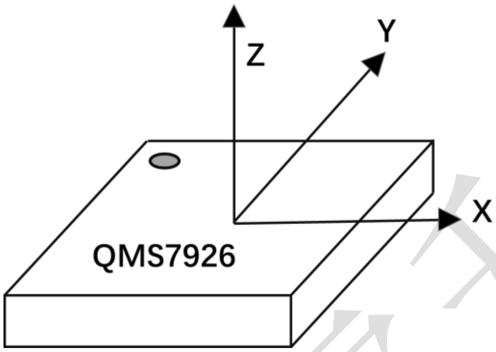
Abstract
Single-Chip Smart Sensor
QMS7926

Advanced Information

QMS7926 is a highly integrated intelligent sensor chip, which are packaged in BGA60,4x4x1.2mm3. Its built-in main functional modules include: high precision and low power consumption triaxial acceleration sensor and signal processing circuit, ARM M0 kernel microprocessor, 2.4G RF transceiver / Bluetooth BLE broadcast module and so on.



◆ Key Features

- 3-Axis Accelerometer with low noise, high accuracy, low power consumption and offset trimming
- Up to 48Mhz ARM® Cortex™-M0 32-bit processor
- 2.4 GHz Bluetooth Low Energy (BLE) transceiver
- Supply voltage range 1.8V to 3.6V

◆ 3-Axis Accelerometer

- 14-Bit ADC with low noise accelerometer sensor
- High resolution allows for motion and tilt sensing
- I2C Interface with Standard and Fast modes
- low power consumption (2-50uA low power conversion current)
- Automatically maintains sensor's sensitivity under wide operation voltage range and compatible with battery powered applications

◆ ARM® Cortex™-M0 32-bit processor

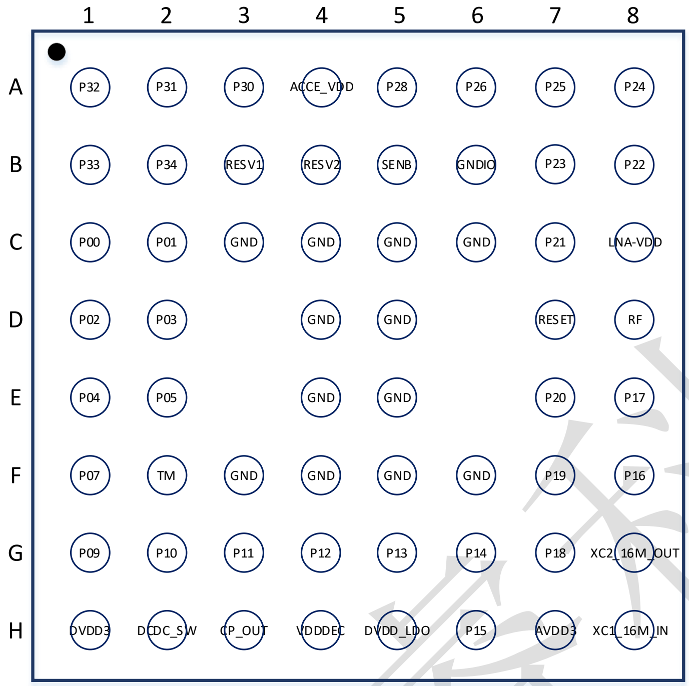
- Memory
  - ➢ 512KB in-system flash memory
  - ➢ 128KB ROM
  - ➢ 138KB SRAM, all programmable retention in sleep mode
- 31 general purpose I/O pins
  - ➢ All pins can be configured as serial interface and programmable IO MUX function mapping
  - ➢ All pins can be configured for wakeup
  - ➢ 16 pins for triggering interrupt
  - ➢ 3 quadrature decoder (QDEC)
  - ➢ 6-channel PWM
  - ➢ 4-channel I2S
  - ➢ 2-channel PDM
  - ➢ 2-channel I2C
  - ➢ 2-channel SPI
  - ➢ 1-channel UART
  - ➢ JTAG
- 6-channel 12bit ADC with analog PGA
- 4-channel 32bit timer, one watchdog timer
- Real timer counter (RTC)
- Power, clock, reset controller
- Flexible power management
  - ➢ Supply voltage range 1.8V to 3.6V
  - ➢ Embedded buck DC-DC
  - ➢ Embedded LDOs
  - ➢ Battery monitor: Supports low battery detection
  - ➢ 2µA @ Sleep Mode with 32KHz RTC
  - ➢ 0.7µA @ OFF Mode (IO wake up only)

- 2.4 GHz Bluetooth Low Energy (BLE) transceiver
  - 2.4 GHz transceiver
    - Compliant to Bluetooth 5.0, ETSI EN 300 328 and EN 300 440 Class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan)
    - Sensitivity:
      -97dBm@BLE 1Mbps data rate
      -103dBm@BLE 125Kbps data rate
    - TX Power -20 to +10dBm in 3dB steps
    - Receiver: 8mA @sensitivity level
    - Transmitter: 8mA @0dBm TX power
    - Single-pin antenna: no RX/TX switching required
    - RSSI (1dB resolution)
  - RC oscillator hardware calibrations
    - 32KHz RC oscillator automatic calibration
    - 32MHz RC oscillator automatic calibration
  - AES-128 encryption hardware
    - AES-ECB
    - AES-CCM
  - Link layer hardware
    - Automatic packet assembly
    - Automatic packet detection and validation
    - Auto Re-transmit
    - Auto ACK
    - Hardware Address Matching
    - Random number generator

- Operating condition
  - Supply voltage range: 1.8V to 3.6V
  - Operating temperature: -40℃ to 85℃

- RoHS Package: BGA-60

Pin Assignment



	1	2	3	4	5	6	7	8
A	P32	P31	P30	ACCE_VDD	P28	P26	P25	P24
B	P33	P34	RESV1	RESV2	SENB	GNDIO	P23	P22
C	P00	P01	GND	GND	GND	GND	P21	LNA-VDD
D	P02	P03		GND	GND		RESET	RF
E	P04	P05		GND	GND		P20	P17
F	P07	TM	GND	GND	GND	GND	P19	P16
G	P09	P10	P11	P12	P13	P14	P18	XC2_16M_OUT
H	DVDD3	DCDC_SW	CP_OUT	VDDDEC	DVDD_LDO	P15	AVDD3	XC1_16M_IN

Pin Configurations

Pin Number	Pin Name	Description
A1	P32	all functions configurable
		*Note: Not support interrupt and ADC function
A2	P31	all functions configurable
		*Note: Not support interrupt and ADC function
A3	P30	all functions configurable
		*Note: Not support interrupt and ADC function
A4	ACCE_VDD	power supply of Accelerometer
A5	P28/SDA	it is connected between P28 of MCU and SDA of Accelerometer internally
		*Note: Not available for other signals.
A6	P26/SCL	it is connected between P26 of MCU and SCL of Accelerometer internally
		*Note: Not available for other signals.
A7	P25	all functions configurable/test_mode_select [1]
		*Note: Not support interrupt function and ADC function
A8	P24	all functions configurable/test_mode_select [0]
		*Note: Not support interrupt function and ADC function
B1	P33	all functions configurable
		*Note: Not support interrupt and ADC function
B2	P34	all functions configurable
		*Note: Not support interrupt and ADC function
B3	RESVE1	reserved for Accelerometer
		*Note: Float or connect to GND

B4	RESVE2	Reserved for Accelerometer
		*Note: Float or connect to GND
B5	SENB	protocol selection of Accelerometer
		*Note: Connect to VDD
B6	GNDIO	test pin of Accelerometer
		*Note: Connect to GND
B7	P23	all functions configurable
		*Note: Not support interrupt function and ADC function
B8	P22	all functions configurable
		*Note: Not support interrupt function and ADC function
C1	P00	all functions configurable/JTAG_TDO
		*Note: Not support ADC function
C2	P01	all functions configurable/JTAG_TDI
		*Note: Not support ADC function
C3	GND	Ground
C4	GND	Ground
C5	GND	Ground
C6	GND	Ground
C7	P21	all functions configurable
		*Note: Not support interrupt function and ADC function
C8	LNA/TRX_VDD	LNA and TRX VDD
D1	P02	all functions configurable/JTAG_TMS
		*Note: Not support ADC function
D2	P03	all functions configurable/JTAG_TCK
		*Note: Not support ADC function
D4	GND	Ground
D5	GND	Ground
D7	RST_N	reset pin
D8	RF	RF antenna
E1	P04	all functions configurable
		*Note: Not support ADC function
E2	P05/AINT1	it is connected between P05 of MCU and INT1 of Accelerometer internally
		*Note: Not available for other signals and keep NC
E4	GND	Ground
E5	GND	Ground
E7	P20	all functions configurable/AIO<9>/Micphone bias output
		*Note: Not support interrupt function
E8	P17/32K_OUT	all functions configurable/AIO<6>/32k crystal output
F1	P07	all functions configurable
		*Note: Not support ADC function
F2	TM	test mode pin

F3	GND	Ground
F4	GND	Ground
F5	GND	Ground
F6	GND	Ground
F7	P19	all functions configurable/AIO<8>/PGA differential negative input
		*Note: Not support interrupt function
F8	P16/32K_IN	all functions configurable/AIO<5>/32K crystal input
G1	P09	all functions configurable
		*Note: Not support ADC function
G2	P10	all functions configurable
		*Note: Not support ADC function
G3	P11	all functions configurable/AIO<0>
G4	P12	all functions configurable/AIO<1>
G5	P13	all functions configurable/AIO<2>
G6	P14	all functions configurable/AIO<3>
G7	P18	all functions configurable/AIO<7>/PGA differential positive input
		*Note: Not support interrupt function
G8	XC2_16M_OUT	16M crystal output
H1	DVDD3	3V power supply for digital IO, DCDC, Charge pump
H2	DCDC_SW	buck dcdc output
H3	CP_OUT	charge pump output
H4	VDDDEC	1.2V VDD_CORE, digital LDO output
H5	DVDD_LDO	digital LDO input
H6	P15	all functions configurable/AIO<4>
H7	AVDD3	3V power supply for analog IO, bg, rcosc, etc
H8	XC1_16M_IN	16M crystal input

CONTENTS

3-Axis Accelerometer

1 INTERNAL SCHEMATIC DIAGRAM

1.1 Internal Schematic Diagram

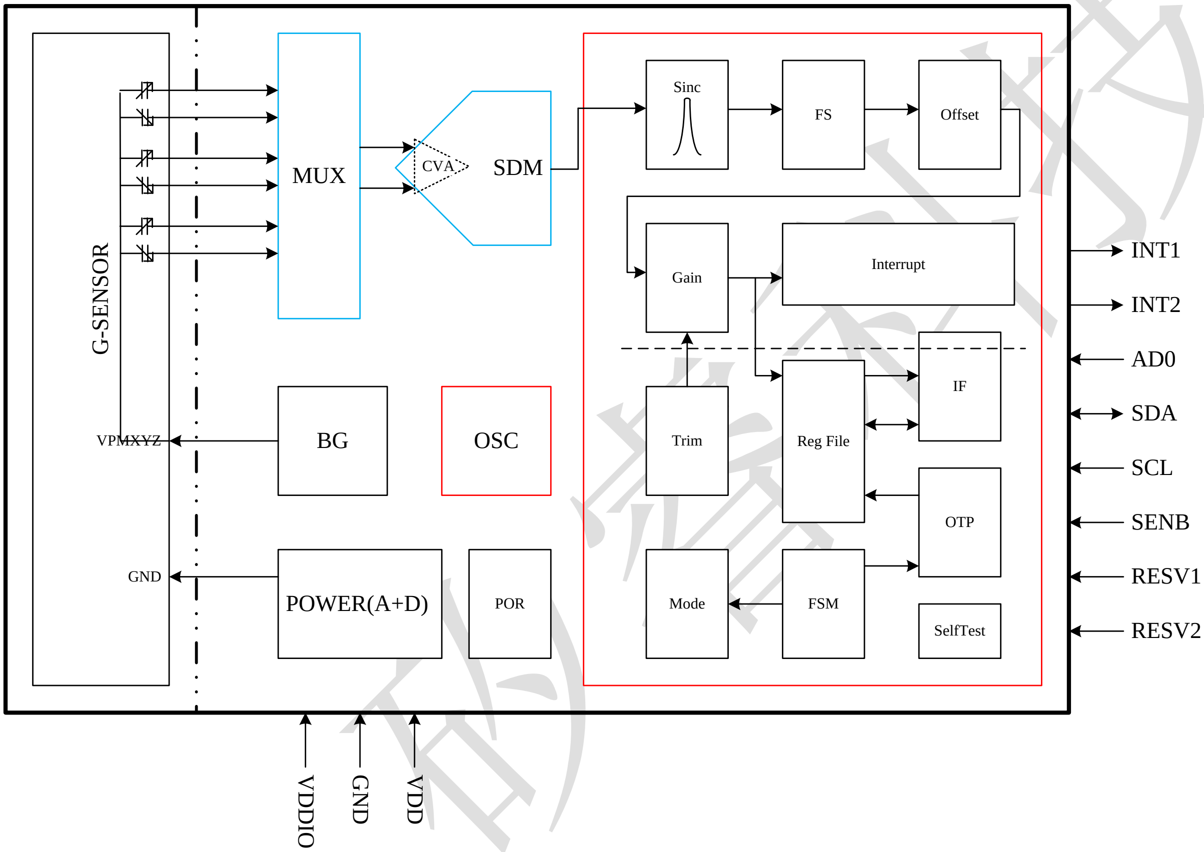


Figure 1. Block Diagram

Table 1. Block Function

Block	Function
Transducer	3-axis acceleration sensor
CVA	Charge-to-Voltage amplifier for sensor signals
Interrupt	Digital interrupt engine, to generate interrupt signal on data conversion, and motion function
FSM	Finite state machine, to control device in different mode
I^2C/SPI	Interface logic data I/O
OSC	Internal oscillator for internal operation
Power	Power block, including LDO

2 SPECIFICATIONS AND I/O CHARACTERISTICS

2.1 Product Specifications

Table 2. Specifications (* Tested and specified at 25°C and 3.0V VDD except stated otherwise.)

