • 0x1 (SEQUENTIAL): Sequential Microwire Transfer

10.9.4.5 S_EN

- **Name:** Slave Enable Register
- **Description:** This register exists only when the SPI is configured as a master device. The register enables the individual slave select output lines from the SPI master.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 10-150 Slave Enable Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	S_EN	RW	0x0	**Slave Select Enable**. Each bit in this register corresponds to a slave select line (CSn) from the SPI master. **Value:** • 0x0 (NOT_SELECTED): No slave selected • 0x1 (SELECTED): Slave is selected **Note:** When the SPI is configured as a slave, this register serves no purpose.

10.9.4.6 BAUD

- **Name:** Baud Rate Register
- **Description:** This register exists only when the SPI is configured as a master device. The register derives the frequency of the serial clock that regulates the data transfer. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 10-151 Baud Rate Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	SSI_CLK_DIV	RW	0x0	**SSI Clock Divider**. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: $F_{sclk\_out} = F_{ssi\_clk}/SSI\_CLK\_DIV$ **Note:** When the SPI is configured as a slave, this register serves no purpose.

10.9.4.7 TX_FIFO_TL

- **Name:** Transmit FIFO Threshold Level

- **Description:** This register controls the threshold value for the transmit FIFO memory.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 10-152 Transmit FIFO Threshold Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	TX_FIFO_THD	RW	0x0	**Transmit FIFO Threshold**. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2 – 256; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

10.9.4.8 RX_FIFO_TL

- **Name:** Receive FIFO Threshold Level
- **Description:** This register controls the threshold value for the receive FIFO memory.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 10-153 Receive FIFO Threshold Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	RX_FIFO_THD	RW	0x0	**Receive FIFO Threshold**. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2 – 256. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

10.9.4.9 TX_FIFO_LEVEL

- **Name:** Transmit FIFO Level Register
- **Description:** This register contains the number of valid data entries in the transmit FIFO memory.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 10-154 Transmit FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	TX_FIFO_LEVEL	R	0x0	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO.

10.9.4.10 RX_FIFO_LEVEL

- **Name:** Receive FIFO Level Register
- **Description:** This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 10-155 Receive FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	RX_FIFO_LEVEL	R	0x0	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO.

10.9.4.11 STAT

- **Name:** Status Register
- **Description:** This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x28
- **Reset Value:** 0x00000006

Table 10-156 Status Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6	DATA_COLN_ERR	R	0x0	**Data Collision Error**. This bit will be set if MISO input is asserted by other master, when the SPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read.
5	TX_ERR	R	0x0	**Transmission Error**. Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the txd line. This bit is cleared when read. **Value:** • 0x0 (NO_ERROR): No Error • 0x1 (TX_ERROR): Transmission Error **Note:** When the SPI is configured as a master, this register serves no purpose.
4	RX_FIFO_FULL	R	0x0	**Receive FIFO 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. **Value:** • 0x0 (NOT_FULL): Receive FIFO is not full • 0x1 (FULL): Receive FIFO is full
3	RX_FIFO_NE	R	0x0	**Receive FIFO Not Empty**. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. **Value:** • 0x0 (EMPTY): Receive FIFO is empty • 0x1 (NOT_EMPTY): Receive FIFO is not empty
2	TX_FIFO_EMPTY	R	0x1	**Transmit FIFO Empty**. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. **Value:** • 0x0 (NOT_EMPTY): Transmit FIFO is not empty • 0x1 (EMPTY): Transmit FIFO is empty

Bits	Field Name	RW	Reset	Description
1	TX_FIFO_NF	R	0x1	**Transmit FIFO Not Full**. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. **Value:** • 0x0 (FULL): Transmit FIFO is full • 0x1 (NOT_FULL): Transmit FIFO is not Full
0	SSI_BUSY	R	0x0	**SSI Busy Flag**. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. **Value:** • 0x0 (INACTIVE): SPI is idle or disabled • 0x1 (ACTIVE): SPI is actively transferring data

10.9.4.12 INT_MASK

- **Name:** Interrupt Mask Register
- **Description:** This read/write register masks or enables all interrupts generated by the SPI. When the SPI is configured as a slave device, the MULTI_M_CIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x2C
- **Reset Value:** 0x0000003F

Table 10-157 Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MULTI_M_CIM	RW	0x1	Multi-Master Contention Interrupt Mask. **Value:** • 0x0 (MASKED): Disable mst_intr interrupt • 0x1 (UNMASKED): Enable mst_intr interrupt
4	RX_FIFO_FIS	RW	0x1	Receive FIFO Full Interrupt Mask **Value:** • 0x0 (MASKED): Disable rxf_intr interrupt • 0x1 (UNMASKED): Enable rxf_intr interrupt
3	RX_FIFO_OIS	RW	0x1	Receive FIFO Overflow Interrupt Mask

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (MASKED): Disable rxo_intr interrupt • 0x1 (UNMASKED): Enable rxo_intr interrupt
2	RX_FIFO_UIS	RW	0x1	Receive FIFO Underflow Interrupt Mask **Value:** • 0x0 (MASKED): Disable rxu_intr interrupt • 0x1 (UNMASKED): Enable rxu_intr interrupt
1	TX_FIFO_OIS	RW	0x1	Transmit FIFO Overflow Interrupt Mask **Value:** • 0x0 (MASKED): Disable txo_intr interrupt • 0x1 (UNMASKED): Enable txo_intr interrupt
0	TX_FIFO_EIS	RW	0x1	Transmit FIFO Empty Interrupt Mask **Value:** • 0x0 (MASsKED): Disable txe_intr interrupt • 0x1 (UNMASKED): Enable txe_intr interrupt

10.9.4.13 INT_STAT

- **Name:** Interrupt Status Register
- **Description:** This register reports the status of the SPI interrupts after they have been enabled.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 10-158 Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MULTI_M_CIS	R	0x0	Multi-Master Contention Interrupt Status **Value:** • 0x0 (INACTIVE): mst_intr interrupt is not active after being enabled • 0x1 (ACTIVE): mst_intr interrupt is full after being enabled
4	RX_FIFO_FIS	R	0x0	Receive FIFO Full Interrupt Status **Value:** • 0x0 (INACTIVE): rxf_intr interrupt is not active after being enabled

Bits	Field Name	RW	Reset	Description
				• 0x1 (ACTIVE): rxf_intr interrupt is full after being enabled
3	RX_FIFO_OIS	R	0x0	Receive FIFO Overflow Interrupt Status **Value:** • 0x0 (INACTIVE): rxo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxo_intr interrupt is active after being enabled
2	RX_FIFO_UIS	R	0x0	Receive FIFO Underflow Interrupt Status **Value:** • 0x0 (INACTIVE): rxu_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxu_intr interrupt is active after being enabled
1	TX_FIFO_OIS	R	0x0	Transmit FIFO Overflow Interrupt Status **Value:** • 0x0 (INACTIVE): txo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txo_intr interrupt is active after being enabled
0	TX_FIFO_EIS	R	0x0	Transmit FIFO Empty Interrupt Status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txe_intr interrupt is active after being enabled

10.9.4.14 RAW_INT_STAT

- **Name:** Raw Interrupt Status Register
- **Description:** This read-only register reports the status of the SPI interrupts prior to be enabled.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 10-159 Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MULTI_M_CRIS	R	0x0	Multi-Master Contention Raw Interrupt Status **Value:** • 0x0 (INACTIVE): mst_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): mst_intr interrupt is full prior to be enabled
4	RX_FIFO_FRIS	R	0x0	Receive FIFO Full Raw Interrupt Status **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (INACTIVE): rxf_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): rxf_intr interrupt is active prior to be enabled
3	RX_FIFO_ORIS	R	0x0	Receive FIFO Overflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): rxo_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): rxo_intr interrupt is active prior be enabled
2	RX_FIFO_URIS	R	0x0	Receive FIFO Underflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): rxu_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): rxu_intr interrupt is active prior to be enabled
1	TX_FIFO_ORIS	R	0x0	Transmit FIFO Overflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): txo_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): txo_intr interrupt is active prior be enabled
0	TX_FIFO_ERIS	R	0x0	Transmit FIFO Empty Raw Interrupt Status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active prior to be enabled • 0x1 (ACTIVE): txe_intr interrupt is active prior be enabled

10.9.4.15 TX_FIFO_OIC

- **Name:** Transmit FIFO Overflow Interrupt Clear Register.
- **Description:** Transmit FIFO Overflow Interrupt Clear Register.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 10-160 Transmit FIFO Overflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_FIFO_OIC	R	0x0	**Clear Transmit FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the txo_intr interrupt; writing has no effect.

10.9.4.16 RX_FIFO_OIC

- **Name:** Receive FIFO Overflow Interrupt Clear Register
- **Description:** Receive FIFO Overflow Interrupt Clear Register.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 10-161 Receive FIFO Overflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_FIFO_OIC	R	0x0	**Clear Receive FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the rxo_intr interrupt; writing has no effect.

10.9.4.17 RX_FIFO_UIC

- **Name:** Receive FIFO Underflow Interrupt Clear Register
- **Description:** Receive FIFO Underflow Interrupt Clear Register.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 10-162 Receive FIFO Underflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_FIFO_UIC	R	0x0	**Clear Receive FIFO Underflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the rxu_intr interrupt; writing has no effect.

10.9.4.18 MULTI_M_IC

- **Name:** Multi-Master Interrupt Clear Register
- **Description:** Multi-Master Interrupt Clear Register
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 10-163 Multi-Master Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	MULTI_M_IC	R	0x0	**Clear Multi-Master Contention Interrupt**. This register reflects the status of the interrupt. A read from this register clears the mst_intr interrupt; writing has no effect.

10.9.4.19 INT_CLR

- **Name:** Interrupt Clear Register
- **Description:** Interrupt Clear Register
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 10-164 Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	INT_CLR	R	0x0	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the txo_intr, rxu_intr, rxo_intr, and the mst_intr interrupts. Writing to this register has no effect.

10.9.4.20 DMA_CTRL

- **Name:** DMA Control Register
- **Description:** The register is used to enable the DMA Controller interface operation.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 10-165 DMA Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	TX_DMA_EN	RW	0x0	**Transmit DMA Enable**. This bit enables/disables the transmit FIFO DMA channel.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (DISABLED): Transmit DMA disabled • 0x1 (ENABLED): Transmit DMA enabled
0	RX_DMA_EN	RW	0x0	**Receive DMA Enable**. This bit enables/disables the receive FIFO DMA channel **Value:** • 0x0 (DISABLED): Receive DMA disabled • 0x1 (ENABLED): Receive DMA enabled

10.9.4.21 DMA_TX_DL

- **Name:** DMA Transmit Data Level
- **Description:** This register controls the threshold value for the transmit FIFO memory.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 10-166 DMA Transmit Data Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	DMA_TX_DL	RW	0x0	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 TX_DMA_EN = 1.

10.9.4.22 DMA_RX_DL

- **Name:** DMA Receive Data Level
- **Description:** This register controls the threshold value for the receive FIFO memory
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 10-167 DMA Receive Data Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	DMA_RX_DL	RW	0x0	**Receive Data Level**. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMA_RX_DL +1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RX_DMA_EN =1.

10.9.4.23 DATA

- **Name:** Data Register
- **Description:** The SPI data register is a 32-bit read/write buffer for the transmit/receive FIFOs. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data is moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 10-168 Data Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	RW	0x0	Data Register. When writing to this register, you must right-justify the data. Read data is automatically right-justified.

10.9.4.24 RX_SMP_DLY

- **Name:** RX Sample Delay Register
- **Description:** This register controls the number of ssi_clk cycles that are delayed (from the default sample time) before the actual sample of the rxd input occurs. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0xF0
- **Reset Value:** 0x00000000

Table 10-169 RX Sample Delay Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	RX_SMP_DLY	RW	0x0	**RX Sample Delay**.

Bits	Field Name	RW	Reset	Description
				This register is used to delay the sample of the rxd input port. Each value represents a single ssi_clk delay on the sample of rxd. The maximum value is 7. **Note:** When the SPI is configured as a slave, this register serves no purpose.

10.9.4.25 SPI_CTRL

- **Name:** SPI Control Register
- **Description:** This register is used to control the serial data transfer in SPI mode of operation. The register is only relevant when SPI_FRAME_FORMAT is set to either 01 or 10. It is not possible to write to this register when the SPI is enabled (SSI_EN=1).
- **Base Address:** 0xA000C200 + x*0x600
- **Offset:** 0xF4
- **Reset Value:** 0x00000200

Table 10-170 SPI Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:11	WAIT_CYCLES	RW	0x0	**Wait cycles** Number of wait cycles in Dual/Quad mode between control frames transmit and data reception. This value is specified as number of SPI clock cycles.
10	RSVD	R		Reserved bits
9:8	INST_LEN	RW	0x2	**Instruction Length** Dual/Quad mode instruction length in bits. **Value:** • 0x0 (INST_LEN_0): 0-bit (No Instruction) • 0x1 (INST_LEN_1): 4-bit Instruction • 0x2 (INST_LEN_2): 8-bit Instruction • 0x3 (INST_LEN_3): 16-bit Instruction
7:6	RSVD	R		Reserved bits
5:2	ADDR_LEN	RW	0x0	**Address Length**. This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (ADDR_LEN_0): 0-bit Address Width • 0x1 (ADDR_LEN_1): 4-bit Address Width • 0x2 (ADDR_LEN_2): 8-bit Address Width • 0x3 (ADDR_LEN_3): 12-bit Address Width • 0x4 (ADDR_LEN_4): 16-bit Address Width • 0x5 (ADDR_LEN_5): 20-bit Address Width • 0x6 (ADDR_LEN_6): 24-bit Address Width • 0x7 (ADDR_LEN_7): 28-bit Address Width • 0x8 (ADDR_LEN_8): 32-bit Address Width • 0x9 (ADDR_LEN_9): 36-bit Address Width • 0xa (ADDR_LEN_10): 40-bit Address Width • 0xb (ADDR_LEN_11): 44-bit Address Width • 0xc (ADDR_LEN_12): 48-bit Address Width • 0xd (ADDR_LEN_13): 52-bit Address Width • 0xe (ADDR_LEN_14): 56-bit Address Width • 0xf (ADDR_LEN_15): 60-bit Address Width
1:0	XFE_FORMAT_AI	RW	0x0	**Address and instruction transfer format**. Selects whether QSPI will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRL0. SPI_FRAME_FORMAT field. **Value:** • 0x0 - Instruction and Address will be sent in Standard SPI Mode. • 0x1 - Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRL0. SPI_FRAME_FORMAT. • 0x2 - Both Instruction and Address will be sent in the mode specified by SPI_FRAME_FORMAT. • 0x3 - Reserved.

10.9.5 Electrical Specifications

10.9.5.1 QSPI Electrical Specifications

Table 10-171 QSPI Electrical Specifications

Parameter	Description	Min	Typ	Max	Units
f_{SCK}	Clock frequency			32	MHz
t_{CSCK}	SCK clock period	31.25			ns

Parameter	Description	Min	Typ	Max	Units
t_{RSCK}	SCK rise time, 15 pF loading			$t_{RF,15pF}$[1]	ns
t_{FSCK}	SCK fall time, 15 pF loading			$t_{RF,15pF}$[1]	ns
t_{WSCKH}	SCK clock high time	7.625[2]			ns
t_{WSCKL}	SCK clock low time	7.625[2]			ns
t_{SUMI}	Data input setup time (MISO to SCK edge)	0			ns
t_{HMI}	Data input hold time (SCK edge to MISO change)	8			ns
t_{VMO}	Data output valid time (SCK edge to MOSI valid)	0			ns
t_{HMO}	MOSI hold time after CLK edge			20	ns

[1] The SCK rise time and the SCK fall time are from Figure 10-24.
[2] The SCK clock high time and the SCK clock low time are calculated by (t_{CSCK}/2)–t_{RSCK} and (t_{CSCK}/2)–t_{FSCK}, respectively.

The QSPI Timing Diagram is as below:

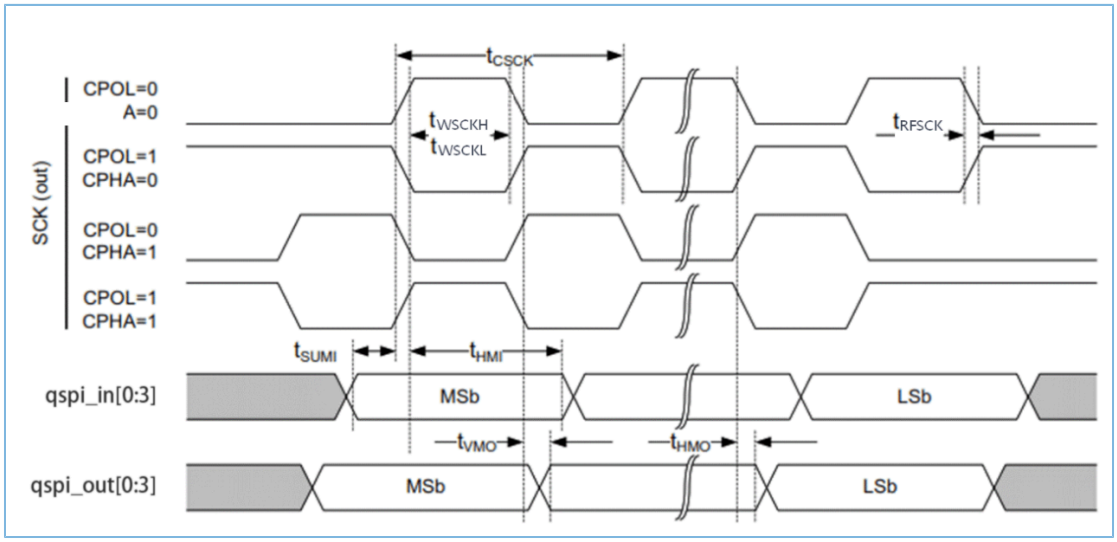

Figure 10-24 QSPI Timing Diagram

10.10 Execute-in-place QSPI (XQSPI)

10.10.1 Introduction

GR551x has a Quad Serial Peripheral Interface (QSPI) module with XIP (Execute-In-Place) support used to interface to an external Flash device that may host executable program code or data. A cache controller with 8 KB of cache memory is used to reduce the power consumption of the external Flash access and reduce the wait states for instruction read when the code is in the cache memory.

10.10.2 Main Features

- XIP read on QSPI to support fast boot and code execution from Flash
- Flash cache memory to optimize power and performance

- AHB slave interface for XIP read transactions
- APB slave interface for configuration
- Quad SPI interface to connect to a stacked FLASH device within the package
- Normal/Fast, Dual/Quad Read modes in XIP
- Flash Erase/Programing when not in XIP

10.10.3 Functional Description

XQSPI block assists the MCU with fast boot up and code execution. It consists of a Flash cache controller and a QSPI XIP Flash controller which is capable of XIP reads from external NOR Flash devices during cache misses. XIP is a way that allows an AHB master (in this case, the MCU) to directly read the contents of Flash devices simply by reading from the address space of the XQSPI controller. During XIP read, cache line is also refilled at Least Frequently Used (LFU) location to reduce future chance of cache miss. The block top level contains three major interfaces:

XQSPI supports common industry-standard QSPI FLASH devices: Winbond W25Q, Micron N25Q, Spansion S25FL, and Macronix MX25L in addition to GD25VQ and GD25LQ families, as well as Gigadevice and PUYA devices.

When XIP is enabled, QSPI can do normal read, fast read, fast read dual out, fast read quad out, fast read dual I/O and fast read quad I/O through AHB bus read transactions. This is code execution path from Flash memory. Cache can greatly reduce the read latency if the code snippet can be found in the cache tag memory, thus improving the performance.

With XIP disabled, CPU can erase and program the FLASH memory through APB bus. This is usually done one-time only or in code maintenance phase and patching.

10.10.4 XIP_Cache Registers

10.10.4.1 CACHE_CTRL0

- **Name:** XQSPI cache Control 0 Register
- **Description:** This register performs the main control of XIP cache.
- **Base Address:** 0xA000D000
- **Offset:** 0x00
- **Reset Value:** 0x00000011

Table 10-172 Control 0 Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4	HIT_MISS_CLR	RW	0x1	Clear Hit/Miss Counters **Value:** • 0x0: Hit/Miss Counters in Normal Mode • 0x1: Hit/Miss Counters in Clear Mode

Bits	Field Name	RW	Reset	Description
3	FIFO_CLR	RW	0x0	Clear LFU FIFO **Value:** • 0x0: FIFO in Normal Mode • 0x1: FIFO in Clear Mode
2	Reserved	R		Reserved bits
1	TAG_FLUSH_EN	RW	0x0	Enable Tag Memory Flush Tag memory will get flushed unless tag_ret is enabled (Tag_ret signal lives in CPU register space). **Value:** • 0x0: Tag memory Flush is disabled. • 0x1: Tag memory Flush is enabled.
0	CACHE_EN	RW	0x1	**Value:** • 0x0: Cache enabled. • 0x1: Cache disabled.

10.10.4.2 CACHE_CRTL1

- **Name:** XQSPI cache Control 1 Register
- **Description:** This register is used to configure Bus Write/Read.
- **Base Address:** 0xA000D000
- **Offset:** 0x04
- **Reset Value:** 0x00000110

Table 10-173 Control 1 Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	BUS_MUX_EN	RW	0x0	**Value:** • 0x0: enable • 0x1: disable
10:8	BUS_MUX_SRC	RW	0x1	Select 1 of 8 debug-bus configurations. Debug-bus is 8-bit bus.
7:4	BIAS_TRIM	RW	0x1	Bias TRIM signal
3:2	RW_MARGIN_CTRL	RW	0x0	Read and write margin control
1	WR_MARGIN	RW	0x0	Write margin

Bits	Field Name	RW	Reset	Description
0	RD_MARGIN	RW	0x0	Read margin

10.10.4.3 CACHE_HIT_COUNT

- **Name:** XQSPI cache Hits Counter Register
- **Description:** Cache Hits counter
- **Base Address:** 0xA000D000
- **Offset:** 0x08
- **Reset Value:** 0x00000000

Table 10-174 Hits Counter Register

Bits	Field Name	RW	Reset	Description
31:0	HIT_COUNT	R	0x0	Cache hit counter

10.10.4.4 CACHE_MISS_COUNT

- **Name:** XQSPI cache Miss Counter Register
- **Description:** Cache Miss counter
- **Base Address:** 0xA000D000
- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 10-175 Miss Counter Register

Bits	Field Name	RW	Reset	Description
31:0	MISS_COUNT	R	0x0	Cache miss counter

10.10.4.5 CACHE_STAT

- **Name:** XQSPI cache Status Register
- **Description:** This is a read-only register used to indicate the current tag flush status.
- **Base Address:** 0xA000D000
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 10-176 Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	R	0x0	Tag_flush_busy status. **Value:** • 0x0: Idle • 0x1: Busy

10.11 Sense Analog-to-Digital Converter (SADC)

10.11.1 Introduction

An 8-channel sense 13-bit 1 Msps ADC that can be used for single ended or for differential measurements. The sense ADC has DMA support to allow for large sample collection without overloading the MCU.

10.11.2 Main Features

- 13-bit Analog-to-Digital Conversion
- 8-channel Analog-to-Digital Conversion:
  - channel0 to channel4 used as analog input
  - channel5 used as temperature sensor input
  - channel6 used as battery sensor input
  - channel7 used as internal calibration (connected to internal reference)
- Up to 16 MHz ADC clock frequency
- Up to 1 Msps conversion rate
- Single-ended or differential measurements
- Internal Temperature and Vbattery Sensors
- Ability to capture ADC samples using DMA, unburdening the MCU
- Programmable internal reference

10.11.3 Functional Description

SADC is a successive approximation register (SAR) based ADC. It can sample one of 8 analog sources at a given time, using an input mux. Two of the channels are used internally for VBAT and Temperature monitoring that can be used for adjusting the analog and RF circuitry configurations for optimum performance. SADC can be reconfigured to perform single-ended or differential measurements. The reference voltage can be internally generated or supplied as one of the external inputs. The internal reference voltage can be programmed as **0.85** V, **1.28** V and **1.6** V. For any

specific reference value (Vref), the maximum ADC input is supposed to be (**2 x Vref**). Other aspects of Analog-to-Digital conversion can be controlled by programming **SNSADC_CFG** register.

10.11.3.1 SADC Operation

- Pairs of successive 13-bit samples from SADC are combined into 32-bit words by 16-bit aligning the samples. 32-bit words are stored into an SADC FIFO of 64 32-bit entries, where each entry would hold 2 samples.
- When the SADC FIFO entry count reaches the 6-bit threshold value specified in **FIFO_THD** register, the samples stored in the FIFO are automatically streamed to the SRAM by way of DMA Controller. (DMA Controller Channel 7 assigned to SADC must be programmed accordingly).
- The SADC FIFO status (not-empty signal and the entry count) can be polled in the **FIFO_STAT** register.
- The SADC FIFO can also be “popped” by one entry, by reading the **FIFO_RD** register.

10.11.3.2 SADC Sample Operation Mode

SADC supports two sample operation modes: single-end measurement mode and differential measurement mode. You can choose the mode by programming **MODE** bit of **CFG** reg.

- In single-end measurement mode, the input signal should be assigned to channel N. In differential measurement mode, the input signals should be assigned to channel P and channel N. You can make the assignment by programming **CH_P** bit and **CH_N** bit of **CFG** register.
- In single-end measurement mode, the output is the measurement of voltage between **CH_N** to GND. The maximum is **2*Vref**, and the minimum is **0** V. In differential measurement mode, the output is the measurement of difference voltage between **CH_P** to **CH_N**.The maximum is **2*Vref**, and the minimum is **-2*Vref**.

10.11.4 Registers

10.11.4.1 FIFO_RD

- **Name:** Sense ADC FIFO Register
- **Description:** Reading this register will pop an entry from the sense ADC FIFO.
- **Base Address:** 0xA000E000
- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 10-177 Sense ADC FIFO Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	R	0x0	When the register is read, data in the FIFO buffer is accessed.

10.11.4.2 FIFO_THD

- **Name:** Sense ADC FIFO Threshold Register

- **Description:** This register is used to set the threshold at which the DMA request is made to the hardware handshake mechanism.
- **Base Address:** 0xA000E000
- **Offset:** 0x100
- **Reset Value:** 0x00000000

Table 10-178 Sense ADC FIFO Threshold Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5:0	THD	RW	0x0	This is used to set the threshold at which the DMA request is made to the hardware handshake mechanism.

10.11.4.3 FIFO_STAT

- **Name:** Sense ADC Status Register
- **Description:** This register contains the number of valid data entries in the ADC FIFO memory.
- **Base Address:** 0xA000E000
- **Offset:** 0x104
- **Reset Value:** 0x00000000

Table 10-179 Sense ADC Status Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	VAL	R	0x0	Identify whether FIFO is empty **Value:** • 0x0: FIFO is empty • 0x1: FIFO is not empty
7	RSVD	R		Reserved bits
6:0	COUNT	R	0x0	Sense ADC FIFO Count, number of 32-bit words in the FIFO. Maximum value is 0x40 or decimal 64.

10.11.4.4 CFG

- **Name:** Sense ADC Configuration Register
- **Description:** This register is used to configure reference, channels and input mode of ADC.
- **Base Address:** 0xA000C500

- **Offset:** 0x08
- **Reset Value:** 0x0708070A

Table 10-180 Sense ADC Configuration Register

Bits	Field Name	RW	Reset	Description
31	RST	RW	0x0	Reset ADC **Value:** • 0x0: None • 0x1: Reset
30	EN	RW	0x0	Enable ADC **Value:** • 0x0: Disable ADC • 0x1: Enable ADC
29:27	REF_SEL	RW	0x0	Reference Voltage Select **Value:** • 0x0: Buffered internal reference • 0x1: Reserved • 0x2: Reserved • 0x3: Using MSIO0 as a reference • 0x4: Using MSIO1 as a reference • 0x5: Using MSIO2 as a reference • 0x6: Using MSIO3 as a reference • 0x7: Reserved **Note:** Because the maximum input for any MSIO is supposed to be VBATT. So in this case, there is no benefit from using an external input higher than VBATT/2.
26:24	REF_HP_MODE	RW	0x7	Used to define current in the reference circuit :Default-<0x0> at 100 Ksps and <0x7> at 1 Msps
23:22	Reserved	R		Reserved bits
21:19	CH_P	RW	0x1	Used to define input for channel P **Value:** • 0x0: MSIO0 as input • 0x1: MSIO1 as input • 0x2: MSIO2 as input • 0x3: MSIO3 as input

Bits	Field Name	RW	Reset	Description
				• 0x4: MSIO4 as input
18:16	CH_N	RW	0x0	Used to define input for channel N **Value:** • 0x0: MSIO0 as input • 0x1: MSIO1 as input • 0x2: MSIO2 as input • 0x3: MSIO3 as input • 0x4: MSIO4 as input • 0x5: Temperature sensor as input • 0x6: Battery sensor as input • 0x7: Internal reference as input
15	EN_TMP	RW	0x0	Enable temperature sensor **Value:** • 0x0: Disable • 0x1: Enable
14	EN_VBAT	RW	0x0	Enable battery sensor **Value:** • 0x0: Disable • 0x1: Enable
13	MODE	RW	0x0	Used to define operation either single ended or differential. **Value:** • 0x0: Differential • 0x1: Single ended
12	EN_CAL	RW	0x0	Used to swap inputs of comparator for offset calibration. **Value:** • 0x0: Disable • 0x1: Enable
11	RSVD	R		Reserved bits
10:8	CODE_IN	RW	0x7	Used to define dynamic range of ADC. Default set to <0x7>, for higher input signal frequencies close to Nyquist rate use <0x1>
7:4	RSVD	R		Reserved bits
3:0	BIAS_RES_CTRL	RW	0xA	Buffered internal reference **Value:** • 0x3: 0.85 V

Bits	Field Name	RW	Reset	Description
				• 0x7: 1.28 V • 0xA: 1.6 V **Note:** For any specific reference value (Vref), the maximum ADC input is supposed to be 2 x Vref.

10.11.4.5 CLK

- **Name:** Sense ADC Clock Register
- **Description:** This register selects ADC clock.
- **Base Address:** 0xA000C500
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 10-181 Sense ADC Clock Register

Bits	Field Name	RW	Reset	Description
31	CLK_EN	RW	0x0	Enable ADC clock **Value:** • 0x0: Disable • 0x1: Enable
30:28	CLK_SEL	RW	0x0	ADC clock select **Value:** • 0x0: 16 MHz • 0x1: 8 MHz • 0x2: 4 MHz • 0x3: 2 MHz • 0x4: 1.6 MHz • 0x5: 1 MHz
27:0	RSVD	R		Reserved bits

10.11.5 Electrical Specifications

Table 10-182 Electrical Specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit.
Physical number of bits	PNOB		13		bits

Parameter	Symbol	Min.	Typ.	Max.	Unit.
Sampling frequency	Freq	0.0625		1.07	Msps
Integrated nonlinearity	INL	-8		8	LSB
Differential nonlinearity	DNL	-8		8	LSB
Signal-to-noise and distortion ratio	SNDR		60		dB
Total harmonic distortion	THD		65		dB
Effective number of bits	ENOB		9.5		bits
Differential offset voltage, 13-bit resolution	Vos	-4		4	LSB
Gain error	Eg	-1		1	%
Sampling capacitor	Cs		2		pF
Sampling rate	F_S	66		1060	ksps
Input impedance	Rin		> 1 M		ohm
Current consumption during ADC conversion @100 ksps	I_ADC		< 20		uA
Spurious Free Dynamic Range	SFDR		68		dB

10.12 I2S

10.12.1 I2S Introduction

The I2S controller consists of Internet Information Services (IIS) protocol to interface with external audio codec. Two 16-word deep FIFO for read path and write path respectively and is capable of handling 12-bit – 32-bit word sizes. DMA controller handles the data movement between FIFO and memory.

GR551x has 2 I2S instances:

- I2S Master (I2S_M)
- I2S Slave (I2S_S)

I2S bus is used in systems that process digital audio signals, such as:

- A/D and D/A converters
- Digital signal processors
- Error correction for compact disc and digital recording
- Digital filters
- Digital I/O interfaces

I2S Bus is a simple three-wire serial bus protocol developed by Philips to transfer stereo audio data. The bus only handles the transfer of audio data; hence control and subcode signals need to be transferred separately using a different bus protocol (such as I2C).

10.12.2 Main Features

- 4-channel:
  - 1 RX channel and 1 TX channel for I2S Slave Controller
  - 1 RX channel and 1 TX channel for I2S Master Controller
- Full duplex communication due to the independence of transmitter and receiver
- Asynchronous clocking of APB bus and I2S SCLK
- Audio data resolutions of 12-, 16-, 20-, 24-, and 32-bit
- SCLK gating for power saving
- Two 16-word FIFOs are provided, one for transmit and one for receive
- DMA supported
- Programmable FIFO thresholds
- Sampling rate: Various sample rates, up to 96 kHz (16-bit WS)

10.12.3 Functional Description

The I2S bus can only handle audio data transmissions; subcoding and controls are handled by another device, such as an I2C. The I2S protocol requires a minimum of three wires – data (SD), word select (WS), and serial clock (SCLK) – keeping the design simple and the pin count minimal. I2S controller also supports full duplex mode, four wires – data_in (SDI), data_out (SDO), word select (WS), and serial clock (SCLK).

I2S Controller supports the standard I2S frame format for transmitting and receiving data —the MSB of a word is sent one SCLK cycle after a word select change.

I2S Controller block diagram is illustrated in Figure 10-25.

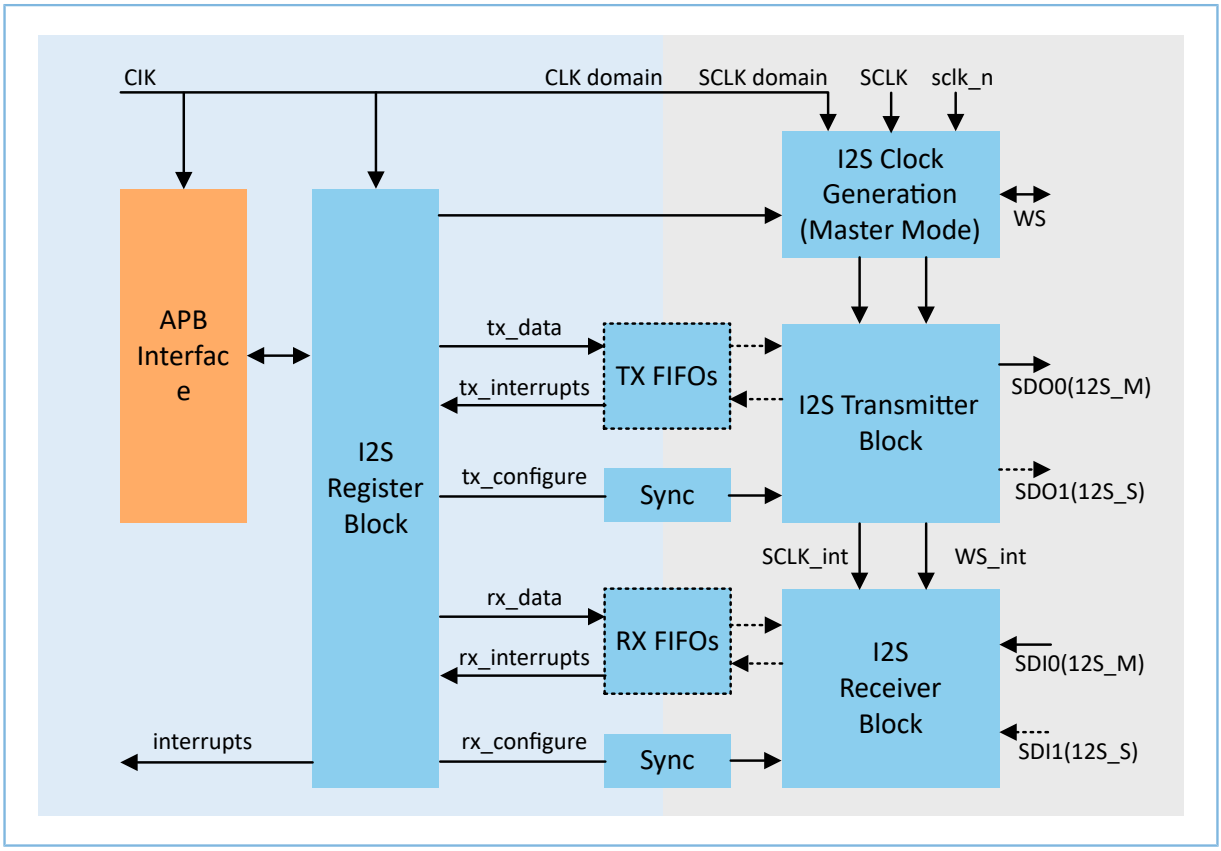

Figure 10-25 I2S Block diagram

10.12.3.1 I2S Terminology

- SCLK – serial clock
- WS – word select
- SDO – serial data out
- SDI – serial data in
- Transmitter – device that places data on the SDO line and is clocked by SCLK and WS
- Receiver – device that receives data from the SDI line and is clocked by SCLK and WS
- Master – when configured as a master, I2S Controller initializes the WS signal and supplies the clock gating and clock enabling signals
- Slave – when configured as a slave, I2S Controller responds to externally generated SCLK and WS signals

10.12.3.1.1 Signals and Formats

The inter-IC Sound (I2S) Bus is a simple three-wire serial bus protocol developed by Philips to transfer stereo audio data. It is used extensively in audio interfaces. It adopts the design of data and clock signal separation, avoiding the distortion induced by time difference.

The I2S Bus has three signals:

- SCLK – serial clock (Also called BCLK, each SCLK corresponding to each bit of audio data)
- WS – word select (Also called LRCK, WS is used to switch the left and right channel data.)
- SD(SDO or SDI) – serial data (MSB ---> LSB: Data is transmitted from high-bit to low-bit)

In I2S Controller, serial data is transmitted in two's complement format with the most significant bit (MSB) first. This means that the transmitter and receiver can have different word lengths, and neither the transmitter nor receiver needs to know what Word size the other can handle. If the word being transferred is too large for the receiver, the least significant bits (LSB) are truncated. Similarly, if the word size is less than what the receiver can handle, the data is zero padded.

The I2S module of GR551x used the I2S Philips standard. For instance, when WS is low, the word being transferred is left stereo data; when WS is high, the word being transferred is right stereo data. For standard I2S formats, the MSB of a word is sent one SCLK cycle after a WS change. Serial data sent by the transmitter can be synchronized with either the negative edge or positive edge of the SCLK signal. However, the receiver must latch the serial data on the rising edge of SCLK.

Figure 10-26 illustrates an example I2S transfer in which I2S Controller is a slave. The IDLE state of WS line is 0. Whenever the WS line makes a transition to 1, it means that after the next transition (0->1), the data starts being received. Therefore, the I2S Controller slave treats the transfer as a START condition. When the stereo data is completely latched (signaled by Word Select Line going 0 again, also treated as start of new data frame), the data is pushed into the internal FIFO.

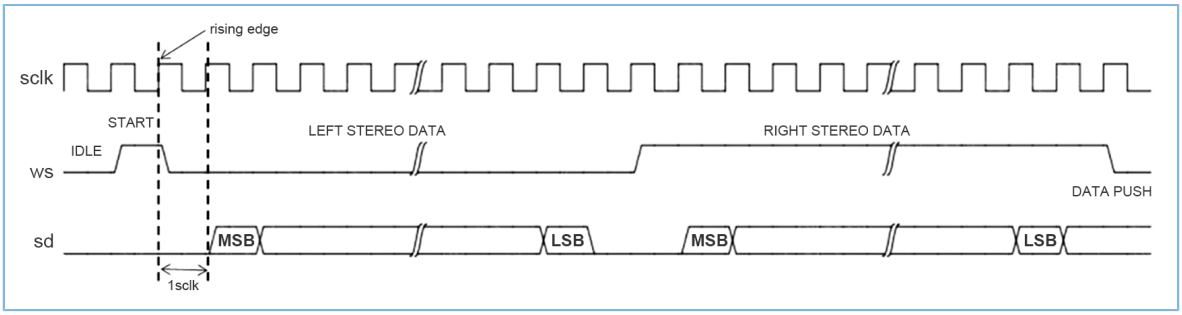

Figure 10-26 I2S formats

10.12.3.1.2 Sampling and CLK

As the frequency of SCLK up to 3072 kHz, the number of SCLK cycles for which the word select the WS line of I2S can configured as 16 SCLK cycles, 24 SCLK cycles or 32 SCLK cycles (configure **SCLK_CFG.WS_SCLK**). If select the 16 SCLK cycles, the sample rate is up to 96 kHz. The calculation formula is as follows:

- sample rate = frequency of SCLK / (2 X Sampling depth (16,24 or 32))
- sample rate = frequency of WS(LRCK)

The I2S module can be used as master or slave. The instance of master mode is I2S_M and the instance of slave mode is I2S_S. When configured as a master, I2S controller initializes the WS signal and supplies the clock gating and clock enabling signals. Otherwise, when configured as a slave, I2S controller responds to externally generated SCLK and WS signals. Working in master mode, the I2S Clock Generation must be enabled by setting the Clock Enable Register (**CLK_EN**) bit0 to '1'.

If the I2S module is used as master, the I2S Clock Configure Register (**CLK_CFG**) should be properly configured. The value of **CLK_CFG** register bit0 – bit11 (DIV) is decided by the sample rate, sampling depth (16, 24 or 32) and clock divider source (bit18).

For example, the **CLK_CFG** bit18 is '1' (32 MHz) and the sample rate is 48 kHz, the calculation formula is as follows:

```
div_TEMP0 = 32MHz *10       / (2 X Sampling Depth X Sample Rate)
if ((div_TEMP0%10) >= 5) /****** Rounding****/
{
    div_TEMP1 = div_TEMP0/10+1;
}
else
{
    div_TEMP1 = div_TEMP0/10;
}
if (div_TEMP1 >= 2)
{
    DIV = div_TEMP1 - 2;
}
else
{
    DIV = 0;
}
```

Table 10-183 CLK_CFG.CLK_SRC_SEL clock configuration examples

Common Configurations			32 MHz			96 MHz		
Sample Rate (k)	Bits (Left + Right)	Desired Bit Clock (kHz)	Desired Divider (Value)	Actual Divider (Value) (CLK_CFG.DIV)	Actual Word Size (SCLK_CFG.WS_SCLK)	Desired Divider (Value)	Actual Divider (Value) (CLK_CFG.DIV)	Actual Word Size (SCLK_CFG.WS_SCLK)
8	2*8	256	125	125	16	375	375	16
8	2*16	256	125	125	16	375	375	16
8	2*24	384	83.333333333	83	24	250	250	24
8	2*32	512	62.5	63	32	187.5	188	32
48	2*8	1,536	20.83333333	21	16	62.5	63	16
48	2*16	1,536	20.83333333	21	16	62.5	63	16
48	2*24	2,304	13.88888889	14	24	41.666666667	42	24
48	2*32	3072	10.41666667	10	32	31.21951	31	32
44.1	2*8	1,411.2	22.67573696	23	16	68.02721088435	68	16
44.1	2*16	1,411.2	22.67573696	23	16	68.02721088435	68	16
44.1	2*24	2,116.8	15.1171579743	15	24	45.3514739229	45	24
44.1	2*32	2,822.4	11.33786848	11	32	34.013605442	34	32

10.12.3.2 I2S Enable

You must enable the I2S Controller before any data can be received or transmitted into the FIFOs. To enable the component, set the I2S Enable (**I2S_EN**) bit of the I2S Enable Register (**EN**) to **1**. When you disable the device, it acts as a global disable. To disable I2S Controller, set **EN[0]** to **0**.

After disable, the following events occur:

- TX and RX FIFOs are cleared, and read/write pointers are reset;
- Any data in the process of being transmitted or received is lost;
- All other programmable enables (such as transmitter/receiver block enables and individual TX/RX channel enables) in the component are overridden;
- Generation of master mode clock signals sclk_en, ws_out and sclk_gate are disabled (for instance, they are held low).

When I2S Controller is enabled and configured as a master, the device always starts in the left stereo data cycle (`WS = 0`), and one SCLK cycle later transitions to the right stereo data cycle (`WS = 1`). This allows for half a frame of SCLKs to write data to the TX FIFOs and to ensure that any connected slave receivers do not miss the start of the data frame (for instance, the WS 1-to-0 transition) once the SCLK restarts. (When I2S Controller is configured as a slave, WS is externally supplied.) On reset, the **EN[0]** is set to **0** (disable).

10.12.3.3 I2S as Transmitter

I2S Master Controller and I2S Slave Controller support 1 I2S transmit (TX) channel. Stereo data pairs (such as, left and right audio data) written to a TX channel by CPU are shifted out serially on the appropriate serial data out line. The shifting is timed with respect to the serial clock (SCLK) and the word select line (WS).

Figure 10-27 illustrates the basic usage flow for an I2S Controller when it acts as a transmitter.

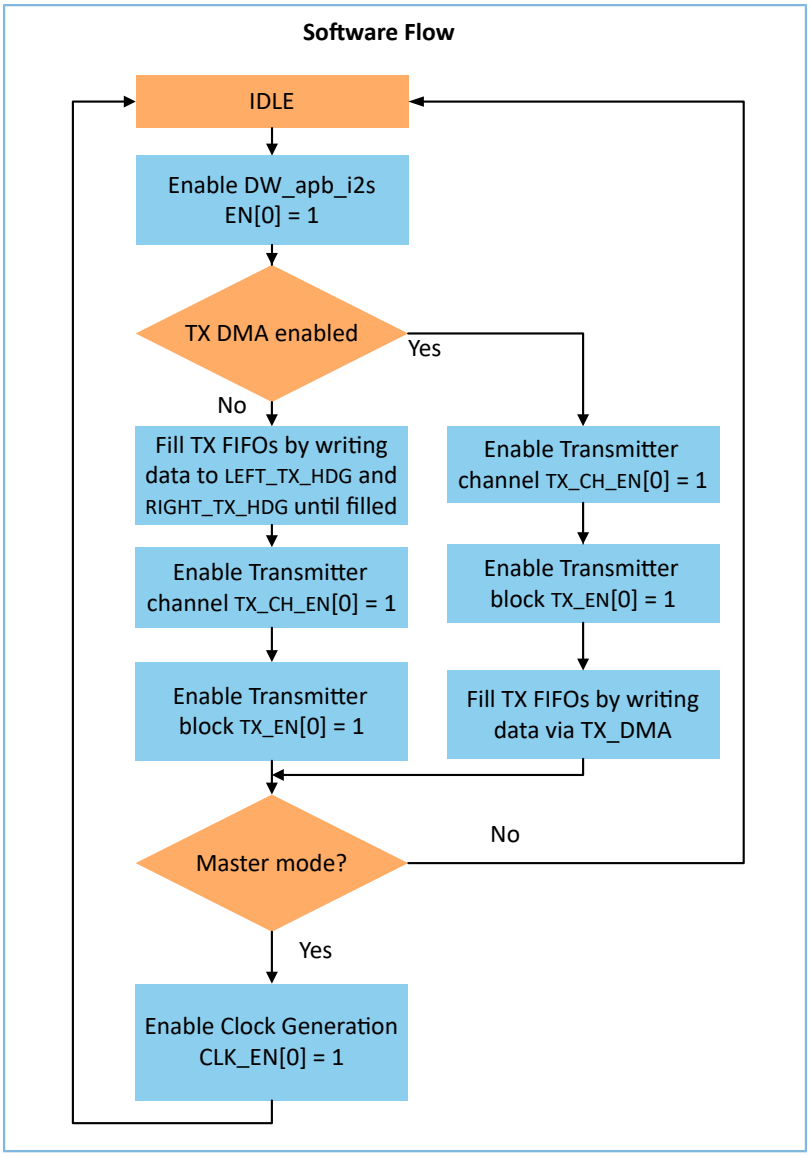

Figure 10-27 Basic Usage flow when I2S as transmitter

10.12.3.4 I2S as Receiver

I2S Master Controller and I2S Slave Controller support 1 I2S receive (RX) channel. Stereo data pairs (such as, left and right audio data) are received serially from a data input line. These data words are stored in RX FIFOs until they are read by CPU. The receiving is timed with respect to the serial clock (SCLK) and the word select line (WS).

Figure 10-28 illustrates the basic usage flow for an I2S Controller when it acts as a receiver.

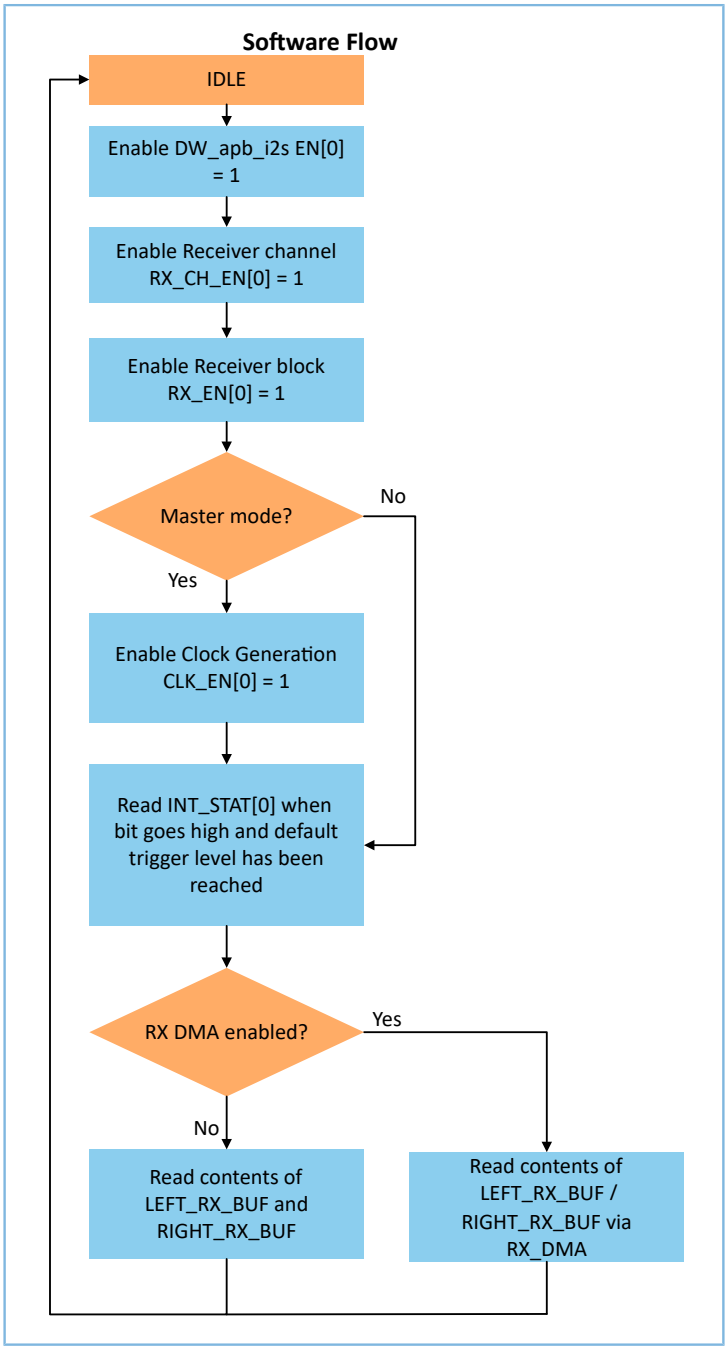

Figure 10-28 Basic Usage Flow when I2S as Receiver

10.12.3.5 I2S interrupts

The **INT_STAT** register, **INT_MASK** register, **RX_OVER** register and **TX_OVER** are provided to manage interrupts. The **INT_MASK** register is used to enable or disable the TX FIFO Overrun interrupt, TX FIFO Empty interrupt, RX FIFO Overrun interrupt and RX FIFO Data Available interrupt. By reading the **INT_STAT** register, the triggered interrupt type can be obtained. If the RX FIFO or TX FIFO data overflows, read the **RX_OVER** register **bit0** or **TX_OVER** register **bit0** to clear the RX FIFO Data Overrun interrupt or TX FIFO Data Overrun interrupt.

10.12.3.6 I2S FIFO and DMA

The I2S module provide four 16-word FIFO, two (I2S_M TX FIFO and I2S_S TX FIFO) for transmit and two (I2S_M RX FIFO and I2S_S RX FIFO) for receive. By configuring the Receive FIFO Configuration Register (**RX_FIFO_CFG**), you can program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. By configuring Transmit FIFO Configuration Register (**TX_FIFO_CFG**), you can program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated.

The I2S module supported four DMA channel, one for I2S slave transmit, one for I2S slave receive, one for I2S master transmit and one for I2S master receive. The Receiver Block DMA Register (**RX_DMA**), Reset Receiver Block DMA Register (**RST_RX_DMA**), Transmitter Block DMA Register (**TX_DMA**) and Reset Transmitter Block DMA Register (**RST_TX_DMA**) are provide to configure these channel.

10.12.4 Registers

10.12.4.1 EN

- **Name:** I2S Enable Register
- **Description:** This register acts as a global enable/disable for I2S.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 10-184 I2S Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	I2S_EN	RW	0x0	I2S enable. This bit enables or disables I2S. A disable on this bit overrides any other block or channel enables and flushes all FIFOs. **Value:** • 0x0: I2S disabled. • 0x1: I2S enabled

10.12.4.2 RX_EN

- **Name:** I2S Receiver Block Enable Register
- **Description:** This register acts as an enable/disable for the I2S Receiver block.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 10-185 I2S Receiver Block Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_EN	RW	0x0	Receiver block enable. This bit enables or disables the receiver. A disable on this bit overrides any individual receive channel enables. **Value:** • 0x0: Receiver disabled • 0x1: Receiver enabled

10.12.4.3 TX_EN

- **Name:** I2S Transmitter Block Enable Register
- **Description:** This register acts as an enable/disable for the I2S Transmitter block.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 10-186 I2S Transmitter Block Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_EN	RW	0x0	Transmitter block enable. This bit enables or disables the transmitter. A disable on this bit overrides any individual transmit channel enables. **Value:** • 0x0: Transmitter disabled • 0x1: Transmitter enabled

10.12.4.4 CLK_EN

- **Name:** Clock Enable Register
- **Description:** This register acts as an enable/disable for the I2S clock generation block.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0xC
- **Reset Value:** 0x00000000

Table 10-187 Clock Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLK_EN	RW	0x0	Clock Generation enable/disable. This bit enables or disables the clock generation signals when I2S is a master. **Value:** • 0x0: Clock generation disabled • 0x1: Clock generation enabled **Note:** When the I2S is configured as a slave, this register serves no purpose.

10.12.4.5 SCLK_CFG

- **Name:** Clock Configuration Register
- **Description:** This register configures the WS when I2S is a master.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 10-188 Clock Configuration Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:3	WS_SCLK	RW	0x0	These bits are used to program the number of SCLK cycles for which the word select line (ws_out) stays in the left or right sample mode. The I2S Clock Generation block must be disabled (CLK_EN [0] = 0) prior to any changes in this value. **Value:** • 0x0: 16 SCLK cycles • 0x1: 24 SCLK cycles • 0x2: 32 SCLK cycles
2:0	SCLK_GAT	RW	0x0	These bits are used to program the gating of SCLK. The programmed gating value must be less than or equal to the largest configured/programmed audio resolution to prevent the truncating of RX/TX data. The I2S Clock Generation block must be disabled (CLK_EN[0] = 0) before making any changes in this value. **Values:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (NO_CLOCK_GATING): Clock gating is disabled • 0x1 (CLOCK_CYCLES_12): Gating after 12 SCLK cycles • 0x2 (CLOCK_CYCLES_16): Gating after 16 SCLK cycles • 0x3 (CLOCK_CYCLES_20): Gating after 20 SCLK cycles • 0x4 (CLOCK_CYCLES_24): Gating after 24 SCLK cycles **Note:** This register is only relevant when component is configured to be a master (I2S_MODE_EN = 1).

10.12.4.6 RX_FIFO_RST

- **Name:** Receiver Block FIFO Reset Register
- **Description:** This register specifies the Receiver Block FIFO Reset Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 10-189 Receiver Block FIFO Reset Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RX_FIFO_RST	W	0x0	Receiver FIFO Reset. Writing a 1 to this register flushes all the RX FIFOs (this is a self-clear bit). The Receiver Block must be disabled before writing to this bit. **Value:** • 0x0: Do not flush the RX FIFO • 0x1: Flush the RX FIFO

10.12.4.7 TX_FIFO_RST

- **Name:** Transmitter Block FIFO Reset Register
- **Description:** This register specifies the Transmitter Block FIFO Reset Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 10-190 Transmitter Block FIFO Reset Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	TX_FIFO_RST	W	0x0	Transmitter FIFO Reset. Writing a 1 to this register flushes all the TX FIFOs (this is a self-clear bit). The Transmitter Block must be disabled prior to writing this bit. **Value:** • 0x0: Do not flush the TX FIFO • 0x1: Flush the TX FIFO

10.12.4.8 LEFT_RX_BUF

- **Name:** Left Receive Buffer Register
- **Description:** This specifies the Left Receive Buffer Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 10-191 Left Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:0	LEFT_RX_BUF	R	0x0	The left stereo data received serially from the receive channel input (sdi). If the RX FIFO is full and the two-stage read operation (for instance, a read from LEFT_RX_BUF followed by a read from RIGHT_RX_BUF) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) **Note:** Before reading this register again, the right stereo data must be read from RIGHT_RX_BUFx or the status/interrupts will not be valid.

10.12.4.9 LEFT_TX_HDG

- **Name:** Left Transmit Holding Register
- **Description:** This specifies the Left Transmit Holding Register
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 10-192 Left Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:0	LEFT_TX_HDG	W	0x0	The left stereo data to be transmitted serially through the transmit channel output (sdo) is written through this register. Writing is a two-stage process: • A write to this register passes the left stereo sample to the transmitter • This MUST be followed by writing the right stereo sample to the RIGHT_TX_HDG register Data must only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.

10.12.4.10 RIGHT_RX_BUF

- **Name:** Right Receive Buffer Register
- **Description:** This specifies the Right Receive Buffer Register
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 10-193 Right Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:0	RIGHT_RX_BUF	R	0x0	The right stereo data received serially from the receive channel input (SDI) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LEFT_RX_BUF followed by a read from RIGHT_RX_BUF) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.) **Note:** Prior to reading this register, the left stereo data must be read from LEFT_RX_BUF, or the status/interrupts will not be valid.

10.12.4.11 RIGHT_TX_HDG

- **Name:** Right Transmit Holding Register
- **Description:** This specifies the Right Transmit Holding Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)

- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 10-194 Right Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:0	RIGHT_TX_HDG	W	0x0	The right stereo data to be transmitted serially through the transmit channel output (SDO) is written through this register. Writing is a two-stage process: • A left stereo sample must be written to the LEFT_TX_HDG register • A write to this register passes the right stereo sample to the transmitter Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.

10.12.4.12 RX_CH_EN

- **Name:** Receive Enable Register
- **Description:** This specifies the Receive Enable Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x28
- **Reset Value:** 0x00000001

Table 10-195 Receive Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_CH_EN	RW	0x1	Receive channel enable. This bit enables/disables a receive channel. On enable, the channel begins receiving on the next left stereo cycle. A global disable of I2S (EN[0] = 0) or the Receiver block (RX_EN[0] = 0) overrides this value. **Value:** • 0x0: Receive Channel Disable • 0x1: Receive Channel Enable

10.12.4.13 TX_CH_EN

- **Name:** Transmit Enable Register
- **Description:** This specifies the Transmit Enable Register.

- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x2c
- **Reset Value:** 0x00000001

Table 10-196 Transmit Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_CH_EN	RW	0x1	Transmit channel enable. This bit enables/disables a transmit channel. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of I2S (EN[0] = 0) or Transmitter block (TX_EN[0] = 0) overrides this value. **Value:** • 0x0: Transmit Channel Disable • 0x1: Transmit Channel Enable

10.12.4.14 RX_CFG

- **Name:** Receive Configuration Register
- **Description:** This specifies the Receive Configuration Register
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x30
- **Reset Value:** 0x00000005

Table 10-197 Receive Configuration Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	WORD_LEN	RW	0x5	These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LEFT_RX_BUF (or RIGHT_RX_BUF) register. Programmed data resolution must be less than or equal to 0x5. If the selected resolution is greater than the 0x5, the receive channel defaults back to 0x5. The channel must be disabled prior to any changes in this value (RX_EN0[0] = 0). **Value:** • 0x0: Ignore the word length • 0x1: 12-bit data resolution of the receiver.

Bits	Field Name	RW	Reset	Description
				• 0x2: 16-bit data resolution of the receiver. • 0x3: 20-bit data resolution of the receiver. • 0x4: 24-bit data resolution of the receiver. • 0x5: 32-bit data resolution of the receiver.

10.12.4.15 TX_CFG

- **Name:** Transmit Configuration Register
- **Description:** This specifies the Transmit Configuration Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x34
- **Reset Value:** 0x00000005

Table 10-198 Transmit Configuration Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	WORD_LEN	RW	0x5	These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first. Programmed resolution must be less than or equal to 0x5. If the selected resolution is greater than 0x5, the transmit channel defaults back to 0x5. The channel must be disabled prior to any changes in this value (TX_EN [0] = 0). **Value:** • 0x0: Ignore the word length • 0x1: 12-bit data resolution of the transmitter. • 0x2: 16-bit data resolution of the transmitter. • 0x3: 20-bit data resolution of the transmitter. • 0x4: 24-bit data resolution of the transmitter. • 0x5: 32-bit data resolution of the transmitter.

10.12.4.16 INT_STAT

- **Name:** Interrupt status Register
- **Description:** This specifies the Interrupt Status Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)

- **Offset:** 0x38
- **Reset Value:** 0x00000010

Table 10-199 Interrupt status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	TX_FIFO_OVER	R	0x0	Status of Data Overrun interrupt for the TX channel. This bit specifies whether the TX FIFO write is valid or an overrun. Attempt to write to full TX FIFO. **Value:** • 0x0: TX FIFO write valid • 0x1: TX FIFO write overrun
4	TX_FIFO_EMPTY	R	0x1	Status of Transmit Empty Trigger interrupt. This bit specifies whether the TX FIFO trigger level has reached or not. TX FIFO is empty. **Value:** • 0x0: TX FIFO trigger level is reached • 0x1: TX FIFO trigger level is not reached
3:2	RSVD	R		Reserved bits
1	RX_FIFO_OVER	R	0x0	Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. **Value:** • 0x0: RX FIFO write valid • 0x1: RX FIFO write overrun
0	RX_DATA_AVL	R	0x0	Status of Receive Data Available interrupt. This bit denotes the status of the RX FIFO trigger level. **Value:** • 0x0: RX FIFO trigger level is not reached • 0x1: RX FIFO trigger level is reached

10.12.4.17 INT_MASK

- **Name:** Interrupt Mask Register
- **Description:** This specifies the Interrupt Mask Register
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x3c

- **Reset Value:** 0x00000033

Table 10-200 Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	TX_FOM	RW	0x1	Mask TX FIFO Overrun interrupt. This bit masks or unmasks a TX FIFO overrun interrupt. **Value:** • 0x0: Enable TX FIFO Overrun interrupt • 0x1: Disable TX FIFO Overrun interrupt
4	TX_FEM	RW	0x1	Mask TX FIFO Empty interrupt. This bit masks or unmasks a TX FIFO Empty interrupt. **Value:** • 0x0: Enable TX FIFO Empty interrupt • 0x1: Disable TX FIFO Empty interrupt
3:2	RSVD	R		Reserved bits
1	RX_FOM	RW	0x1	Mask RX FIFO Overrun interrupt. This bit masks or unmasks an RX FIFO Overrun interrupt. **Value:** • 0x0: Enable RX FIFO Overrun interrupt • 0x1: Disable RX FIFO Overrun interrupt
0	RX_DAM	RW	0x1	Mask RX FIFO Data Available interrupt. This bit masks or unmasks an RX FIFO Data Available interrupt. **Value:** • 0x0: Enable RX FIFO data available interrupt • 0x1: Disable RX FIFO data available interrupt

10.12.4.18 RX_OVER

- **Name:** Receive Overrun Register
- **Description:** This specifies the Receive Overrun Register
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 10-201 Receive Overrun Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_CLR_FDO	R	0x0	Read this bit to clear the RX FIFO Data Overrun interrupt. **Value:** • 0x0: RX FIFO write valid • 0x1: RX FIFO write overrun

10.12.4.19 TX_OVER

- **Name:** Transmit Overrun Register
- **Description:** This specifies the Transmit Overrun Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 10-202 Transmit Overrun Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_CLR_FDO	R	0X0	Read this bit to clear the TX FIFO Data Overrun interrupt. **Value:** • 0x0: TX FIFO write valid • 0x1: TX FIFO write overrun

10.12.4.20 RX_FIFO_CFG

- **Name:** Receive FIFO Configuration Register
- **Description:** This specifies the Receive FIFO Configuration Register
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x48
- **Reset Value:** 0x00000008

Table 10-203 Receive FIFO Configuration Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
3:0	RX_FIFO_TL	RW	0x8	These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated. Trigger Level = Programmed Value + 1 **Value:** • 0x0: Interrupt trigger when FIFO level is 1. • 0x1: Interrupt trigger when FIFO level is 2. • 0x2: Interrupt trigger when FIFO level is 3. • 0x3: Interrupt trigger when FIFO level is 4. • 0x4: Interrupt trigger when FIFO level is 5. • 0x5: Interrupt trigger when FIFO level is 6. • 0x6: Interrupt trigger when FIFO level is 7. • 0x7: Interrupt trigger when FIFO level is 8. • 0x8: Interrupt trigger when FIFO level is 9. • 0x9: Interrupt trigger when FIFO level is 10. • 0xA: Interrupt trigger when FIFO level is 11. • 0xB: Interrupt trigger when FIFO level is 12. • 0xC: Interrupt trigger when FIFO level is 13. • 0xD: Interrupt trigger when FIFO level is 14. • 0xE: Interrupt trigger when FIFO level is 15. • 0xF: Interrupt trigger when FIFO level is 16.

10.12.4.21 TX_FIFO_CFG

- **Name:** Transmit FIFO Configuration Register
- **Description:** This specifies the Transmit FIFO Configuration Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x4C
- **Reset Value:** 0x00000008

Table 10-204 Transmit FIFO Configuration Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	TX_FIFO_TL	RW	0x8	These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated.

Bits	Field Name	RW	Reset	Description
				Trigger Level = TX_FIFO_TL **Value:** • 0x0: Interrupt trigger when FIFO level is 1. • 0x1: Interrupt trigger when FIFO level is 2. • 0x2: Interrupt trigger when FIFO level is 3. • 0x3: Interrupt trigger when FIFO level is 4. • 0x4: Interrupt trigger when FIFO level is 5. • 0x5: Interrupt trigger when FIFO level is 6. • 0x6: Interrupt trigger when FIFO level is 7. • 0x7: Interrupt trigger when FIFO level is 8. • 0x8: Interrupt trigger when FIFO level is 9. • 0x9: Interrupt trigger when FIFO level is 10. • 0xA: Interrupt trigger when FIFO level is 11. • 0xB: Interrupt trigger when FIFO level is 12. • 0xC: Interrupt trigger when FIFO level is 13. • 0xD: Interrupt trigger when FIFO level is 14. • 0xE: Interrupt trigger when FIFO level is 15. • 0xF: Interrupt trigger when FIFO level is 16.

10.12.4.22 RX_FIFO_FLUSH

- **Name:** Receive FIFO Flush Register
- **Description:** This specifies the Receive FIFO Flush Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 10-205 Receive FIFO Flush Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RX_FIFO_RST	W	0x0	Receive Channel FIFO Reset. Writing a 1 to this register flushes an individual RX FIFO (This is a self-clear bit.). A RX channel or block must be disabled prior to writing to this bit. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Do not flush an individual RX FIFO. • 0x1: Flush an individual RX FIFO.

10.12.4.23 TX_FIFO_FLUSH

- **Name:** Transmit FIFO Flush Register
- **Description:** This specifies the Transmit FIFO Flush Register.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 10-206 Transmit FIFO Flush Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	TX_FIFO_RST	W	0x0	Transmit Channel FIFO Reset. Writing a 1 to this register flushes an individual TX FIFO (This is a self-clear bit.). A TX channel or block must be disabled prior to writing to this bit. **Value:** • 0x0: Do not flush an individual TX FIFO. • 0x1: Flush an individual TX FIFO.