Parameter	Conditions	Min	Typ	Max	Unit
Supply voltage VDD	VDD, for internal blocks	1.71	3.3	3.6	V
I/O voltage VDDIO	VDDIO, for IO only	1.71	3.3	VDD	V
Standby current	VDD and VDDIO on		1		μA
Low power current	ODR=268 Hz		50		μA
	ODR=134 Hz		25.3		
	ODR=67 Hz		12.9		
	ODR=33.6 Hz		6.7		
	ODR=13.4 Hz		2.9		
	ODR=6.7 Hz		1.7		
Low noise current	ODR=32.5 Hz		100		μA
	ODR=21.6 Hz		83.3		
	ODR=13 Hz		50		
	ODR=6.5 Hz		25		
BW	Programmable bandwidth		0.16~168		Hz
Data output rate (ODR)	2*BW		0.32~336		Samples /sec
Startup time	From the time when VDD reaches to 90% of final value to the time when device is ready for conversion		2		ms
Wakeup time	From the time device enters into active mode to the time device is ready for conversion		1		ms
Operating temperature		-40		85	℃
Acceleration Full Range			±2/±4/±8/ ±16/±32		g
Sensitivity	FS=±2g		4096		LSB/g
	FS=±4g		2048		
	FS=±8g		1024		
	FS=±16g		512		
	FS=±32g		256		
Sensitivity Temperature Drift	FS=±2g, Normal VDD Supplies		±0.02		%/℃

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity tolerance	Gain accuracy		±4		%
Zero-g offset	FS=±2g, Normal VDD Supplies		±80		mg
Zero-g offset Temperature Drift	FS=±2g, Normal VDD Supplies		±2		mg/℃
Noise density	FS=±2g, run state		200		µg/√Hz
Nonlinearity	FS=±2g, Best fit straight line,		±0.5		%FS
Cross Axis Sensitivity			1		%

2.2 Absolute Maximum Ratings

Table 3. Absolute Maximum Ratings (Tested at 25°C except stated otherwise.)

Parameters	Condition	Min	Max	Units
VDD		-0.3	5.4	V
VDDIO		-0.3	5.4	V
ESD	HBM		2	kV
Shock Immunity	Duration < 200µS		10000	g
Storage temperature		-50	150	℃

2.3 I/O Characteristics

Table 4. I/O Characteristics

Parameter	Symbol	Pin	Condition	Min.	TYP.	Max.	Unit
Voltage Input High Level 1	$V_{IH}1$	SDA, SCL		0.7*VDDIO		VDDIO +0.3	V
Voltage Input Low Level 1	$V_{IL}1$	SDA, SCL		-0.3		0.3*VDDIO	V
Voltage Output High Level	V_{OH}	INT1, INT2	Output Current ≥-100µA	0.8*VDDIO			V
Voltage Output Low Level	V_{OL}	INT1, INT2, SDA	Output Current ≤100µA(INT) Output Current ≤1mA (SDA)			0.2*VDDIO	V

3 BASIC DEVICE OPERATION

3.1 Acceleration sensor

The acceleration sensor circuit consists of tri-axial sensors and application specific support circuits to measure the acceleration of device. When a DC power supply is applied to the sensor, the sensor converts any accelerating incident in the sensitive axis directions to charge output.

3.2 Power Management

Device has two power supply pins. VDD is the main power supply for all of the internal blocks, including analog and digital. VDDIO is a separate power supply, for digital interface only.

The device contains a power-on-reset generator. It generates reset pulse as power on, which can load the register's default value, for the device to function properly.
To make sure the POR block functions well, we should have such constrains on the timing of VDD and VDDIO.

The device should turn-on both power pins in order to operate properly. When the device is powered on, all registers are reset by POR, then the device transits to the standby mode and waits for further commends.

Table 6 provides references for four power states.

Table 6. Power States

Power State	VDD	VDDIO	Power State description
1	0V	0V	Device Off, No Power Consumption
2	0V	1.71V~3.6V	Not allowed. User need to make sure that VDDIO is less than VDD. Otherwise, there will be leakage from VDDIO to VDD through internal ESD devices
3	1.71V~3.6V	0V	Device Off, Same Current as Standby Mode
4	1.71V~3.6V	1.71V~VDD	Device On, Normal Operation Mode, Enters Standby Mode after POR

3.3 Power On/Off Time

Device has two power supply pins and two ground pins. VDD is the main power supply for all of the internal blocks, including analog and digital. VDDIO is a separate power supply, for digital interface only. GND is 0V supply for all of internal blocks, and GNDIO for digital interface.

There is no limitation on the voltage levels of VDD and VDDIO relative to each other, as long as they are within operating range.

The device contains a power-on-reset generator. It generates reset pulse as power on, which can load the register's default value, for the device to function properly.

To make sure the POR block functions well, we should have such constrains on the timing of VDD.

The power on/off time related to the device is in **Table 7**

Table 7. Time Required for Power On/Off

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
POR Completion Time	PORT	Time Period After VDD and VDDIO at Operating Voltage to Ready for I^{2}C Commend and Analogy Measurement.			250	µs
Power off Voltage	SDV	Voltage that Device Considers to be Power Down.			0.2	V
Power on Interval	PINT	Time Period Required for Voltage Lower Than SDV to Enable Next POR	100			µs
Power on Time	PSUP	Time Period Required for Voltage from SDV to 90% of final value			50	ms

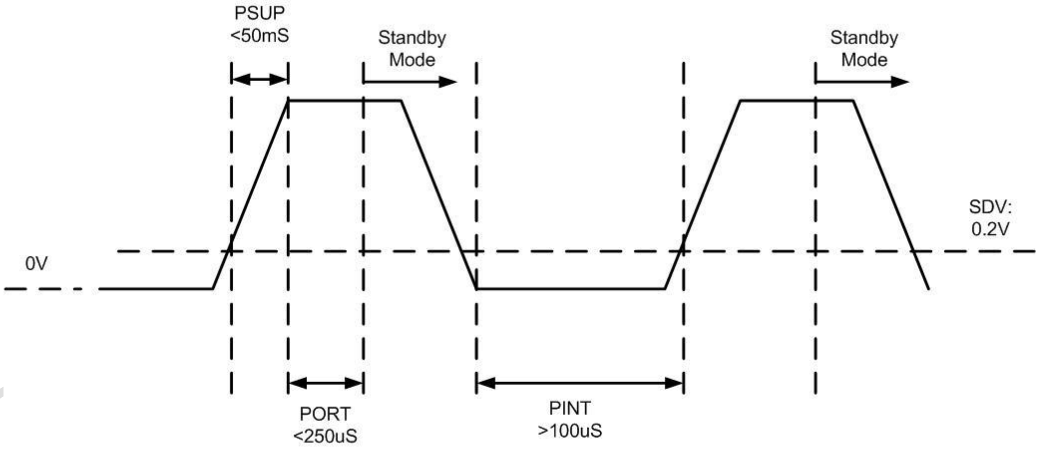


Figure 9. Power On/Off Timing

3.4 Communication Bus Interface I²C and Its Addresses

This device will be connected to a serial interface bus as a slave device under the control of a master device, such as the processor. Control of this device is carried out via I²C.

This device is compliant with I²C -Bus Specification, document number: 9398 393 40011. As an I²C compatible device, this device has a 7-bit serial address and supports I²C protocols. This device supports standard and fast speed modes, 100 kHz and 400 kHz, respectively. External pull-up resistors are required to support all these modes.

There are two I^{2}C addresses selected by connecting pin 1 (AD0) to GND or VDDIO. The first six MSB are hardware configured to “001001” and the LSB can be configured by AD0.

Table 8. I^{2}C Address Options

AD0 (pin 1)	I^{2}C Slave Address(HEX)	I^{2}C Slave Address(BIN)
Connect to GND	12	0010010

4 MODES OF OPERATION

4.1 Modes Transition

QMA7981 has two different operational modes, controlled by register (0x11), MODE_BIT. The main purpose of these modes is for power management. The modes can be transited from one to another, as shown below, through I^{2}C commands. The default mode after power-on is standby mode.

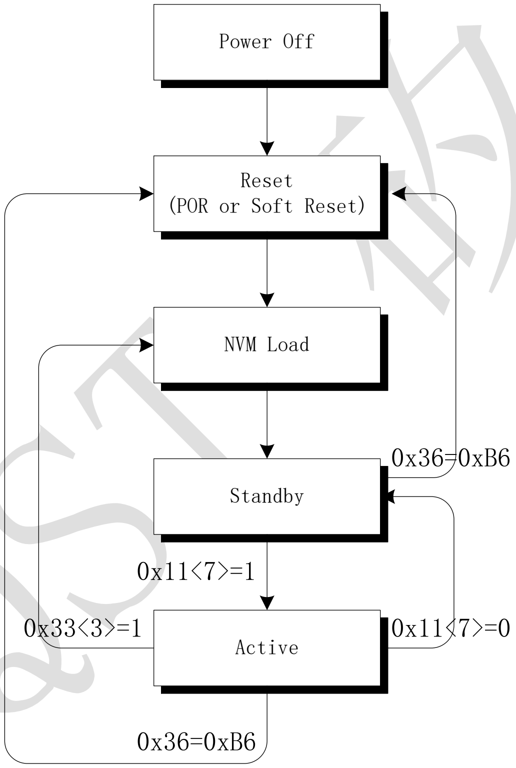


Figure 10. Basic operation flow after power-on

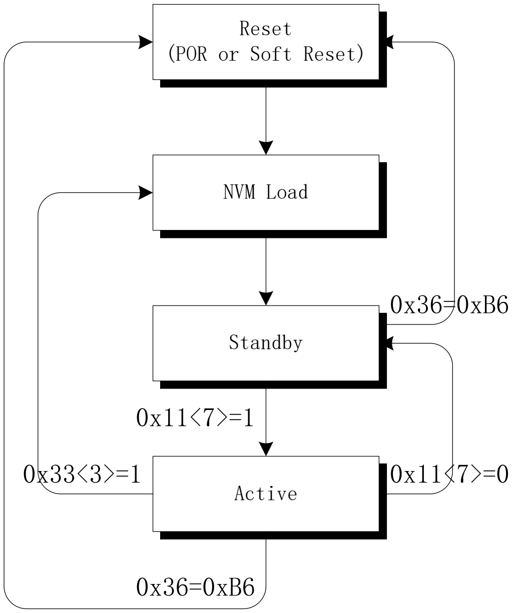


Figure 11. The work mode transferring

The default mode after power on is standby mode. Through I²C instruction, device can switch between standby mode and active mode. With SOFTRESET by writing 0xB6 into register 0x36, all of the registers will get default values. SOFTRESET can be done both in active mode and in standby mode. Also, by writing 1 in NVM_LOAD (0x33<3>) when device is in active mode, the NVM related image registers will get default value from NVM, however, other registers will keep the values of their own.

4.2 Description of Modes

4.2.1 Active Mode

In active mode, the ADC digitizes the charge signals from transducer, and digital signal processor conditions these signals in digital domain, processes the interrupts, and send data to Data registers (0x01~0x06).

4.2.2 Standby Mode

In standby mode, most of the blocks are off, while device is ready for access through I²C. Standby mode is the default mode after power on or soft reset. Device can enter into this mode by set the soft reset register (0x36) to 0xB6 or set the MODE_BIT (0x11<7>) to logic 0.

Besides the above two modes, the device also contains NVM loading state. This state is used to reset the value of the NVM related image registers. There are two bits related to this state. When NVM_LOAD (0x33<3>) is set to 1, NVM loading starts. When the device is in NVM loading state, NVM_RDY (0x33<2>) is set to logic 0 by device. After NVM loading is finished, NVM_RDY (0x33<2>) is set back to logic 1 by device, and NVM_LOAD is reset to 0 by device automatically. NVM loading can only happen when NVM_LOAD is set to 1 in active mode. If the user sets this NVM_LOAD bit to 1 in standby mode, the device will not take the action until it enters into active state by setting MODE_BIT (0x11<7>) to logic 1.

After loading NVM, the device will enter into standby mode directly.

The loading time for NVM is about 100uS.

5 Functions and interrupts

ASIC support interrupts, such as STEP_INT, DRDY_INT, ANY_MOT_INT, SIG_MOT_INT, NO_MOT_INT, RAISE_INT, etc.

5.1 STEP_ INT

The STEP/STEP_QUIT detect that the user is entering/exiting step mode. When the user enter into step mode, at least one axis sensor data will vary periodically, by numbering the variation periods the step counter can be calculated.

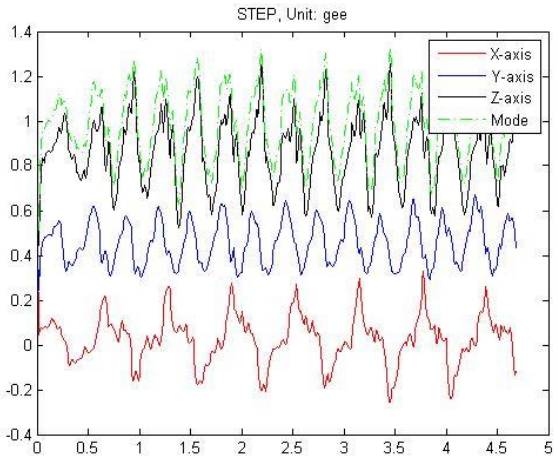


Figure 12. STEP/STEP_QUIT

Median data (max+min) /2 is called dynamic threshold, the max and min data can be updated by certainly samples, the sample number can be set by register STEP_SAMPLE_CNT(0x12<6:0>). When the sensor data decreasing (or increasing) through the dynamic threshold, a user run step is detected.

Register STEP_PRECISION (0x13<6:0>) is used as threshold when updating the new collected sensor data. Sensor data below the threshold will be discarded, this helps removing unstable variations causing failed detection.

The run step event happened at certain interval timing. All of the events outside the timing window will not be regarded as a run step and the step counter will not counted. The timing window can be set by register STEP_TIME_UP(0x15) and STEP_TIME_LOW(0x14), the conversion ODR numbers ranged from STEP_TIME_LOW *ODR to 8* STEP_TIME_UP*ODR .

STEP_COUNT_PEAK<2:0> is used to set a fixed peak value for step detection, 0.05G~0.4G can be set. STEP_COUNT_P2P<2:0> is used to set a peak to peak threshold for step detection, 0.3G~1G can be set.

To remove unstable variation which will cause false STEP event detection, the step counter considers steps as valid step events only after some continuous steps detected; the start threshold can be set by 0x1F<7:5>. Also, the step counter register STEP_CNT<23:0> ({0x0E,0x08,0x07}) will be updated immediately by the setting number, and interrupt STEP is also generated.

The related interrupt status bit is STEP_INT (0x0A<3>) and SIG_STEP (0x0A<6>). When the interrupt is generated, the value of STEP_INT will be set to logic 1, which will be cleared after the interrupt status register is read by user. STEP_IEN/SIG_STEP_IEN (0x16<3>/0x16<6>) is the enable bit for the STEP_INT/SIG_STEP_INT. Also, to get this interrupt on PIN_INT1 and/or PIN_INT2, we need to set INT1_STEP (0x19<3>)/INT1_SIG_STEP (0x19<6>)or INT2_STEP (0x1B<3>) /INT2_SIG_STEP (0x1B<6>) to logic 1, to map the internal interrupt to the interrupt PINs.