10.12.4.24 RX_DMA

- **Name:** Receiver Block DMA Register
- **Description:** The RX_DMA register allows access to receive channel via a single point rather than through the LEFT_RX_BUF and RIGHT_RX_BUF registers. The receive channel is targeted in a cyclical fashion (starting at the lowest numbered enabled channel) and takes two reads (left and right stereo data) before the component points to the next channel.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x1c0
- **Reset Value:** 0x00000000

Table 10-207 Receiver Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_DMA	RW	0x0	Receiver Block DMA Register.

Bits	Field Name	RW	Reset	Description
				These bits are used to cycle repeatedly through the enabled receive channel, reading stereo data pairs.

10.12.4.25 RST_RX_DMA

- **Name:** Reset Receiver Block DMA Register
- **Description:** The RX_DMA can be reset to the enabled Channel via the RST_RX_DMA register. The RST_RX_DMA register can be written to at any stage of the RX_DMA's read cycle, however, it has no effect when the component is in the middle of a stereo pair read.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x1C4
- **Reset Value:** 0x00000000

Table 10-208 Reset Receiver Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RST_RX_DMA	W	0x0	Reset Receiver Block DMA Register. Writing a 1 to this self-clearing register resets the RX_DMA register mid-cycle to point to the enabled Receive channel. **Value:** • 0x0: No effect. • 0x1: Reset receiver block DMA register.

10.12.4.26 TX_DMA

- **Name:** Transmitter Block DMA Register
- **Description:** The TX_DMA register functions similar to the RX_DMA register and allows write access to all of the enabled Transmit channel via a single point rather than through the LEFT_TX_HDG and RIGHT_TX_HDG registers.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x1c8
- **Reset Value:** 0x00000000

Table 10-209 Transmitter Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_DMA	RW	0x0	Transmitter Block DMA Register.

Bits	Field Name	RW	Reset	Description
				These bits are used to cycle repeatedly through the enabled transmit channel to allow writing of stereo data pairs.

10.12.4.27 RST_TX_DMA

- **Name:** Reset Transmitter Block DMA Register
- **Description:** This register provides the same functionality as the RST_RX_DMA register but targets TX_DMA instead.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x1cc
- **Reset Value:** 0x00000000

Table 10-210 Reset Transmitter Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RST_TX_DMA	W	0x0	Reset Transmitter Block DMA Register. Writing a 1 to this self-clearing register resets the TX_DMA register mid-cycle to point to the enabled Transmit channel. **Value:** • 0x0: No effect. • 0x1: Reset transmitter block DMA register.

10.12.4.28 CLK_CFG

- **Name:** I2S Clock Configure Register
- **Description:** This register is used to configure I2S master clock.
- **Base Address:** 0xA000CA00 (master) and 0xA000F000 (slave)
- **Offset:** 0x268
- **Reset Value:** 0x00040008

Table 10-211 I2S Clock Configure Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R		Reserved bits
18	CLK_SRC_SEL	RW	0x1	clock divider source select 0:96 M, 1:32 M
17	RSVD	R		Reserved bits
16	DIV_EN	RW	0x0	Enable I2S clock divider

Bits	Field Name	RW	Reset	Description
15:12	RSVD	R		Reserved bits
11:0	DIV	RW	0x8	ratio = 1/(div + 2). Duty cycle is not 50 when the number is odd. For example: div = 0, ratio = 1/2

10.12.5 Electrical Specifications

Table 10-212 Electrical Specifications

Symbol	Description	Min.	Typ.	Max.	Unit
f_{SCK}	I2S clock frequency			3072	kHz
t_{SUSDI}	Data input setup time (from SDI to SCK)	20			ns
t_{HDSDI}	Data input hold time (from SCK to SDI change)	15			ns
t_{VSDO}	Data output valid time (from SCK to SDO valid)			20	ns
DC_{SCK}	SCK Clock duty cycle	45		55	%

The I2S Timing Diagram is in Figure 10-29:

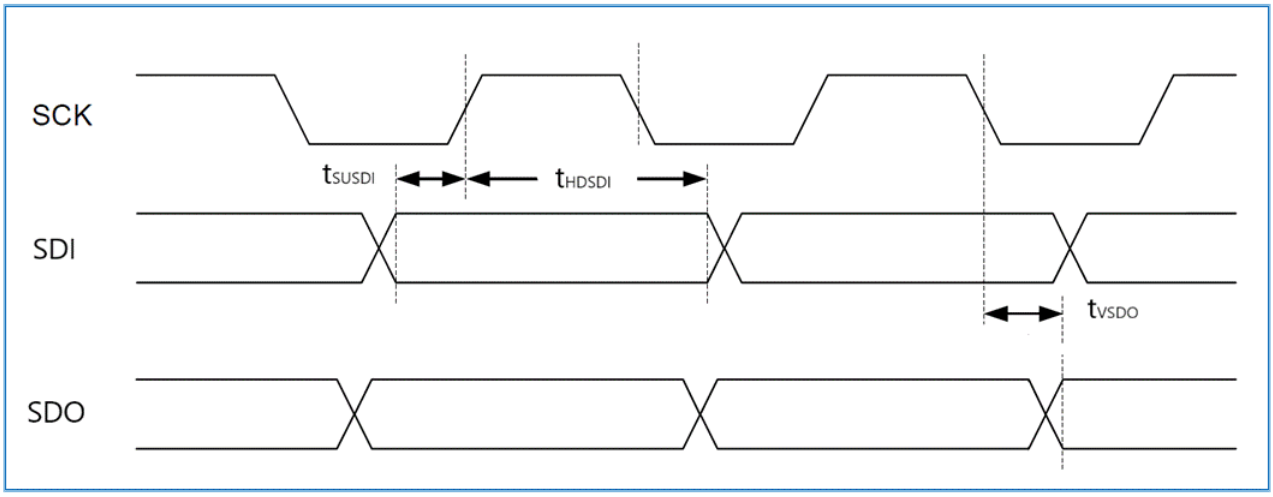

Figure 10-29 I2S Timing Diagram

10.13 ISO/IEC 7816-3 Master

10.13.1 Introduction

GR551x has one instance of SIM Interface block that implements ISO/IEC 7816 standard. ISO/IEC 7816 is an international standard related to electronic identification cards with contacts, especially smart cards, managed jointly by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

10.13.2 Main Features

- UICC power save (clock stop)
- UICC cold and warm reset
- Guard time handling
- Wait time evaluation
- Low power design

10.13.3 Functional Description

The SIM interface module provides an interface to an external Universal Integrated Circuit Card (UICC) containing the Universal Subscriber Identity Module (USIM) application. The corresponding software module implements the SIM terminal transport, Card Application Toolkit (CAT) and application layer. It has a software interface to access all required USIM functions.

10.13.3.1 Block Diagram

The block diagram of SIM Interface is illustrated in Figure 10-30. A main clock gate disables the module clock during any module inactivity for a minimal dynamic power consumption. The SIM clock is derived from the module clock and base for the Elementary Time Unit (ETU) generation. SIM clock and ETU are tracked by counters.

A shift register in the I/O module deserializes/serializes the bit stream and calculates/checks the parity. Data bytes are served by/to a Byte Direct Memory Access (DMA). The overall transmission is managed by the control unit. Events are signaled via interrupts. The UICC presence is monitored by the presence observer module.

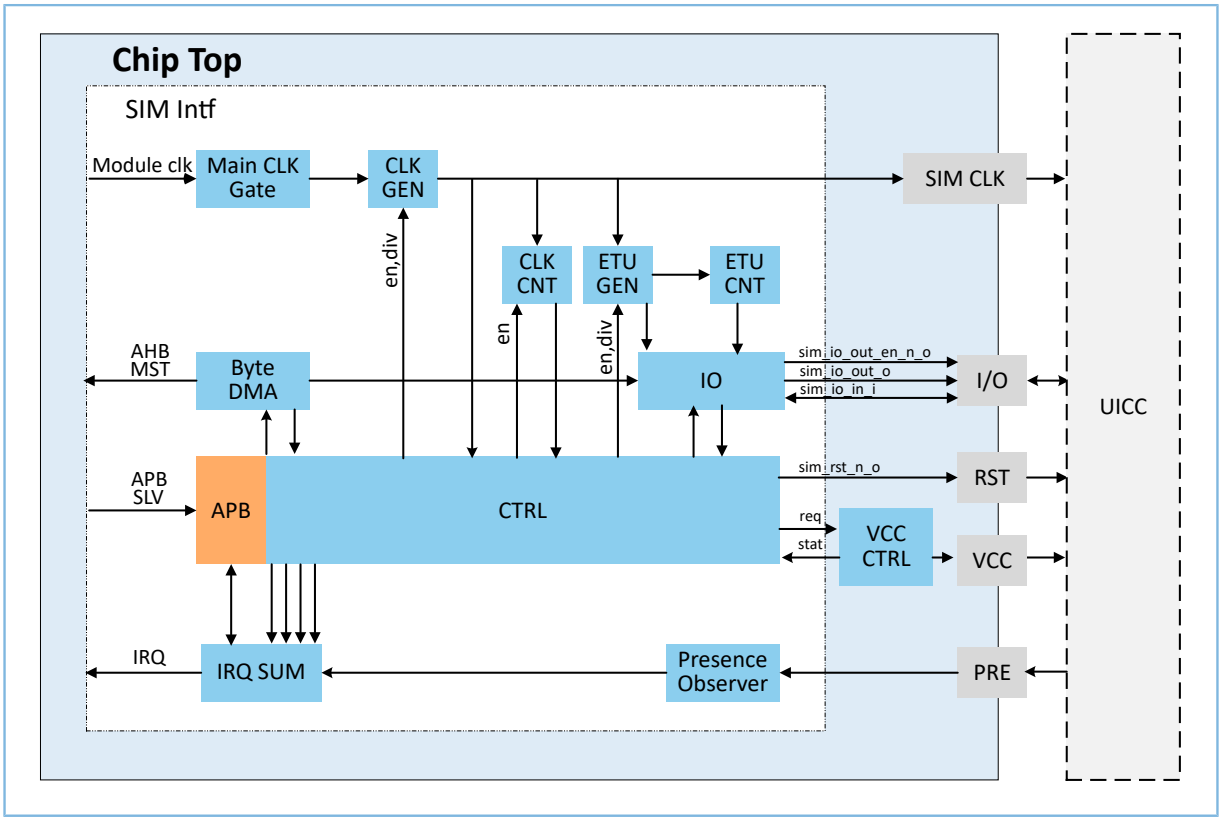

Figure 10-30 SIM Interface block diagram

10.13.4 Hardware Behavior

The SIM interface hardware module supports the physical, data link, and transport layer defined in *ETSI TS 131 101*.

10.13.4.1 Physical Layer

The SIM interface hardware module interfaces the UICC with the following ports:

- Clock (CLK)
- Reset (RST)
- Input/Output (I/O)
- UICC presence (PRE)

The detailed electrical required characteristic of the ports can be taken from *ETSI TS 131 101*.

10.13.4.1.1 CLK

The SIM clock is derived from the module clock by dividing it by **CLK_CFG.CLK_DIV**.

$$f = \frac{f_{mod}}{clk\_div+1} \quad (5)$$

The SIM clock frequency shall be in the range of 1 to 5 MHz, 4 MHz during UICC initialization and 5 MHz during normal operation. The clock division has to be chosen accordingly.

10.13.4.1.2 RST

The UICC active low reset pin shall be driven by the SIM interface output, and RST controlled by the register **CTRL.ACTION**.

10.13.4.1.3 I/O

The bidirectional I/O pad on chip top level is implemented as open drain with a recommended pull up to VCC of 20 kΩ. If supported by the UICC the I/O pad can use low impedance drivers to save power.

10.13.4.1.4 PRE

UICC card holder often serve a card presence contact. Any card presence change will trigger an interrupt (see **STAT.PRESENCE_STAT** and **STAT.IRQ_PRESENCE)**.

10.13.4.2 Power States

The SIM interface module is responsible to execute the cold reset, warm reset, clock stop and the power down sequence. Software triggers power state changes by writing register field **CTRL. ACTION**.

10.13.4.2.1 Activation on Cold Reset

The activation sequence and cold reset sequence is shown in Figure 10-31. At first VCC is ramped up. With T_a the clock starts while the reset is still asserted. On T_b the reset is released and the UICC has to start with the ATR within t_c.

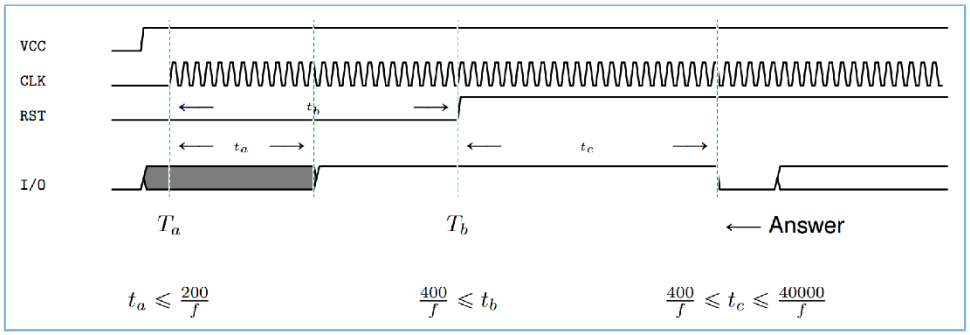

Figure 10-31 Activation on cold reset

The I/O pin is not actively driven by the SIM interface during the entire cold reset procedure. The high level is ensured by the pull up resistor. Therefore t_a is not implemented in the SIM interface.

t_b is hard-coded to 400 clock cycles.

The minimum t_c is not checked by the SIM interface module.

The maximum t_c is defined by register fields **CLK_CFG.ETU_DIV** and **TIMES_CFG.WAIT_TIME** with respect to (6). The reset values are chosen according the specification to receive the ATR without any modifications. **CLK_CFG.ETU_DIV** is **371** and **TIMES.WAITTIME** is **107** (see (7)).

$$t_c = \frac{(etu\_div+1)\times(waittime+1)}{f_{mod}} \quad (6)$$

$$waittime = \left\lceil \frac{40000}{(etu\_div+1)} - 1 \right\rceil = \lceil 106.52 \rceil = 107 \quad (7)$$

If maximum t_c is not met by the UICC, interrupt **STAT.IRQ_RX_ERR** is triggered.

10.13.4.2.2 Warm Reset

The warm reset sequence is shown in Figure 10-32.

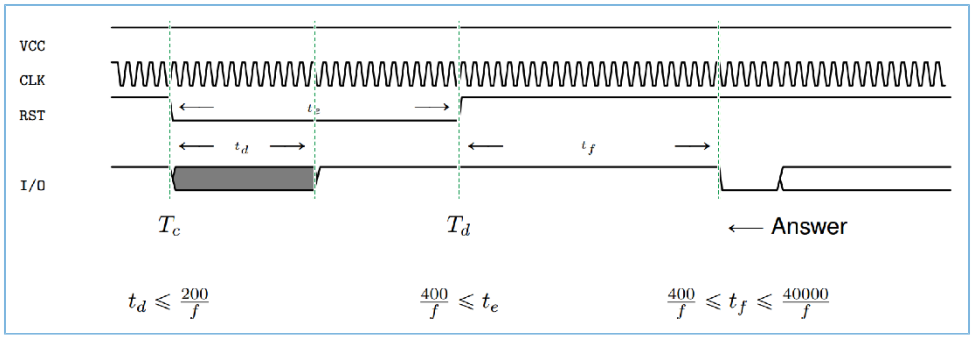

Figure 10-32 Warm reset

The warm reset is implemented identical to the cold reset, with $T_c = T_a$, $T_d = T_b$, $t_d = t_a$, $t_e = t_b$ and $t_f = t_c$.

Due to the fact that the registers **CLK_CFG** and **TIMES_CFG** may have changed after the ETU evaluation, the reset values have to be restored by firmware before triggering the warm reset.

10.13.4.2.3 Clock Stop

The clock stop sequence is shown in Figure 10-33.

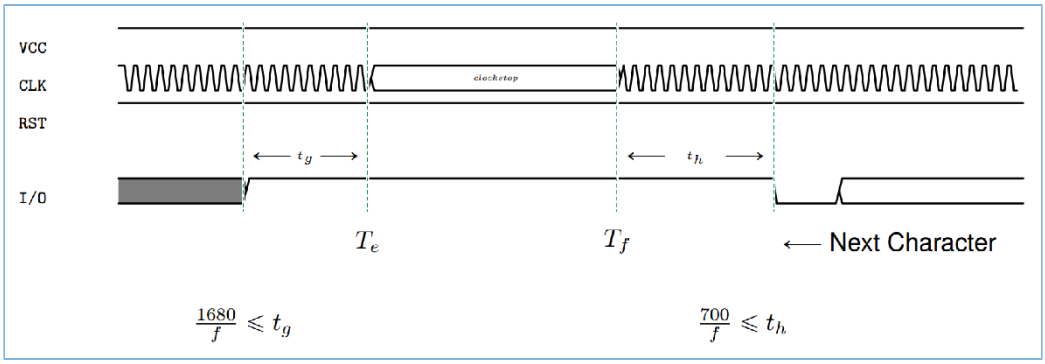

Figure 10-33 Clock stop

The times t_g and t_h are ensured by the SIM interface hardware module, with $t_g = \frac{1680}{f}$ and $t_h = \frac{700}{f}$.

10.13.4.2.4 Deactivation

The UICC deactivation sequence is shown in Figure 10-34.

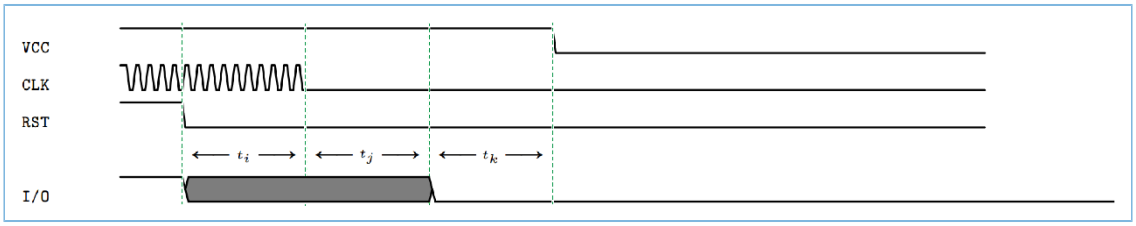

Figure 10-34 Deactivation

The times t_i and t_j are ensured by the SIM interface hardware module, with $t_i = \frac{100}{f}$ and $t_j = \frac{100}{f_{mod}}$.

The timing t_k depends on the VCC source.

10.13.4.3 Data Link Layer

10.13.4.3.1 Elementary Time Unit (ETU)

The nominal duration of one moment on the electrical circuit I/O is named Elementary Time Unit (ETU).

$$\frac{F}{D} * \frac{1}{f} \tag{8}$$

The ETU shall be equal to F/D clock cycles on the electrical circuit CLK where F and D are the transmission parameters. F is the clock rate conversion integer and D is the baud rate adjustment integer. F, D and f are negotiated during the initialization and setup by firmware. Possible values for F and D are shown in Table 10-213.

Table 10-213 Values for F and D

F	D	F/D
372	1	372
372	2	186
372	3	124
372	4	93
512	1	512
512	2	256
512	4	128
512	8	64
512	16	32
512	32	16
512	64	8

f is configured as described in Section 10.13.4.1.1 CLK.

The chosen ETU has to be written to register field **CLK_CFG.ETU_DIV**.

$$etu\_div = \frac{F}{D} - 1 \tag{9}$$

Fractional ETU times are not supported.

10.13.4.3.2 Character Frame

The character transmission timing is shown in Figure 10-35. A character consists of:

- Start Bit
- 8 Data Bits
- Parity Bit
- 2 Stop Bits with error signaling

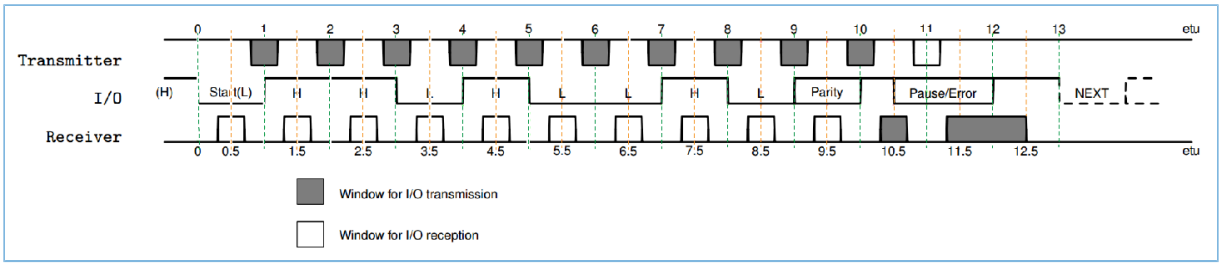

Figure 10-35 Character Timing

10.13.4.3.3 Error Signal and Character Repetition

- **RX**

In case of detected parity error during reception, the behavior depends on register field **STAT.RX_RETRY_MAX**. If the number of retries is not exceeded, the I/O pin is pulled low for one clock cycle to trigger a retransmission. Otherwise or if RETRY_MAX = 0, the received byte is discarded and the reception is aborted with a **STAT.IRQ_RETRY_ERR** interrupt. The internal retry counter is restarted with every character.

The maximum number of retries during the reception of one byte is displayed in register field **STAT.RX_RETRY_MAX** and cleared via **CTRL.RX_RETYR_MC**.

- **TX**

In case of an error signaling from the receiver, the behavior depends on register field **STAT.TX_RETRY_MAX**. If the number of retries is not exceeded, the character is retransmitted. Otherwise or if RETRY_MAX = 0, the transmission is aborted with a **STAT.IRQ_RETRY_ERR** interrupt. The internal retry counter is restarted with every character.

The maximum number of retries during the transmission of one byte is displayed in register field **STAT.TX_RETRY_MAX** and cleared via **CTRL.TX_RETYR_MC**.

10.13.4.3.4 Guard Time and Wait Time

The guard time specifies the minimum time between the leading edge of two consecutive characters, which has to be taken into account by the transmitter. The wait time specifies the maximum time between the leading edge of two consecutive characters, which has to be monitored by the receiver before issuing a timeout.

The guard time is adjusted by register field **TIMES_CFG.GUARD_TIME** and respected before transmitting a character. This includes the scenario, when the SIM interface has received a character and starts a transmission afterwards. The minimal guard time is 12. Register field **TIMES_CFG.GUARD_TIME** values smaller than that are ignored.

The wait time is adjusted by register field **TIMES_CFG.WAIT_TIME** and respected before aborting a character reception.

10.13.4.4 Answer to Reset

After a cold or warm reset the UICC answers with the Answer to Reset (ATR). The module receives the ATR and decodes the first byte TS. TS marks the used encoding. If enabled in register field data_cfg.detect_coding the coding is set accordingly in **DATA_CFG.CODING**. All subsequent bytes are decoded with ISO/IEC 7816-3.

The class selection procedure and the Protocol and Parameter Select (response to the ATR) (PPS) procedure is controlled by firmware. After the PPS procedure, the firmware is responsible to configure the transport protocol, the clock frequency, the I/O mode and the ETU.

10.13.4.5 Receive and Transmit Buffer

The SIM interface hardware module writes all received data bytes via a Byte DMA to a memory. Transmitted data bytes are read via the same Byte DMA.

10.13.4.5.1 Byte DMA

The Byte DMA accesses memory addresses word-wise and interacts with the I/O module on a byte base. In transmitting process, one entire word is read and served byte-wise. Irrelevant bytes are simply ignored. In receiving process, the word is cleared at the beginning and filled byte-wise. After the 4th byte or after receiving the last byte, the entire word is written to the memory.

10.13.4.5.2 Address Ranges

Receive and transmit buffer share the same start address (register **START_ADDR**), which has to be aligned to **4**.

The internal word address counting is limited to **18** Bit. Therefore the maximum receive and transmit buffer sizes are limited to $2^{(18+2)}$ Bytes (1 MB). The buffer shall not cross a 1 MB address boundary.

10.13.4.5.3 Receive Buffer

The receive buffer end address is controlled by register **RX_END_ADDR** and has to be calculated according to (10) for an expected number of bytes N_{RX}.

$$rx\_end\_addr = strt\_addr + N_{RX} - 1 \quad (10)$$

A done interrupt will be issued after receiving N_{RX} bytes.

If less bytes are received and the wait time is expired, an **rx_err** interrupt is triggered. The register addr points to the last received byte, which can be used to calculate the number of received bytes n_{rx}.

$$n_{rx} = addr - strt\_addr + 1 \quad (11)$$

If more bytes are served by the UICC than expected, an **rx_err** interrupt is triggered too, but the incoming bytes are ignored and addr stays untouched.

10.13.4.5.4 Transmit Buffer

The transmit buffer end address is controlled by register **TX_END_ADDR** and has to be calculated according to (12) for an expected number of bytes NTX.

$$tx\_end\_addr = strt\_addr + N_{TX} - 1 \quad (12)$$

10.13.4.5.5 DMA Error

A DMA error interrupt **dma_err** is triggered under the following conditions:

- Invalid address
- Bus slave responses with an error
- Word could not be read within one ETU
- Word could not be written within one character

10.13.5 Action State

By default, the SIM interface hardware module does nothing, unless the firmware requests an action by writing register field **CTRL.ACTION**.

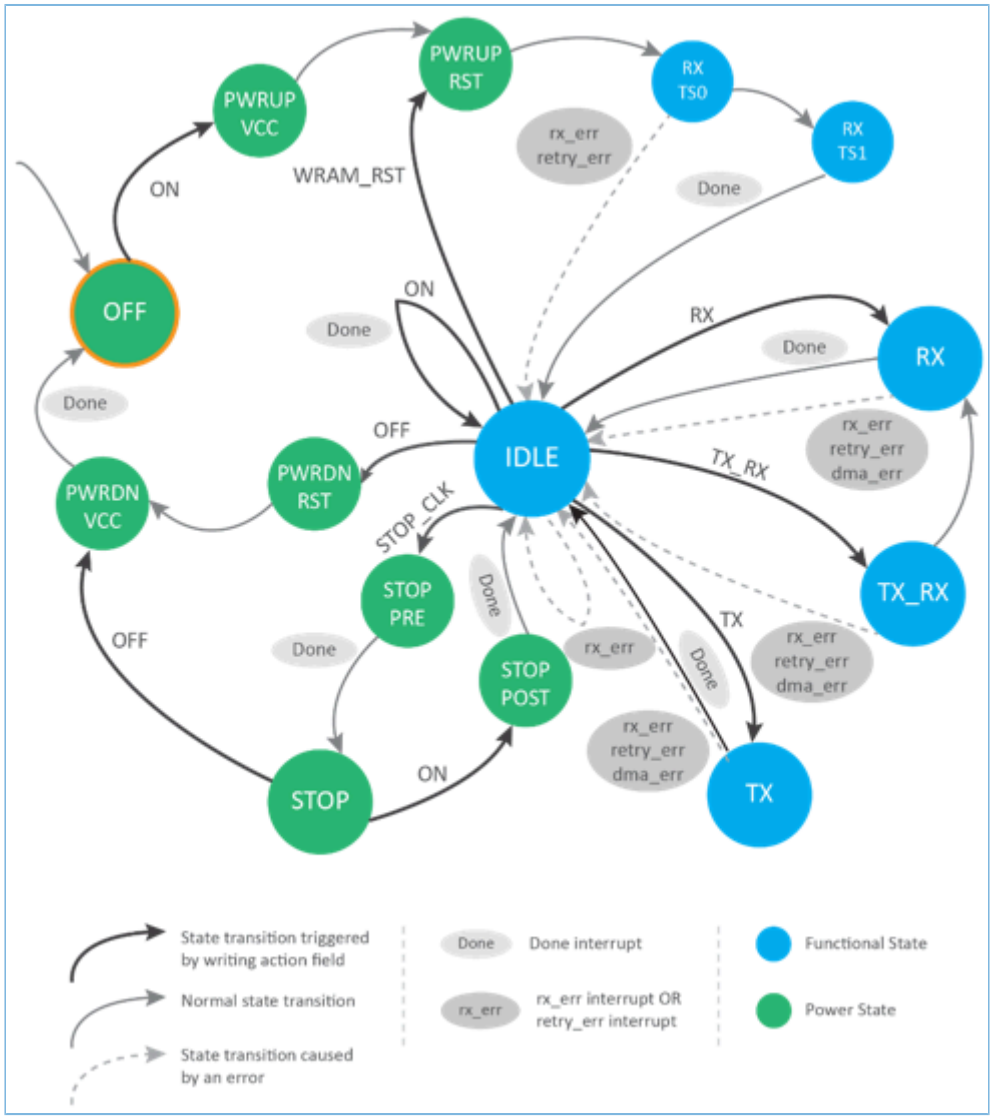

Figure 10-36 Main SIM Interface states

The main state machine is shown in Figure 10-36, with the following major states:

- **OFF:** UICC is unpowered.
- **IDLE:** UICC is powered and ready to communicate.
- **RX:** Receive characters from UICC and store them in the receive buffer.
- **TX:** Read characters from the transmit buffer and transmit them to UICC.
- **TX_RX:** Read characters from the transmit buffer and transmit them to UICC, continue with RX.
- **STOP:** UICC is powered but clock is stopped.

The main state machine operation bases on the following rules:

- The reset state is OFF.
- The action NONE (0) does nothing.

- The states OFF, IDLE, and STOP are inactive states with busy = 0.
- All other states are active states with busy = 1.
- The busy flag is visible in register field **STAT.BUSY**.
- It is only allowed to request the state transitions in Figure 10-36.
- Any other action will trigger a state_err interrupt, but not abort the ongoing action.
- On entering of an inactive state a done interrupt is triggered (if no error occurs).
- In case of an rx_err, retry_err or dma_err error, the corresponding interrupt is issued, and the idle state is entered.
- Requesting ON during IDLE does nothing except triggering a done interrupt.

10.13.6 Transport Layer

The module supports the automatic transmission and reception of multiple bytes. With the inbuilt DMA the TX data is take from or the RX data is written directly into the memory. For TX the firmware writes the payload into the memory and configures the module with payload start address and payload length. The module transfers the payload automatically and signalizes the end of the transmission with an interrupt. For RX the firmware configures the module with expected payload length and payload address. The module writes the received payload at the configured memory address. The number of received bytes is stored inside the module and interrupts for begin of transmission and end of transmission are generated.

ISO/IEC 7816-3 defines three types of data transport modes:

- Plain mode
- T = 0 mode
- T = 1 mode

10.13.6.1 Plain Mode

In plain mode, the module transmits or receives multiple bytes controlled by software. Thus the module is explicitly set in TX, RX or TX followed by RX mode.

10.13.6.2 T = 0 Mode

T = 0 mode is not implemented in hardware on purpose. SIM implementations vary in details. Therefore the implementation is done in firmware using plain mode.

10.13.6.3 T = 1 Mode

T = 1 mode is not implemented in hardware on purpose. SIM implementations vary in details. Therefore the implementation is done in firmware using plain mode.

10.13.7 Registers

10.13.7.1 CTRL

- **Name:** Control Register
- **Description:** This register is only to write.
- **Base Address:** 0xA000F200
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 10-214 Control Register

Bits	Field Name	RW	Reset	Description
31	IRQ_TEST_SET	W	0x0	Interrupt Test Set. This register sets test interrupt **Value:** • 0x0: No effect • 0x1: Set
30	IRQ_TEST_CLR	W	0x0	Interrupt Test Clear. This register clears test interrupt (has higher priority than 'IRQ_TEST_SET') **Value:** • 0x0: No effect • 0x1: Clear
29:26	RSVD	W		Reserved bits
25	IRQ_PRESENCE_CLR	W	0x0	Interrupt source presence clear. This register clears interrupt source 'STAT.IRQ_PRESENCE'. **Value:** • 0x0: No effect • 0x1: Clear
24	IRQ_STAT_EC	W	0x0	Interrupt source state_err clear. This register clears interrupt source 'STAT.IRQ_STAT_ERR'. **Value:** • 0x0: No effect • 0x1: Clear
23	IRQ_DMA_EC	W	0x0	Interrupt source dma_err clear. This register clears interrupt source 'STAT.IRQ_DMA_ERR'. **Value:** • 0x0: No effect

Bits	Field Name	RW	Reset	Description
				• 0x1: Clear
22	IRQ_RETYR_EC	W	0x0	Interrupt source retry_err clear. This register clears interrupt source 'STAT.IRQ_RETYR_ERR'. **Value:** • 0x0: No effect • 0x1: Clear
21	IRQ_RX_EC	W	0x0	Interrupt source rx_err clear. This register clears interrupt source 'STAT.IRQ_RX_ERR'. **Value:** • 0x0: No effect • 0x1: Clear
20	IRQ_DONE_CLR	W	0x0	Interrupt source done clear. This register clears interrupt source 'STAT.IRQ_DONE'. **Value:** • 0x0: No effect • 0x1: Clear
19:13	RSVD	W		Reserved bits
12	TX_RETYR_MC	W	0x0	Transmit Retries Maximum Clear. This register clears 'STAT.TX_RETRY_MAX'. **Value:** • 0x0: No effect • 0x1: Clear
11:9	RSVD	W		Reserved bits
8	RX_RETYR_MC	W	0x0	Receive Retries Maximum Clear. This register clears 'STAT.RX_RETRY_MAX'. **Value:** • 0x0: No effect • 0x1: Clear
7:3	RSVD	W		Reserved bits
2:0	ACTION	W	0x0	**Value:** • 0x0: Do Nothing. • 0x1: Switch off. • 0x2: Stop the clock. • 0x3: Switch on and receive ATR. Re-enable clock if clock is stopped.

Bits	Field Name	RW	Reset	Description
				• 0x4: Trigger warm reset and receive ATR. • 0x5: Receive. • 0x6: Transmit. • 0x7: Transmit, followed by RX.

10.13.7.2 STAT

- **Name:** Status Register
- **Description:** This register record interrupt, IO and power status.
- **Base Address:** 0xA000F200
- **Offset:** 0x04
- **Reset Value:** 0x00020000

Table 10-215 Status Register

Bits	Field Name	RW	Reset	Description
31	RSVD	R		Reserved bits
30	IRQ_TEST	R	0x0	Interrupt Test. Test interrupt for connection check. **Value:** • 0x0: No interrupt • 0x1: Active
29:26	RSVD	R		Reserved bits
25	IRQ_PRESENCE	R	0x0	Interrupt Presence. SIM card presence changed. Inserted or removed. **Value:** • 0x0: No interrupt • 0x1: Active
24	IRQ_STAT_ERR	R	0x0	Interrupt State Error. Action requested while busy or unsupported transition. **Value:** • 0x0: No interrupt • 0x1: Active
23	IRQ_DMA_ERR	R	0x0	Interrupt DMA Error. DMA read/write operation could not be issued. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: No interrupt • 0x1: Active
22	IRQ_RETRY_ERR	R	0x0	Interrupt Retry Error. Maximum number of retries exceeded. **Value:** • 0x0: No interrupt • 0x1: Active
21	IRQ_RX_ERR	R	0x0	Interrupt RX Error. No or incomplete or unexpected data. **Value:** • 0x0: No interrupt • 0x1: Active
20	IRQ_DONE	R	0x0	Interrupt Done. Requested operation has been completed. **Value:** • 0x0: No interrupt • 0x1: Active
19:18	RSVD	R		Reserved bits
17	PRESENCE_STAT	R	0x1	Status of presence IO. **Value:** • 0x0: Absent • 0x1: Presence
16	BUSY	R	0x0	Status of SIM interface. **Value:** • 0x0: Idle • 0x1: Busy
15	RSVD	R		Reserved bits
14:12	TX_RETRY_MAX	R	0x0	Transmit Retries Maximum. Maximum number of seen transmit retries after error signaling by SIM.
11	RSVD	R		Reserved bits
10:8	RX_RETRY_MAX	R	0x0	Receive Retries Maximum. Maximum number of seen receive retries after parity error.
7	RSVD	R		Reserved bits
6:4	IO_STAT	R	0x0	IO States.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: IO pin is driven low. • 0x1: IO pin is high impedance and driven high by the pull-up. • 0x4: Wait for start bit. • 0x5: Receive. • 0x6: Transmit. • 0x7: Wait until next character to ensure configured guard time.
3:0	PWR_STAT	R	0x0	Power States. **Value:** • 0x0: SIM is unpowered. • 0x1: Power up SIM. RST asserted (low). Clock stopped. IO is tristate. • 0x2: Power up SIM. RST asserted (low). Clock is running. IO is high. • 0x3: Power up SIM. RST asserted (low). Clock is running. IO is tristate. • 0x4: Power up SIM. RST asserted (low). Clock stopped. IO is low. • 0x5: Preparing clock stop. • 0x6: Clock stopped. • 0x7: Exiting clock stop • 0x8: SIM is idle, no communication is ongoing. • 0x9: RX TS0 Character. • 0xA: RX TS1 Character. • 0XB: Receive. • 0xC: Transmit. • 0xD: Transmit and Receive.

10.13.7.3 CLK_CFG

- **Name:** Clock Configuration Register
- **Description:** This register configure clock.
- **Base Address:** 0xA000F200
- **Offset:** 0x08
- **Reset Value:** 0x002F0173

Table 10-216 Clock Configuration Register

Bits	Field Name	RW	Reset	Description
31	CLK_STOP_SEL	RW	0x0	Clock Stop Select. Value of the clock output during stopped clock. **Value:** • 0x0: Low • 0x1: High
30:24	RSVD	R		Reserved bits
23:16	CLK_DIV	RW	0x2F	Clock Division. Divide system clock by this value + 1.
15:10	RSVD	R		Reserved bits
9:0	ETU_DIV	RW	0x173	Divide SIM clock by this value+1 to define ETU length. The reset value is the one, needed for the ATR.

10.13.7.4 TIMES_CFG

- **Name:** Times Configuration Register
- **Description:** This register configures the timing
- **Base Address:** 0xA000F200
- **Offset:** 0x10
- **Reset Value:** 0x006B0000

Table 10-217 Times Configuration Register

Bits	Field Name	RW	Reset	Description
31:30	RSVD	R		Reserved bits
29:12	WAIT_TIME	RW	0x6B	Wait time in [ETU]. Maximum card response time (leading edge to leading edge).
11:10	RSVD	R		Reserved bits
9:0	GUARD_TIME	RW	0x0	Guard time in [ETU]. Time between the leading edges of two consecutive characters.

10.13.7.5 DATA_CFG

- **Name:** Data Configuration Register
- **Description:** This register configures the communication
- **Base Address:** 0xA000F200

- **Offset:** 0x14
- **Reset Value:** 0x00000002

Table 10-218 Data Configuration Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6:4	RETRY_LIMIT	RW	0x0	Retries Limit. Maximum number of issued retries before giving up.
3:2	RSVD	R		Reserved bits
1	DETECT_CODING	RW	0x1	Detect Coding Convention. Automatically detect coding convention during ATR reception. **Value:** • 0x0: Disable • 0x1: Enable
0	CODING	RW	0x0	Coding Convention. **Value:** • 0x0: Default. High = 1, LSB first. • 0x1: Inverse. High = 0, MSB first.

10.13.7.6 ADDR

- **Name:** Address Register
- **Description:** This register records current address relative to base_addr.
- **Base Address:** 0xA000F200
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 10-219 Address Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19:2	ADDR	R	0x0	Address. Current address relative to base_addr.
1:0	ADDR_FRAC	R	0x0	Address Fraction. Byte selection.

10.13.7.7 START_ADDR

- **Name:** Start Address Register
- **Description:** This register configures read/write memory address. RX and TX buffer has to be aligned to 4 bytes, and buffer address [31:2] = START_ADDR[31:2]
- **Base Address:** 0xA000F200
- **Offset:** 0x1c
- **Reset Value:** 0x00000000

Table 10-220 Start Address Register

Bits	Field Name	RW	Reset	Description
31:20	BASE_ADDR	RW	0x0	Base Address. Base Address for RX and TX Buffer.
19:2	START_ADDR	RW	0x0	Start Address. Start address of RX and TX buffer, relative to base_addr.
1:0	RSVD	R		Reserved bits

10.13.7.8 RX_END_ADDR

- **Name:** RX End Register
- **Description:** This register configures RX data end address
- **Base Address:** 0xA000F200
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 10-221 RX End Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19:2	RX_END_ADDR	RW	0x0	RX End Address. End address of receive buffer, relative to base_addr.
1:0	RX_END_AF	RW	0x0	RX End Address Fraction. Byte selection.

10.13.7.9 TX_END_ADDR

- **Name:** TX End Register
- **Description:** This register configures TX data end address
- **Base Address:** 0xA000F200

- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 10-222 TX End Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19:2	TX_END_ADDR	R	0x0	TX End Address. End address of transmit buffer, relative to base_addr.
1:0	TX_END_AF	R	0x0	TX End Address Fraction. Byte selection.

10.13.8 Electrical Specifications

Table 10-223 Electrical Specifications

Symbol	Description	Min.	Typ.	Max.	Unit
f_{SIMCLK}	UICC CLK frequency	1		5	MHz
VCC	UICC Voltage class supported	1.8		3.8	V

10.14 DMA

10.14.1 Introduction

GR551x has one instance of Direct Memory Access (DMA) Controller that performs fast data transfers from/to peripherals to/from on-chip SRAM over the on-chip bus. DMA Controller is highly programmable and allows the CPU to offload the burden of serving the slow-rate peripherals.

10.14.2 Main Features

- 8 DMA Channels
- 16 handshaking Interfaces
- 1 AHB Master Interface for moving data
- 1 AHB Slave Interface for configuration
- Maximum block size of 4 K for each channel
- Low power mode with global clock gating and channel clock gating

10.14.3 Functional description

Figure 10-37 shows the following functional groupings of the main interfaces to the DMA Controller block:

- DMA hardware request interface

- 8 channels
- 32 bytes FIFO per channel for source and destination
- Arbiter
- AHB master interface
- AHB slave interface

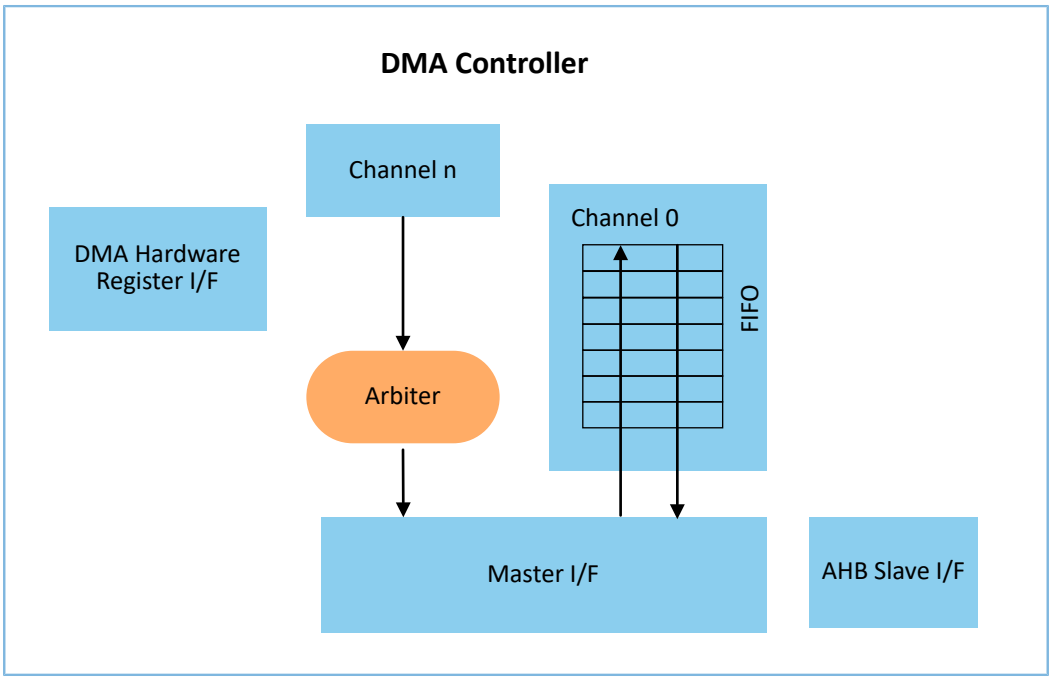

Figure 10-37 DMA Controller block diagram

Figure 10-38 illustrates a peripheral-to-peripheral DMA transfer, where peripheral A (source) uses a hardware handshaking interface, and peripheral B (destination) uses a software handshaking interface. For example, the request to send data to peripheral B is originated by the CPU, while writing to peripheral B is handled by the DMA Controller. The channel source and destination arbitrate independently for the AHB master interface, along with other channels.

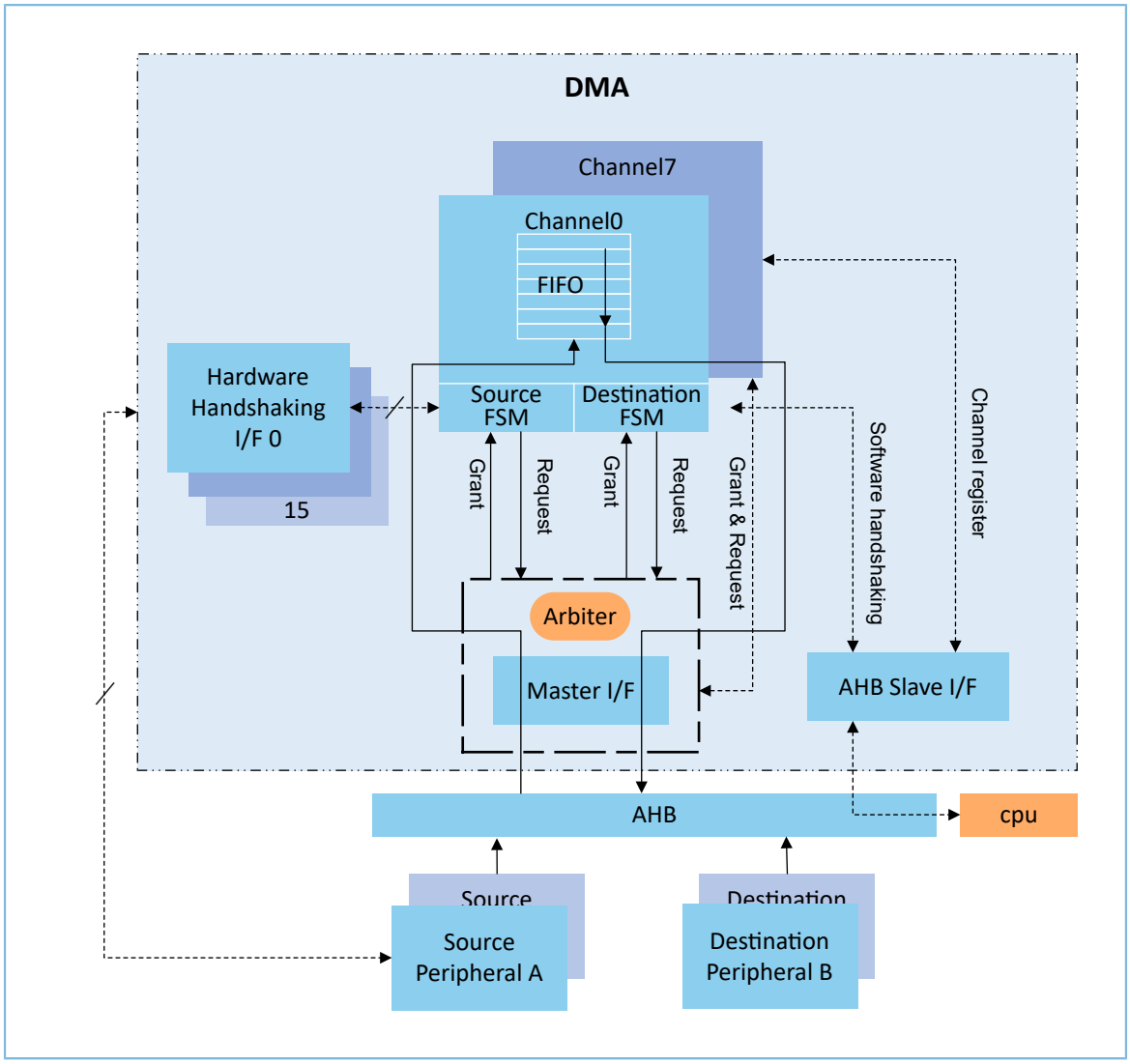

Figure 10-38 Peripheral-to-Peripheral DMA Transfer on Same AHB Layer

10.14.3.1 DMA Hardware Interface Assignment

Each channel can then communicate with the peripheral connected to that interface through the assigned hardware handshaking interface, the hardware handshaking interfaces are assigned as shown in Table 10-224. The developer can program the **CFG_CHx.SRC_PER** or **CFG_CHx.DEST_PER** to assign a handshaking interface for each channel source and destination.

Table 10-224 DMAC Hardware Interface Assignment

CFG_CHx.DEST_PER	CFG_CHx.SRC_PER	Peripheral Name	Peripheral Description
0	M2P: any value (0 – 14) P2P: any value except 0	SPIM TX	SPI Master Transmit
P2M: any value (0 – 14) P2P: any value except 1	1	SPIM RX	SPI Master Receive

CFG_CHx.DEST_PER	CFG_CHx.SRC_PER	Peripheral Name	Peripheral Description
2	M2P: any value (0 – 14) P2P: any value except 2	SPIS TX	SPI Slave Transmit
P2M: any value (0 – 14) P2P: any value except 3	3	SPIS RX	SPI Slave Receive
4	M2P: any value (0 – 14) P2P: any value except 4	QSPI0 TX	QSPI Master Transmit
P2M: any value (0 – 14) P2P: any value except 5	5	QSPI0 RX	QSPI Master Receive
6	M2P: any value (0 – 14) P2P: any value except 6	I2C0 TX	I2C Transmit
P2M: any value (0 – 14) P2P: any value except 7	7	I2C0 RX	I2C Receive
8	M2P: any value (0 – 14) P2P: any value except 8	I2C1 TX or I2S_S TX	I2C or I2S Slave Transmit
P2M: any value (0 – 14) P2P: any value except 9	9	I2C1 RX or I2S_S RX	I2C or I2S Slave Receive
10	M2P: any value (0 – 14) P2P: any value except 10	UART0 TX	UART Transmit
P2M: any value (0 – 14) P2P: any value except 11	11	UART0 RX	UART Receive
12	M2P: any value (0 – 14) P2P: any value except 12	QSPI1 TX or I2S_M TX	QSPI Master or I2S Master Transmit
P2M: any value (0 – 14) P2P: any value except 13	13	QSPI1 RX or I2S_M RX	QSPI Master or I2S Master Receive
P2M: any value (0 – 14) P2P: any value except 14	14	SNSADC	Sensor ADC

10.14.3.2 DMA Setting Up Transfers

10.14.3.2.1 Transfer Type Flow Control

GR551x only support DMA as flow controller. There are four transfer types as follow:

- Memory to memory
- Memory to peripheral
- Peripheral to memory

- Peripheral to peripheral

10.14.3.2.2 Transfer Width

For memory, source transfer width must be less than or equal to AHB master interface data bus width, normally 8, 16 or 32 bits. For a non-memory, source transfer width is equal to the peripheral FIFO Width.

For memory, destination transfer width must be less than or equal to AHB master interface data bus width, normally 8, 16 or 32 bits. For a non-memory, destination transfer width is equal to the peripheral FIFO Width.

10.14.3.2.3 Source and Destination Address Increment

- Source address increment: indicates whether to increase or decrease the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to **No change**.
- Destination address increment: whether to increment or decrement the destination address on every source transfer. If the device is writing data to a destination peripheral FIFO with a fixed address, then set this field to **No change**.

10.14.3.2.4 Channel Priority

Each channel has a programmable priority in range 0 – 7, lowest to highest. Channel priority work for most scenario except in master bus. It is worth noting that a request for the master bus interface can be made at any time, but is granted only after the current AHB transfer (burst or single) is completed. Therefore, when the master interface is transferring data for a low priority channel while a high priority channel requests service, it completes the current AHB transfer (burst or single) for the low priority channel before switching to transfer data for the high one.

10.14.3.2.5 DMA Transfer

DMA has 8 channels and each channel has a 32-bit FIFO. The 8 FIFOs are independent of each other. After the DMA channel is enabled, the DMA obtains data from the source according to the bit width configured for the source port, stores the data into the DMA FIFO, takes out the data from the DMA FIFO according to the bit width configured for the destination port, and sends the data to the destination port. Regardless of the configured transmission bit width, the data obtained from the source is compactly stored, which can also maximize the use of DMA FIFO storage space. For example, the transmission bit width is 16-bit wide and the FIFO bit width is 32-bit wide, so that one depth of DMA FIFO will store two pieces of 16-bit wide data. When DMA is used as flow control, the total size of data in bytes (blk_size_bytes_dma) transmitted by DMA in a single transmission is equal to the product of the source transmission bit width (src_single_size_bytes) and the block size (BLOCK_XFE_SIZE).

$$src\_single\_size\_bytes = CTRL\_CHx.SRC\_XFE\_WIDTH / 8$$

$$blk\_size\_bytes\_dma = BLOCK\_XFE\_SIZE * src\_single\_size\_bytes$$

Table 10-225 Programmable data width and endian behavior

Source Port Bit Width	Destination Port Bit Width	Block Size (DMA as Flow Control)	Source Content: Address/Data	Transfer Operations	Destination Content: Address/Data
8	8	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1: Read B0[7:0] @0x0 to DMA then write B0[7:0] @0x0 2: Read B1[7:0] @0x1 to DMA then write B1[7:0] @0x1 3: Read B2[7:0] @0x2 to DMA then write B2[7:0] @0x2 4: Read B3[7:0] @0x3 to DMA then write B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
8	16	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1: Read B0[7:0] @0x0 to DMA 2: Read B1[7:0] @0x1 to DMA then write B1B0[15:0] @0x0 3: Read B2[7:0] @0x2 to DMA 4: Read B3[7:0] @0x3 to DMA then write B3B2[15:0] @0x2	@0x0/B1B0 @0x2/B3B2
8	32	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1: Read B0[7:0] @0x0 to DMA 2: Read B1[7:0] @0x1 to DMA 3: Read B2[7:0] @0x2 to DMA 4: Read B3[7:0] @0x3 to DMA then write B3B2B1B0[31:0] @0x0	@0x0/B3B2B1B0
16	8	2	@0x0/B1B0 @0x2/B3B2	1: Read B1B0[15:0] @0x0 to DMA then write B0[7:0] @0x0 2: write B1[7:0] @0x1 3: Read B3B2[15:0] @0x2 to DMA then write B2[7:0] @0x2 4: write B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
16	16	2	@0x0/B1B0 @0x2/B3B2	1: Read B1B0[15:0] @0x0 to DMA then write B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 to DMA then write B3B2[15:0] @0x2	@0x0/B1B0 @0x2/B3B2
16	32	2	@0x0/B1B0 @0x2/B3B2	1: Read B1B0[15:0] @0x0 to DMA 2: Read B3B2[15:0] @0x2 to DMA then write B3B2B1B0[31:0] @0x0	@0x0/B3B2B1B0

Source Port Bit Width	Destination Port Bit Width	Block Size (DMA as Flow Control)	Source Content: Address/Data	Transfer Operations	Destination Content: Address/ Data
32	8	1	@0x0/B3B2B1B0	1: Read B3B2B1B0[31:0] @0x0 to DMA then write B0[7:0] @0x0 2: write B1[7:0] @0x1 3: write B2[7:0] @0x2 4: write B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
32	16	1	@0x0/B3B2B1B0	1: Read B3B2B1B0[31:0] @0x0 to DMA then write B1B0[15:0] @0x0 2: write B3B2[15:0] @0x2	@0x0/B1B0 @0x2/B3B2
32	32	1	@0x0/B3B2B1B0	1: Read B3B2B1B0[31:0] @0x0 to DMA then write B3B2B1B0[31:0] @0x0	@0x0/B3B2B1B0

10.14.3.3 Multi-block DMA transfer

Multi-block DMA transfers are supported through Auto-reloading.

During Auto-reloading, the channel registers are reloaded with their initial values at the completion of each block and the new values used for the new block.

Some or all of the **SRC_ADDR_CHx**, **DEST_ADDR_CHx**, and **CTRL_CHx** channel registers are reloaded from their initial value at the start of a block transfer.

By configuring the Configuration Register (**CFG_CHx.RELOAD_SRC**), you can program if the **SRC_ADDR_CHx** register can be automatically reloaded from its initial value at the end of every block for multi-block transfers.

By configuring the Configuration Register (**CFG_CHx.RELOAD_DST**), you can program if the **DEST_ADDR_CHx** register can be automatically reloaded from its initial value at the end of every block for multi-block transfers.

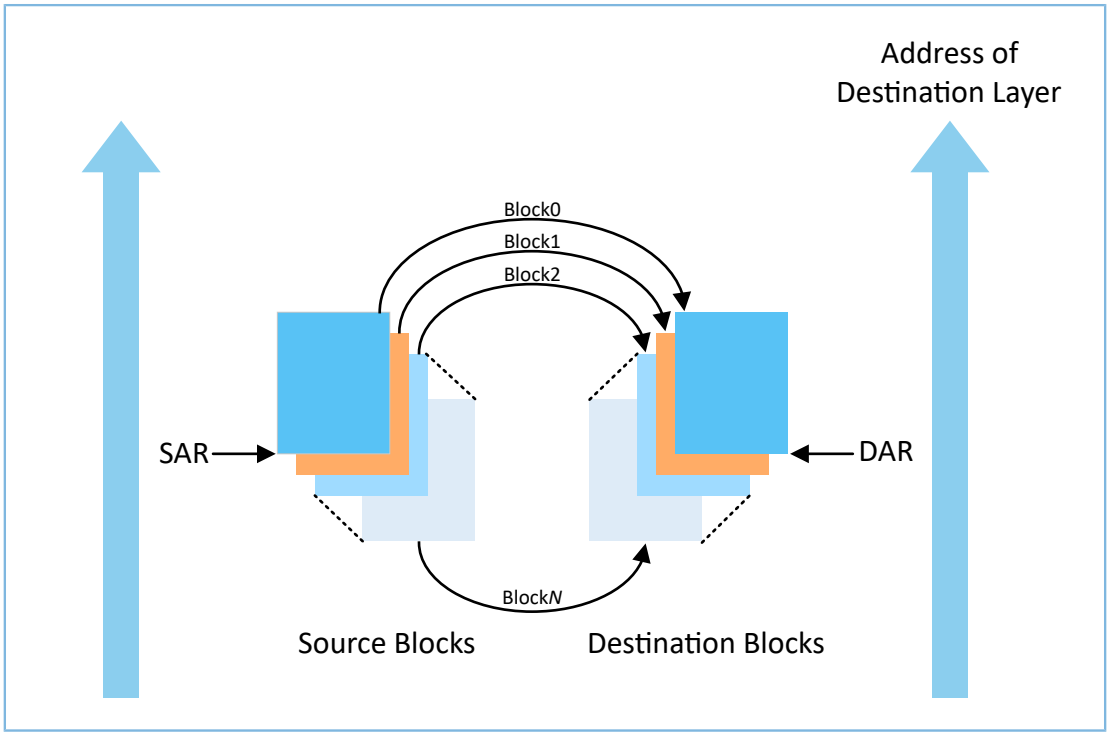

Figure 10-39 Multi-Block DMA Transfer with Source and Destination Address Auto-Reloaded

10.14.3.4 DMA Interrupt

The **INT_RSTAT**, **INT_STAT**, **INT_MASK**, **INT_CLR** registers are provided to manage interrupts. The **INT_MASK** register is used to enable or disable the block transfer complete interrupt, DMA transfer complete interrupt and error interrupt. By reading the **INT_STAT** registers or **INT_RSTAT** registers, you can check whether the interruption occurred. To clear interrupts, write **1** to the **INT_CLR** registers.

- **INT_MASK** registers: enable or disable the block transfer complete interrupt, DMA transfer complete interrupt and error interrupt.
- **INT_STAT** or **INT_RSTAT** registers: reply whether the interruption occurred.
- **INT_CLR** registers: write **1** to **INT_CLR** to clear interrupts.

10.14.4 Registers

10.14.4.1 Channel_x_Registers

10.14.4.1.1 SRC_ADDR_CHx

- **Name:** Source Address for Channel x
- **Description:** The starting source address is programmed by software before the DMA channel is enabled. While the DMA transfer is in progress, this register is updated to reflect the source address of the current AHB transfer.

Note:

You must program the S_ADDR address to be aligned to CTRL_CHx.SRC_XFE_WIDTH.

- **Base Address:** 0xA0013000
- **Reset Value:** 0x0
- **Offset:** 0x0 + x*0x58

Table 10-226 Source Address for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	SRC_ADDR	RW	0x0	Current Source Address of DMA transfer. Updated after each source transfer. The SRC_ADDR_INC field in the CTRL_CHx register determines whether the address increments, decrements, or is left unchanged on every source transfer through the block transfer. **Volatile:** true

10.14.4.1.2 DEST_ADDR_CHx

- **Name:** Destination Address Register for Channel x
- **Description:** The starting destination address is programmed by software before the DMA channel is enabled. While the DMA transfer is in progress, this register is updated to reflect the destination address of the current AHB transfer.

Note:

You must program the DEST_ADDR_CH to be aligned to CTRL_CHx.DEST_XFE_WIDTH.

- **Base Address:** 0xA0013000
- **Offset:** 0x8 + x*0x58
- **Reset Value:** 0x0

Table 10-227 Destination Address Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	DEST_ADDR	RW	0x0	Current Destination address of DMA transfer. Updated after each destination transfer. The DEST_ADDR_INC field in the CTRL_CHx register determines whether the address increments,

Bits	Field Name	RW	Reset	Description
				decrements, or is left unchanged on every destination transfer throughout the block transfer. **Volatile:** true

10.14.4.1.3 CTRL_CHx

- **Name:** Control Register for Channel x
- **Description:** This register contains fields that control the DMA transfer.

Note:

You need to program this register prior to enabling the channel.

- **Base Address:** 0xA0013000
- **Offset:** 0x18 + x*0x58
- **Reset Value:** 0x0000000200304801

Table 10-228 Control Register for Channel x

Bits	Field Name	RW	Reset	Description
63:45	RSVD	R	0x0	Reserved field **Volatile:** true
44	DONE	RW	0x0	Done bit. If status write-back is enabled, the upper word of the control register, CTRL_CHx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTRL_CHx.DONE bit to see when a block transfer is completed. The LLI CTRL_CHx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel. LLI accesses are always 32-bit accesses (Hsize = 2) aligned to 32-bit boundaries and cannot be changed or programmed to anything other than 32-bit. **Volatile:** true
43:32	BLOCK_XFE_SIZE	RW	0x2	Block Transfer Size. When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_XFE_SIZE indicates the total number of single

Bits	Field Name	RW	Reset	Description
				transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. **Width:** The width of single transaction is determined by CTRL_CHx.SRC_XFE_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what the flow controller is. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at 0xFFF, but the actual block size can be greater. **Volatile:** true
31:23	Rsvd_1_CTRL	R	0x0	Reserved field **Volatile:** true
22:20	XFE_TYPE_FC	RW	0x3	Transfer Type and Flow Control. Flow control can be assigned to the DMA, the source peripheral, or the destination peripheral. **Value:** • 0x0: Transfer type is Memory to Memory and Flow Controller is DMA • 0x1: Transfer type is Memory to Peripheral and Flow Controller is DMA • 0x2: Transfer type is Peripheral to Memory and Flow Controller is DMA • 0x3: Transfer type is Peripheral to Peripheral and Flow Controller is DMA **Volatile:** true
19:17	Rsvd_CTRL	R	0x0	Reserved field **Volatile:** true
16:14	SRC_MSIZE	RW	0x1	Source Burst Transaction Length. Number of data items, each of width CTRL_CHx.SRC_XFE_WIDTH, to be read from the source every time a burst transferred request is made from either the corresponding hardware or software handshaking interface. **Note:** This value is not related to the AHB bus master HBURST bus. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Number of data items to be transferred is 1 • 0x1: Number of data items to be transferred is 4 • 0x2: Number of data items to be transferred is 8 **Volatile:** true
13:11	DEST_MSIZE	RW	0x1	Destination Burst Transaction Length. Number of data items, each of width CTRL_CHx.DEST_XFE_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. **Note:** This value is not related to the AHB bus master HBURST bus. **Value:** • 0x0: Number of data items to be transferred is 1 • 0x1: Number of data items to be transferred is 4 • 0x2: Number of data items to be transferred is 8 **Volatile:** true
10:9	SRC_ADDR_INC	RW	0x0	Source Address Increment. Indicate whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change". **Value:** • 0x0: Increments the source address • 0x1: Decrements the source address • 0x2: No change in the source address • 0x3: No change in the source address **Volatile:** true
8:7	DEST_ADDR_INC	RW	0x0	Destination Address Increment. Indicate whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to \"No Change\". **Value:** • 0x0: Increments the destination address • 0x1: Decrements the destination address

Bits	Field Name	RW	Reset	Description
				• 0x2: No change in the destination address • 0x3: No change in the destination address **Volatile:** true
6:4	SRC_XFE_WIDTH	RW	0x0	Source Transfer Width. Mapped to AHB bus Hsize. For a non-memory peripheral, typically the peripheral (source) FIFO width. **Value:** • 0x0: Source transfer width is 8 bits • 0x1: Source transfer width is 16 bits • 0x2: Source transfer width is 32 bits **Volatile:** true
3:1	DEST_XFE_WIDTH	RW	0x0	Destination Transfer Width. Mapped to AHB bus Hsize. For a non-memory peripheral, typically the peripheral (destination) FIFO width. **Value:** • 0x0: Destination transfer width is 8 bits • 0x1: Destination transfer width is 16 bits • 0x2: Destination transfer width is 32 bits **Volatile:** true
0	INT_EN	RW	0x1	Interrupt Enable Bit. If set, then all interrupt-generating sources are enabled. Functions as a global mask bit for all interrupts for the channel; raw* interrupt registers still assert if `CTRL_CHx.INT_EN = 0`. **Value:** • 0x0: Interrupt is disabled • 0x1: Interrupt is enabled **Volatile:** true

10.14.4.1.4 CFG_CHx

- **Name:** Configuration Register for Channel x
- **Description:** This register contains fields that configure the DMA transfer. The channel configuration register remains fixed for all blocks of a multi-block transfer.

Note:

You need to program this register prior to enabling the channel.

- **Base Address:** 0xA0013000
- **Offset:** 0x40 + x*0x58
- **Reset Value:** 0x0000000400000e00 + (x*0x20)

Table 10-229 Configuration Register for Channel x

Bits	Field Name	RW	Reset	Description
63:47	RSVD	R		Reserved field
46:43	DEST_PER	RW	0x0	Destination hardware interface. Assigns a hardware handshaking interface to the destination of channel x if the CFG_CHx.HSG_SEL_DEST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. **Note:** For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
42:39	SRC_PER	RW	0x0	**Source Hardware Interface.** Assigns a hardware handshaking interface to the source of channel x if the CFG_CHx.HSG_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. **Note:** For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
38:37	RSVD	R		Reserved field
36:34	PROT_CTRL	RW	0x0	Protection Control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches.

Bits	Field Name	RW	Reset	Description
				There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals. Mapping of HPROT bus is as follows: • 0x1 to HPROT[0] • CFG_CHx.PROT_CTRL[1] to HPROT[1] • CFG_CHx.PROT_CTRL[2] to HPROT[2] • CFG_CHx.PROT_CTRL[3] to HPROT[3]
33	FIFO_MODE	RW	0x0	FIFO Mode Select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced. **Value:** • 0x0: Space/data available for single AHB transfer of the specified transfer width • 0x1: Data available is greater than or equal to half the FIFO depth for destination transfers and space available is greater than half the FIFO depth for source transfers. The exceptions are at the end of a burst transaction request or at the end of a block transfer.
32	FLOW_CTRL_MODE	RW	0x0	Flow Control Mode. Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller. **Value:** • 0x0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled • 0x1: Source transaction requests are not serviced until a destination transaction request occurs. In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.
31	RELOAD_DEST	RW	0x0	Automatic Destination Reload. The DEST_ADDR_CHx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. **Value:** • 0x0: Destination Reload Disabled • 0x1: Destination Reload Enabled
30	RELOAD_SRC	RW	0x0	Automatic Source Reload.