5.2 DRDY_INT

The width of the acceleration data is 14 bits, in two's complement representation. The data of each axis is split into 2 parts, the MSB part (one byte contains bit 13 to bit 6) and the LSB part (one byte contains bit 5 to bit 0). Reading data should start with LSB part. When user is reading the LSB byte of data, to ensure the integrity of the acceleration data, the content of MSB can be locked, by setting SHADOW_DIS (0x21<6>) to logic 0. This lock function can be disabled by setting SHADOW_DIS to logic 1. Without lock, the MSB and LSB content will be updated by new value immediately. The bit NEW_DATA in the LSB byte is the flag of the new data. If new data is updated, this NEW_DATA flag will be 1, and will be cleared when corresponding MSB or LSB is read by user.

Also, the user should note that even with SHADOW_DIS=0, the data of 3 axes are not guaranteed from the same time point.

The device supports four different acceleration measurement ranges. The range is setting through RANGE (0x0F<3:0>), and the details as following:

RANGE	Acceleration range	Resolution
0001	2g	244ug/LSB
0010	4g	488ug/LSB
0100	8g	977ug/LSB
1000	16g	1.95mg/LSB
1111	32g	3.91mg/LSB
Others	2g	244ug/LSB

The interrupt for the new data serves for the synchronous data reading for the host. It is generated after storing a new value of z-axis acceleration data into data register. This interrupt will be cleared automatically when the next data conversion cycle starts, and the interrupt will be effective about 64*MCLK, and automatically cleared.
The interrupt mode for the new data is fixed to be non-latched.

5.3 ANY_MOT_INT

Any motion Any motion detection uses slope between two successive data to detect the changes in motion. It generates interrupt when a preset threshold ANY_MOT_TH (0x2E) is exceeded.

The time difference between two successive data depends on the output data rate (ODR).

$$\mathrm{Slope(t1)} = \big(acc(t1) - acc(t0)\big) * ODR$$

The any motion detection criteria are fulfilled and interrupt is generated if any of enabled channels exceeds ANY_MOT_TH for ANY_MOT_DUR (0x2C<1:0>) consecutive times.

As long as all the enabled channels data fall or stay below ANY_MOT_TH for ANY_MOT_DUR consecutive times, the interrupt will be reset unless the interrupt signal is latched.

The any motion detection engine will send out the signals of axis which triggered the interrupt (ANY_MOT_FIRST_X (0x09<0>), ANY_MOT_FIRST_Y (0x09<1>), ANY_MOT_FIRST_Z (0x09<2>)) and the sign of the motion (ANY_MOT_SIGN (0x09<3>))

5.4 SIG_MOT_INT

A significant motion is a motion due to a change in user location.
The algorithm is as following:

1) Look for movement, same setting as any motion detection
2) If movement detected, sleep for T_Skip (0x2F<3:2>)
3) Look for movement
    a) If no movement detected within T Proof (0x2F<5:4>), go back to 1
    b) If movement detected, report a significant movement, and generate the interrupt

The significant motion detection and any motion detection are exclusive, user can select either one through SIG_MOT_SEL (0x2F<0>).
If significant motion is detected, the engine will set SIG_MOT_INT (0x0A<0>).

5.5 NO_MOT_INT

No-motion interrupt is generated if the slope (absolute value of acceleration difference) on all selected axes is smaller than the programmable threshold for a programmable time. Figure shows the timing for the no-motion interrupt. Register (0x2C) NO_MOT_DUR defines the delay times before the no-motion interrupt is generated. Table lists the delay times adjustable with register (0x2C) NO_MOT_DUR.

The no-motion interrupt is enabled per axis by writing logic 1 to bits (0x18) NO_MOTION_EN_X, (0x18) NO_MOTION_EN_Y, and (0x18) NO_MOTION_EN_Z, respectively. The no-motion threshold is set through the (0x2D) NO_MOT_TH register. The meaning of an LSB of (0x2D) NO_MOT_TH depends on the selected g-range: it corresponds to 3.91mg in 2g-range (7.81mg in 4g-range, 15.6mg in 8g-range, 31.25mg in 16g-range, 62.5mg in 32g-range). Therefore the maximum value is 996mg in 2g-range (2g in 4g-range, 4g in 8g-range, 8g in 16g-range, and 16g in 32g-range). The time difference between the successive acceleration samples depends on the selected ODR and equates to 1/ODR.

5.6 RAISE_INT

Raise wake algorithm is used to detect the action of raise hand (or hand down). The interrupt is enabled by writing logic 1 to bits (0X16[1]) RAISE_EN, (0X16[2]) HD_EN. User can adjust the sensitivity through the registers. The register RAISE_WAKE_SUM_TH(0X2A[5:0]) defines the strength of hand action (raise and down). The register RAISE_DIFF_TH(0X2A[7:6],0X2B[1:0]) defines the differential values of twice actions, when the hand behavior almost done the differential value will be smaller and we can use this register to set the threshold. RAISE_WAKE_PERIOD and RAISE_WAKE_TIMEOUT_TH define the duration of the total hand action.

5.7 Interrupt configuration

The device has the above 3 interrupt engines. Each of the interrupts can be enabled and configured independently. If the trigger condition of the enabled interrupt fulfilled, the corresponding interrupt status bit will be set to logic 1, and the mapped interrupt pin will be activated. The device has two interrupt PINs, INT1 and INT2. Each of the interrupts can be mapped to either PIN or both PINs.

The interrupt status registers INT_ST(0x09~0x0d) will update when a new data word is written into the data registers. If an interrupt is disabled, the related active interrupt status bit is disabled immediately.

When interrupt condition is fulfilled, related bit of interrupt will be set, until the associated interrupt condition is no more valid. Read operation to related register will also clear the register.

Device supports 2 interrupt modes, non-latched, and latched mode. The interrupt modes are set through LATCH_INT (0x21<0>).

In non-latched mode, the mapped interrupt pin will be set and/or cleared same as associated interrupt register bit. Also, the mapped interrupt pin can be cleared with read operation to any of the INT_ST(0x09~0x0d).

Exception to this is the new data interrupt and step interrupt, which are automatically reset after a fixed time (T_Pulse = 64/MCLK), no matter LATCH_INT (0x21<0>) is set to 0 or 1.

In latched mode, the clearings of mapped pins are determined by INT_RD_CLR (0x21<7>).
If the condition for trigging the interrupt still holds, the interrupt status will be set again with the next change of the data registers.

Mapping the interrupt pins can be set by INT_MAP (0x19~0x1B).

The electrical interrupt pins can be set INT_PIN_CONF (0x20<3:0>). The active logic level can be set to 1 or 0, and the interrupt pin can be set to open-drain or push-pull.

6 I2C COMMUNICATION PROTOCOL

6.1 I^{2}C Timings

Table 9 and Figure 11 describe the I^{2}C communication protocol times

Table 9. I^{2}C Timings

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
SCL Clock	f_{scl}		0		400	kHz
SCL Low Period	t_{low}		1			µs
SCL High Period	t_{high}		1			µs
SDA Setup Time	t_{sudat}		0.1			µs
SDA Hold Time	t_{hddat}		0		0.9	µs
Start Hold Time	t_{hdsta}		0.6			µs
Start Setup Time	t_{susta}		0.6			µs
Stop Setup Time	t_{susto}		0.6			µs
New Transmission Time	t_{buf}		1.3			µs
Rise Time	t_r					µs
Fall Time	t_f					µs

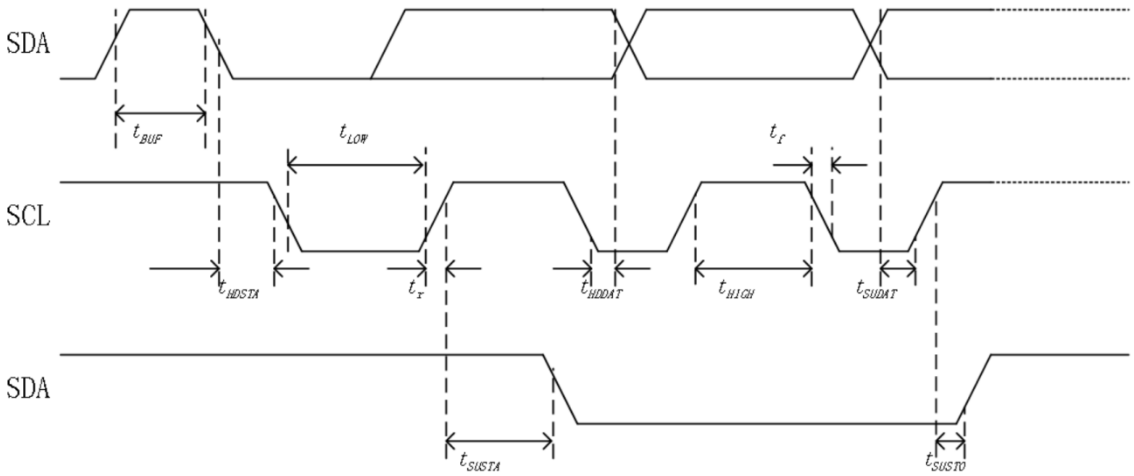


Figure 13. I^{2}C Timing Diagram

6.2 I^{2}C R/W Operation

6.2.1 Abbreviation

Table 10. Abbreviation

SACK	Acknowledged by slave
MACK	Acknowledged by master
NACK	Not acknowledged by master
RW	Read/Write

6.2.2 Start/Stop/Ack

START: Data transmission begins with a high to transition on SDA while SCL is held high. Once I^{2}C transmission starts, the bus is considered busy.

STOP:STOP condition is a low to high transition on SDA line while SCL is held high.

ACK: Each byte of data transferred must be acknowledged. The transmitter must release the SDA line during the acknowledge pulse while the receiver mush then pull the SDA line low so that it remains stable low during the high period of the acknowledge clock cycle.

NACK: If the receiver doesn’t pull down the SDA line during the high period of the acknowledge clock cycle, it’s recognized as NACK by the transmitter.

6.2.3 I^{2}C Write

I^{2}C write sequence begins with start condition generated by master followed by 7 bits slave address and a write bit (R/W=0). The slave sends an acknowledge bit (ACK=0) and releases the bus. The master sends the one byte register address. The slave again acknowledges the transmission and waits for 8 bits data which shall be written to the specified register address. After the slave acknowledges the data byte, the master generates a stop signal and terminates the writing protocol.

Table 11. I^{2}C Write

START	Slave Address							RW	SACK	Register Address (0x11)								SACK	Data (0x80)								SACK	STOP
	0	0	1	0	0	1	0	0		0	0	0	1	0	0	0	1		1	0	0	0	0	0	0	0		

6.2.4 I^{2}C Read

I^{2}C write sequence consists of a one-byte I^{2}C write phase followed by the I^{2}C read phase. A start condition must be generated between two phase. The I^{2}C write phase addresses the slave and sends the register address to be read. After slave acknowledges the transmission, the master generates again a start condition and sends the slave address together with a read bit (R/W=1). Then master releases the bus and waits for the data bytes to be read out from slave. After each data byte the master has to generate an acknowledge bit (ACK = 0) to enable further data transfer.

A NACK from the master stops the data being transferred from the slave. The slave releases the bus so that the master can generate a STOP condition and terminate the transmission.

The register address is automatically incremented and more than one byte can be sequentially read out. Once a new data read transmission starts, the start address will be set to the register address specified in the current I²C write command.

Table 12. I²C Read

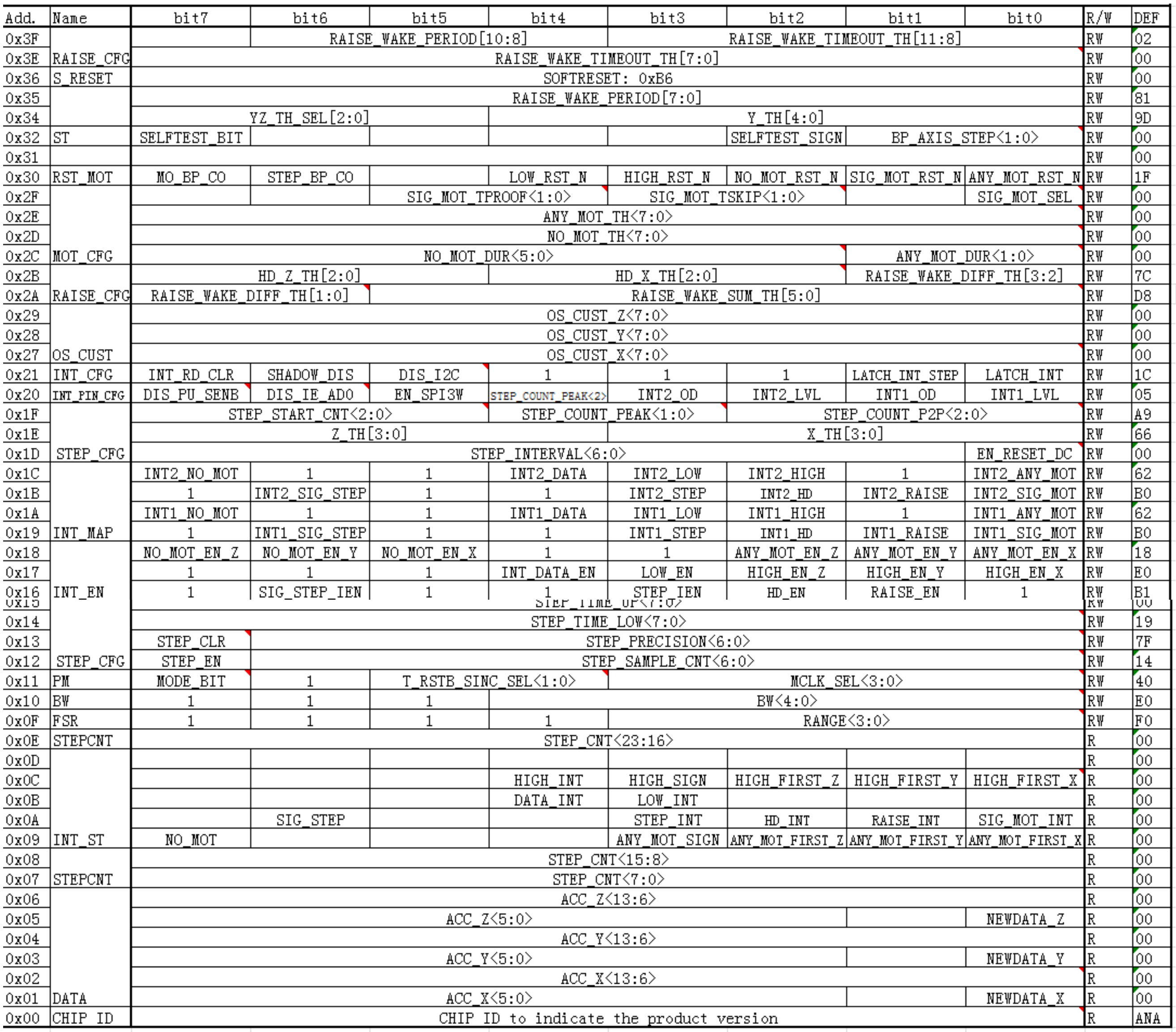
<table>
<tr><td rowspan="2">START</td><td colspan="7">Slave Address</td><td>R
W</td><td rowspan="2">SACK</td><td colspan="8">Register Address
(0x00)</td><td rowspan="2">SACK</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="2">START</td><td colspan="7">Slave Address</td><td>R
W</td><td rowspan="2">SACK</td><td colspan="8">Data
(0x00)</td><td rowspan="2">MACK</td><td colspan="8">Data
(0x01)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="2">MACK</td><td colspan="8">Data
(0x02)</td><td rowspan="2">MACK</td><td colspan="8">………………………………</td><td rowspan="2">MACK</td><td colspan="8">Data
(0x07)</td><td rowspan="2">NACK</td><td rowspan="2">STOP</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td colspan="8">………………………………</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

7 REGISTERS

7.1 Register Map

The table below provides a list of the 8-bit registers embedded in the device and their respective function and addresses

Table 13. Register Map



Add.	Name	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	R/W	DEF
0x3F			RAISE_WAKE_PERIOD[10:8]			RAISE_WAKE_TIMEOUT_TH[11:8]				RW	02
0x3E	RAISE_CFG	RAISE_WAKE_TIMEOUT_TH[7:0]								RW	00
0x36	S_RESET	SOFTRESET: 0xB6								RW	00
0x35		RAISE_WAKE_PERIOD[7:0]								RW	81
0x34		YZ_TH_SEL[2:0]			Y_TH[4:0]					RW	9D
0x32	ST	SELFTEST_BIT					SELFTEST_SIGN	BP_AXIS_STEP<1:0>		RW	00
0x31										RW	00
0x30	RST_MOT	MO_BP_CO	STEP_BP_CO		LOW_RST_N	HIGH_RST_N	NO_MOT_RST_N	SIG_MOT_RST_N	ANY_MOT_RST_N	RW	1F
0x2F				SIG_MOT_TPROOF<1:0>		SIG_MOT_TSKIP<1:0>			SIG_MOT_SEL	RW	00
0x2E		ANY_MOT_TH<7:0>								RW	00
0x2D		NO_MOT_TH<7:0>								RW	00
0x2C	MOT_CFG	NO_MOT_DUR<5:0>						ANY_MOT_DUR<1:0>		RW	00
0x2B		HD_Z_TH[2:0]			HD_X_TH[2:0]			RAISE_WAKE_DIFF_TH[3:2]		RW	7C
0x2A	RAISE_CFG	RAISE_WAKE_DIFF_TH[1:0]		RAISE_WAKE_SUM_TH[5:0]						RW	D8
0x29		OS_CUST_Z<7:0>								RW	00
0x28		OS_CUST_Y<7:0>								RW	00
0x27	OS_CUST	OS_CUST_X<7:0>								RW	00
0x21	INT_CFG	INT_RD_CLR	SHADOW_DIS	DIS_I2C	1	1	1	LATCH_INT_STEP	LATCH_INT	RW	1C
0x20	INT_PIN_CFG	DIS_PU_SENB	DIS_IE_AD0	EN_SPI3W	STEP_COUNT_PEAK<2>	INT2_OD	INT2_LVL	INT1_OD	INT1_LVL	RW	05
0x1F		STEP_START_CNT<2:0>			STEP_COUNT_PEAK<1:0>		STEP_COUNT_P2P<2:0>			RW	A9
0x1E		Z_TH[3:0]				X_TH[3:0]				RW	66
0x1D	STEP_CFG	STEP_INTERVAL<6:0>							EN_RESET_DC	RW	00
0x1C		INT2_NO_MOT	1	1	INT2_DATA	INT2_LOW	INT2_HIGH	1	INT2_ANY_MOT	RW	62
0x1B		1	INT2_SIG_STEP	1	1	INT2_STEP	INT2_HD	INT2_RAISE	INT2_SIG_MOT	RW	B0
0x1A		INT1_NO_MOT	1	1	INT1_DATA	INT1_LOW	INT1_HIGH	1	INT1_ANY_MOT	RW	62
0x19	INT_MAP	1	INT1_SIG_STEP	1	1	INT1_STEP	INT1_HD	INT1_RAISE	INT1_SIG_MOT	RW	B0
0x18		NO_MOT_EN_Z	NO_MOT_EN_Y	NO_MOT_EN_X	1	1	ANY_MOT_EN_Z	ANY_MOT_EN_Y	ANY_MOT_EN_X	RW	18
0x17		1	1	1	INT_DATA_EN	LOW_EN	HIGH_EN_Z	HIGH_EN_Y	HIGH_EN_X	RW	E0
0x16	INT_EN	1	SIG_STEP_IEN	1	1	STEP_IEN	HD_EN	RAISE_EN	1	RW	B1
0x15		STEP_TIME_UP<7:0>								RW	00
0x14		STEP_TIME_LOW<7:0>								RW	19
0x13		STEP_CLR	STEP_PRECISION<6:0>							RW	7F
0x12	STEP_CFG	STEP_EN	STEP_SAMPLE_CNT<6:0>							RW	14
0x11	PM	MODE_BIT	1	T_RSTB_SINC_SEL<1:0>		MCLK_SEL<3:0>				RW	40
0x10	BW	1	1	1	BW<4:0>					RW	E0
0x0F	FSR	1	1	1	1	RANGE<3:0>				RW	F0
0x0E	STEPCNT	STEP_CNT<23:16>								R	00
0x0D										R	00
0x0C					HIGH_INT	HIGH_SIGN	HIGH_FIRST_Z	HIGH_FIRST_Y	HIGH_FIRST_X	R	00
0x0B					DATA_INT	LOW_INT				R	00
0x0A			SIG_STEP			STEP_INT	HD_INT	RAISE_INT	SIG_MOT_INT	R	00
0x09	INT_ST	NO_MOT				ANY_MOT_SIGN	ANY_MOT_FIRST_Z	ANY_MOT_FIRST_Y	ANY_MOT_FIRST_X	R	00
0x08		STEP_CNT<15:8>								R	00
0x07	STEPCNT	STEP_CNT<7:0>								R	00
0x06		ACC_Z<13:6>								R	00
0x05		ACC_Z<5:0>							NEWDATA_Z	R	00
0x04		ACC_Y<13:6>								R	00
0x03		ACC_Y<5:0>							NEWDATA_Y	R	00
0x02		ACC_X<13:6>								R	00
0x01	DATA	ACC_X<5:0>							NEWDATA_X	R	00
0x00	CHIP ID	CHIP ID to indicate the product version								R	ANA

7.2 Register Definition

Register 0x00 (CHIP ID)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
CHIP_ID<7:0>								RW	0xEX

This register is used to identify the device

Register 0x01 ~ 0x02 (DXL, DXM)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
DX<5:0>							NEWDATA_X	R	0x00
DX<13:6>								R	0x00

DX: 14bits acceleration data of x-channel. This data is in two’s complement.

NEWDATA_X: 1, acceleration data of x-channel has been updated since last reading
0, acceleration data of x-channel has not been updated since last reading

Register 0x03 ~ 0x04 (DYL, DYM)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
DY<5:0>							NEWDATA_Y	R	0x00
DY<13:6>								R	0x00

DY: 14bits acceleration data of y-channel. This data is in two’s complement.

NEWDATA_Y: 1, acceleration data of y-channel has been updated since last reading
0, acceleration data of y-channel has not been updated since last reading

Register 0x05 ~ 0x06 (DZL, DZM)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
DZ<5:0>							NEWDATA_Z	R	0x00
DZ<13:6>								R	0x00

DZ: 14bits acceleration data of z-channel. This data is in two’s complement.

NEWDATA_Z: 1, acceleration data of z-channel has been updated since last reading
0, acceleration data of z-channel has not been updated since last reading

Register 0x07 ~ 0x08 (STEP_CNT)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_CNT<7:0>								R	0x00
STEP_CNT<15:8>								R	0x00

STEP_CNT<15:0> 16 bits of step counter, out of total 24bits data. The MSB data are in 0x0e

Register 0x09 (INT_ST0)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
NO_MOT	SIG_STEP			ANY_MOT_SIGN	ANY_MOT_FIRST_Z	ANY_MOT_FIRST_Y	ANY_MOT_FIRST_X	R	0x00

NO_MOT:

1, no_motion interrupt active
0, no_motion interrupt inactive

ANY_MOT_SIGN: 1, sign of any_motion triggering signal is negative
0, sign of any_motion triggering signal is positive

ANY_MOT_FIRST_Z:

1, any_motion interrupt is triggered by Z axis
0, any_motion interrupt is not triggered by Z axis

ANY_MOT_FIRST_Y:
1, any_motion interrupt is triggered by Y axis
0, any_motion interrupt is not triggered by Y axis

ANY_MOT_FIRST_X:
1, any_motion interrupt is triggered by X axis
0, any_motion interrupt is not triggered by X axis

Register 0x0a (INT_ST1)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
	SIG_STEP			STEP_INT	HD_INT	RAISE_INT	SIG_MOT_INT	R	0x00

SIG_STEP: 1, significant step is active
0, significant step is inactive
STEP_INT: 1, step valid interrupt is active
0, step quit interrupt is inactive
HD_INT: 1, hand down interrupt is active
0, hand down interrupt is inactive
RAISE_INT: 1, raise hand interrupt is active
0, raise hand interrupt is inactive
SIG_MOT_INT: 1, significant interrupt is active
0, significant interrupt is inactive

Register 0x0b (INT_ST2)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
			DATA_INT					R	0x00

DATA_INT: 1, data ready interrupt active
0, data ready interrupt inactive

Register 0x0e (STEP_CNT)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_CNT<23:16>								R	0x00

STEP_CNT<23:16>: 8bit MSB data of step counter, out of total 24bits data. The LSB data are in 0x07 and 0x08

Register 0x0f (FSR)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
1	1	1		RANGE<3:0>				RW	0xF0

RANGE<3:0>: set the full scale of the accelerometer. Setting as following

RANGE<3:0>	Acceleration range	Resolution
0001	2g	244ug/LSB
0010	4g	488g/LSB
0100	8g	977ug/LSB
1000	16g	1.95mg/LSB
1111	32g	3.91mg/LSB
Others	2g	244ug/LSB

Register 0x10 (BW)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
1	1	1	BW<4:0>					RW	0xE0

BW<4:0>: bandwidth setting, as following

BW<4:0>	ODR
xx000	MCLK/7695
xx001	MCLK/3855
xx010	MCLK/1935
xx011	MCLK/975
xx100	
xx101	MCLK/15375
xx110	MCLK/30735
xx111	MCLK/61455
Others	MCLK/7695

Register 0x11 (PM)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
MODE_BIT	1	T_RSTB_SINC_SEL<1:0>		MCLK_SEL<3:0>				RW	0x40

MODE_BIT: 1, set device into active mode
0, set device into standby mode

T_RSTB_SINC_SEL<1:0>: Reset clock setting. The preset time is reserved for CIC filter in digital
11, T_RSTB_SINC=8*MCLK
10, T_RSTB_SINC=6*MCLK
01, T_RSTB_SINC=4*MCLK
00, T_RSTB_SINC=3*MCLK

MCLK_SEL<3:0>: set the master clock to digital

MCLK_SEL<3:0>	Freq of MCLK
0000	500KHz
0001	333KHz
0010	200KHz
0011	100KHz
0100	50KHz
0101	25KHz
0110	12.5KHz
0111	5KHz
1xxx	Reserved

Register 0x12 (STEP_CONF0)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_EN	STEP_SAMPLE_CNT<6:0>							RW	0x14

STEP_EN: enable step counter, this bit should be set when using step counter

STEP_SAMPLE_CNT:

sample count setting for dynamic threshold calculation. The actual value is STEP_SAMPLE_CNT<6:0>*8, default is 0xC, 96 sample count

Register 0x13 (STEP_CONF1)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_CLR	STEP_PRECISION<6:0>							RW	0x7F

STEP_CLR: clear step count in register 0x07 ,0x08 and 0x0E

STEP_PRECISION<6:0>:

threshold for acceleration change of two successive sample which is used to update sample_new register in step counter, the actual g value is STEP_PRECISION<6:0>*LSB*16 when STEP_PRECISION<6:0>!=0000000 & !=1111111

When STEP_PRECISION<6:0>=0000000, always use P2P/8
When STEP_PRECISION<6:0>=1111111, always use P2P/16
When STEP_PRECISION<6:0>=?, always use P2P/4

Register 0x14 (STEP_CONF2)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_TIME_LOW<7:0>								RW	0x19

STEP_TIME_LOW<7:0>: the short time window for a valid step, the actual time is STEP_TIME_LOW<7:0>*(1/ODR)

Register 0x15 (STEP_CONF3)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_TIME_UP<7:0>								RW	0x00

STEP_TIME_UP<7:0>: time window for quitting step counter, the actual time is
STEP_TIME_UP<7:0>*8*(1/ODR)

Register 0x16 (INT_EN0)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
1	SIG_STEP_IEN	1	1	STEP_IEN	HD_EN	RAISE_EN	1	RW	0xB1

SIG_STEP_IEN: 1, enable significant step interrupt
0, disable significant step interrupt
STEP_IEN: 1, enable step valid interrupt
0, disable step valid interrupt
HD_EN: 1, enable hand-down interrupt
0, disable hand-down interrupt
RAISE_EN: 1, enable raise-hand interrupt
0, disable raise-hand interrupt

Register 0x17 (INT_EN1)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
1	1	1	INT_DATA_EN					RW	0xE0

INT_DATA_EN: 1, enable data ready interrupt
0, disable data ready interrupt

Register 0x18 (INT_EN2)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
NO_MOT_EN_Z	NO_MOT_EN_Y	NO_MOT_EN_X	1	1	ANY_MOT_EN_Z	ANY_MOT_EN_Y	ANY_MOT_EN_X	RW	0x18

NO_MOT_EN_Z: 1, enable no_motion interrupt on Z axis
0, disable no_motion interrupt on Z axis
NO_MOT_EN_Y: 1, enable no_motion interrupt on Y axis
0, disable no_motion interrupt on Y axis
NO_MOT_EN_X: 1, enable no_motion interrupt on X axis
0, disable no_motion interrupt on X axis
ANY_MOT_EN_Z: 1, enable any_motion interrupt on Z axis
0, disable any_motion interrupt on Z axis
ANY_MOT_EN_Y: 1, enable any_motion interrupt on Y axis
0, disable any_motion interrupt on Y axis
ANY_MOT_EN_X: 1, enable any_motion interrupt on X axis
0, disable any_motion interrupt on X axis