Bits	Field Name	RW	Reset	Description
				The SRC_ADDR_CHx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. **Value:** • 0x0: Source Reload Disabled • 0x1: Source Reload Enabled
29:20	Rsvd_CFG	R	0x0	Reserved field
19	SRC_HSG_POL	RW	0x0	Source Handshaking Interface Polarity. **Value:** • 0x0: Source Handshaking Interface Polarity is Active high • 0x1: Source Handshaking Interface Polarity is Active low
18	DEST_HSG_POL	RW	0x0	Destination Handshaking Interface Polarity. **Value:** • 0x0: Destination Handshaking Interface Polarity is Active high • 0x1: Destination Handshaking Interface Polarity is Active low
17:12	Rsvd_CFG	R	0x0	Reserved field
11	HSG_SEL_SRC	RW	0x0	Source Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces - hardware or software - is active for source requests on this channel. If the source peripheral is memory, then this bit is ignored. **Value:** • 0x0: Hardware handshaking interface. Software initiated transaction requests are ignored. • 0x1: Software handshaking interface. Hardware initiated transaction requests are ignored.
10	HSG_SEL_DEST	RW	0x0	Destination Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces - hardware or software - is active for destination requests on this channel. If the destination peripheral is memory, then this bit is ignored. **Value:** • 0x0: Hardware handshaking interface. Software initiated transaction requests are ignored. • 0x1: Software handshaking interface. Hardware initiated transaction requests are ignored.
9	FIFO_EMPTY	R	0x0	Channel FIFO status.

Bits	Field Name	RW	Reset	Description
				Indicate if there is data left in the channel FIFO. It can be used in conjunction with CFG_CHx to cleanly disable a channel. **Value:** • 0x0: Channel FIFO is not empty • 0x1: Channel FIFO is empty
8	CH_SUSP	RW	0x0	Channel Suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. It can also be used in conjunction with CFG_CHx to cleanly disable a channel without losing any data. **Value:** • 0x0: DMA transfer from the source is not suspended • 0x1: Suspend DMA transfer from the source
7:5	CH_PRIOR	RW		Channel Priority. A priority of 7 is the highest priority and 0 is the lowest. This field must be programmed within the range 0 to 3. A programmed value outside this range will cause erroneous behavior. **Value:** • 0x0: Channel priority is 0 • 0x1: Channel priority is 1 • 0x2: Channel priority is 2 • 0x3: Channel priority is 3
4:0	RSVD	R		Reserved field

10.14.4.2 Interrupt_Registers

10.14.4.2.1 INT_RSTAT_TC

- **Name:** Raw Status for Transfer Complete Interrupt
- **Description:** Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated to per channel; for example, INT_RSTAT_TC[2] is the Channel 2 raw transfer complete interrupt.
- Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_TC register.

Note:

Write access is available to this register or software for testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address:** 0xA0013000
- **Offset:** 0x2c0
- **Reset Value:** 0x0

Table 10-230 Raw Status for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	Rsvd_INT_RSTAT_TC	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw Status for Transfer Complete Interrupt **Value:** • 0x0: Inactive Raw Interrupt Status • 0x1: Active Raw Interrupt Status

10.14.4.2.2 INT_RSTAT_BTC

- **Name:** Raw Status for Block Transfer Complete Interrupt
- **Description:** Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated to per channel; for example, INT_RSTAT_BTC[2] is the Channel 2 raw block complete interrupt.

  Each bit in this register is cleared by writing a 1 to the corresponding location in the CLR_BLK register.

Note:

Write access is available to this register or software for testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address:** 0xA0013000
- **Offset:** 0x2c8
- **Reset Value:** 0x0

Table 10-231 Raw Status for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_BT	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw Status for Block Transfer Complete Interrupt **Value:** • 0x0: Inactive Raw Interrupt Status • 0x1: Active Raw Interrupt Status

10.14.4.2.3 INT_RSTAT_STC

- Name: Raw Status for Source Transaction Complete Interrupt

- Description: Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, INT_RSTAT_STC[2] is the Channel 2 raw source transaction complete interrupt.
- Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_STC register.

Note:

Write access is available to this register or software testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address:** 0xA0013000
- **Offset:** 0x2D0
- **Reset Value:** 0x0

Table 10-232 Raw Status for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_STC	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw Status for Source Transaction Complete Interrupt **Value:** • 0x0: Inactive Raw Interrupt Status • 0x1: Active Raw Interrupt Status

10.14.4.2.4 INT_RSTAT_DTC

- **Name:** Raw Status for Destination Transaction Complete Interrupt
- **Description:** Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, INT_RSTAT_DTC[2] is the Channel 2 raw destination transaction complete interrupt.

  Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_DTC register.

Note:

Write access is available to this register or software testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address:** 0xA0013000
- **Offset:** 0x2d8
- **Reset Value:** 0x0

Table 10-233 Raw Status for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_DT	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw Status for Destination Transaction Complete Interrupt **Value:** • 0x0: Inactive Raw Interrupt Status • 0x1: Active Raw Interrupt Status

10.14.4.2.5 INT_RSTAT_ERR

- **Name:** Raw Status for Error Interrupt
- **Description:** Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, INT_RSTAT_ERR[2] is the Channel 2 raw error interrupt.

  Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_ERR register.

Note:

Write access is available to this register or software testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address:** 0xA0013000
- **Offset:** 0x2E0
- **Reset Value:** 0x0

Table 10-234 Raw Status for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_ER	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw Status for Error Interrupt **Value:** • 0x0: Inactive Raw Interrupt Status • 0x1: Active Raw Interrupt Status

10.14.4.2.6 INT_STAT_TC

- **Name:** Status for Transfer Complete Interrupt
- **Description:** Channel DMA Transfer complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, INT_STAT_TC [2] is the Channel 2 source DMA transfer complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

- **Base Address:** 0xA0013000
- **Offset:** 0x2E8
- **Reset Value:** 0x0

Table 10-235 Status for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_TC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for Transfer Complete Interrupt **Value:** • 0x0: Inactive Interrupt Status • 0x1: Active Interrupt Status

10.14.4.2.7 INT_STAT_BTC

- **Name:** Status for Block Transfer Complete Interrupt
- **Description:** Channel Block complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, INT_STAT_BTC[2] is the Channel 2 block complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.
- **Base Address:** 0xA0013000
- **Offset:** 0x2F0
- **Reset Value:** 0x0

Table 10-236 Status for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_BTC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for Block Transfer Complete Interrupt **Value:** • 0x0: Inactive Interrupt Status • 0x1: Active Interrupt Status

10.14.4.2.8 INT_STAT_STC

- **Name:** Status for Source Transaction Complete Interrupt
- **Description:** Channel Source Transaction complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, INT_STAT_STC[2] is the Channel 2 source transaction complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

- **Base Address:** 0xA0013000
- **Offset:** 0x2F8
- **Reset Value:** 0x0

Table 10-237 Status for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_STC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for Source Transaction Complete Interrupt **Value:** • 0x0: Inactive Interrupt Status • 0x1: Active Interrupt Status

10.14.4.2.9 INT_STAT_DTC

- **Name:** Status for Destination Transaction Complete Interrupt
- **Description:** Channel destination transaction complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, INT_STAT_DTC[2] is the Channel 2 status destination transaction complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.
- **Base Address:** 0xA0013000
- **Offset:** 0x300
- **Reset Value:** 0x0

Table 10-238 Status for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_DTC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for Destination Transaction Complete Interrupt **Value:** • 0x0: Inactive Interrupt Status • 0x1: Active Interrupt Status

10.14.4.2.10 INT_STAT_ERR

- **Name:** Status for Error Interrupt
- **Description:** Channel Error interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, INT_STAT_ERR[2] is the Channel 2 status Error interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

- **Base Address:** 0xA0013000
- **Offset:** 0x308
- **Reset Value:** 0x0

Table 10-239 Status for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_ERR	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for Error Interrupt **Value:** • 0x0: Inactive Interrupt Status • 0x1: Active Interrupt Status

10.14.4.2.11 INT_MASK_TC

- **Name:** MASK for Transfer Complete Interrupt
- **Description:** The contents of the Raw Status register INT_RSTAT_TC is masked with the contents of the Mask register INT_MASK_TC. Each bit of register is allocated per channel; for example, INT_MASK_TC[2] is the mask bit for the Channel 2 transfer complete interrupt.

  A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_TC register writes a 1 into INT_MASK_TC[0], while INT_MASK_TC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_TC[7:0] unchanged.

  Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.
- **Base Address:** 0xA0013000
- **Offset:** 0x310
- **Reset Value:** 0x0

Table 10-240 MASK for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_TC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value:** • 0x0: Interrupt mask write disable • 0x1: Interrupt mask write enable
7:0	INT_MASK	RW	0x0	Mask for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: Mask the interrupts • 0x1: Unmask the interrupts

10.14.4.2.12 INT_MASK_BTC

- **Name:** Mask for Block Transfer Complete Interrupt
- **Description:** The contents of the Raw Status register INT_RSTAT_BTC is masked with the contents of the Mask register INT_MASK_BTC. Each bit of register is allocated per channel; for example, INT_MASK_BTC[2] is the mask bit for the Channel 2 block complete interrupt.

  A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_BTC register writes a 1 into INT_MASK_BTC[0], while INT_MASK_BTC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_BTC[7:0] unchanged.

  Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.
- **Base Address:** 0xA0013000
- **Offset:** 0x318
- **Reset Value:** 0x0

Table 10-241 Mask for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_BTC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value:** • 0x0: Interrupt mask write disable • 0x1: Interrupt mask write enable
7:0	INT_MASK	RW	0x0	Mask for Block Transfer Complete Interrupt **Value:** • 0x0: Mask the interrupts • 0x1: Unmask the interrupts

10.14.4.2.13 INT_MASK_STC

- **Name:** MASK for Source Transaction Complete Interrupt

- **Description:** The contents of the Raw Status register INT_RSTAT_STC is masked with the contents of the Mask register INT_MASK_STC. Each bit of register is allocated per channel; for example, INT_MASK_STC[2] is the mask bit for the Channel 2 source transaction complete interrupt.

  When the source peripheral of DMA channel i is memory, then the source transaction complete interrupt, INT_MASK_STC[i], must be masked to prevent an erroneous triggering of an interrupt on the int_combined signal.

  A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, write hex 01x1 to the INT_MASK_STC register and write a 1 into INT_MASK_STC[0], while INT_MASK_STC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_STC[7:0] unchanged.

  Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

- **Base Address:** 0xA0013000
- **Offset:** 0x320
- **Reset Value:** 0x0

Table 10-242 MASK for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_STC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value:** • 0x0: Interrupt mask write disable • 0x1: Interrupt mask write enable
7:0	INT_MASK	RW	0x0	Mask for Source Transaction Complete Interrupt **Value:** • 0x0: Mask the interrupts • 0x1: Unmask the interrupts

10.14.4.2.14 INT_MASK_DTC

- **Name:** Mask for Destination Transaction Complete Interrupt

- **Description:** The contents of the Raw Status register INT_RSTAT_DTC is masked with the contents of the Mask register INT_MASK_DTC. Each bit of register is allocated per channel; for example, INT_MASK_DTC[2] is the mask bit for the Channel 2 destination transaction complete interrupt.

  When the destination peripheral of DMA channel i is memory, then the destination transaction complete interrupt, INT_MASK_DTC[i], must be masked to prevent an erroneous triggering of an interrupt on the int_combined(_n) signal.

  A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_DTC register writes a 1 into INT_MASK_DTC[0], while INT_MASK_DTC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_DTC[7:0] unchanged.

  Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

- **Base Address:** 0xA0013000
- **Offset:** 0x328
- **Reset Value:** 0x0

Table 10-243 Mask for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_DTC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value:** • 0x0: Interrupt mask write disable • 0x1: Interrupt mask write enable
7:0	INT_MASK	RW	0x0	Mask for Destination Transaction Complete Interrupt **Value:** • 0x0: Mask the interrupts • 0x1: Unmask the interrupts

10.14.4.2.15 INT_MASK_ERR

- **Name:** Mask for Error Interrupt
- **Description:** The contents of the Raw Status register INT_RSTAT_ERR is masked with the contents of the Mask register INT_MASK_ERR. Each bit of register is allocated per channel; for example, INT_MASK_ERR[2] is the mask bit for the Channel 2 error interrupt.

  A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_ERR register writes a 1 into

INT_MASK_ERR[0], while INT_MASK_ERR[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_ERR[7:0] unchanged.

Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

- **Base Address:** 0xA0013000
- **Offset:** 0x330
- **Reset Value:** 0x0

Table 10-244 Mask for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_ERR	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value:** • 0x0: Interrupt mask write disable • 0x1: Interrupt mask write enable
7:0	INT_MASK	RW	0x0	Mask for Error Interrupt **Value:** • 0x0: Mask the interrupts • 0x1: Unmask the interrupts

10.14.4.2.16 INT_CLR_TC

- **Name:** Clear for Transfer Complete Interrupt
- **Description:** Each bit in the INT_RSTAT_TC and INT_STAT_TC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. Each bit is allocated per channel; for example, INT_CLR_TC[2] is the clear bit for the Channel 2 transfer done interrupt. Writing a 0 has no effect. This registers are not readable.
- **Base Address:** 0xA0013000
- **Offset:** 0x338
- **Reset Value:** 0x0

Table 10-245 Clear for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_TC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Transfer Complete Interrupt **Value:** • 0x0: No effect

Bits	Field Name	RW	Reset	Description
				• 0x1: Clears interrupts

10.14.4.2.17 INT_CLR_BTC

- **Name:** Clear for Block Transfer Complete Interrupt
- **Description:** Each bit in the INT_RSTAT_BTC and INT_STAT_BTC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. Each bit is allocated per channel; for example, INT_CLR_BTC[2] is the clear bit for the Channel 2 block done interrupt. Writing a 0 has no effect. These registers are not readable.
- **Base Address:** 0xA0013000
- **Offset:** 0x340
- **Reset Value:** 0x0

Table 10-246 Clear for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_BTC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Block Transfer Complete Interrupt

10.14.4.2.18 INT_CLR_STC

- **Name:** Clear for Source Transaction Complete Interrupt
- **Description:** Each bit in the INT_RSTAT_STC and INT_STAT_STC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. Each bit is allocated per channel; for example, INT_CLR_STC[2] is the clear bit for the Channel 2 source transaction done interrupt. Writing a 0 has no effect. This registers are not readable.
- **Base Address:** 0xA0013000
- **Offset:** 0x348
- **Reset Value:** 0x0

Table 10-247 Clear for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_STC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Source Transaction Complete Interrupt **Value:** • 0x0: No effect • 0x1: Clears interrupts

10.14.4.2.19 INT_CLR_DTC

- **Name:** Clear for Destination Transaction Complete Interrupt
- **Description:** Each bit in the INT_RSTAT_DTC and INT_STAT_DTC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. Each bit is allocated per channel; for example, INT_CLR_DTC[2] is the clear bit for the Channel 2 destination transaction done interrupt. Writing a 0 has no effect. This registers are not readable.
- **Base Address:** 0xA0013000
- **Offset:** 0x350
- **Reset Value:** 0x0

Table 10-248 Clear for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_DTC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Destination Transaction Complete Interrupt **Value:** • 0x0: No effect • 0x1: Clears interrupts

10.14.4.2.20 INT_CLR_ERR

- **Name:** Clear for Error Interrupt
- **Description:** Each bit in the INT_RSTAT_ERR and INT_STAT_ERR is cleared on the same cycle by writing a 1 to the corresponding location in the registers. Each bit is allocated per channel; for example, INT_CLR_ERR[2] is the clear bit for the Channel 2 error interrupt. Writing a 0 has no effect. This registers are not readable.
- **Base Address:** 0xA0013000
- **Offset:** 0x358
- **Reset Value:** 0x0

Table 10-249 Clear for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_ERR	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Error Interrupt **Value:** • 0x0: No effect • 0x1: Clears interrupts

10.14.4.2.21 INT_STAT_ET

- **Name:** Status for each Interrupt type

- **Description:** The contents of each of the five Status registers INT_STAT_TC, INT_STAT_BTC, INT_STAT_STC, INT_STAT_DTC, INT_STAT_ERR are ORed to produce a single bit for each interrupt type in the Combined Status register (INT_STAT_ET). This register is read-only.
- **Base Address:** 0xA0013000
- **Offset:** 0x360
- **Reset Value:** 0x0

Table 10-250 Status for each Interrupt type

Bits	Field Name	RW	Reset	Description
63:5	RSVD_INT_STAT_ET	R	0x0	Reserved field
4	ERR	R	0x0	OR of the contents of INT_STAT_ERR **Value:** • 0x0: OR of the contents of INT_STAT_ERR register is 0 • 0x1: OR of the contents of INT_STAT_ERR register is 1
3	DEST_XFE_CPLT	R	0x0	OR of the contents of INT_STAT_DTC **Value:** • 0x0: OR of the contents of INT_STAT_DTC register is 0 • 0x1: OR of the contents of INT_STAT_DTC register is 1
2	SRC_XFE_CPLT	R	0x0	OR of the contents of INT_STAT_STC **Value:** • 0x0: OR of the contents of INT_STAT_STC register is 0 • 0x1: OR of the contents of INT_STAT_STC register is 1
1	BLK_XFE_CPLT	R	0x0	OR of the contents of INT_STAT_BTC register **Value:** • 0x0: OR of the contents of INT_STAT_BTC register is 0 • 0x1: OR of the contents of INT_STAT_BTC register is 1
0	XFE_CPLT	R	0x0	OR of the contents of INT_STAT_TC register **Value:** • 0x0: OR of the contents of INT_STAT_TC register is 0 • 0x1: OR of the contents of INT_STAT_TC register is 1

10.14.4.3 Software_Handshake_Registers

10.14.4.3.1 REQ_SST

- **Name:** Source Software Transaction Request register

- **Description:** A bit is assigned for each channel in this register. REQ_SST[n] is ignored when software handshaking is not enabled for the source of channel n.

  A channel SRC_REQ bit is written only if the corresponding channel write enable bit in the SRC_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register. For example, writing hex 0101 writes a 1 into REQ_SST[0], while REQ_SST[7:1] remains unchanged. Writing hex 00xx leaves REQ_SST[7:0] unchanged. This allows software to set a bit in the REQ_SST register without performing a read-modified write operation.

- **Base Address:** 0xA0013000
- **Offset:** 0x368
- **Reset Value:** 0x0

Table 10-251 Source Software Transaction Request register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_SST	R	0x0	Reserved field
15:8	SRC_REQ_WE	RW	0x0	Source Software Transaction Request write enable **Value:** • 0x0: Source request write Disable • 0x1: Source request write Enable
7:0	SRC_REQ	RW	0x0	Source Software Transaction Request **Value:** • 0x0: Source request is not active • 0x1: Source request is active

10.14.4.3.2 REQ_DST

- **Name:** Destination Software Transaction Request Register
- **Description:** A bit is assigned for each channel in this register. REQ_DST[n] is ignored when software handshaking is not enabled for the source of channel n.

  A channel DEST bit is written only if the corresponding channel write enable bit in the DEST_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.

- **Base Address:** 0xA0013000
- **Offset:** 0x370
- **Reset Value:** 0x0

Table 10-252 Destination Software Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_DST	R	0x0	Reserved field

Bits	Field Name	RW	Reset	Description
15:8	DEST_REQ_WE	RW	0x0	Destination Software Transaction Request write enable **Value:** • 0x0: Destination request write Disable • 0x1: Destination request write Enable
7:0	DEST_REQ	RW	0x0	Destination Software Transaction Request **Value:** • 0x0: Destination request is not active • 0x1: Destination request is active

10.14.4.3.3 REQ_SGL_ST

- **Name:** Source Single Transaction Request Register
- **Description:** A bit is assigned for each channel in this register. REQ_SGL_ST is ignored when software handshaking is not enabled for the source of channel n.

  A channel SRC_SGL_REQ bit is written only if the corresponding channel write enable bit in the SRC_SGL_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.
- **Base Address:** 0xA0013000
- **Offset:** 0x378
- **Reset Value:** 0x0

Table 10-253 Source Single Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_SGL_ST	R	0x0	Reserved field
15:8	SRC_SGL_REQ_WE	RW	0x0	Source Single Transaction Request write enable **Value:** • 0x0: Single write Disable • 0x1: Single write Enable
7:0	SRC_SGL_REQ	RW	0x0	Source Single Transaction Request **Value:** • 0x0: Source request is not active • 0x1: Source request is active

10.14.4.3.4 REQ_SGL_DT

- **Name:** Destination Single Transaction Request Register

- **Description:** A bit is assigned for each channel in this register. REQ_SGL_DT is ignored when software handshaking is not enabled for the destination of channel n.

  A channel DEST_SGL_REQ bit is written only if the corresponding channel write enable bit in the DST_SGL_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.

- **Base Address:** 0xA0013000
- **Offset:** 0x380
- **Reset Value:** 0x0

Table 10-254 Destination Single Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_SGL_DT	R	0x0	Reserved field
11:8	DST_SGL_REQ_WE	RW	0x0	Destination Single Transaction Request write enable **Value:** • 0x0: Destination write Disable • 0x1: Destination write Enable
7:0	DEST_SGL_REQ	RW	0x0	Destination Single Transaction Request **Value:** • 0x0: Destination Single or burst request is not active • 0x1: Destination Single or burst request is active

10.14.4.3.5 REQ_LST_ST

- **Name:** Source Last Transaction Request Register
- **Description:** A bit is assigned for each channel in this register. REQ_LST_ST is ignored when software handshaking is not enabled for the source of channel n, or when the source of channel n is not a flow controller.

  A channel LST_SRC bit is written only if the corresponding channel write enable bit in the LST_SRC_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.

- **Base Address:** 0xA0013000
- **Offset:** 0x388
- **Reset Value:** 0x0

Table 10-255 Source Last Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_LST_ST	R	0x0	Reserved field
15:8	LST_SRC_WE	RW	0x0	Source Last Transaction Request write enable **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Source last transaction request write disable • 0x1: Source last transaction request write enable
7:0	LST_SRC	RW	0x0	Source Last Transaction Request register **Value:** • 0x0: Not last transaction in current block • 0x1: Last transaction in current block

10.14.4.3.6 REQ_LST_DT

- **Name:** Destination Last Transaction Request Register
- **Description:** A bit is assigned for each channel in this register. REQ_LST_DT is ignored when software handshaking is not enabled for the destination of channel n or when the destination of channel n is not a flow controller.

  A channel LST_DEST bit is written only if the corresponding channel write enable bit in the LST_DEST_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.
- **Base Address:** 0xA0013000
- **Offset:** 0x390
- **Reset Value:** 0x0

Table 10-256 Destination Last Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_LST_DT	R	0x0	Reserved field
15:8	LST_DEST_WE	RW	0x0	Destination Last Transaction Request write enable **Value:** • 0x0: Destination last transaction request write disable • 0x1: Destination last transaction request write enable
7:0	LST_DEST	RW	0x0	Destination Last Transaction Request **Value:** • 0x0: Not last transaction in current block • 0x1: Last transaction in current block

10.14.4.4 Miscellaneous_Registers

10.14.4.4.1 CFG

- **Name:** DMA Configuration Register

- **Description:** This register is used to enable the DMA, which must be done before any channel activity can begin.

  If the global channel enable bit is cleared while any channel is still active, then CFG.DMA_EN still returns 1 to indicate that there are channels still active until hardware has terminated all activity on all channels, at which point the CFG.DMA_EN bit returns 0.

- **Base Address:** 0xA0013000
- **Offset:** 0x398
- **Reset Value:** 0x0

Table 10-257 DMA Configuration Register

Bits	Field Name	RW	Reset	Description
63:1	RSVD_CFG	R	0x0	Reserved field
0	DMA_EN	RW	0x0	DMA Enable bit. **Value:** • 0x0: DMA Disabled • 0x1: DMA Enabled

10.14.4.4.2 CH_EN

- **Name:** DMA Channel Enable Register
- **Description:** This is the DMA Channel Enable Register. If software needs to set up a new channel, then it can read this register in order to find out which channels are currently inactive; it can then enable an inactive channel with the required priority.

  All bits of this register are cleared to 0 when the global DMA channel enable bit, CFG[0], is 0. When the global channel enable bit is 0, then a write to the CH_EN register is ignored and a read will always read back 0.

  The channel enable bit, CH_EN.CH_EN, is written only if the corresponding channel write enable bit, CH_EN.CH_EN_WE, is asserted on the same AHB write transfer. For example, writing hex 01x1 writes a 1 into CH_EN[0], while CH_EN[7:1] remains unchanged. Writing hex 00xx leaves CH_EN[7:0] unchanged. Note that a read-modified write is not required.

- **Base Address:** 0xA0013000
- **Offset:** 0x3a0
- **Reset Value:** 0x0

Table 10-258 DMA Channel Enable Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_CH_EN	R	0x0	Reserved field
15:8	CH_EN_WE	W	0x0	Channel enable register

Bits	Field Name	RW	Reset	Description
7:0	CH_EN	RW	0x0	**Channel Enable.** The CH_EN.CH_EN bit is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer. **Value:** • 0x0: Disable the channel • 0x1: Enable the channel

10.15 PWM

10.15.1 Introduction

GR551x has two instances of Pulse Width Modulation (PWM) module. PWM generates successive pulses with variable duty cycles, which can be converted to analog levels by external discrete components, e.g. controlling LED brightness via a PWM signal.

10.15.2 Main Features

- Configurable output frequency and dynamic frequency configuration.
- Three independent PWM outputs: PWMA, PWMB and PWMC.
- Two operation modes: flicker mode and breath mode.
- Two alignment modes: edge alignment mode and center alignment mode.
- Configurable duty cycles in flicker mode.
- Three configuration modes: synchronize-all, synchronize-separate, and asynchronous.
- Polarity switching to support LED-positive and LED-negative modes.

10.15.3 Functional Description

PWM is capable of configurable output frequency dynamically.

PWM block supports three independent outputs: pwma, pwmb and pwmc.

PWM Block has two operation modes: flicker mode and breath mode.

In flicker mode, PWM module outputs successive pulses with certain frequency and duty as configured. In breath mode, the duty of PWM output periodically changes from 0% to 100%, and then 100% to 0% uniformly. The change period is configurable.

under flicker mode, the duty of PWM can be configured in the range of 0% to 100%. The three outputs can be configured independently and they share one output frequency.

Under breath mode, a configurable duration of breath hold state is supported. The breath hold state is set between two adjacent breath processes (i.e. duty change: 0%→100%→0%). In the hold state, the LED driven by PWM stays off.

In flicker mode, edge alignment and center alignment are supported.

Choose edge alignment mode, all open channels are aligned at the beginning of each duty cycle as shown in the figure below.

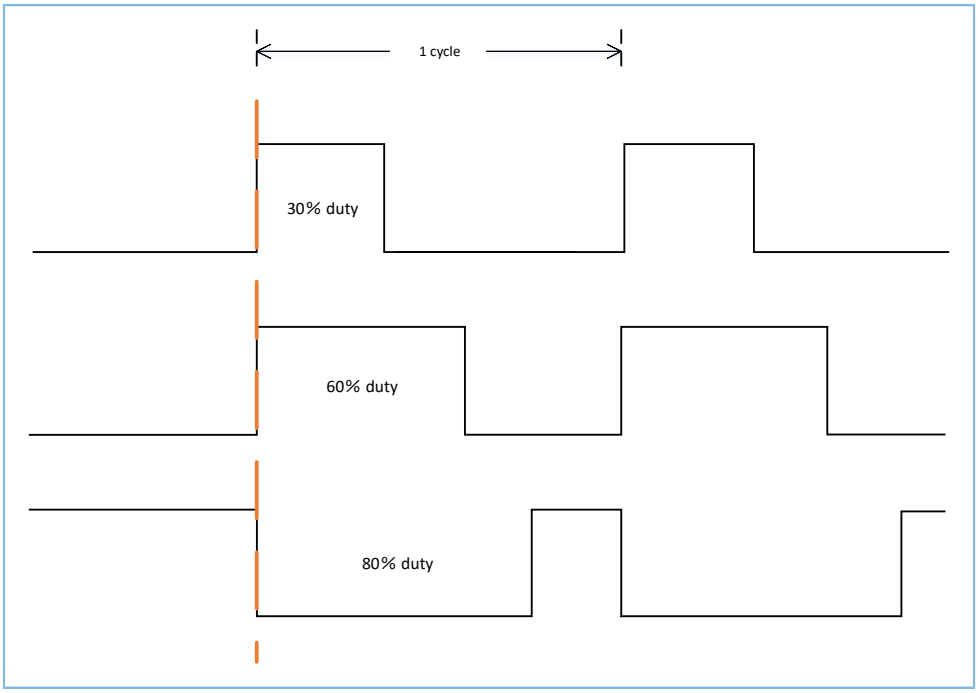

Figure 10-40 PWM edge alignment

Choose center alignment mode, all open channels are aligned in the middle of each duty cycle as shown in the figure below.

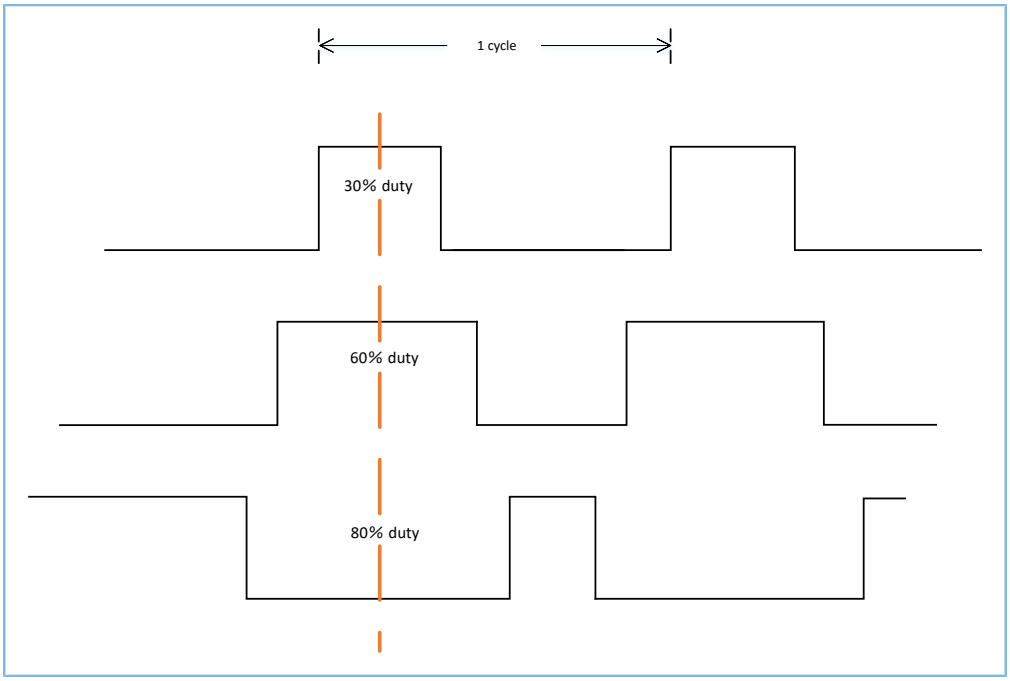

Figure 10-41 PWM center alignment

The configured data can be updated into active registers in three modes: synchronize-all mode (enabled by **UPDATE_SYNC_AE**), synchronize separate mode (enabled by **UPDATE_SYNC_SXX**) and asynchronous mode.

Both led-positive-drive mode and led-negative-drive mode are supported and can be configured in DRV_POS_x. In led-positive-drive mode, the LED driven by PWM is lightened when PWM outputs logic **1**, and goes off when PWM outputs logic **0**. In LED-negative mode, the contrary is the case.

10.15.3.1 Block Diagram

The PWM block diagram is illustrated in Figure 10-42.

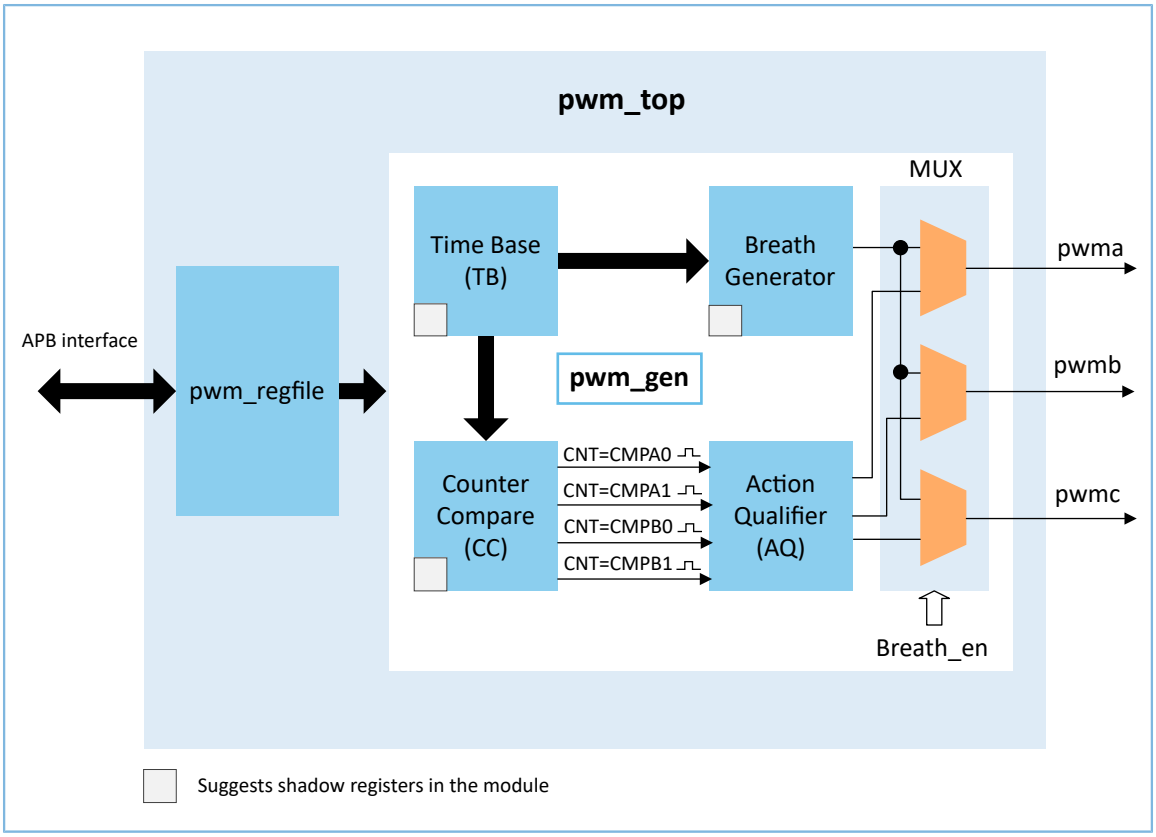

Figure 10-42 PWM Block Diagram

10.15.3.2 Operation

There are several shadow registers in PWM module. The write operation will first change the shadow registers, and then will update the value of active registers from the shadow registers under certain condition.

Active registers update conditions:

1. Under synchronize-all update mode (UPDATE_SYNC_AE = 0x1), all effective registers of flicker mode or breath mode will be updated simultaneously. In flicker mode (BREATH_EN = 0), after writing to **PRD**, **CMPXX** and **AQCTRL** will update from shadow registers simultaneously when the Time Base counter counts to PRD - 1.

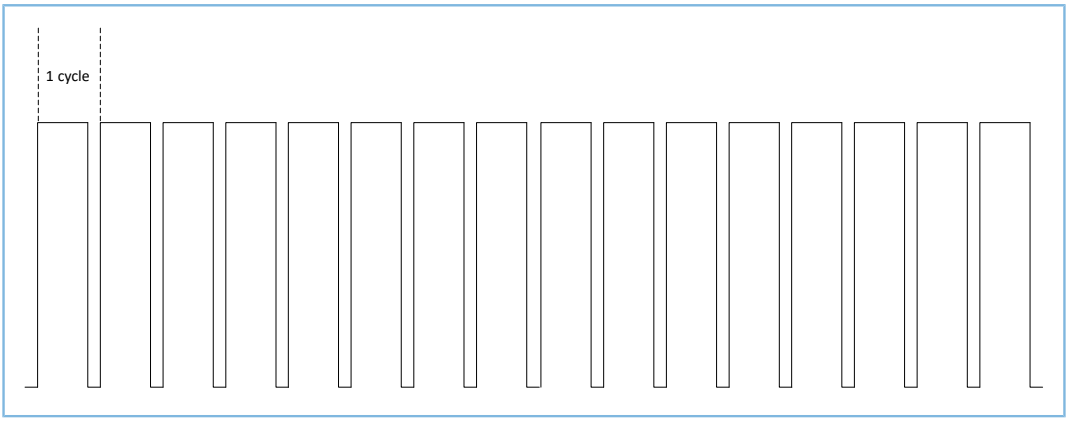

Figure 10-43 Flicker mode waveform

2. In breath mode (BREATH_EN = 1), after writing to **BRPRD**, **PRD**, **BRPRD** and **HOLD** will update from shadow registers simultaneously when the current breath process ends (the moment PWM duty reduces to 0%).

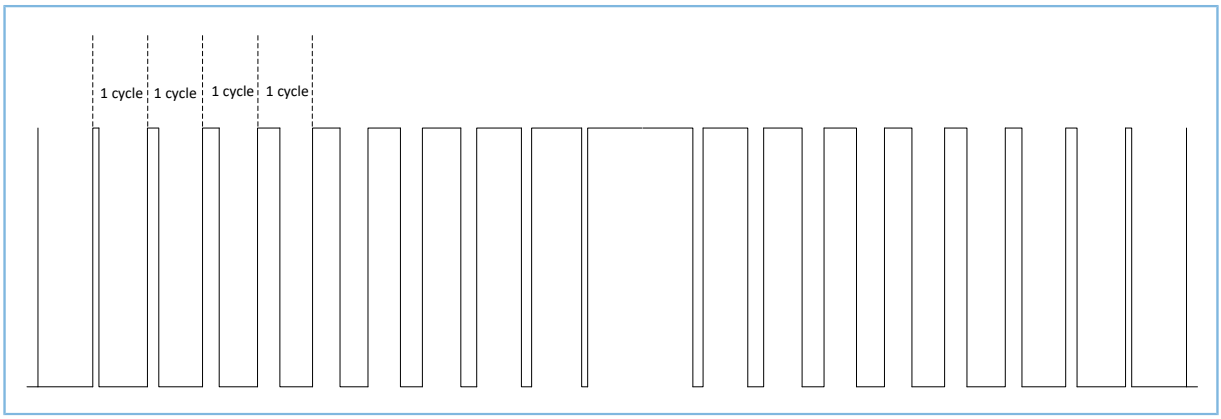

Figure 10-44 Breath mode waveform

Note:

PAUSE cannot be updated in this mode.

3. In synchronize-separate update mode (UPDATE_SYNC_SXX = 0x1), **CMPXX**, **AQCTRL** and **PAUSE** will update when the Time Base counter counts to PRD - 1 if the corresponding **UPDATE_SYNC_SXX** bit is set. **BRPRD** and **HOLD** will update when the current breath process ends and the corresponding **UPDATE_SYNC_SXX** bit is set. **PRD** will update when Time Base counter counts to PRD - 1 in flicker mode, and will update when current breath ends in breath mode if **UPDATE_SYNC_SPRD** is set.
4. In asynchronous update mode, all registers will update instantly.
5. On the rising edge of **FLIC_EN** or **BREATH_EN** (FLIC_EN = EN & BREATH_EN, BREATH_EN = EN & BREATH_EN), PRD will update instantly; on the rising edge of **FLIC_EN**, **CMPXX**, **AQCTRL** and **PAUSE** will update instantly; on the rising edge of **BREATH_EN**, **BRPRD** and **HOLD** will update instantly.

In initial state, **UPDATE** is configured to set update mode, then **PRD**, **CMPXX**, **AQCTRL**, **BRPRD** and **HOLD** registers are configured, and finally, MODE is configured to choose the operation mode and enable the module.

If the configurations need to be changed under sync-all update mode, CMPXX and AQCTRL in flicker mode, or HOLD in breath mode should be written before modifying the values of **PRD** and **BRPRD**.

10.15.3.3 Duty Cycle Calculation

The PWM duty cycle can be configured as an integer from 0 to 100, depending on the following calculation formulas of the duty cycle accuracy "n".

- In PWM_ALIGNED_EDGE mode:

$$n = f_{PWM} \div f_{SYSCLK} \times 100$$

- In PWM_ALIGNED_CENTER mode:

$$n = f_{PWM} \div f_{SYSCLK} \times 200$$

For example, when the clock frequency is 64 MHz and the duty cycle is in the range of 1 to 100, the maximum PWM frequency can only be set to 640 KHz in the PWM_ALIGNED_EDGE mode, and can only be set to 320 KHz in the PWM_ALIGNED_CENTER mode.

In a system clock with frequency 64 MHz, duty cycle errors at different PWM frequencies are as follows:

Table 10-259 Duty cycle error in PWM_ALIGNED_EDGE mode

PWM Frequency	PWM Duty Cycle Error
640 KHz	0.51%
320 KHz	0.31%
160 KHz	0.16%
80 KHz	0.08%
40 KHz	0.04%
20 KHz	0.02%
10 KHz	0.01%
5 KHz	0.01%
2.5 KHz	<0.01%

Table 10-260 Duty cycle error in PWM_ALIGNED_CENTER mode

PWM Frequency	PWM Duty Cycle Error
320 KHz	0.51%
160 KHz	0.16%
80 KHz	0.08%
40 KHz	0.04%
20 KHz	0.02%
10 KHz	0.01%
5 KHz	0.01%
2.5 KHz	<0.01%

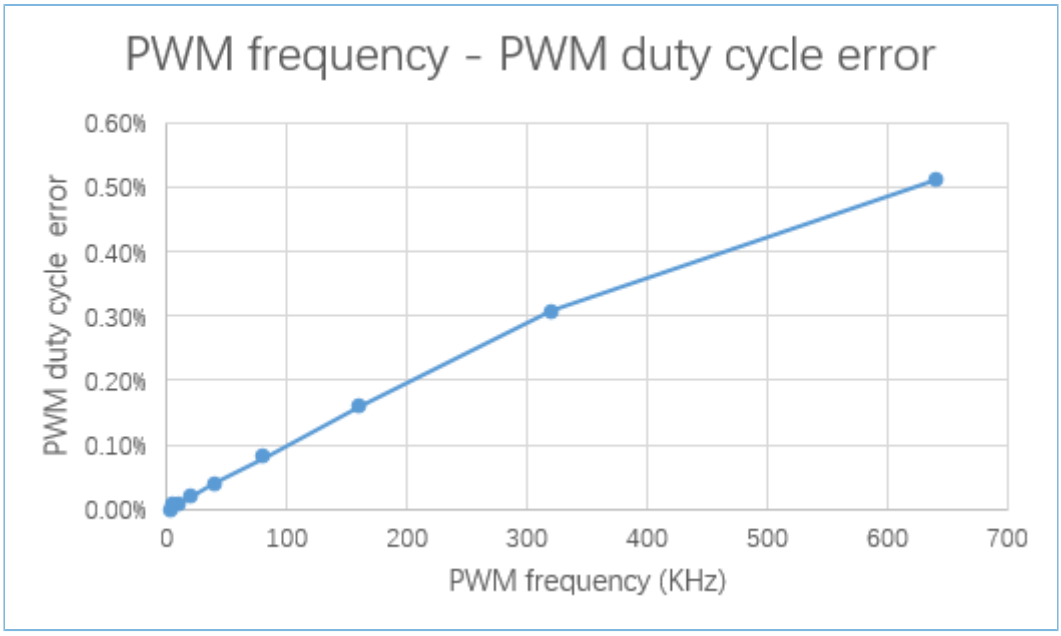

Figure 10-45 PWM frequency - PWM duty cycle error curve

10.15.4 Registers

10.15.4.1 MODE

- **Name:** Mode Register
- **Description:** This register sets PWM operation mode
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Note:

x* is used to identify which PWM Module is x^0, or x^1.

Table 10-261 Mode Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	PD_C_EN	RW	0x0	PWMC positive-drive mode enable. **Value:** • 0x0: negative-drive mode • 0x1: positive-drive mode
4	PD_B_EN	RW	0x0	PWMB positive-drive mode enable. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: negative-drive mode • 0x1: positive-drive mode
3	PD_A_EN	RW	0x0	PWMA positive-drive mode enable. **Value:** • 0x0: negative-drive mode • 0x1: positive-drive mode
2	BREATH_EN	RW	0x0	Breath mode enable. **Value:** • 0x0: flicker mode • 0x1: breath mode
1	PAUSE	RW	0x0	PWM pause signal. **Value:** • 0x0: Ongoing • 0x1: Pause
0	EN	RW	0x0	Enable PWM. **Value:** • 0x0: Disable • 0x1: Enable

10.15.4.2 UPDATE

- **Name:** Update Register
- **Description:** Synchronous update enable register
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 10-262 Update Register

Bits	Field Name	RW	Reset	Description
30x1:19	RSVD	R		Reserved bits
18	UPDATE_SYNC_SAQCTRL	RW	0x0	Separate update enable of AQCTRL **Value:** • 0x0: Disable • 0x1: Enable
17	UPDATE_SYNC_SHOLD	RW	0x0	Separate update enable of HOLD

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: Disable • 0x1: Enable
16	UPDATE_SYNC_SBRPRD	RW	0x0	Separate update enable of BRPRD **Value:** • 0x0: Disable • 0x1: Enable
15	UPDATE_SYNC_SPAUSE	RW	0x0	Separate update enable of PAUSE **Value:** • 0x0: Disable • 0x1: Enable
14	UPDATE_SYNC_SCMPC1	RW	0x0	Separate update enable of CMPC1 **Value:** • 0x0: Disable • 0x1: Enable
13	UPDATE_SYNC_SCMPC0	RW	0x0	Separate update enable of CMPC0 **Value:** • 0x0: Disable • 0x1: Enable
12	UPDATE_SYNC_SCMPB1	RW	0x0	Separate update enable of CMPB1 **Value:** • 0x0: Disable • 0x1: Enable
11	UPDATE_SYNC_SCMPB0	RW	0x0	Separate update enable of CMPB0 **Value:** • 0x0: Disable • 0x1: Enable
10	UPDATE_SYNC_SCMPA1	RW	0x0	Separate update enable of CMPA1 **Value:** • 0x0: Disable • 0x1: Enable
9	UPDATE_SYNC_SCMPA0	RW	0x0	Separate update enable of CMPA0 **Value:** • 0x0: Disable

Bits	Field Name	RW	Reset	Description
				• 0x1: Enable
8	UPDATE_SYNC_SPRD	RW	0x0	Separate update enable of PRD **Value:** • 0x0: Disable • 0x1: Enable
7:2	RSVD	R		Reserved bits
1	UPDATE_SYNC_AE	RW	0x0	Synchronous update enable **Value:** • 0x0: Disable • 0x1: Enable
0	UPDATE_SYNC_AG	R	0x0	Synchronous update ongoing **Value:** • 0x0: Not ongoing • 0x1: Ongoing

10.15.4.3 PRD

- **Name:** Period Register
- **Description:** This register is used to set the PWM cycle
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 10-263 Period Register

Bits	Field Name	RW	Reset	Description
31:0	PRD	RW	0x0	The period of PWM output, PRD=f_{CLK}/f_{PWM}

10.15.4.4 CMPA0

- **Name:** Compare A0 Register
- **Description:** Compare register 0 of channel A
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0xC
- **Reset Value:** 0x00000000

Table 10-264 Compare A0 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPA0	RW	0x0	PWMA duty control register0

10.15.4.5 CMPA1

- **Name:** Compare A1 Register
- **Description:** Compare register 1 of channel A
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 10-265 Compare A1 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPA1	RW	0x0	PWMA duty control register1

10.15.4.6 CMPB0

- **Name:** Compare B0 Register
- **Description:** Compare register 0 of channel B
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 10-266 Compare B0 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPB0	RW	0x0	PWMB duty control register0

10.15.4.7 CMPB1

- **Name:** Compare B1 Register
- **Description:** Compare register 1 of channel B
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 10-267 Compare B1 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPB1	RW	0x0	PWMB duty control register1

10.15.4.8 CMPC0

- **Name:** Compare C0 Register
- **Description:** Compare register 0 of channel C
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 10-268 Compare C0 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPC0	RW	0x0	PWMC duty control register0

10.15.4.9 CMPC1

- **Name:** Compare C1 Register
- **Description:** Compare register 1 of channel C
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 10-269 Compare C1 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPC1	RW	0x0	PWMC duty control register1

10.15.4.10 AQCTRL

- **Name:** Action Qualifier Control Register
- **Description:** This register is used to set actions when counter reaches the compare value.
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 10-270 Action Qualifier Control Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11:10	AQ_CTRL_C1	RW	0x0	Action of event CNT_CMPC1 control register; • 0x0: do nothing • 0x1: clear • 0x2: set • 0x3: toggle
9:8	AQ_CTRL_C0	RW	0x0	Action of event CNT_CMPC0 control register; • 0x0: do nothing • 0x1: clear • 0x2: set • 0x3: toggle
7:6	AQ_CTRL_B1	RW	0x0	Action of event CNT_CMPB1 control register; • 0x0: do nothing • 0x1: clear • 0x2: set • 0x3: toggle
5:4	AQ_CTRL_B0	RW	0x0	Action of event CNT_CMPB0 control register; • 0x0: do nothing • 0x1: clear • 0x2: set • 0x3: toggle
3:2	AQ_CTRL_A1	RW	0x0	Action of event CNT_CMPA1 control register; • 0x0: do nothing • 0x1: clear • 0x2: set • 0x3: toggle
1:0	AQ_CTRL_A0	RW	0x0	Action of event CNT_CMPA0 control register; • 0x0: do nothing • 0x1: clear • 0x2: set • 0x3: toggle

10.15.4.11 BRPRD

- **Name:** Breath Period Register
- **Description:** This register configures the period of breath mode.
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x28
- **Reset Value:** 0x00000000

Table 10-271 Breath Period Register

Bits	Field Name	RW	Reset	Description
31:0	BRPRD	RW	0x0	Breath period register, i.e. the required time (number of clock) that the duty changes from 0% to 100% in breath mode.

10.15.4.12 HOLD

- **Name:** Hold Register
- **Description:** This register configures the hold period on breath mode.
- **Base Address:** 0xA000C900 + x*0x300
- **Offset:** 0x2C
- **Reset Value:** 0x00000000

Table 10-272 Hold Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits
23:0	HOLD	RW	0x0	Breath hold control register. The value should be the required number of clock in breath hold state.

11 Security Cores

11.1 Advanced Encryption Standard (AES)

11.1.1 Introduction

The Advanced Encryption Standard (AES) co-processor encrypts or decrypts data, using an algorithm and implementation fully compliant with the AES defined in Federal Information Processing Standards (FIPS) publication 197. Electronic Codebook (ECB) and Cipher Block Chaining (CBC) are supported for key size of 128-, 192-, and 256-bit.

The AES co-processor has both 32-bit AHB and APB interfaces. It supports DMA transfers for incoming and outgoing data (through a dedicated integrated DMA).

11.1.2 Main Features

- Compliance with NIST “Advanced Encryption standard (AES), FIPS Publication 197” from November 2001
- Supports 128-, 192-, and 256-bit key.
- Supports encryption mode and decryption mode.
- Supports the method of interrupt and query register to report status.
- Supports Electronic Codebook (ECB) and Cipher Block chaining (CBC) mode.
- Supports MCU and DMA operation.
- Supports key fetching by key port or configured by MCU.

11.1.3 Registers

11.1.3.1 CTRL

- **Name:** AES Controller Register
- **Description:** This register acts as a global enable/disable for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 11-1 AES Controller Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	FKEY_EN	W	0x0	Enable AES fetch key by itself through AHB master interface or key port. This register can be cleared by itself when key_valid is set to 1.
2	DMA_MODE_EN	RW	0x0	DMA mode start enable: DMA mode start N block data encryption or decryption.

Bits	Field Name	RW	Reset	Description
1	MCU_MODE_EN	RW	0x0	MCU mode start enable: MCU mode start a block data encryption or decryption.This signal should be cleared to zero before a new block input data is ready to start.
0	MODULE_EN	RW	0x0	Enable AES module **Value:** • 0x0: Disable • 0x1: Enable

11.1.3.2 CFG

- **Name:** AES Configuration Register
- **Description:** This register acts as a configuration for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 11-2 AES Configuration Register

Bits	Field Name	RW	Reset	Description
31:13	RSVD	R		Reserved bits
12:11	KEY_TYPE	RW	0x0	Key type selection for encryption / decryption **Value:** • 0x0: Configured by MCU(default) • 0x1: Fetched through AHB interface • 0x2: Fetched through key port • 0x3: Reserved
10:8	OPT_MODE	RW	0x0	Selection for operation mode **Value:** • 0x0: Electronic Codebook (ECB) mode • 0x1: Cipher Block Chaining (CBC) mode • 0x2 – 0x7: Reserved for future application
7	ENDIAN	RW	0x0	Selection for data endian ctrl **Value:** • 0x0: Reverse to big-endian(default) • 0x1: No reverse

Bits	Field Name	RW	Reset	Description
6	FIRST_BLK	W	0x0	This register should be set to 1 before starting the first block in normal CBC and DMA CBC modes, and this register will be cleared by itself.
5	LOAD_SEED	W	0x0	Load seed for LFSR, seed for LFSR will be reloaded when this register was written with 1. This register can be cleared by itself.
4	DEC_ENC_SEL	RW	0x0	Selection for encryption/decryption **Value:** • 0x0: Decryption(default) • 0x1: Encryption
3	FULL_MASK_EN	RW	0x0	Full mask enable signal **Value:** • 0x0: Disable (default) • 0x1: Enable (DPA Resistance)
2	RSVD	R		Reserved bits
1:0	KEY_MODE	RW	0x0	Key mode selection for encryption / decryption **Value:** • 0x0: 128 bits (default) • 0x1: 192 bits • 0x2: 256 bits • 0x3: Reserved

11.1.3.3 STAT

- **Name:** AES Status Register
- **Description:** This is a read-only register used to indicate the current status of AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 11-3 AES Status Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	KEY_STAT	R	0x0	Key status, key is ready to read or not **Value:** • 0x0: Not ready • 0x1: Ready

Bits	Field Name	RW	Reset	Description
2	DMA_XFE_ERR	R	0x0	DMA transfer error, Write 1 to clear **Value:** • 0x0: Not ready • 0x1: Ready
1	DMA_XFE_CPLT	R	0x0	DMA transfer complete **Value:** • 0x0: Not complete • 0x1: Complete
0	READY	R	0x0	AES result data out ready or not. This signal may be cleared when MCUEN (CTRL[1]) is disabled. **Value:** • 0x0: Not ready • 0x1: Ready

11.1.3.4 INT

- **Name:** AES Interrupt Register
- **Description**: This register enables or disables all interrupts generated by the AES. Also it reports the status of the AES interrupts after they have been enabled.
- **Base Address**: 0xA0015400
- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 11-4 AES Interrupt Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	CPLT_INT_EN	RW	0x0	AES result data complete interrupt **Value:** • 0x0: Disable • 0x1: Enable
0	CPLT_INT_FLAG	R WC	0x0	AES result data complete interrupt flag. Write 1 to clear. **Read:** • 0x0: Not interrupt • 0x1: Interrupt **Write:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Not effect • 0x1: Clear

11.1.3.5 XFE_SIZE

- **Name:** AES Transfer Size Register
- **Description:** This register acts as a size for AES DMA mode transfer.
- **Base Address:** 0xA0015400
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 11-5 AES Transfer Size Register

Bits	Field Name	RW	Reset	Description
31:15	RSVD	R		Reserved bits
14:0	SIZE	RW	0x0	Total transfer size, up to 32 KB • 0x000f: 1 block • 0x001f: 2 blocks • 0x002f: 3 blocks … • 0x7fff: 2048 blocks

11.1.3.6 RD_START_ADDR

- **Name:** AES Read Start Address Register
- **Description:** This register acts as a start address for AES DMA reading.
- **Base Address:** 0xA0015400
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 11-6 AES Read Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	DMA mode, read start address of transfer

11.1.3.7 WR_START_ADDR

- **Name:** AES Write Start Address Register

- **Description:** This register acts as a start address for AES DMA writing.
- **Base Address:** 0xA0015400
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 11-7 AES Write Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	DMA mode, write start address of transfer

11.1.3.8 KEY_ADDR

- **Name:** AES Key Address Register
- **Description:** This register acts as a key address for AES encryption or decryption
- **Base Address**: 0xA0015400
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 11-8 AES Key Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	AES key address in memory

11.1.3.9 DATA_OUT0

- **Name:** AES Data Output 0 Register
- **Description:** This register is a 32-bit read- only buffer for the result data[127:96] from AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 11-9 AES Data Output 0 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT0	R	0x0	AES result data[127:96]

11.1.3.10 DATA_OUT1

- **Name:** AES Data Output 1 Register
- **Description:** This register is a 32-bit read-only buffer for the result data[95:64] from AES.

- **Base Address:** 0xA0015400
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 11-10 AES Data Output 1 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT1	R	0x0	AES result data[95:64]

11.1.3.11 DATA_OUT2

- **Name:** AES Data Output 2 Register
- **Description:** This register is a 32-bit read-only buffer for the result data[63:32] from AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x28
- **Reset Value:** 0x00000000

Table 11-11 AES Data Output 2 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT2	R	0x0	AES result data[63:32]

11.1.3.12 DATA_OUT3

- **Name:** AES Data Output 3 Register
- **Description:** This register is a 32-bit read-only buffer for the result data[31:0] from AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x2C
- **Reset Value:** 0x00000000

Table 11-12 AES Data Output 3 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT3	R	0x0	AES result data[31:0]

11.1.3.13 KEY0

- **Name:** AES Key 0 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[255:224] for AES.
- **Base Address:** 0xA0015400

- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 11-13 AES Key 0 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY0	W	0x0	AES key[255:224]

11.1.3.14 KEY1

- **Name:** AES Key 1 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[223:192] for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 11-14 AES Key 1 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY1	W	0x0	AES key[223:192]

11.1.3.15 KEY2

- **Name:** AES Key 2 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[191:160] for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 11-15 AES Key 2 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY2	W	0x0	AES key[191:160]

11.1.3.16 KEY3

- **Name:** AES Key 3 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[159:128] for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x3C

- **Reset Value:** 0x00000000

Table 11-16 AES Key 3 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY3	W	0x0	AES key[159:128]

11.1.3.17 KEY4

- **Name:** AES Key 4 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[127:96] for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 11-17 AES Key 4 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY4	W	0x0	AES key[127:96]

11.1.3.18 KEY5

- **Name:** AES Key 5 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[95:64] for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 11-18 AES Key 5 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY5	W	0x0	AES key[95:64]

11.1.3.19 KEY6

- **Name:** AES Key 6 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[63:32] for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 11-19 AES Key 6 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY6	W	0x0	AES key[63:32]

11.1.3.20 KEY7

- **Name:** AES Key 7 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[31:0] for AES.
- **Base Address:** 0xA0015400
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 11-20 AES Key 7 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY7	W	0x0	AES key[31:0]

11.1.3.21 INIT_SSI

- **Name:** AES Sbox Initial Seed Input Register
- **Description:** This register acts as an initial seed input for Sbox.
- **Base Address:** 0xA0015400
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 11-21 AES Sbox Initial Seed Input Register

Bits	Field Name	RW	Reset	Description
31:0	SEED	RW	0x0	Sbox initial seed input

11.1.3.22 INIT_SSO

- **Name:** AES Sbox Initial Seed Output Register
- **Description:** This register acts as an initial seed output for Sbox.
- **Base Address:** 0xA0015400
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 11-22 AES Sbox Initial Seed Output Register

Bits	Field Name	RW	Reset	Description
31:0	SEED	RW	0x0	Sbox initial seed output

11.1.3.23 MASK_SSI

- **Name:** AES Sbox Seed Input Mask Register
- **Description:** This register acts as a mask for Sbox initial seed input mask.
- **Base Address:** 0xA0015400
- **Offset:** 0x58
- **Reset Value:** 0x00000000

Table 11-23 AES Sbox Seed Input Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	RW	0x0	Sbox initial seed input mask

11.1.3.24 MASK_SSO

- **Name:** AES Sbox Seed Output Mask Register
- **Description:** This register acts as a mask for Sbox initial seed output mask.
- **Base Address:** 0xA0015400
- **Offset:** 0x5C
- **Reset Value:** 0x00000000

Table 11-24 AES Sbox Seed Output Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	RW	0x0	Sbox initial seed output mask

11.1.3.25 INIT_V0

- **Name:** AES Initialization Vector 0 Register
- **Description:** This register acts as an initialization vector[127:96] for CBC mode.
- **Base Address:** 0xA0015400
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 11-25 AES Initialization Vector 0 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[127:96] for CBC mode

11.1.3.26 INIT_V1

- **Name:** AES Initialization Vector 1 Register
- **Description:** This register acts as an initialization vector[95:64] for CBC mode.
- **Base Address:** 0xA0015400
- **Offset:** 0x64
- **Reset Value:** 0x00000000

Table 11-26 AES Initialization Vector 1 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[95:64] for CBC mode

11.1.3.27 INIT_V2

- **Name:** AES Initialization Vector 2 Register
- **Description:** This register acts as an initialization vector[63:32] for CBC mode.
- **Base Address:** 0xA0015400
- **Offset:** 0x68
- **Reset Value:** 0x00000000

Table 11-27 AES Initialization Vector 2 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[63:32] for CBC mode

11.1.3.28 INIT_V3

- **Name:** AES Initialization Vector 3 Register
- **Description:** This register acts as an initialization vector[31:0] for CBC mode.
- **Base Address:** 0xA0015400
- **Offset:** 0x6C
- **Reset Value:** 0x00000000

Table 11-28 AES Initialization Vector 3 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[31:0] for CBC mode

11.1.3.29 DATA_IN0

- **Name:** AES Data Input 0 Register
- **Description:** This register is a 32-bit only write buffer for the input data[127:96] for encryption or decryption.
- **Base Address:** 0xA0015400
- **Offset:** 0x70
- **Reset Value:** 0x00000000

Table 11-29 AES Data Input 0 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN0	W	0x0	Input data[127:96] for encryption or decryption. This register is valid in MCU mode.

11.1.3.30 DATA_IN1

- **Name:** AES Data Input 1 Register
- **Description:** This register is a 32-bit only write buffer for the input data[95:64] for encryption or decryption.
- **Base Address:** 0xA0015400
- **Offset:** 0x74
- **Reset Value:** 0x00000000

Table 11-30 AES Data Input 1 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN1	W	0x0	Input data[95:64] for encryption or decryption. This register is valid in MCU mode.

11.1.3.31 DATA_IN2

- **Name:** AES Data Input 2 Register
- **Description:** This register is a 32-bit only write buffer for the input data[63:32] for encryption or decryption.
- **Base Address:** 0xA0015400
- **Offset:** 0x78
- **Reset Value:** 0x00000000

Table 11-31 AES Data Input 2 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN2	W	0x0	Input data[63:32] for encryption or decryption. This register is valid in MCU mode.

11.1.3.32 DATA_IN3

- **Name:** AES Data Input 3 Register
- **Description:** This register is a 32-bit only write buffer for the input data[31:0] for encryption or decryption.
- **Base Address:** 0xA0015400
- **Offset:** 0x7C
- **Reset Value:** 0x00000000

Table 11-32 AES Data Input 3 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN3	W	0x0	Input data[31:0] for encryption or decryption. This register is valid in MCU mode.

11.1.3.33 KEYPORT_MASK

- **Name:** AES Keyport Mask Register
- **Description:** This register acts as a mask for key from key port.
- **Base Address:** 0xA0015400
- **Offset:** 0x80
- **Reset Value:** 0x00000000

Table 11-33 AES Keyport Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	W	0x0	Mask for key from key port

11.1.4 Data Flow

11.1.4.1 MCU Control Mode

11.1.4.1.1 ECB Mode

1. Configure the following items according to the application.

   (1). **CFG.KEY_MODE** (128-, 192-, 256-bit key)

(2). **CFG.DEC_ENC_SEL** (encryption/decryption mode)

(3). `CFG.OPT_MODE = 3'b000` (ECB mode)

(4). **CFG.FULL_MASK_EN**

(5). **CFG.KEY_TYPE** (if the key type is configured as MCU mode, jump to step 2, otherwise jump to step 3.

2. Configure **KEY0** – **KEY7** (big-endian alignment, 128-bit can only configure **KEY0** – **KEY3**, 192-bit mode can only configure **KEY0** – **KEY5**, 256-bit mode can only configure **KEY0** – **KEY7**). Then jump to step 6.
3. Configure **KEYPORT_MASK** (configured only when `CFG.KEY_TYPE = 2'b10`) and **KEY_ADDR** (offset address in key RAM).
4. Configure `CTRL.FKEY_EN = 1`, enable module to fetch key from **KEY_ADDR** through AHB master interface or key port.
5. Query the **STAT.KEY_STAT**
6. Configure the random seeds: **INIT_SSI**, **INIT_SSO**, **MASK_SSI** and **MASK_SSO**.
7. Enable or disable **INT.CPLT_INT_EN** and set **INT.CPLT_INT_FLAG = 1** to clear interrupt flag.
8. Write a block of data (128-bit) to **DATA_IN0** – **DATA_IN3**.
9. Set **CTRL.MCU_MODE_EN = 1** and start AES.
10. Query the **STAT.READY** or wait for the **INT.CPLT_INT_FLAG** interrupt signal.
    Read **DATA_OUT0** – **DATA_OUT3** 128-bit cipher text or plaintext (big-endian mode).
11. Set **CTRL.MCU_MODE_EN = 0**.
12. If you use interrupt mode, set **INT.CPLT_INT_FLAG = 1** to clear interrupt flag.
13. If there is any data to be encrypted/decrypted, repeat steps 6 to 12.

11.1.4.1.2 CBC Mode

1. Configure the following items according to the application.

   (1). **CFG.KEY_MODE** (128, 192, 256-bit key)

   (2). **CFG.DEC_ENC_SEL** (encryption/ decryption mode)

   (3). **CFG.OPT_MODE = 3'b001**(CBC mode)

   (4). **CFG.FULL_MASK_EN**

   (5). Set **CFG.LOAD_SEED = 1** (the register will clear the data itself and decide whether to load seed depending on the system application requirements). For the first CBC block, you need to set **CFG.FIRST_BLK = 1** (this register can be cleared by itself)

   (6). **CFG.KEY_TYPE** (if the key type is configured as MCU mode, jump to step 2 , otherwise jump to step 3.

2. Configure **KEY0** – **KEY7** (big-endian alignment, 128-bit can only configure **KEY0** – **KEY3**, 192-bit mode can only configure **KEY0** – **KEY5**, 256-bit mode can only configure **KEY0** – **KEY7**). Then jump to step 6.

3. Configure **KEYPORT_MASK** (configured only when **CFG.KEY_TYPE = 2'b10**), **KEY_ADDR**(offset address in key ram)
4. Configure **CTRL.FKEY_EN = 1**, enable module to fetch key from **KEY_ADDR** through AHB master interface or key port.
5. Query the **STAT.KEY_STAT**.
6. Configure Initialization Vector in **INIT_V0** – **INIT_V3** 128-bit CBC mode.
7. Configure the random seeds: **INIT_SSI**, **INIT_SSO**, **MASK_SSI** and **MASK_SSO**.
8. Enable or disable **INT.CPLT_INT_EN**, set **INT.CPLT_INT_FLAG = 1** to clear interrupt flag.
9. Write a block of data (128-bit) to **DATA_IN0**– **DATA_IN3**.
10. Set **CTRL.MCU_MODE_EN = 1** and start AES.
11. Query the **STAT.READY** or wait for the **INT.CPLT_INT_FLAG** interrupt signal.
12. Read **DATA_OUT0** – **DATA_OUT3** 128-bit cipher text or plaintext (big-endian mode).
13. Set **CTRL.MCU_MODE_EN = 0**.
14. If you use interrupt mode, set **INT.CPLT_INT_FLAG = 1** to clear interrupt flag.
15. If there is any data to be encrypted/decrypted, repeat steps 7 to 14.

11.2 Hash Message Authentication Code (HMAC)

11.2.1 Introduction

The Hash Message Authentication Code (HMAC) co-processor authenticates data, using an algorithm and implementation fully compliant with The Keyed-Hash Message Authentication Code (HMAC) defined in Federal Information Processing Standards (FIPS) Publication 198. Multiple modes are supported (SHA256, HMAC-SHA256), for key size of 256 bits.

The HMAC co-processor has both 32-bit AHB and APB interfaces. It supports DMA transfers for incoming and outgoing data (through a dedicated integrated DMA).