Register 0x19 (INT_MAP0)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
1	INT1_SIG_STEP	1	1	INT1_STEP	INT1_HD	INT1_RAISE	INT1_SIG_MOT	RW	0xB0

INT1_SIG_STEP:

1, map significant step interrupt to INT1 pin
0, not map significant step interrupt to INT1 pin

INT1_STEP:

1, map step valid interrupt to INT1 pin
0, not map step valid interrupt to INT1 pin

INT1_HD:

1, map hand down interrupt to INT1 pin
0, not map hand down interrupt to INT1 pin

INT1_RAISE:

1, map raise hand interrupt to INT1 pin
0, not map raise hand interrupt to INT1 pin

INT1_SIG_MOT:

1, map significant interrupt to INT1 pin
0, not map significant interrupt to INT1 pin

Register 0x1a (INT_MAP1)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
INT1_NO_MOT	1	1	INT1_DATA			1	INT1_ANY_MOT	RW	0x62

INT1_NO_MOT: 1, map no_motion interrupt to INT1 pin
0, not map no_motion interrupt to INT1 pin

INT1_DATA: 1, map data ready interrupt to INT1 pin
0, not map data ready interrupt to INT1 pin

INT1_ANY_MOT: 1, map any motion interrupt to INT1 pin
0, not map any motion interrupt to INT1 pin

Register 0x1b (INT_MAP2)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
1	INT2_SIG_STEP	1	1	INT2_STEP	INT2_HD	INT2_RAISE	INT2_SIG_MOT	RW	0xB0

INT2_SIG_STEP: 1, map significant step interrupt to INT2 pin
0, not map significant step interrupt to INT2 pin

INT2_STEP: 1, map step valid interrupt to INT2 pin
0, not map step valid interrupt to INT2 pin

INT2_HD: 1, map hand down interrupt to INT2 pin
0, not map hand down interrupt to INT2 pin

INT2_RAISE: 1, map raise hand interrupt to INT2 pin
0, not map raise hand interrupt to INT2 pin

INT2_SIG_MOT: 1, map significant interrupt to INT2 pin
0, not map significant interrupt to INT2 pin

Register 0x1c (INT_MAP3)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
INT2_NO_MOT	1	1	INT2_DATA			1	INT2_ANY_MOT	RW	0x62

INT2_NO_MOT: 1, map no motion interrupt to INT2 pin
0, not map no motion interrupt to INT2 pin
INT2_DATA: 1, map register data ready interrupt to INT2 pin
0, not map register data ready interrupt to INT2 pin
INT2_ANY_MOT: 1, map any motion interrupt to INT2 pin
0, not map any motion interrupt to INT2 pin

Register 0x1d (SIG_STEP_TH)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_INTERVAL<7:0>								RW	0x00

STEP_INTERVAL <7:0>:threshold of significant step. When MOD(STEP_CNT, STEP_INTERVAL)=0, SIG_STEP_INT will be generated.

Register 0x1e (raise hand: X_TH Z_TH)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
Z_TH<3:0>				X_TH<3:0>				RW	0x66

X_TH<3:0>: 0~7.5, LSB 0.5 (unit: m/s2)
Z_TH<3:0>: -8~7, LSB 1 (unit: m/s2)

Register 0x1f

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
STEP_START_CNT<2:0>			STEP_COUNT_PEAK<1:0>		STEP_COUNT_P2P<2:0>			RW	0xA9

STEP_START_CNT<2:0>: th_step_pattern = 0/4/8/12/16/24/32/40
STEP_COUNT_PEAK<2:0>: FIXED_PEAK = 0.05g + 0.05g * STEP_COUNT_PEAK<2:0>.
This FIXED_PEAK is used in algorithm of STEP COUNTER.
STEP_COUNT_PEAK<2> is in register 0x20<4> and
STEP_COUNT_PEAK[2:0]= {0x20[4], 0x1F[4:3]}
STEP_COUNT_P2P<2:0>: FIXED_P2P = 0.3g + 0.1g * STEP_COUNT_P2P<2:0>.
STEP_COUNT_P2P[3:0]= {0x1F[2:0]}

Register 0x20 (INTPIN_CONF)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
DIS_PU_SENB	DIS_IE_AD0	EN_SPI3W	STEP_COUNT_PEAK<2>	INT2_OD	INT2_LVL	INT1_OD	INT1_LVL	RW	0x05

DIS_PU_SENB: 1, disable pull-up resistor of PIN_SENB
0, enable pull-up resistor of PIN_SENB
DIS_IE_AD0: 1, disable input of AD0
0, not disable input of AD0
EN_SPI3W: 1, enable 3W SPI
0, 4W SPI
STEP_COUNT_PEAK<2>: Definition in 0x1F<4:3>
INT2_OD: 1, open-drain for INT2 pin
0, push-pull for INT2 pin
INT2_LVL: 1, logic high as active level for INT2 pin
0, logic low as active level for INT2 pin
INT1_OD: 1, open-drain for INT1 pin
0, push-pull for INT1 pin
INT1_LVL: 1, logic high as active level for INT1 pin
0, logic low as active level for INT1 pin

Register 0x21 (INT_CFG)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
INT_RD_ CLR	SHADOW _DIS	DIS_I2C	1	1	1	LATCH_INT _STEP	LATCH_ INT	RW	0x1C

INT_RD_CLR: 1, clear all the interrupts in latched-mode, when any read operation to any of registers from 0x09 to 0x0D
0, clear the related interrupts, only when read the register INT_ST (0x09 to 0x0D), no matter the interrupts in latched-mode, or in non-latched-mode.
Reading 0x09 will clear the register 0x09 only and the others keep the status
SHADOW_DIS: 1, disable the shadowing function for the acceleration data
0, enable the shadowing function for the acceleration data.
When shadowing is enabled, the MSB of the acceleration data is locked, when corresponding LSB of the data is reading.
This can ensure the integrity of the acceleration data during the reading.
The MSB will be unlocked when the MSB is read.
DIS_I2C: 1: disable I2C. Setting this bit to 1 in SPI mode is recommended
0: enable I2C
LATCH_INT_STEP: 1, step related interrupt is in latch mode
0, step related interrupt is in non-latch mode
LATCH_INT: 1, interrupt is in latch mode
0, interrupt is in non-latch mode

Register 0x27 (OS_CUST_X)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
OS_CUST_X<7:0>								RW	0x00

OS_CUST_X<7:0>: offset calibration of X axis for user, the LSB depends on full-scale of the device which is 3.9mg in 2g range,
7.8mg in 4g range, 15.6mg in 8g range, 31.2mg in 16g, and 62.5mg in 32g

Register 0x28 (OS_CUST_Y)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
OS_CUST_Y<7:0>								RW	0x00

OS_CUST_Y<7:0>: offset calibration of Y axis for user, the LSB depends on full-scale of the device which is 3.9mg in 2g range,
7.8mg in 4g range, 15.6mg in 8g range, 31.2mg in 16g, and 62.5mg in 32g

Register 0x29 (OS_CUST_Z)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
OS_CUST_Z<7:0>								RW	0x00

OS_CUST_Z<7:0>: offset calibration of Z axis for user, the LSB depends on full-scale of the device which is 3.9mg in 2g range,
7.8mg in 4g range, 15.6mg in 8g range, 31.2mg in 16g, and 62.5mg in 32g

Register 0x2a (RAISE_WAKE_SUM_TH RAISE_WAKE_DIFF_TH)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
RAISE_WAKE_DIFF_TH<1:0>		RAISE_WAKE_SUM_TH<5:0>						RW	0xD8

RAISE_WAKE_SUM_TH <5:0>: 0 ~ 31.5 (LSB 0.5 m/s2)

RAISE_WAKE_DIFF_TH<3:0>	UNIT (m/s^2)
0	0.2
1	0.3
2	0.4
3	0.5
4	0.6
5	0.7
6	0.8
7	0.9
8	1.0
9	1.1
10	1.2
default	0.2

Register 0x2b (RAISE_WAKE_DIFF_TH HD_X_TH HD_Z_TH)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
HD_Z_TH<2:0>			HD_X_TH<2:0>			RAISE_WAKE_DIFF_TH<3:2>		RW	0x7C

HD_X_TH<2:0> : hand down x threshold, 0~7 (m/s2)
HD_Z_TH<2:0> : hand down z threshold, 0~7 (m/s2)

Register 0x2c (MOT_CONF0)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
NO_MOT_DUR<5:0>						ANY_MOT_DUR<1:0>		RW	0x00

NO_MOT_DUR<5:0>: no motion interrupt will be triggered when slope < NO_MOT_TH for the times which defined by NO_MOT_DUR<5:0>

Duration = (NO_MOT_DUR<3:0> + 1) * 1s, if NO_MOT_DUR<5:4> =b00
Duration = (NO_MOT_DUR<3:0> + 4) * 5s, if NO_MOT_DUR<5:4> =b01
Duration = (NO_MOT_DUR<3:0> + 10) * 10s, if NO_MOT_DUR<5:4> =b1x

ANY_MOT_DUR<1:0>: any motion interrupt will be triggered when slope > ANY_MOT_TH for (ANY_MOT_DUR<1:0> + 1) samples

Register 0x2d (MOT_CONF1)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
NO_MOT_TH<7:0>								RW	0x00

NO_MOT_TH<7:0>: Threshold of no-motion interrupt. The threshold definition is as following
TH= NO_MOT_TH<7:0> * 16 * LSB

Register 0x2e (MOT_CONF2)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
ANY_MOT_TH<7:0>								RW	0x00

ANY_MOT_TH<7:0>: Threshold of any motion interrupt. The threshold definition is as following
TH= ANY_MOT_TH<7:0> * 16 * LSB

Register 0x2f (MOT_CONF3)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
		SIG_MOT_TPROOF<1:0>		SIG_MOT_TSKIP<1:0>			SIG_MOT_SEL	RW	0x00

SIG_MOT_TPROOF<1:0>: 00, T_PROOF=0.25s
01, T_PROOF=0.5s
10, T_PROOF=1s
11, T_PROOF=2s

SIG_MOT_TSKIP<1:0>: 00, T_SKIP=1.5s
01, T_SKIP=3s
10, T_SKIP=6s
11, T_SKIP=12s

SIG_MOT_SEL: 1, select significant motion interrupt
0, select any motion interrupt

Register 0x30

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
MO_BP_CO	STEP_BP_CO				NO_MOT_RST_N	SIG_MOT_RST_N	ANY_MOT_RST_N	RW	0x1F

MO_BP_CO: 1, motion detector will use data without OS_CUST
0, motion detector will use data with OS_CUST

STEP_BP_CO: 1, pedometer will use data without OS_CUST
0, pedometer will use data with OS_CUST

NO_MOT_RST_N : 0, Reset no motion detector. After reset, user should write 1 back.

SIG_MOT_RST_N: 0, Reset significant motion detector. After reset, user should write 1 back.

ANY_MOT_RST_N: 0, Reset any motion detector. After reset, user should write 1 back.

Register 0x32 (ST)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
SELFTEST_BIT					SELFTEST_SIGN	BP_AXIS_STEP<1:0>		RW	0x00

SELFTEST_BIT: 1, self-test enabled. When self-test enabled, a delay of 3ms is necessary for the value settling.
0, normal

SELFTEST_SIGN: 1, set self-test excitation positive
0, set self-test excitation negative

BP_AXIS_STEP<1:0>: 11, bypass Z axis, use only X and Y axes data for step counter algorithm
10, bypass Y axis, use only X and Z axes data for step counter algorithm
01, bypass X axis, use only Y and Z axes data for step counter algorithm
00, use all of 3 axes data for step counter algorithm

Register 0x34 (Y_TH YZ_TH_SEL)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
YZ_TH_SEL<2:0>			Y_TH<4:0>					RW	0x9D

Y_TH: -16 ~ 15 (m/s2)

YZ_TH_SEL<2:0>	UNIT (m/s2)
0	7.0
1	7.5
2	8.0
3	8.5
4	9.0

5	9.5
6	10.0
7	10.5

Register 0x35 (RAISE_WAKE_PERIOD)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
RAISE_WAKE_PERIOD<7:0>								RW	0x81

RAISE_WAKE_PERIOD<10:0> * ODR period = wake count (EX. ODR = 1ms, 0X35 = 100 ➔ wake count = 0.1 sec)

Register 0x36 (SR)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
SOFT_RESET								RW	0x00

SOFT_RESET: 0xB6, soft reset all of the registers. After soft-reset, user should write 0x00 back

Register 0x3e (RAISE_WAKE_TIMEOUT_TH)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
RAISE_WAKE_TIMEOUT_TH<7:0>								RW	0x00

RAISE_WAKE_TIMEOUT_TH<11:0> * ODR period = timeout count (EX. ODR = 1ms, 0X3e = 100 ➔ timeout count = 0.1 sec)

Register 0x3f (RAISE_WAKE_TIMEOUT_TH RAISE_WAKE_PERIOD RAISE_WAKE_EN)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Default
	RAISE_WAKE_PERIOD<10:8>			RAISE_WAKE_TIMEOUT_TH<11:8>				RW	0x02

MCU Introduction

QMS7926 is a System on Chip (SoC) for Bluetooth® low energy applications. QMS7926 has 32-bit ARM® Cortex™-M0 CPU with 138KSRAM/Retention SRAM and an ultra-low power, high performance, multi-mode radio. QMS7926 can support BLE with security, application and over-the-air download update. Serial peripheral IO and integrated application IP enables customer product to be built with minimum bill-of-material (BOM) cost.