11.2.2 Main Features

- Compatible with SHA-256
- Support custom initial hash value
- Support querying registers and interrupts reporting status
- Support MCU mode and DMA mode
- Support fetching key by module itself or configured by MCU

11.2.3 Registers

11.2.3.1 CTRL

- **Name:** HMAC Controller Register
- **Description:** This register acts as a global enable/disable for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 11-34 HMAC Controller Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	LST_TX	W	0x0	Last block in MCU mode or last DMA transfer. This signal may be cleared by itself when hmac_ready was set.
2	KEY_EN	W	0x0	Enable HMAC fetch key by itself through AHB master interface or key port. This signal may be cleared by itself when hmac_key_valid was set.
1	DMA_START	RW	0x0	DMA mode start enable. Starting DMA transfer, this signal should be cleared after DMA has finished all transfer.
0	EN	RW	0x0	HMAC enable, high valid for whole HMAC processing, user must disable it once all HMAC blocks complete. This signal should be asserted until all blocks complete.

11.2.3.2 CFG

- **Name:** HMAC Configuration Register
- **Description:** This register acts as a configuration for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 11-35 HMAC Configuration Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	PRIVT_MOD	RW	0x0	To resist DPA, need to select private mode **Value:** • 0x0: Standard mode • 0x1: Private mode
4	CALC_TYPE	RW	0x0	Select calculation type

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: HMAC • 0x1: SHA
3:2	KEY_TYPE	RW	0x0	Select key type • 0x0: Configured by MCU • 0x1: Fetched through AHB interface • 0x2: Fetched through key port • 0x3: Reserved
1	ENDIAN	RW	0x0	Selection for endian control **Value:** • 0x0: Reverse to big_endian (default) • 0x1: No reverse
0	HASH	RW	0x0	Select enable for user define Hash **Value:** • 0x0: Initial Hash from standard (default) • 0x1: User defined initial Hash enable

11.2.3.3 STAT

- **Name:** HMAC Status Register
- **Description:** This is a read-only register used to indicate the current status of HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 11-36 HMAC Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	DMA_TX_DONE	RW	0x0	HMAC DMA transfer done or not
4	HMAC_READY	RW	0x0	HMAC process status **Value:** • 0x0: Result data is not ready • 0x1: Result data is valid
3	KEY_VALID	R	0x0	HMAC has fetched key or not

Bits	Field Name	RW	Reset	Description
2	DMA_TX_ERR	R WC	0x0	HMAC DMA transfer error. Write 1 to clear
1	DMA_MSG_DONE	R	0x0	If the number of all block message is bigger than transfer size, it indicates all block message was sent when this signal was set.
0	HASH_READY	R	0x0	Hash process status **Value:** • 0x0: Result data is not ready • 0x1: Result data is valid

11.2.3.4 XFE_SIZE

- **Name:** HMAC Transfer Size Register
- **Description:** This register acts as a size for HMAC DMA mode transfer.
- **Base Address:** 0xA0015800
- **Offset:** 0xC
- **Reset Value:** 0x00000000

Table 11-37 HMAC Transfer Size Register

Bits	Field Name	RW	Reset	Description
31:15	RSVD	R		Reserved bits
14:0	SIZE	RW	0x0	Total transfer size, up to 32 KB • 0x003F: 1 block • 0x007F: 2 blocks • 0x00BF: 3 blocks … • 0x7FFF: 512 blocks

11.2.3.5 INT

- **Name:** HMAC Interrupt Register
- **Description:** This register enables or disables all interrupts generated by the HMAC. Also it reports the status of the HMAC interrupts after they have been enabled.
- **Base Address:** 0xA0015800
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 11-38 HMAC Interrupt Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	EN	RW	0x0	HMAC result data complete interrupt **Value:** • 0x0: Disable • 0x1: Enable
0	DONE	R WC	0x0	HMAC result data complete interrupt flag. Write 1 to clear. **Read:** • 0x0: Not interrupt • 0x1: Interrupt **Write:** • 0x0: Not effect • 0x1: Clear

11.2.3.6 RD_START_ADDR

- **Name:** HMAC Read Start Address Register
- **Description:** This register acts as a start address for HMAC DMA reading.
- **Base Address:** 0xA0015800
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 11-39 HMAC Read Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	DMA mode, read start address of transfer

11.2.3.7 WR_START_ADDR

- **Name:** HMAC Write Start Address Register
- **Description:** This register acts as a start address for HMAC DMA writing.
- **Base Address:** 0xA0015800
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 11-40 HMAC Write Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	DMA mode, write start address of transfer

11.2.3.8 USER_HASH_0

- **Name:** HMAC User Hash 0 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0xA0015800
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 11-41 HMAC User Hash 0 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_0	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[255:224]

11.2.3.9 USER_HASH_1

- **Name:** HMAC User Hash 1 Register
- **Description:** This register acts as a user hash value for HASH calculation.
- **Base Address:** 0xA0015800
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 11-42 HMAC User Hash 1 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_1	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[223:192]

11.2.3.10 USER_HASH_2

- **Name:** HMAC User Hash 2 Register
- **Description:** This register acts as a user hash value for HASH calculation.
- **Base Address:** 0xA0015800

- **Offset:** 0x28
- **Reset Value:** 0x00000000

Table 11-43 HMAC User Hash 2 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_2	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[191:160]

11.2.3.11 USER_HASH_3

- **Name:** HMAC User Hash 3 Register
- **Description:** This register acts as a user hash value for HASH calculation.
- **Base Address:** 0xA0015800
- **Offset:** 0x2C
- **Reset Value:** 0x00000000

Table 11-44 HMAC User Hash 3 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_3	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[159:128]

11.2.3.12 USER_HASH_4

- **Name:** HMAC User Hash 4 Register
- **Description:** This register acts as a user hash value for HASH calculation
- **Base Address:** 0xA0015800
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 11-45 HMAC User Hash 4 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_4	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[127:96]

11.2.3.13 USER_HASH_5

- **Name:** HMAC User Hash 5 Register
- **Description:** This register acts as a user hash value for HASH calculation.
- **Base Address:** 0xA0015800
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 11-46 HMAC User Hash 5 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_5	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[95:64]

11.2.3.14 USER_HASH_6

- **Name**: HMAC User Hash 6 Register
- **Description:** This register acts as a user hash value for HASH calculation.
- **Base Address:** 0xA0015800
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 11-47 HMAC User Hash 6 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_6	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[63:32]

11.2.3.15 USER_HASH_7

- **Name:** HMAC User Hash 7 Register
- **Description:** This register acts as a user hash value for HASH calculation.
- **Base Address:** 0xA0015800
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 11-48 HMAC User Hash 7 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_7	RW	0x0	If UHASH is selected, you can configure user defined hash with USER_HASH_0/1/2/3/4/5/6/7. User defined hash value[31:0]

11.2.3.16 DATA_OUT

- **Name:** HMAC Data Output Register
- **Description:**This register is a 32-bit only read buffer for the result data from HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 11-49 HMAC Data Output Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	R	0x0	MCU mode, user can read out result with this register: When all blocks of message were processed, 256 bits HMAC data should be read out 8/9 times. And HMAC data can be read when hmac_ready is valid.

11.2.3.17 DATA_IN

- **Name:** HMAC Data Input Register
- **Description:** This register is a 32-bit only write buffer for the input data for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 11-50 HMAC Data Input Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	W	0x0	MCU mode, user can configure input data with this register: 32-bit data should be sent 16/17 times.

11.2.3.18 KEY0

- **Name:** HMAC Key 0 Register

- **Description:** This register is a 32-bit only write buffer for the key data[255:224] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 11-51 HMAC Data Input Register

Bits	Field Name	RW	Reset	Description
31:0	KEY0	W	0x0	Key was set as 256 bits (8 WORDs) WORD0

11.2.3.19 KEY1

- **Name:** HMAC Key 1 Register
- **Description:** This register is a 32-bit only write buffer for the key data[223:192] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 11-52 HMAC Key 1 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY1	W	0x0	Key was set as 256 bits (8 WORDs) WORD1

11.2.3.20 KEY2

- **Name:** HMAC Key 2 Register
- **Description:** This register is a 32-bit only write buffer for the key data[191:160] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 11-53 HMAC Key 2 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY2	W	0x0	Key was set as 256 bits (8 WORDs) WORD2

11.2.3.21 KEY3

- **Name:** HMAC Key 3 Register
- **Description:** This register is a 32-bit only write buffer for the key data[159:128] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 11-54 HMAC Key 3 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY3	W	0x0	Key was set as 256 bits (8 WORDs) WORD3

11.2.3.22 KEY4

- **Name:** HMAC Key 4 Register
- **Description:** This register is a 32-bit only write buffer for the key data[127:96] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x58
- **Reset Value:** 0x00000000

Table 11-55 HMAC Key 4 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY4	W	0x0	Key was set as 256 bits (8 WORDs) WORD4

11.2.3.23 KEY5

- **Name:** HMAC Key 5 Register
- **Description:** This register is a 32-bit only write buffer for the key data[95:64] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x5C
- **Reset Value:** 0x00000000

Table 11-56 HMAC Key 5 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY5	W	0x0	Key was set as 256 bits (8 WORDs) WORD5

11.2.3.24 KEY6

- **Name:** HMAC Key 6 Register
- **Description:** This register is a 32-bit only write buffer for the key data[63:32] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 11-57 HMAC Key 6 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY6	W	0x0	Key was set as 256 bits (8 WORDs) WORD6

11.2.3.25 KEY7

- **Name:** HMAC Key 7 Register
- **Description:** This register is a 32-bit only write buffer for the key data[31:0] for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x64
- **Reset Value:** 0x00000000

Table 11-58 HMAC Key 7 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY7	W	0x0	Key was set as 256 bits (8 WORDs) WORD7

11.2.3.26 KEY_ADDR

- **Name:** HMAC Key Address Register
- **Description:** This register acts as a key address for HMAC.
- **Base Address:** 0xA0015800
- **Offset:** 0x68
- **Reset Value:** 0x00000000

Table 11-59 HMAC Key Address Register

Bits	Field Name	RW	Reset	Description
31:0	KEY_ADDR	RW	0x0	HMAC key address in memory

11.2.3.27 KEYPORT_MASK

- **Name:** HMAC Keyport Mask Register
- **Description:** This register acts as a mask for key from keyport.
- **Base Address:** 0xA0015800
- **Offset:** 0x6C
- **Reset Value:** 0x00000000

Table 11-60 HMAC Keyport Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	W	0x0	Mask for key from key port

11.3 Public Key Cryptography (PKC)

11.3.1 Introduction

The Public Key Cryptography (PKC) controller module completes the basic underlying modular arithmetic and the FIPS standard 256-point ECC point multiplication in the public key algorithm.

11.3.2 Main Features

- Supports FIPS-180-3 standard and Scalar Multiplication for P-256 elliptic curve.
- Supports Montgomery Modular Multiplication for configurable 256 – 2048 bits length.
- Supports Partial Montgomery Inversion for configurable 256 – 2048 bits length.
- Supports Modular Addition for configurable 256 – 2048 bits length.
- Supports Modular Subtraction for configurable 256 – 2048 bits length.
- Supports Modular Comparison for configurable 256 – 2048 bits length.
- Supports Modular Left-shift Operation for configurable 256 – 2048 bits length.
- Supports Big Integer Multiplication for configurable 256 – 1024 bits length.
- Supports Big Integer Addition for configurable 256 – 2048 bits length.
- Supports Hardware Dummy Multiplication.
- Supports random clocking gating.
- Supports register read-write protection and the register cannot be read nor written in the DISABLE module state.
- Supports interrupt and query.

11.3.3 Registers

11.3.3.1 CTRL

- **Name:** PKC Controller Register
- **Description:** This register acts as a global enable/disable for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 11-61 PKC Controller Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	RST	RW	0x0	Write 0 and then write 1 to this bit, force PKC core to reset
7:5	RSVD	R		Reserved bits
4	SW_CTRL	RW	0x0	Only used in software mode **Value:** • 0x0: disable software controller • 0x1: enable software controller
3:2	RSVD	R		Reserved bits
1	START	RW	0x0	Only used in hardware mode. After MCU configure all parameters, write 0 and then write 1 to this bit, PKC start to work
0	EN	RW	0x0	PKC core enable **Value:** • 0x0: disable PKC • 0x1: enable PKC

11.3.3.2 CFG0

- **Name:** PKC Configuration 0 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x4
- **Reset Value:** 0x00100000

Table 11-62 PKC Configuration 0 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
24:16	R_POINT	RW	0x10	R point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	K_POINT	RW	0x0	K point in sp_ram

11.3.3.3 CFG1

- **Name:** PKC Configuration 1 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x8
- **Reset Value:** 0x00280020

Table 11-63 PKC Configuration 1 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R2_POINT	RW	0x28	R^2 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	P_POINT	RW	0x20	P point in sp_ram

11.3.3.4 CFG2

- **Name:** PKC Configuration 2 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0xC
- **Reset Value:** 0x00380030

Table 11-64 PKC Configuration 2 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	GY_POINT	RW	0x38	Gy point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	GX_POINT	RW	0x30	Gx point in sp_ram

11.3.3.5 CFG3

- **Name:** PKC Configuration 3 Register
- **Description**: This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x10
- **Reset Value:** 0x00480040

Table 11-65 PKC Configuration 3 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R0X_POINT	RW	0x48	R0x point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	GZ_POINT	RW	0x40	Gz point in sp_ram

11.3.3.6 CFG4

- **Name:** PKC Configuration 4 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x14
- **Reset Value:** 0x00580050

Table 11-66 PKC Configuration 4 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R0Z_POINT	RW	0x58	R0z point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	R0Y_POINT	RW	0x50	R0y point in sp_ram

11.3.3.7 CFG5

- **Name:** PKC Configuration 5 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x18
- **Reset Value:** 0x00680060

Table 11-67 PKC Configuration 5 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R1Y_POINT	RW	0x68	R1y point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	R1X_POINT	RW	0x60	R1x point in sp_ram

11.3.3.8 CFG6

- **Name:** PKC Configuration 6 Register
- **Description:** This register acts as a configuration for PKC
- **Base Address:** 0xA0014000
- **Offset:** 0x1C
- **Reset Value:** 0x00780070

Table 11-68 PKC Configuration 6 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	TEMP1_POINT	RW	0x78	Tmp1 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	R1Z_POINT	RW	0x70	R1z point in sp_ram

11.3.3.9 CFG7

- **Name:** PKC Configuration 7 Register
- **Description:** This register acts as a configuration for PKC
- **Base Address:** 0xA0014000
- **Offset:** 0x20
- **Reset Value:** 0x00880080

Table 11-69 PKC Configuration 7 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	TEMP3_POINT	RW	0x88	Tmp3 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	TEMP2_POINT	RW	0x80	Tmp2 point in sp_ram

11.3.3.10 CFG8

- **Name:** PKC Configuration 8 Register
- **Description:** This register acts as a configuration for PKC
- **Base Address:** 0xA0014000
- **Offset:** 0x24
- **Reset Value:** 0x00980090

Table 11-70 PKC Configuration 8 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	TEMP5_POINT	RW	0x98	Tmp5 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	TEMP4_POINT	RW	0x90	Tmp4 point in sp_ram

11.3.3.11 CFG9

- **Name:** PKC Configuration 9 Register
- **Description:** This register acts as a configuration for PKC
- **Base Address:** 0xA0014000
- **Offset:** 0x28
- **Reset Value:** 0x00A800A0

Table 11-71 PKC Configuration 9 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	CONT1_POINT	RW	0xA8	Constant 1 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	TEMP6_POINT	RW	0xA0	Tmp6 point in sp_ram

11.3.3.12 CFG10

- **Name:** PKC Configuration 10 Register
- **Description:** This register acts as a configuration for PKC
- **Base Address:** 0xA0014000
- **Offset:** 0x2C
- **Reset Value:** 0x00B800B0

Table 11-72 PKC Configuration 10 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	X2_POINT	RW	0xB8	X2 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	X1_POINT	RW	0xB0	X1 point in sp_ram

11.3.3.13 CFG11

- **Name:** PKC Configuration 11 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x30
- **Reset Value:** 0x00C800C0

Table 11-73 PKC Configuration 11 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	KT_POINT	RW	0xC8	software compute 2^256 mod p and write to sp_ram
15:9	RSVD	R		Reserved bits
8:0	MIT_POINT	RW	0xC0	Only used in hardware mode, mi point in sp_ram

11.3.3.14 CFG12

- **Name:** PKC Configuration 12 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x34
- **Reset Value:** 0x00D800D0

Table 11-74 PKC Configuration 12 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	B_POINT	RW	0xD8	Curve parameter b in Montgomery field
15:9	RSVD	R		Reserved bits
8:0	A_POINT	RW	0xD0	Curve parameter a in Montgomery field

11.3.3.15 CFG13

- **Name:** PKC Configuration 13 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x38
- **Reset Value:** 0x00000001

Table 11-75 PKC Configuration 13 Register

Bits	Field Name	RW	Reset	Description
31:0	CONSTQ	RW	0x1	Constant used in Montgomery Multiplication

11.3.3.16 SW_CTRL

- **Name:** PKC Software Controller Register
- **Description:** This register acts as a controller for software operation.
- **Base Address:** 0xA0014000
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 11-76 PKC Software Controller Register

Bits	Field Name	RW	Reset	Description
31:10	RSVD	R		Reserved bits
9	RCG_EN	RW	0x0	Enable random clock gating **Value:** • 0x0: Disable • 0x1: Enable
8	DM_EN	RW	0x0	Enable dummy multiplication **Value:** • 0x0: Disable • 0x1: Enable
7	RSVD	R		Reserved bits
6:4	MODE	RW	0x0	Only used in software mode. **Value:** • 0x0: Montgomery Multiplication • 0x1: Mod Inversion

Bits	Field Name	RW	Reset	Description
				• 0x2: Mod Addition • 0x3: Mod Subtraction • 0x4: Mod Comparison • 0x5: Mod Shift • 0x6: Big Integer Multiplication • 0x7: Bit Integer Addition
3:1	RSVD	R		Reserved bits
0	START	RW	0x0	Only used in software mode. MCU writes 0 to this bit and then writes 1 to this bit to start PKC

11.3.3.17 SW_CFG0

- **Name:** PKC Software Configuration 0 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x44
- **Reset Value:** 0x00080000

Table 11-77 PKC Software Configuration 0 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MMB_POINT	RW	0x8	Modular multiplication B point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MMA_POINT	RW	0x0	Modular multiplication A point in sp_ram

11.3.3.18 SW_CFG1

- **Name:** PKC Software Configuration 1 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x48
- **Reset Value:** 0x00180010

Table 11-78 PKC Software Configuration 1 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MMC_POINT	RW	0x18	Modular multiplication C point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MMP_POINT	RW	0x10	Modular multiplication P point in sp_ram

11.3.3.19 SW_CFG2

- **Name:** PKC Software Configuration 2 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x4C
- **Reset Value:** 0x00280020

Table 11-79 PKC Software Configuration 2 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MASB_POINT	RW	0x28	Modular addition/subtraction B point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MASA_POINT	RW	0x20	Modular addition/subtraction A point in sp_ram

11.3.3.20 SW_CFG3

- **Name:** PKC Software Configuration 3 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x50
- **Reset Value:** 0x00380030

Table 11-80 PKC Software Configuration 3 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MASC_POINT	RW	0x38	Modular addition/subtraction C point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MASP_POINT	RW	0x30	Modular addition/subtraction P point in sp_ram

11.3.3.21 SW_CFG4

- **Name:** PKC Software Configuration 4 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x54
- **Reset Value:** 0x00480040

Table 11-81 PKC Software Configuration 4 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MIV_POINT	RW	0x48	Modular inversion V point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MIU_POINT	RW	0x40	Modular inversion U point in sp_ram

11.3.3.22 SW_CFG5

- **Name:** PKC Software Configuration 5 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x58
- **Reset Value:** 0x00580050

Table 11-82 PKC Software Configuration 5 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MIX2_POINT	RW	0x58	Modular inversion X2 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MIT_POINT	RW	0x50	Modular inversion X1 point in sp_ram

11.3.3.23 SW_CFG6

- **Name:** PKC Software Configuration 6 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x5C
- **Reset Value:** 0x00000060

Table 11-83 PKC Software Configuration 6 Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8:0	MITP	RW	0x60	Modular inversion TEMP point in sp_ram

11.3.3.24 SW_CFG7

- **Name:** PKC Software Configuration 7 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x60
- **Reset Value:** 0x00000007

Table 11-84 PKC Software Configuration 7 Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5:0	LEN	RW	0x7	Operator word length configuration, in software mode: • 0x07: 256 bits • 0x08: 288 bits … • 0x3F: 2048 bits In hardware mode fixed to 0x07

11.3.3.25 SW_CFG8

- Name: PKC Software Configuration 8 Register
- Description: This register acts as a configuration for PKC in software mode.
- Base Address: 0xA0014000
- Offset: 0x64
- Reset Value: 0x00000000

Table 11-85 PKC Software Configuration 8 Register

Bits	Field Name	RW	Reset	Description
31:13	RSVD	R		Reserved bits
12:0	MIK_OUT	R	0x0	K out in modular inversion operation

11.3.3.26 SW_CFG9

- **Name:** PKC Software Configuration 9 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x68
- **Reset Value:** 0xAABBCCDD

Table 11-86 PKC Software Configuration 9 Register

Bits	Field Name	RW	Reset	Description
31:0	RDMS	RW	0xAABBCCDD	Random dummy multiplication seed

11.3.3.27 SW_CFG10

- **Name:** PKC Software Configuration 10 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x6C
- **Reset Value:** 0x00780070

Table 11-87 PKC Software Configuration 10 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	BMB_POINT	RW	0x78	Big multiplication B point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	BMA_POINT	RW	0x70	Big multiplication A point in sp_ram

11.3.3.28 SW_CFG11

- **Name:** PKC Software Configuration 11 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x70
- **Reset Value:** 0x00880080

Table 11-88 PKC Software Configuration 11 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	BAA_POINT	RW	0x88	Big addition A point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	BMC_POINT	RW	0x80	Big multiplication C point in sp_ram

11.3.3.29 SW_CFG12

- **Name:** PKC Software Configuration 12 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x74
- **Reset Value:** 0x00980090

Table 11-89 PKC Software Configuration 12 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	BAC_POINT	RW	0x98	Big addition C point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	BAB_POINT	RW	0x90	Big addition B point in sp_ram

11.3.3.30 SW_CFG13

- **Name:** PKC Software Configuration 13 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0xA0014000
- **Offset:** 0x78
- **Reset Value:** 0x11223344

Table 11-90 PKC Software Configuration 13 Register

Bits	Field Name	RW	Reset	Description
31:0	RCG_SEED	RW	0x11223344	Random clock gating seed

11.3.3.31 INT_STAT

- **Name:** PKC Interrupt Status Register

- **Description:** This is a read-only register used to indicate the interrupt status of PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x80
- **Reset Value:** 0x00000000

Table 11-91 PKC Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	BIAO_INT_FLAG	R WC	0x0	PKC big integer add overflow flag, write 1 to clear
1	ERR_INT_EN	R WC	0x0	PKC error interrupt flag, write 1 to clear
0	CPLT_INT_FLAG	R WC	0x0	PKC complete interrupt flag, write 1 to clear

11.3.3.32 INT_EN

- **Name:** PKC Interrupt Enable Register
- **Description:** This register enables or disables all interrupts generated by the PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x84
- **Reset Value:** 0x00000000

Table 11-92 PKC Interrupt Enable Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	BIAO_INT_EN	RW	0x0	PKC big integer add overflow interrupt enable
1	ERR_INT_EN	RW	0x0	PKC error interrupt enable
0	CPLT_INT_EN	RW	0x0	PKC complete interrupt enable

11.3.3.33 STAT

- **Name:** PKC Status Register
- **Description:** This is a read-only register used to indicate the status of PKC.
- **Base Address:** 0xA0014000
- **Offset:** 0x88

- **Reset Value:** 0x00000000

Table 11-93 PKC Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	BUSY	R	0x0	PKC busy status

11.4 True Random Number Generator (TRNG)

11.4.1 Introduction

True Random Number Generator (TRNG) generates random numbers used in encryption and decryption. Post-processing is added in TRNG module to ensure the quality of random number and amend the bias of TRNG.

The goal of true random data generated by TRNG is to pass NIST SP 800-22 standard test suite.

11.4.2 Main Features

- Supports True and Pseudo Random Number Generation
- Random number quality can pass NIST SP 800-22 standard test suite.
- Supports multiple post-processing mechanism, including bit-skipping, bit-counting, and Von-Neumann.
- Supports multiple combination of random numbers.
- Supports both interrupt and query mode.

11.4.3 Registers

11.4.3.1 CTRL

- **Name:** TRNG Controller Register
- **Description:** This register acts as a global enable/disable for TRNG.
- **Base Address:** 0xA0017800
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 11-94 TRNG Controller Register

Bits	Name	Access	Reset Value	Description
31:1	RSVD	R		Reserved bits
0	RUN	RW	0x0	TRNG work enabled signal, valid at HIGH. • 0x0: disable TRNG module

Bits	Name	Access	Reset Value	Description
				• 0x1: TRNG starts working to execute a operation of reading random number

11.4.3.2 STAT

- **Name:** TRNG Status Register
- **Description:** This is a read-only register used to indicate the current status of TRNG.
- **Base Address:** 0xA0017800
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 11-95 TRNG Status Register

Bits	Name	Access	Reset Value	Description
31:1	RSVD	R		Reserved bits
0	READY	R WC	0x0	TRNG status flag bit, queried by CPU. **Value:** • 0x0: TRNG output data is not valid. • 0x1: TRNG output data is valid. This register is automatically cleared by hardware after READY is set HIGH. This register is automatically cleared by hardware after read DATA.

11.4.3.3 DATA

- **Name:** TRNG Data Register
- **Description:** This register is a 32-bit read buffer for the random.
- **Base Address:** 0xA0017800
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 11-96 TRNG Data Register

Bits	Name	Access	Reset Value	Description
31:0	DATA	R	0x0	TRNG data. This register is automatically cleared by hardware after read.

11.4.3.4 MCULOCK

Not used in GR551x.

11.4.3.5 LONG_RUN_STAT

- **Name:** TRNG Long Run Status Register
- **Description:** This is a read-only register used to indicate the current status of long run mode.
- **Base Address:** 0xA0017800
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 11-97 TRNG Long Run Status Register

Bits	Name	Access	Reset Value	Description
31:9	RSVD	R		Reserved bits
8:1	LONG_RUN_COUNT	R	0x0	Counts of TRNG long run test. This value is incremented by 1 for every long run detected by TRNG. After TRNG being disabled or reset, the register is cleared.
0	LONG_RUN_FLAG	R	0x0	Flag of TRNG long run test **Value:** • 0x0: Long run never occurred. • 0x1: Long run was occurred in random number sequence After TRNG being disabled or reset, this register is cleared

11.4.3.6 CFG

- **Name:** TRNG Configuration Register
- **Description:** This register acts as a configuration for TRNG.
- **Base Address:** 0xA0017800
- **Offset:** 0x14
- **Reset Value:** 0x0000B246

Table 11-98 TRNG Configuration Register

Bits	Name	Access	Reset Value	Description
31:16	RSVD	R		Reserved bits
15	FRO_EN	RW	0x1	Ring oscillator TRNG enabled signal **Value:** • 0x0: Disabled • 0x1: Enabled
14	LFS_EN	RW	0x0	Low frequency sampling TRNG enabled signal

Bits	Name	Access	Reset Value	Description
				Value: • 0x0: Disabled • 0x1: Enabled Reserved in GR551x.
13	HW_INT_EN	RW	0x1	TRNG hardware interrupt enable. Active High.
12:10	LFSR_SEED_CFG	RW	0x4	LFSR seed configuration mode, select source of LFSR seed **Value:** • 0x0: LFSR seed is from the switching current s0 • 0x1: LFSR seed is from the switching current s1 • 0x2: LFSR seed is from the switching current s2 • 0x3: LFSR seed is from the switching current s3 • 0x4: LFSR seed is from the oscillator s0 • 0x5: LFSR seed is from the Low Frequency Sampling s0 • 0x6: LFSR seed is configured by users • 0x7: Random numbers do not use LFSR and completely use true random mode Value is limited to 0x4 or 0x6 in GR551x.
9	LFSR_MODE_CFG	RW	0x1	LFSR configuration mode: • 0x0: 59 bit LFSR • 0x1: 128 bit LFSR
8:7	P_MODE_CFG	RW	0x0	TRNG post-process configuration **Value:** • 0x0: No post-process • 0x1: bit skipping • 0x2: bit counting • 0x3: Von-Neumann
6:4	LFSR_XOR_SEL	RW	0x4	In Output mode 8, select TRNG with LFSR XOR 0x4: LFSR $\oplus$ Ring Oscillator s0 **Note:** • the $\oplus$ means XOR. • Value is limited to 0x4 in GR551x.
3:0	OUT_MODE	RW	0x6	Select TRNG Output mode

Bits	Name	Access	Reset Value	Description
				Value: • 0x4: Digital TRNG direct output, ring oscillator s0 • 0x6: LFSR and TRNG cyclic sampling and parity generation • 0x7: LFSR and TRNG cyclic sampling • 0x8: LFSR ⊕ TRNG • 0x9: LFSR direct output Value is limited to 0x4, 0x6, 0x7, 0x8 or 0x9 in GR551x. If the LFSR_SEED_CFG bit is 0x06, it cannot be configured as 0x04.

11.4.3.7 SRC_CFG

- **Name:** TRNG Source Configuration Register
- **Description:** This register acts as a configuration for source of TRNG.
- **Base Address:** 0xA0017800
- **Offset:** 0x18
- **Reset Value:** 0x00007864

Table 11-99 TRNG Source Configuration Register

Bits	Name	Access	Reset Value	Description
31:15	RSVD	R		Reserved bits
14:11	FRO_CHAIN_SEL	RW	0xF	Selection of Chain4 – Chain7 for FRO **Value:** • 0x0: select long ring • 0x1: select short ring
10:8	RSVD	R		Reserved bits
7:0	WAIT_TIME	RW	0x64	The waiting time that TRNG input reaches stable. Default value is 100 time cycles.

11.4.3.8 FRO_CFG

- **Name:** TRNG FRO Configuration Register
- **Description:** This register acts as a configuration for FRO.
- **Base Address:** 0xA0017800
- **Offset:** 0x1C

- **Reset Value:** 0x0000FFFF

Table 11-100 TRNG FRO Configuration Register

Bits	Name	Access	Reset Value	Description
31:16	RSVD	R		Reserved bits
15:8	TEST_IN	RW	0xFF	Test input signal of each Chain for FRO The default value is 1.
7:0	CHAIN_EN	RW	0xFF	Enable signal of each Chain in ring oscillator module It is valid when FROEN is HIGH.

11.4.3.9 USER_SEED

- **Name:** TRNG User Seed Register
- **Description:** This register acts as a user seed value for TRNG.
- **Base Address:** 0xA0017800
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 11-101 TRNG User Seed Register

Bits	Name	Access	Reset Value	Description
31:16	RSVD	R		Reserved bits
15:0	USER_SEED	W	0x0	RNG seed configured by user. • Write four times to user_seed[58:0] to configure a 59-bit random number. • Write eight times to user_seed[127:0] to configure a 128-bit random number.

11.4.3.10 LONG_RUN_CFG

- **Name:** TRNG Long Run Configuration Register
- **Description:** This register acts as a configuration for long run mode.
- **Base Address:** 0xA0017800
- **Offset:** 0x24
- **Reset Value:** 0x00000035

Table 11-102 TRNG Long Run Configuration Register

Bits	Name	Access	Reset Value	Description
31:6	RSVD	R		Reserved bits

Bits	Name	Access	Reset Value	Description
5:1	LONG_RUN_THD	RW	0x1A	Threshold configuration of RNG long run test. The run which is greater or equal to this value will be detected. Default value is 26 and the maximum value is 31.
0	THD_LR_TEST	RW	0x1	RNG long run test enable **Value:** • 0x0: Disabled • 0x1: Enabled

12 Communication Subsystem

12.1 Supported Features

GR551x has a communication subsystem that supports Bluetooth Low Energy 5.1.

12.2 Transceiver

The transceiver consists of a receiver path with Low Noise Amplifier (LNA), Mixer, Baseband (BB) amplifier and an Analog to Digital Converter (ADC).

The digitized signals are sent to the digital frontend (DFE) for demodulation. The digital frontend provides AGC feedback signals to adjust the gain of the LNA and BB amplifier to maximize the SNR at the demodulation.

On the transmit side, the digital signal from the DFE is used to modulate the PLL and delivers the modulated carrier to the PA which is configurable by the digital gain settings to deliver the required output power.

RF frequency and digital clocks are generated from the XO.

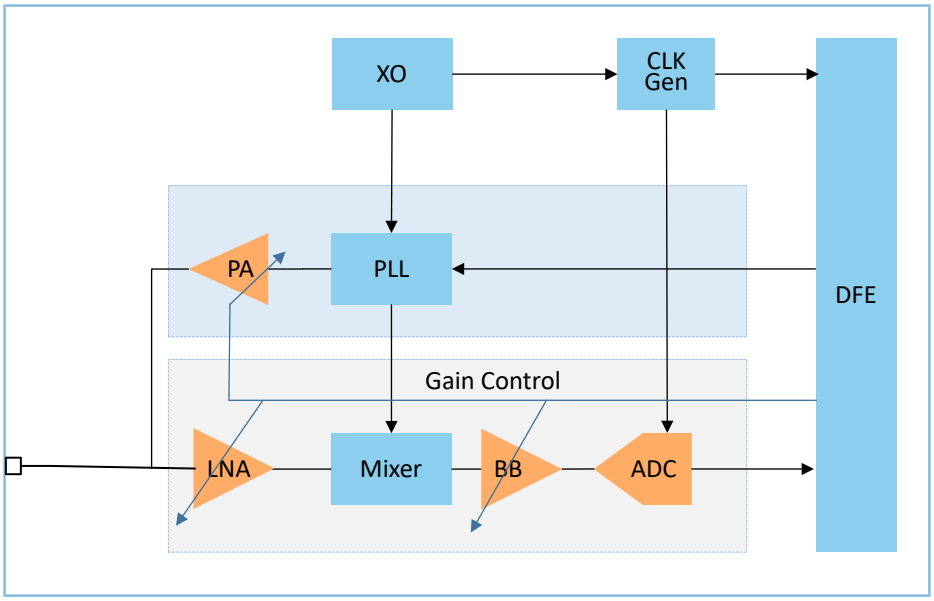

Figure 12-1 Transceiver architecture

12.3 Digital Bluetooth LE Subsystem

The digital side of Bluetooth LE subsystem is shown in Figure 12-2.