1 Product Overview

1.1 Block Diagram

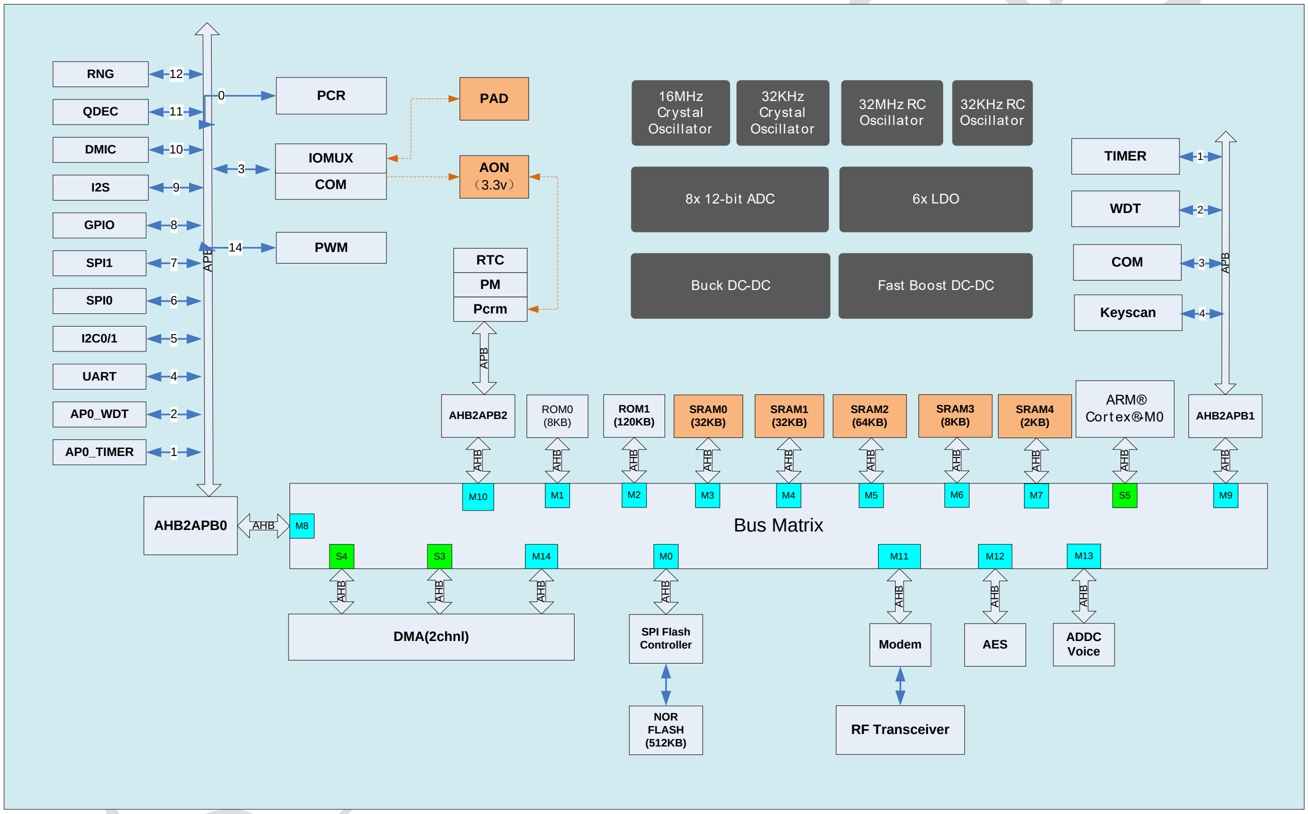


Figure 1: QMS7926 block diagram

2 System Blocks

The system block diagram of QMS7926 is shown in **Figure 1**.

2.1 CPU

The QMS7926 has an ARM Cortex-M0 CPU. The CPU, memories, and all peripherals are connected by AMBA bus fabrics.

The ARM® Cortex™-M0 CPU has a 16-bit instruction set with 32-bit extensions (Thumb-2® technology) that delivers high-density code with a small-memory-footprint. By using a single-cycle 32-bit multiplier, a 3-stage pipeline and a Nested Vector Interrupt Controller (NVIC), the ARM Cortex™-M0 CPU makes program execution simple and highly efficient.

The CPU will play controller role in BLE modem and run all user applications. The following main features are listed below.

- Up to 48Mhz ARM Cortex™-M0 processor core.
    - Low gate count and high energy efficient.
    - ARMv6M architecture, Thumb ISA but no ARM ISA.
    - No cache and no TCM.
    - Up to 32 interrupts embedded NVIC.
    - SysTick timer.
    - Sleep/deep sleep mode.
    - Support low power WFI and WFE
- 4 32-bit general purpose timers and 1 watchdog timer (WDT).
- 120KB ROM for boot and protocol stack.
- 138KB retention SRAM for program and data.
- AHB to APB Bridge for peripherals and registers.
- Clock and reset controller.
- AHB debug access port interface and DAP ROM.
- APB interface to/from BLE modem.
- Dynamic and static clock gating to save power.
- No TRACE.

Some of these features are shared with the AP subsystem.

2.2 Memory

QMS7926 has total 128KB ROM, 138KB SRAM and up to 512KB FLASH. The physical address space of these memories is shown in **Figure2**.

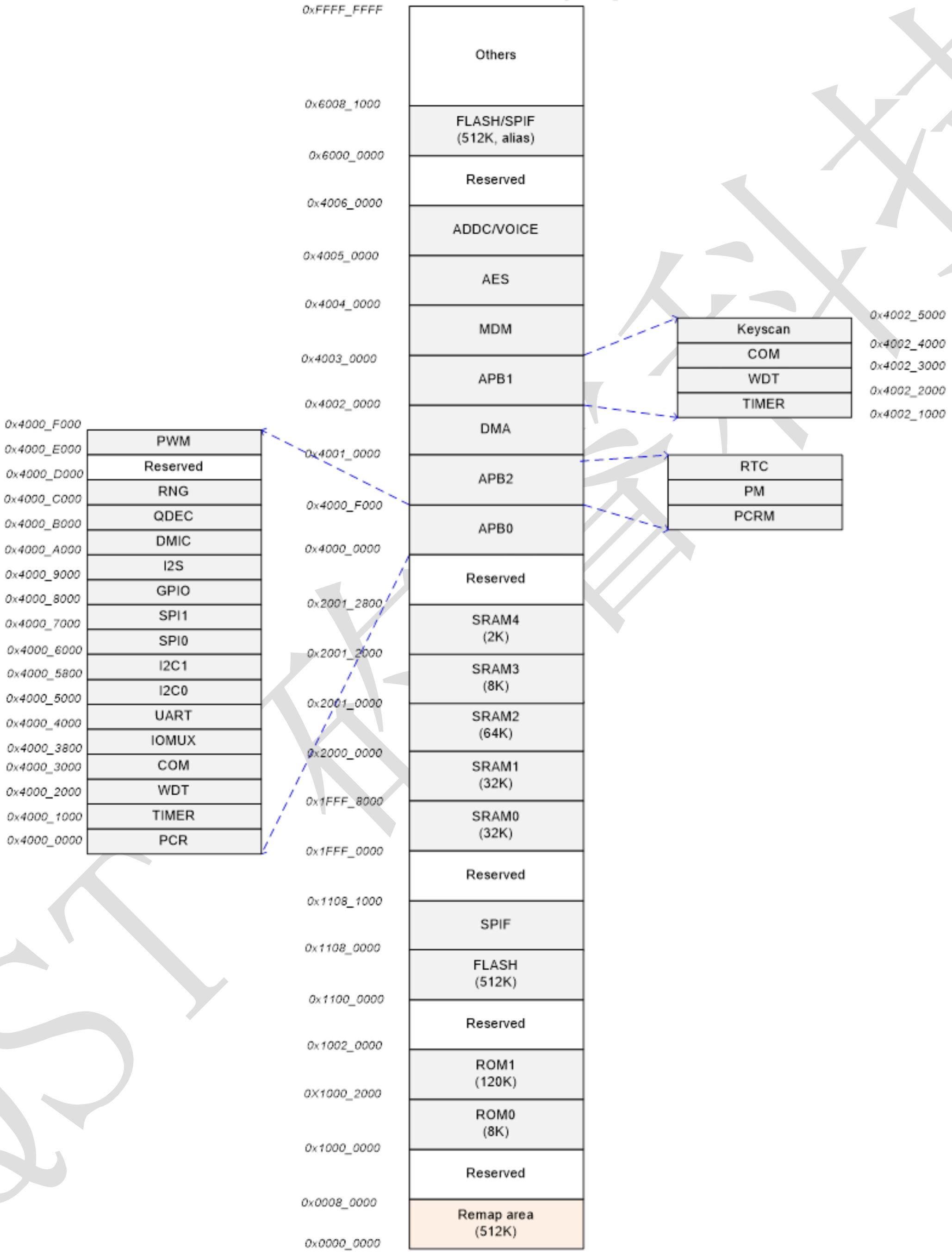


Figure 2: QMS7926 memory space

2.2.1 ROM

QMS7926 has 2 ROMs.

	SIZE	CONTENT
ROM0	8KB	Reserved
ROM1	120KB	Boot ROM for M0. Protocol stack. Common peripheral drivers.

Table 1: List of ROMs

2.2.1 SRAM

QMS7926 has 5 SRAM blocks. All 5 SRAM blocks have retention capability. which can be configured individually. All SRAM blocks can be used to store program or data.

	SIZE	CONTENT
SRAM0	32KB	
SRAM1	32KB	
SRAM2	64KB	
SRAM3	8KB	
SRAM4	2KB	

Table 2: List of SRAMs

2.2.3 FLASH

QMS7926 has FLASH to provide non-volatile program and data storage. The size of the FLASH can be 256KB or 512KB. QMS7926 supports 2-wire reading.

2.2.4 Memory Address Mapping

Name	Size (KB)	Master	Physical Address	CM4 Alias	M0 Remap		
					0	1	2
ROM0	8	M0	1000_0000~1000_1FFF	0x0			
ROM1	120	M0	1000_2000~1001_FFFF		0x0		
RAM0	32	M0	1FFF_0000~1FFF_7FFF				
RAM1	32	M0	1FFF_8000~1FFF_FFFF				
RAM2	64	M0	2000_0000~2000_FFFF			0x0	
RAM3	8	M0	2001_0000~2001_1FFF				
RAM4	2	M0	2001_2000~2001_27FF				
FLASH	512	M0	1100_0000~1107_FFFF				0x0
			6000_0000~6007_FFFF				

Table 3: Memory address mapping

2.3 Boot and Execution Modes

During the boot, the ROM1 is aliased to 0x0 address. The M0 starts to execute the program from the ROM1.

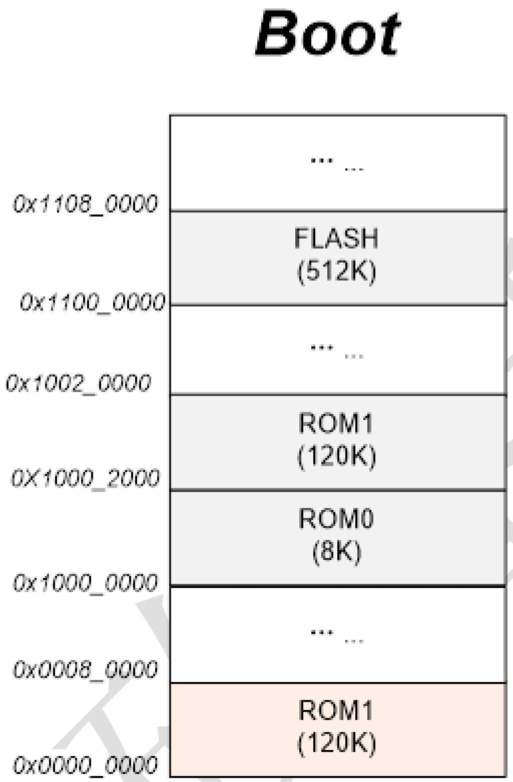


Figure 3: QMS7926 boot mode

2.3.1 Mirror Mode

The mirror mode is not tied to the chip variations. Any chip variation can use mirror mode to execute program. In the mirror mode, the program is copied from the FLASH to the SRAM, then is executed in the SRAM. For the M0 processor, one of the SRAM blocks must be aliased to 0x0 address.

2.3.2 FLASH Mode

The FLASH mode is not tied to the chip variations. Any chip variation can use FLASH mode to execute program. In the FLASH mode, the program is executed in the FLASH. For the M0 processor, the FLASH must be aliased to 0x0 address.

2.3.3 Boot loader

The boot loader in the ROM has the basic structure as shown below. The content in the FLASH should be specifically defined to allow boot loader to identify whether the FLASH content is valid, as shown in the example below. If the FLASH is valid, the ROM boot loader will put the chip in the normal mode and start normal program execution. If the FLASH is not valid, the boot loader will enter FLASH programming mode.

Address	Variable	Content
0	PRODUCT_MODE	Identify the chip mode
4	CODE_BASE	The base address of the code
8	CODE_LEN	The length of the code
C	BOOT_MODE	Identify mirror or FLASH mode

Table 4: Flash content example

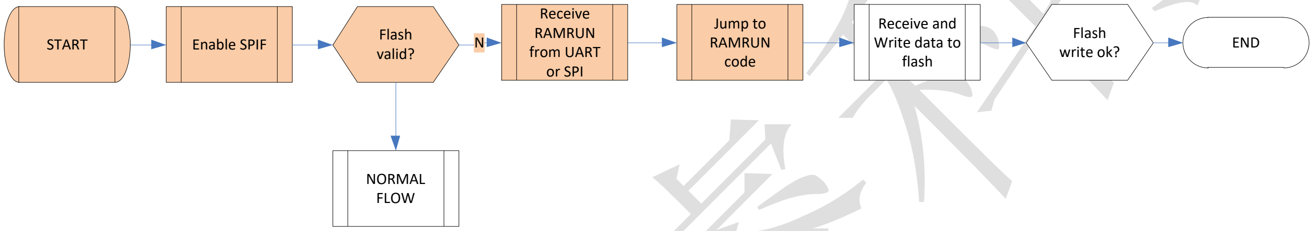


Figure 4: Bootloader flow

2.4 Power, Clock and Reset (PCR)

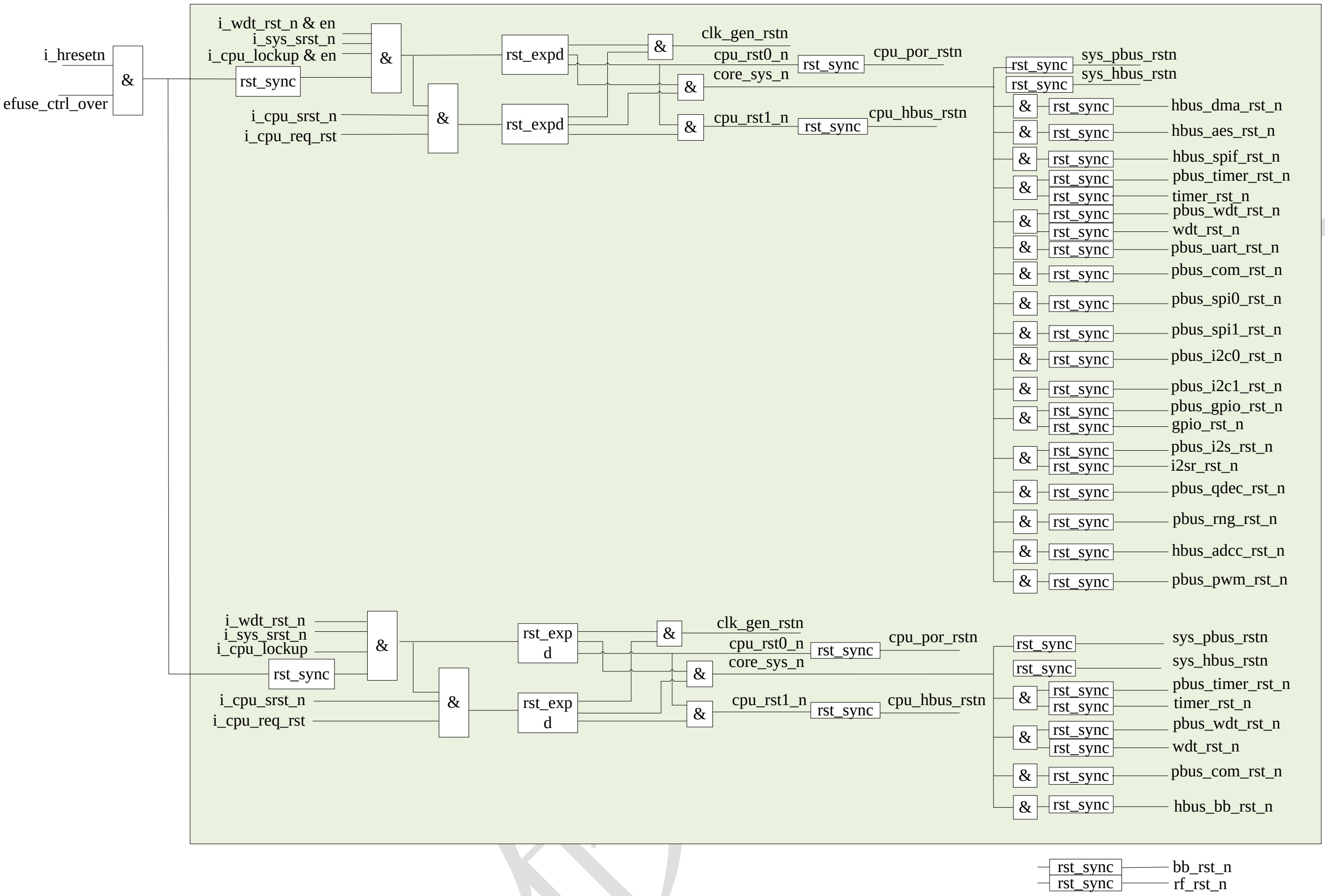


Figure 5: QMS7926 power, clock and reset

2.5 Power Management (POWER)

The power management system is highly flexible with functional blocks such as the CPU, radio transceiver, and peripherals saving separate power state control in addition to the System Sleep mode and OFF modes. When in System Normal mode, all functional blocks will independently be turned on depending on needed application functionality.

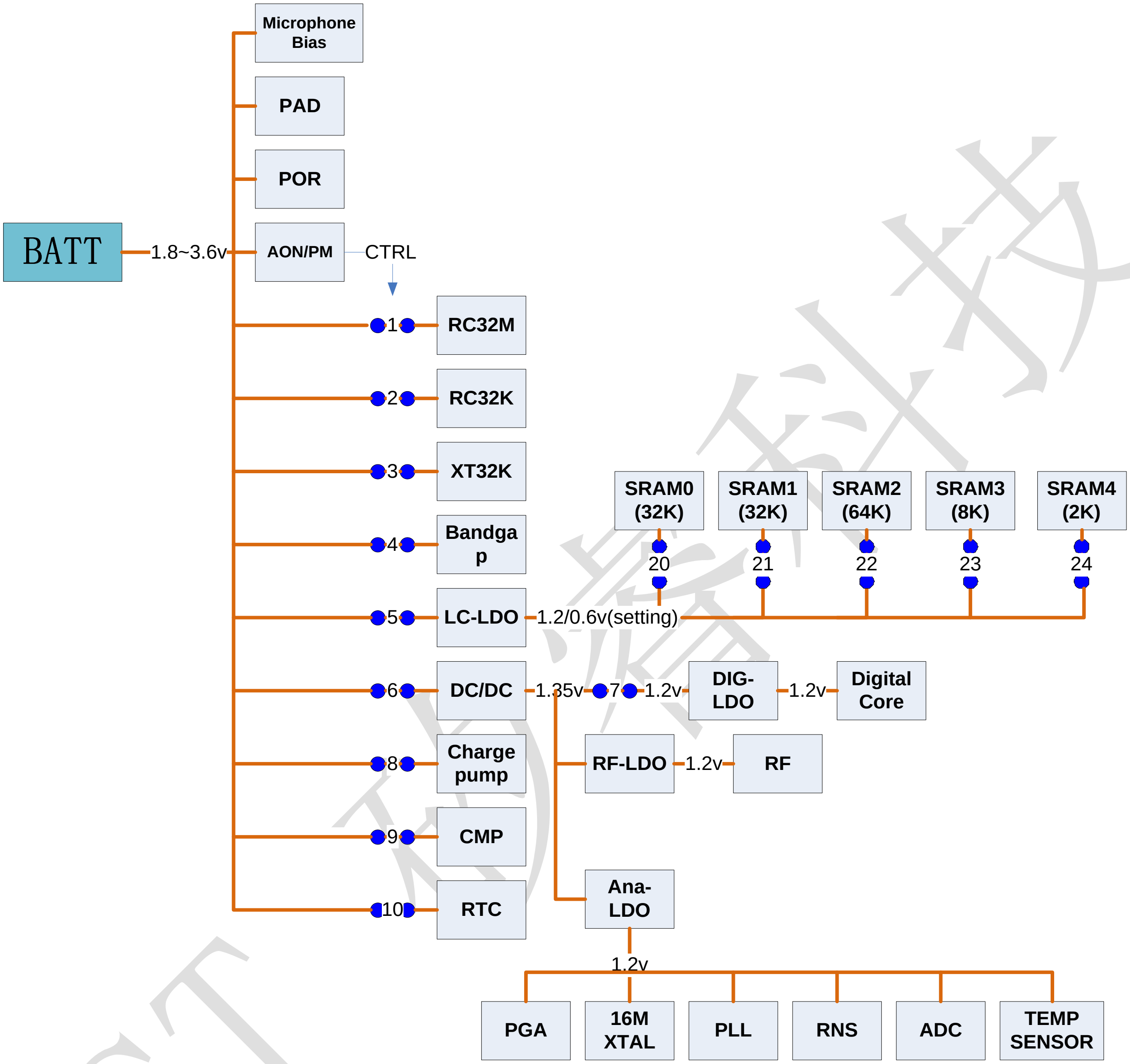


Figure 6: Power system

The following diagram is Normal, Sleep and Off mode. Switches are optional depending on user's request.

Switch	Normal	Sleep	Off
1RC32M	On	Off	Off
2RC32K	On	Optional	Off
3XT32K	On	Optional	Off
4bandgap	On	Off	Off

5LC-LDO	On	on	Off
6DC/DC	On	Off	Off
7DIG-LDO	On	Off	Off
8charge pump	On	Off	Off
9CMP	On	Optional	Off
10RTC	On	Optional	Off
20SRAM-32K	1.2v	0.6v	0
21SRAM-32K	1.2v	0.6v	0
22SRAM-64K	1.2v	0.6v	0
23SRAM-8K	1.2v	0.6v	0
24SRAM-2K	1.2v	0.6v	0

Table 5: Flash Switches of different power modes

2.6 Low Power Features

2.6.1 Operation and Sleep States

2.6.1.1 Normal State

2.6.1.2 Clock Gate State

The CPU executes WFI/WFE to enter clock gate state. After wake-up from clock-gate state, the CPU continues to execute the program from where it stopped. The wake-up sources includes interrupts and events. The wake-up sources are configured by the software according to applications.

2.6.1.3 System Sleep State

The wake-up sources include:

- IO
- RTC
- RESET
- UVLO reset

2.6.1.4 System Off State

The wake-up sources include：

- IOs
- RESET
- UVLO reset

2.6.2 State Transition

2.6.2.1 Entering Clock Gate State and Wake-up

CPU executes WFI/WFE.

2.6.2.2 Entering Sleep/off States and Wake-up

The PM registers identify whether the CPU is in mirror mode or FLASH mode before sleep or off, and record the remap and vectors. The CPU configures the corresponding PM registers to put the chip into sleep or off mode. After wake-up, the chip enters boot mode to execute boot code in the ROM. The ROM code checks the mode before sleep/off and the remap information, perform corresponding configurations, and starts to execute the program.

2.7 Interrupts

Interrupt Name	M0 Interrupt Number
Reserved	0
Reserved	1
cp_timer_irq	2
cp_wdt_irq	3
bb_irq	4
kscan_irq	5
rtc_irq	6
Reserved	7
Reserved	8
timer_irq	9
wdt_irq	10
uart_irq	11
i2c0_irq	12
i2c1_irq	13
spi0_irq	14
spi1_irq	15
gpio_irq	16
i2s_irq	17
spif_irq	18
dmac_intr	19
dmac_inttc	20
dmac_interr	21
fpidc	22
fpdzc	23
fpioc	24
fpufc	25
fpofc	26
fpixc	27
aes_irq	28
adcc_irq	29
qdec_irq	30
rng_irq	31

Table 6: Interrupts

2.8 Clock Management (CLOCK)

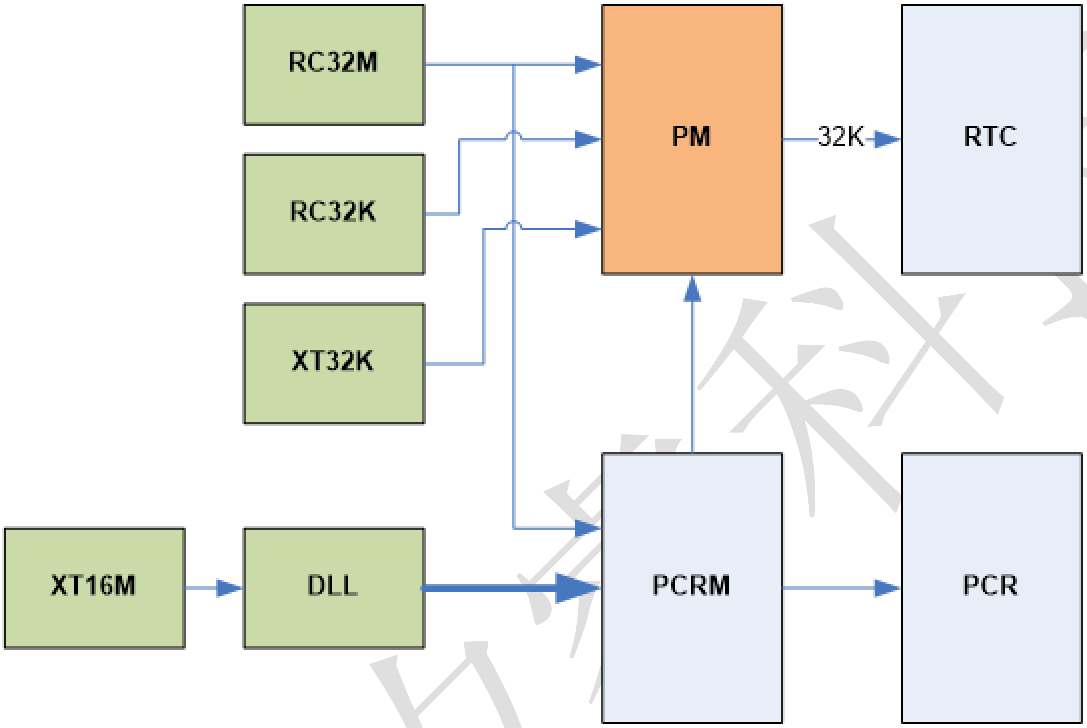


Figure 7: Clock management

There are two crystal clock sources: 16MHz crystal oscillator (XT16M) and 32.768kHz crystal oscillator (XT32k), of which the 32.768k crystal oscillator is optional. There are also two on chip RC oscillators: 32MHz RC oscillator (RC32M) and 32kHz RC oscillator (RC32k), both of which can be calibrated with respect to 16MHz crystal oscillator. If 32.768kHz crystal is not installed, RC32k oscillator would be periodically calibrated and used for RTC. At initial power up or wake up before XT16M oscillator starts up, RC32M is used as the main clock. An on-chip DLL generates higher frequency clocks such as 32/48/64/96MHz from the XT16M clock source.

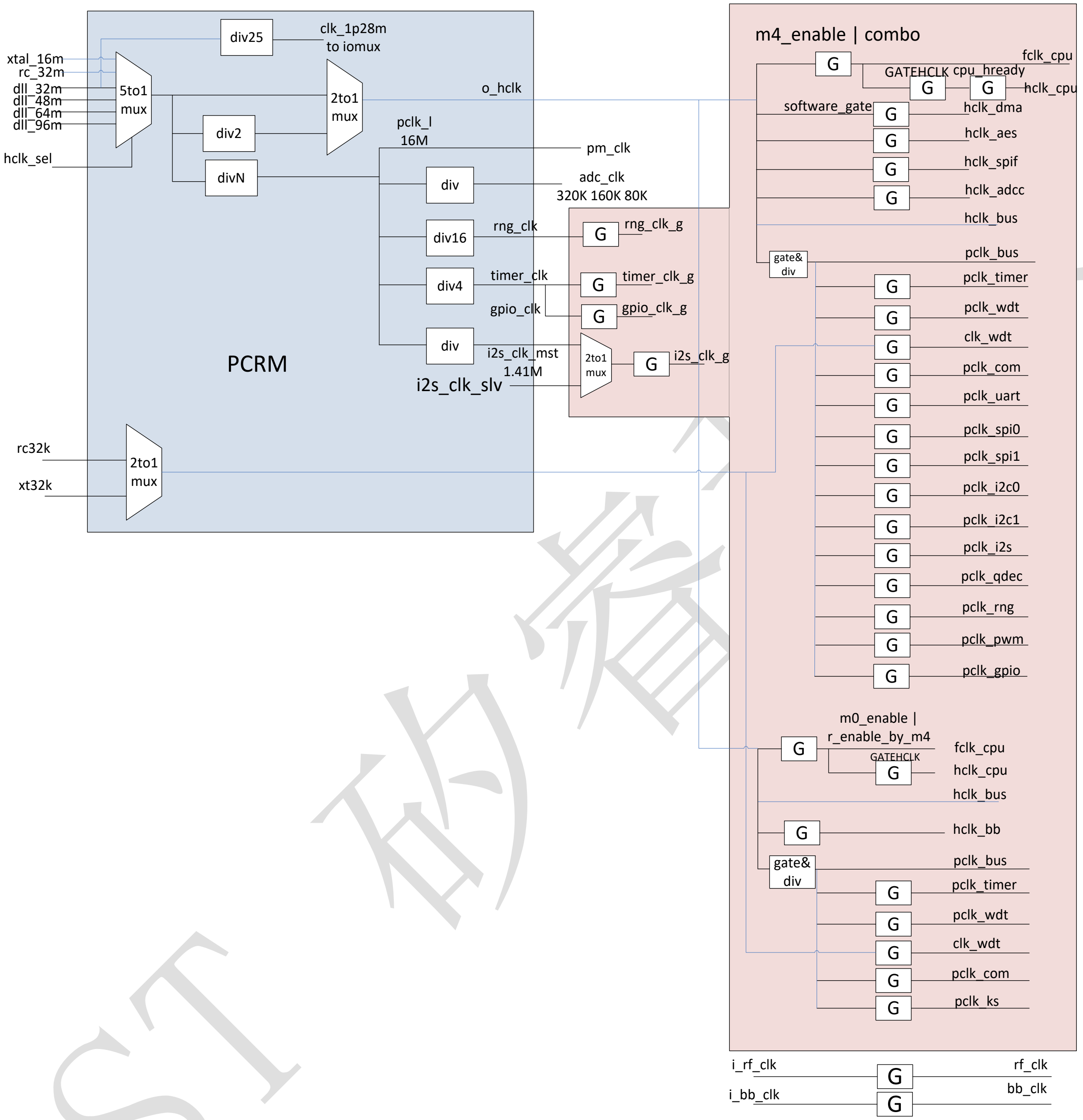


Figure 8: Clock structure diagram

2.9 IOMUX

The IOMUX provides a flexible I/O configuration, as the ports of most of the peripherals can be configured and mapped to any of the physical I/O pads (I/O at die boundary). These peripheral modules include I2C 0-1, I2S, UART, PWM 0-5, SPI 0-1, Quadrature Decoder etc. However for other specific purpose peripherals, their IOs mappings are fixed when they are enabled. These specific purpose peripherals include JTAG, analog ios, GPIOs and key scan.