On the receive side, the digitized samples from the transceiver are filtered by a digital filter to remove the unwanted blockers, and the RSSI information is extracted. Based on the signal strength, a feedback signal is sent back to the transceiver to adjust the gain of the LNA and BB accordingly. The modem detects the frame start using the preamble and the frequency demodulator extracts the bits inside the Bluetooth LE packet. Bit stream processing block checks the CRC of the packet content and writes it to the packet memory along with any control or status information (RSSI and RF Channel). The crypto engine decrypts the packet if it is encrypted using AES-128. The MAC firmware running on the chip's MCU gets the packet information and processes it accordingly.

On the transmit side, the MAC firmware prepares the data to be transmitted and maps it to an event with a specific time stamp according to its connection. The Event Scheduler sets the event time and configures the packet controller to fill the data into packets and writes it to the packet memory along with its control information. The memory controller sends the packet from the memory to the modem and configures the RF channel frequency for the data packets using the packet control parameters. The packet is encrypted; CRC is added and modulated, and then sent to the frontend that passes it through a Gaussian filter and hands it to the RF transceiver.

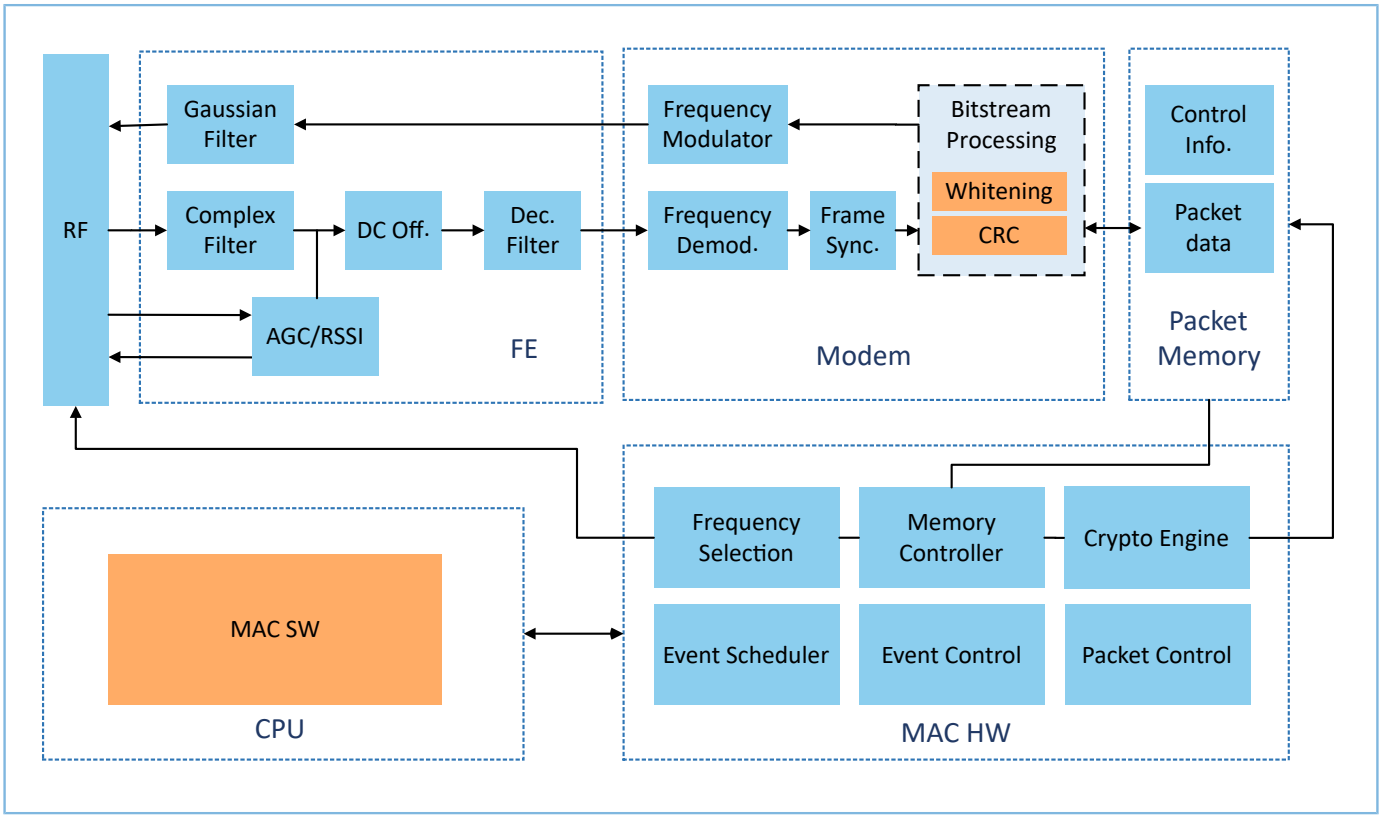

Figure 12-2 Digital Bluetooth LE subsystem architecture

12.4 Performance

12.4.1 Receiver Performance

Table 12-1 Receiver Performance

Parameter		Min	Typ	Max	Unit
Frequency		2,402		2,480	MHz
Sensitivity	1 Mbps		-97		dBm
	2 Mbps		-93		
	LR2		-99.5		
	LR8		-103		
Maximum receive signal level			0		

Parameter	Min	Typ	Max	Unit
CCI		-12		dBc
ACI (N±1)		2		
N-2 Blocker (Image)		25		
N+2 Blocker		30		
N-3 Blocker (Adj. Image)		30		
N+3 Blocker		40		
N±4 or greater		41		

12.4.2 Transmitter Performance

Table 12-2 Transmitter Performance

Parameter	Min.	Typ.	Max.	Unit
Frequency	2,402		2,480	MHz
Output power range	-20	0	7	dBm
In-band Spurious (N±2)		-22	-20	
In-band Spurious (N±3)		-43	-40	
2^{nd} harmonic P_{out}		-42	-40	
3^{rd} harmonic P_{out}		-45	-40	
4^{th} harmonic P_{out}		-50	-40	
5^{th} harmonic P_{out}		-45	-40	
Frequency deviation	185	200		kHz

13 Absolute Maximum Ratings

Maximum ratings specify the limits to voltage, current, temperature, Flash, etc., which applied to GR551x. Exposure to conditions beyond these ratings may affect the lifetime and reliability of the device.

Table 13-1 GR551x absolute maximum ratings

Type	Parameter	Description	Conditions	Min.	Max.	Unit
Voltage	VBAT_LIM	limiting battery supply voltage	Pin VBATL	-0.3	4.7	V
	VDDIO	Limiting external input IO voltage	Pin VDDIO	-0.3	3.8	V
	VPIN_LIM_3V3	Limiting voltage on a pin	3.3 V I/O pins	-0.3	3.9	V
	VPIN_LIM_VDD	Limiting voltage on a pin	VDD I/O pins VDD < 3.6 V	-0.3	VDD+0.3	V
	VESD_HBM	Electrostatic discharge voltage (Human Body Model)	QFN56 BAG68 BGA55 QFN40	-	2000	V
	VESD_CDM	Electrostatic discharge voltage (Charged Device Model)	QFN56 BAG68 BGA55 QFN40	-	500	V
Radio	RF input level	-	-	-	10	dBm
Temperature	TSTG	Storage temperature	-	-40	125	°C
Flash	Endurance	Flash memory	-	10000		Write/Erase cycles
	Retention	Flash memory	-	10 years at 40°C	-	-

14 Package Information

GR551x offers QFN56, BGA68, BGA55, and QFN40 packages to support different environmental requirements.

14.1 QFN56

GR551x QFN56, including GR5515IGND QFN56, GR5515IENDU QFN56, and GR5515I0NDA QFN56, is a 56-pin and 7 x 7 x 0.75 (mm) QFN package. It is qualified for MSL3.

Table 14-1 QFN56 package information

Parameter	Value	Unit	Tolerance
Package Size	7 x 7	mm	±0.1 mm
QFN Pad Count	56		
Total Thickness	0.75	mm	±0.05 mm
QFN Pad Pitch	0.40		
Pad Width	0.20		
Exposed Pad Size	5.2 x 5.2		±0.1 mm

The Figure 14-1 shows the QFN56 package outlines.

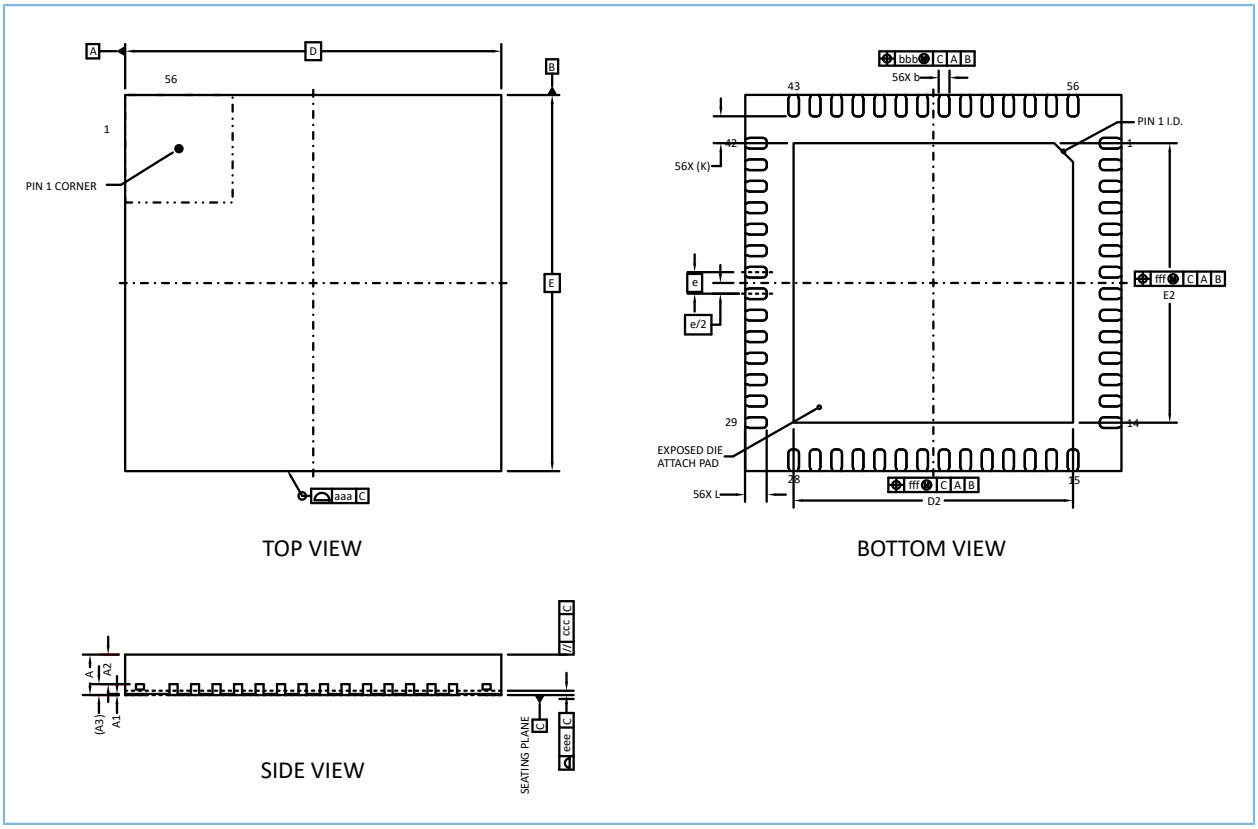

Figure 14-1 QFN56 package outlines

Note:

Drawing is not to scale.

Table 14-2 QFN56 package dimensions

Symbol	Dimensions in mm			Dimensions in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.700	0.750	0.800	0.028	0.030	0.032
A1	0.000	0.020	0.050	0.000	0.001	0.002
A2	-	0.550	-	-	0.022	-
A3	0.203 REF.			0.008 REF.		
b	0.150	0.200	0.250	0.006	0.008	0.010
D	7.000 BSC.			0.276 BSC.		
E	7.000 BSC.			0.276 BSC.		
e	0.400 BSC.			0.016 BSC.		
D2	5.100	5.200	5.300	0.201	0.205	0.209
E2	5.100	5.200	5.300	0.201	0.205	0.209
L	0.300	0.400	0.500	0.012	0.016	0.020
K	0.500 REF.			0.020 REF.		
aaa	0.100			0.004		
ccc	0.100			0.004		
eee	0.080			0.003		
bbb	0.070			0.003		
fff	0.100			0.004		

Note:

Values in inches are converted from values in millimeter and rounded to 3 decimal digits.

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

14.2 BGA68

GR551x BGA68, including GR5515RGBD BGA68, is a 68-pin and 5.3 x 5.3 x 0.88 (mm) package. It is qualified for MSL3.

Table 14-3 BGA68 package information

Parameter	Value	Unit	Tolerance
Package Size	5.3 x 5.3	mm	±0.1 mm
BGA Ball Count	68		
Total Thickness	0.88	mm	±0.1 mm

Parameter	Value	Unit	Tolerance
BGA Ball Pitch	0.50		
Ball Diameter	0.25		
Ball Height	0.18		

Figure 14-2 below shows the BGA68 package outlines.

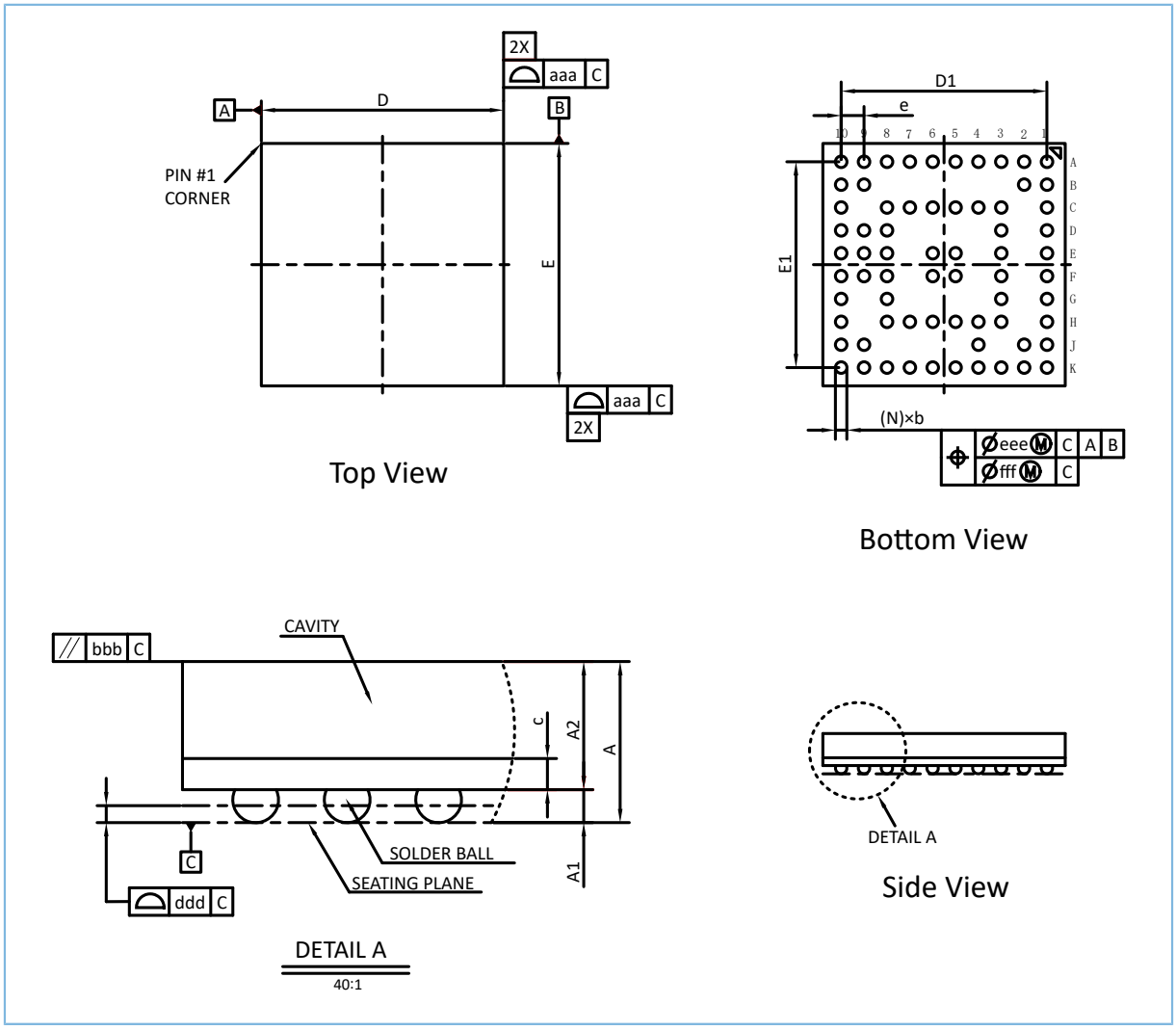

Figure 14-2 BGA68 package outlines

Note:

Drawing is not to scale.

Table 14-4 BGA68 package dimensions

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.780	0.880	0.980	0.031	0.035	0.039
A1	0.130	0.180	0.230	0.005	0.007	0.009
A2	0.650	0.700	0.750	0.026	0.028	0.030
c	0.140	0.170	0.200	0.006	0.007	0.008
D	5.200	5.300	5.400	0.205	0.209	0.213
E	5.200	5.300	5.400	0.205	0.209	0.213
D1	--	4.500	--	--	0.177	--
E1	--	4.500	--	--	0.177	--
e	--	0.500	--	--	0.020	--
b	0.200	0.250	0.300	0.008	0.010	0.012
aaa	0.100			0.004		
bbb	0.100			0.004		
ddd	0.080			0.003		
eee	0.150			0.006		
fff	0.050			0.002		

Note:

Values in inches are converted from values in millimeters and rounded to 3 decimal digits.

14.3 BGA55

GR551x BGA55, including GR5515GGBD BGA55, is a 55-pin and 3.5 x 3.5 x 0.60 (mm) BGA package. It is qualified for MSL3.

Table 14-5 BGA55 package information

Parameter	Value	Unit	Tolerance
Package Size	3.5 x 3.5	mm	±0.1 mm
BGA Ball Count	55		
Total Thickness	0.60	mm	±0.05 mm
BGA Ball Pitch	0.40		
Ball Diameter	0.20		
Ball Height	0.12		±0.03 mm

The figure below shows the BGA55 package outlines.

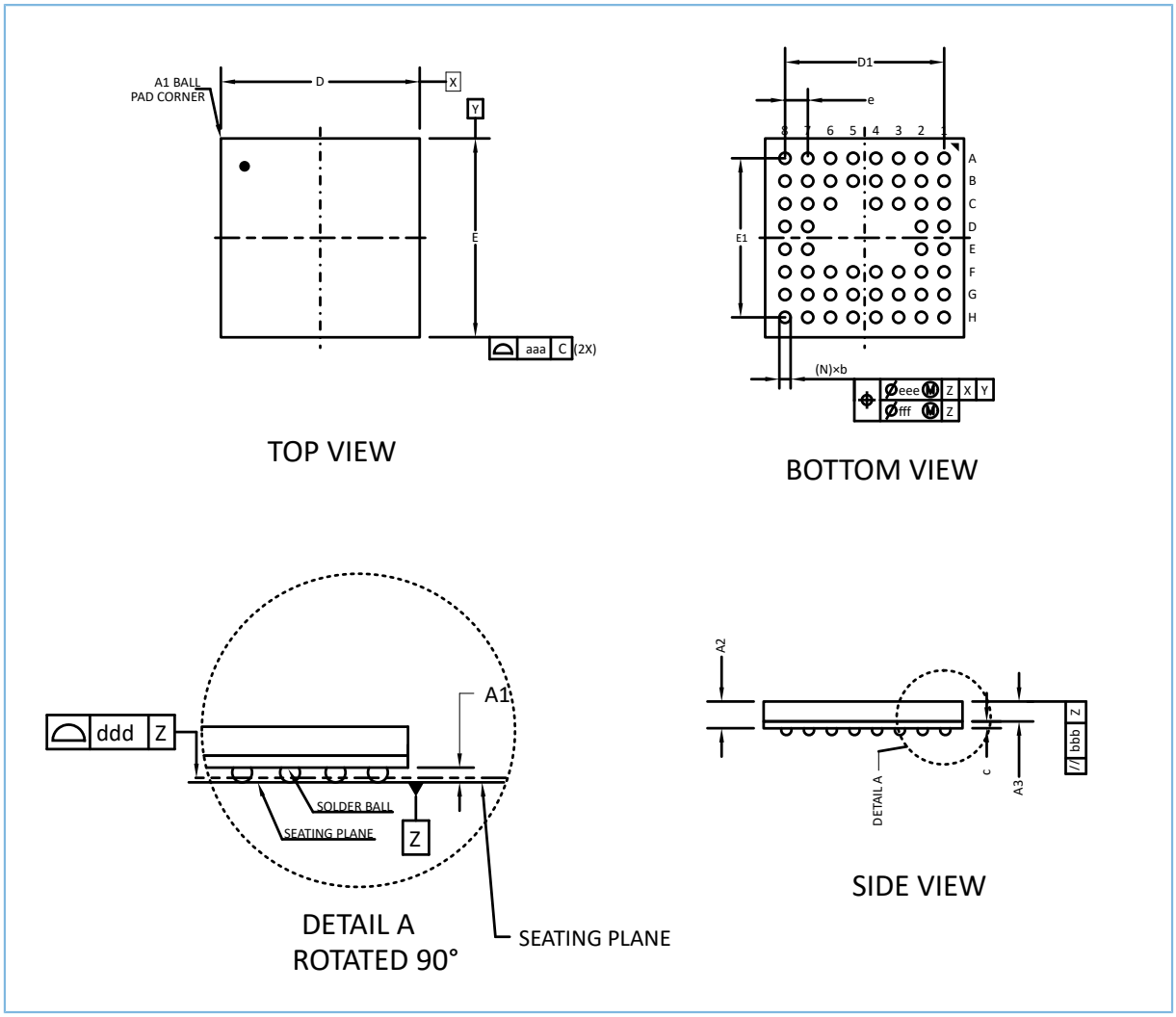

Figure 14-3 BGA55 package outlines

Note:

Drawing is not to scale.

Table 14-6 BGA55 package dimensions

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.550	0.600	0.650	0.022	0.024	0.026
A1	0.090	0.120	0.150	0.004	0.005	0.006
A2	0.435	0.475	0.505	0.017	0.019	0.020
A3	0.350 REF.			0.014 REF.		
c	0.125 REF.			0.005 REF.		
D	-	3.500	-	-	0.138	-

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
E	-	3.500	-		0.138	
D1	-	2.800	-	-	0.110	-
E1	-	2.800	-	-	0.110	-
e	-	0.400	-	-	0.016	-
b	0.150	0.200	0.250	0.006	0.008	0.010
aaa	0.100			0.004		
bbb	0.100			0.004		
ddd	0.080			0.003		
eee	0.150			0.006		
fff	0.050			0.002		

Note:

Values in inches are converted from values in millimeters and rounded to 3 decimal digits.

14.4 QFN40

GR551x QFN40, including GR5513BEND QFN40 and GR5513BENDU QFN40, is a 40-pin and 5 x 5 x 0.75 (mm) package. It is qualified for MSL3.

Table 14-7 QFN40 package information

Parameter	Value	Unit	Tolerance
Package Size	5 x 5	mm	±0.1 mm
QFN Pad Count	40		
Total Thickness	0.75	mm	±0.05 mm
QFN Pad Pitch	0.40		
Pad Width	0.20		±0.05 mm
Exposed Pad Size	3.7 x 3.7		±0.1 mm

The figure below shows the QFN40 package outlines.

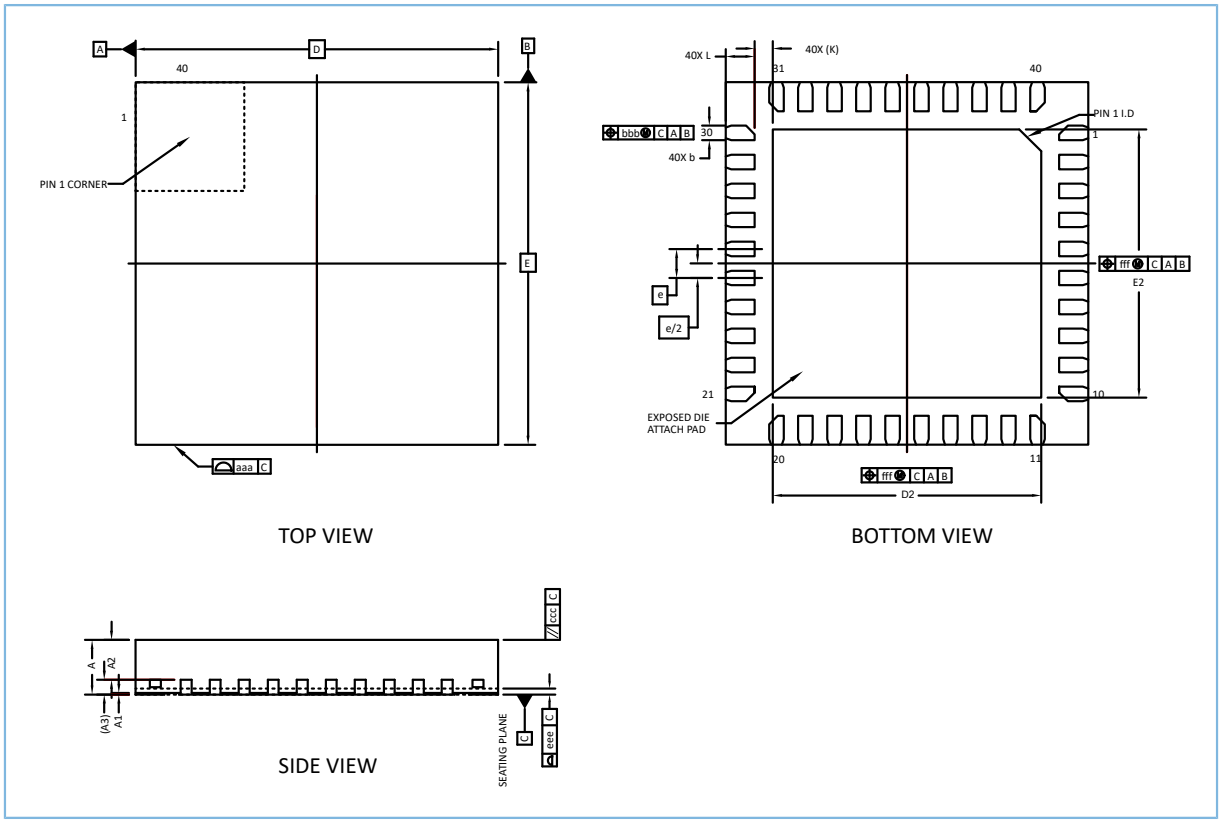

Figure 14-4 QFN40 package outlines

Note:

Drawing is not to scale.

Table 14-8 QFN40 package dimensions

Symbol	Dimensions in mm			Dimensions in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.700	0.750	0.800	0.028	0.030	0.032
A1	0.000	0.020	0.050	0.000	0.001	0.002
A2	-	0.550	-	-	0.022	-
A3	0.203 REF.			0.008 REF.		
b	0.150	0.200	0.250	0.006	0.008	0.010
D	5.000 BSC.			0.197 BSC.		
E	5.000 BSC.			0.197 BSC.		
e	0.400 BSC.			0.016 BSC.		
D2	3.600	3.700	3.800	0.142	0.146	0.150
E2	3.600	3.700	3.800	0.142	0.146	0.150
L	0.300	0.400	0.500	0.012	0.016	0.020

Symbol	Dimensions in mm			Dimensions in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
K	0.250 REF.			0.010 REF.		
aaa	0.100			0.004		
ccc	0.100			0.004		
eee	0.080			0.003		
bbb	0.100			0.004		
fff	0.100			0.004		

Note:

Values in inches are converted from values in millimeter and rounded to 3 decimal digits.

15 Ordering Information

GR551x offers devices below for different application requirements.

Table 15-1 GR551x ordering information

Features	GR5515IGND	GR5515IENDU	GR5515I0NDA	GR5515RGBD	GR5515GGBD	GR5513BEND	GR5513BENDU
CPU	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F
RAM	256 KB	256 KB	256 KB	256 KB	256 KB	128 KB	128 KB
Flash	1 MB	512 KB	N/A	1 MB	1 MB	512 KB	512 KB
Package (mm)	QFN56 (7 x 7 x 0.75)	QFN56 (7 x 7 x 0.75)	QFN56 (7 x 7 x 0.75)	BGA68 (5.3 x 5.3 x 0.88)	BGA55 (3.5 x 3.5 x 0.60)	QFN40 (5 x 5 x 0.75)	QFN40 (5 x 5 x 0.75)
I/O Number	39	39	39	39	29	22	22

Note:

- GR5515IENDU and GR5513BENDU are embedded with wide-voltage Flash, with Flash power supply from 1.65 V to 3.6 V.
- GR5515I0NDA supports external wide-voltage Flash (range: 1.65 V–3.6 V).

16 Glossary

Table 16-1 Glossary

Name	Description
ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AMS	Analog Mix Signal
AON	Always-on
AONWDG	Always-on Watchdog Timer
APB	Advanced Peripheral Bus
BB	Baseband
Bluetooth LE	Bluetooth Low Energy
BOD	Brown-out Detector
DPA	Differential Power Analysis
IFS	Inter Frame Spacing
LDO	Low Dropout
LNA	Low Noise Amplifier
LPD	Low Power Domain
NVM	Non-volatile memory
NRND	Not Recommended for New Designs
PLL	Phase Locked Loop
PMU	Power Management Unit
POR	Power-on Reset
RNG	RING Oscillator
SoC	System-on-Chip
TPMS	Tire Pressure Monitor System
W1C	Write 1 to Clear
XO	Crystal Oscillator
VIO	I/O Voltage
Typ	Typical
SNR	Signal to Noise Ratio
PA	Power Amplifier
IRQ	Interrupt Request
LSB	Least Significant Bit
MSB	Most Significant Bit

Name	Description
DFE	Digital Frontend

17 Reference Documents

The following documents are referred in this datasheet:

Table 17-1 Reference Documents

Documents	Description
GR551x Hardware Design Guidelines	Introduce GR551x hardware design.
GR551x Developer User Guide	Introduce GR551x SDK and its related resources to developers.
ETSI TS 131 101	Universal Mobile Telecommunications System (UMTS); LTE; UICC-terminal interface; Physical and logical characteristics https://www.etsi.org/deliver/etsi_ts/131100_131199/131101/15.00.00_60/ts_131101v150000p.pdf

18 Legal and Contact Information

Trademarks and Permissions

GOODIX and other Goodix trademarks are trademarks of Shenzhen Goodix Technology Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Disclaimer

Information contained in this document is intended for your convenience only and is subject to change without prior notice. It is your responsibility to ensure its application complies with technical specifications.

Shenzhen Goodix Technology Co., Ltd. (hereafter referred to as “Goodix”) makes no representation or guarantee for this information, express or implied, oral or written, statutory or otherwise, including but not limited to representation or guarantee for its application, quality, performance, merchantability or fitness for a particular purpose. Goodix shall assume no responsibility for this information and relevant consequences arising out of the use of such information.

Without written consent of Goodix, it is prohibited to use Goodix products as critical components in any life support system. Under the protection of Goodix intellectual property rights, no license may be transferred implicitly or by any other means.

Shenzhen Goodix Technology Co., Ltd.

Headquarters: Floor 12-13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

TEL: +86-755-33338828 Zip Code: 518000

Website: www.goodix.com

19 Revision History

Table 19-1 Revision history

Version	Date	Description
1.0	2019-12-27	Initial release.
1.3	2020-03-16	Updated some register names based on the design specification and updated the pinout diagram to the top view.
1.5	2020-05-30	• Updated SoC models, pinout packages, and the memory of SoC. • Updated some descriptions of peripherals, such as I2S, DMA. • Added "Absolute Maximum Ratings".
1.6	2020-06-30	• Updated the package information of QFN56, BGA68, BGA55, and QFN40. • Updated a few values for parameters in Electrical Specifications: t_{HD_DAT} in I2C, t_{VMO} and t_{HMO} in SPI and QSPI. • Updated the max value of supply voltage from 4.35 V to 3.8 V, and updated the typical value of I/O voltage from 3.6 V to 3.3 V. • Removed ADC reference voltage 2.0 V.
1.7	2020-08-30	• Added the GR5515I0ND SoC, and updated the pinout package and the feature, such as Flash, supply voltage, and I/O voltage. • Updated the value of input resistance of ADC in "Electrical Specifications". • Added "DMA Transfer".
1.8	2020-11-30	Optimized the descriptions in "Pin Mux", and corrected the tolerance value of Exposed Pad Size in "QFN56".
1.9	2020-12-30	• Added two alignment modes in "PWM". • Corrected register names in "I2S" and "ISO/IEC 7816-3 Master". • Added "Duty Cycle Calculation".
2.0	2021-03-02	• Updated function descriptions on the TEST_MODE pin for all GR551x SoCs in "Pinout". • Added the descriptions on I/O number for all GR551x SoCs in "GR551x Overview". • Updated descriptions on I/O voltage of GR5515I0ND in "Features", "GR5515I0ND", "QFN40", and "Power Management".
2.1	2021-04-16	Added chip uid and package type imformation in "NVM Storage (eFuse)".
2.2	2021-06-01	• Added a note in "Function Description of PWM". • Added a note for classifying GR5515RGBD as NRND.
2.3	2021-08-20	• Introduced GR5515IENDU and GR5515I0NDA and updated relevant chapters/sections (“GR551x Overview”, “Features”, “Pinout”, “Package Information” and "Ordering Information") accordingly. • Updated the pins name of RTC_N, RTC_P, XON, XOP to RTC_IN, RTC_OUT,XON_OUT, XO_IN in "Pinout". • Updated the description of Sleep mode in "Features".

Version	Date	Description
2.4	2022-02-20	Introduced GR5513BENDU, a wide-voltage SoC.
2.5	2023-01-19	• Deleted the GR5515I0ND SoC. • Updated the GR5515RGBD status from "NRND" to "Active". • Added a note for classifying GR5513BEND as "NRND".