Figure 11 below shows the IOMUX functional diagram.

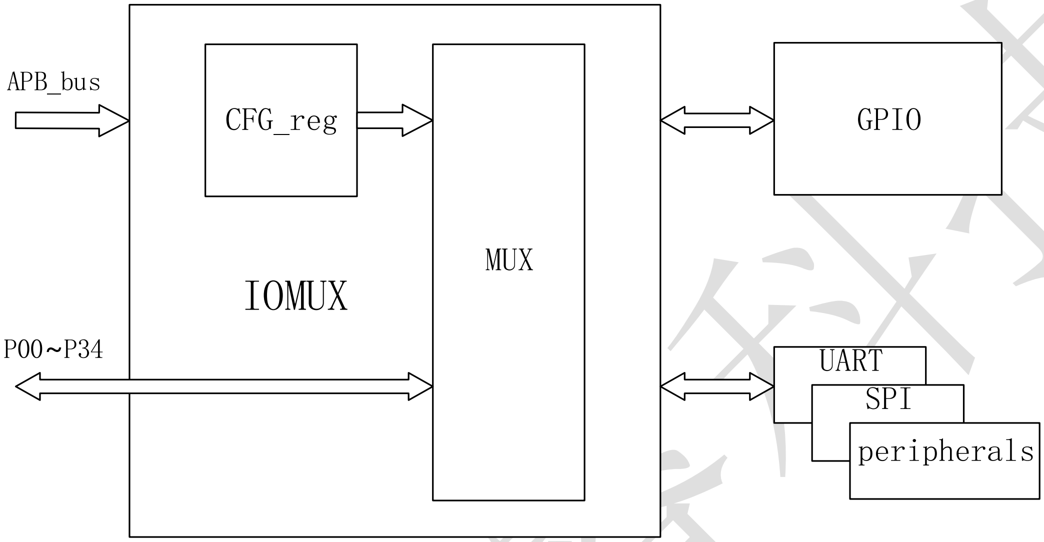


Figure 9: IOMUX structure diagram

There are 34 configurable pads which are from P00 to P07 and from P09 to P34. P08 pad is assigned for TM pin which is a test mode pin. The table blow shows the mapping of the peripheral IOs that can be mapped through IOMUX. These include I2C 0-1, I2S, UART, PWM 0-5, SPI 0-1, Quadrature Decoder, 1.28MHz clock and dmic out.

Signal Name	IO	FULLMUX
iic0_scl	B	0
iic0_sda	B	1
iic1_scl	B	2
iic1_sda	B	3
i2s_sck	B	4
i2s_ws	B	5
i2s_sdo0	O	6
i2s_sdo1	O	35
i2s_sdo2	O	36
i2s_sdo3	O	37
i2s_sdi0	I	7
i2s_sdi1	I	38
i2s_sdi2	I	39

i2s_sdi3	I	40
uart_tx	O	8
uart_rx	I	9
pwm0	O	10
pwm1	O	11
pwm2	O	12
pwm3	O	13
pwm4	O	14
pwm5	O	15
spi_0_sck	B	16
spi_0_ssn	B	17
spi_0_tx	O	18
spi_0_rx	I	19
spi_1_sck	B	20
spi_1_ssn	B	21
spi_1_tx	O	22
spi_1_rx	I	23
chax	I	24
chbx	I	25
chix	I	26
chay	I	27
chby	I	28
chiy	I	29
chaz	I	30
chbz	I	31
chiz	I	32
clk_1p28m	O	33
adcc_dmic_out	I	34

Table 7: Peripheral IO mapped through IOMUX

On the other hand, there are also special purpose peripherals, whose IOs are fixed to certain physical pads, when these peripheral functions are enabled. These special purpose peripherals include: JTAG, analog I/Os (ADC inputs), GPIO, and key scan. When they are enabled, their IOs are mapped to physical pads according to the following table (by default JTAG is enabled).

Number	GPIO				Name
0	GPIO_P00	jtag_dout	GPIO		mk_in[0]
1	GPIO_P01	jtag_din	GPIO		mk_out[0]
2	GPIO_P02	jtag_tm	GPIO		mk_in[1]

3	GPIO_P03	jtag_clk	GPIO		mk_out[1]
4	GPIO_P04	GPIO			mk_out[9]
5	GPIO_P05	GPIO			mk_in[10]
6	GPIO_P06	GPIO			mk_out[10]
7	GPIO_P07	GPIO			mk_in[11]
8	TEST_MODE				
9	GPIO_P09	GPIO			mk_out[4]
10	GPIO_P10	GPIO			mk_in[4]
11	GPIO_P11	GPIO		analog_io[0]	mk_out[11]
12	GPIO_P12	GPIO		analog_io[1]	mk_in[12]
13	GPIO_P13	GPIO		analog_io[2]	mk_out[12]
14	GPIO_P14	GPIO		analog_io[3]	mk_out[2]
15	GPIO_P15	GPIO		analog_io[4]	mk_in[2]
16	GPIO_P16	XTALI(ANA)	GPIO		mk_out[16]
17	GPIO_P17	XTALO(ANA)	GPIO		mk_out[17]
18	GPIO_P18	GPIO		analog_io[7]	mk_in[5]
19	GPIO_P19	GPIO		analog_io[8]	mk_in[13]
20	GPIO_P20	GPIO		analog_io[9]	mk_out[5]
21	GPIO_P21	GPIO			mk_out[13]
22	GPIO_P23	GPIO			mk_in[6]
23	GPIO_P25	GPIO			mk_in[3]
24	GPIO_P26	GPIO			mk_out[14]
25	GPIO_P28	GPIO			mk_out[8]
26	GPIO_P31	spi_t_ssn	GPIO		mk_out[7]
27	GPIO_P32	spi_t_rx	GPIO		mk_in[7]
28	GPIO_P33	spi_t_tx	GPIO		mk_out[6]
29	GPIO_P34	spi_t_sck	GPIO		mk_in[8]

Table 8: Peripheral IO mapped through IOMUX (special purpose)

In the IOMUX table above, the first column is the IO pad mapping in default mode, when no IOMUX function is selected and no special purpose peripherals such as analog IO, GPIO<0:3>, key scan, are enabled. In this mode, pin<0:3> are used for JTAG.

When analog IOs are enabled, pins<11:15>, <18:20> are connected to internal analog IOs. More specifically, analog_io<0:4><9> are connected to ADC inputs, analog_io<7,8> are connected to PGA inputs.

In JTAG mode, data output for JTAG test mode is mapped to P00; data input for JTAG test mode is mapped to P01; mode control input for JTAG test mode is mapped to P02; clock input for JTAG test mode is mapped to P03.

2.9.1 Register table

Detailed IOMUX register table and physical IO pad control are shown below.

Base address: 4000_3800

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x0			**r_analog_io**	
[31:10]	RW	22'h0	reserved	
[9:0]	RW	10'h60	r_analog_io_en	Analog IO enable
0xc			**full_mux0**	**register description**
[31:0]	RW	32'h0	r_func_io_en[31:0]	full mux enable. [8] must set to 0
0x10			**full_mux1**	**register description**
[31:3]	RW	29'h0	reserved	
[2:0]	RW	3'h0	r_func_io_en[34:32]	full mux enable
0x14			**gpio_papb**	**register description**
[31:17]	RW	15'h0	reserved	
[16]	RW	1'h0	r_gpio_pb_16_en	gpio_16 enable
[15]	RW	1'h0	r_gpio_pb_15_en	gpio_15 enable
[14]	RW	1'h0	r_gpio_pb_14_en	gpio_14 enable
[13]	RW	1'h0	r_gpio_pb_13_en	gpio_13 enable
[12:4]	RW	9'h0	reserved	
[3]	RW	1'h0	r_gpio_pa_03_en	gpio_03 enable
[2]	RW	1'h0	r_gpio_pa_02_en	gpio_02 enable
[1]	RW	1'h0	r_gpio_pa_01_en	gpio_01 enable
[0]	RW	1'h0	r_gpio_pa_00_en	gpio_00 enable
0x18			**func_io0**	**register description**
[31:30]	RW	2'h0	reserved	
[29:24]	RW	6'h0	r_func_io03_sel	pad 3 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io02_sel	pad 2 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io01_sel	pad 1 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io00_sel	pad 0 full mux function select
0x1c			**func_io1**	**register description**

[31:30]	RW	2'h0	reserved	
[29:24]	RW	6'h0	r_func_io07_sel	pad 7 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io06_sel	pad 6 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io05_sel	pad 5 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io04_sel	pad 4 full mux function select
0x20			**func_io2**	**register description**
[31:30]	RW	2'h0	reserved	
[29:24]	RW	6'h0	r_func_io11_sel	pad 11 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io10_sel	pad 10 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io09_sel	pad 9 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io08_sel	pad 8 full mux function select. not used. can delete
0x24			**func_io3**	**register description**
[31:30]	RW	2'h0	reserved	
[29:24]	RW	6'h0	r_func_io15_sel	pad 15 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io14_sel	pad 14 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io13_sel	pad 13 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io12_sel	pad 12 full mux function select
0x28			**func_io4**	**register description**
[31:30]	RW	2'h0	reserved	
[29:24]	RW	6'h0	r_func_io19_sel	pad 19 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io18_sel	pad 18 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io17_sel	pad 17 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io16_sel	pad 16 full mux function select
0x2c			**func_io5**	**register description**
[31:30]	RW	2'h0	reserved	

[29:24]	RW	6'h0	r_func_io23_sel	pad 23 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io22_sel	pad 22 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io21_sel	pad 21 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io20_sel	pad 20 full mux function select
0x30			**func_io6**	**register description**
[31:30]	RW	2'h0	reserved	
[29:24]	RW	6'h0	r_func_io27sel	pad 27 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io26_sel	pad 26 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io25_sel	pad 25 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io24_sel	pad 24 full mux function select
0x34			**func_io7**	**register description**
[31:30]	RW	2'h0	reserved	
[29:24]	RW	6'h0	r_func_io31sel	pad 31 full mux function select
[23:22]	RW	2'h0	reserved	
[21:16]	RW	6'h0	r_func_io30_sel	pad 30 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io29_sel	pad 29 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io28_sel	pad 28 full mux function select
0x38			**func_io8**	**register description**
[31:22]	RW	10'h0	reserved	
[21:16]	RW	6'h0	r_func_io34_sel	pad 34 full mux function select
[15:14]	RW	2'h0	reserved	
[13:8]	RW	6'h0	r_func_io33_sel	pad 33 full mux function select
[7:6]	RW	2'h0	reserved	
[5:0]	RW	6'h0	r_func_io32_sel	pad 32 full mux function select
0x4C			**key_scan_in_en**	**register description**
[31:16]	RW	16'h0	reserved	
[15:0]	RW	16'h0	r_kscan_in_en	key scan in enable
0x50			**key_scan_out_en**	**register description**
[31:18]	RW	14'h0	reserved	
[17:0]	RW	18'h0	r_kscan_out_en	key scan out enable

Table 9: Detailed IOMUX register

2.9.2 Register table

Physical IO PAD control registers:

Base address: 4000_F000

0xF008			**IOCTL0**
[31:30]	RW	2'd0	
[29:28]	RW	2'b0	pull up/down control of pin 09
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[27]	RW	1'b0	wake up polarity select of pin 09
			0: active POSEDGE
			1: active NEGEDGE
[26:24]	RW	3'b110	P08 is used for test mode config pin
[23:22]	RW	2'b0	pull up/down control of pin 07
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[21]	RW	1'b0	wake up polarity select of pin 07
			0: active POSEDGE
			1: active NEGEDGE
[20:19]	RW	2'b0	pull up/down control of pin 06
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[18]	RW	1'b0	wake up polarity select of pin 06
			0: active POSEDGE
			1: active NEGEDGE
[17:16]	RW	2'b0	pull up/down control of pin 05
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[15]	RW	1'b0	wake up polarity select of pin 05
			0: active POSEDGE
			1: active NEGEDGE

[14:13]	RW	2'b0	pull up/down control of pin 04 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[12]	RW	1'b0	wake up polarity select of pin 04 0: active POSEDGE 1: active NEGEDGE
[11:10]	RW	2'b11	pull up/down control of pin 03 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[9]	RW	1'b0	wake up polarity select of pin 03 0: active POSEDGE 1: active NEGEDGE
[8:7]	RW	2'b0	pull up/down control of pin 02 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[6]	RW	1'b0	wake up polarity select of pin 02 0: active POSEDGE 1: active NEGEDGE
[5:4]	RW	2'b0	pull up/down control of pin 01 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[3]	RW	1'b0	wake up polarity select of pin 01 0: active POSEDGE 1: active NEGEDGE
[2:1]	RW	2'b0	pull up/down control of pin 00 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[0]	RW	1'b0	wake up polarity select of pin 00 0: active POSEDGE

			1: active NEGEDGE
0xF00C			**IOCTL1**
[31:30]	RW	2'd0	
[29:28]	RW	2'b0	pull up/down control of pin 19
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[27]	RW	1'b0	wake up polarity select of pin 19
			0: active POSEDGE
			1: active NEGEDGE
[26:25]	RW	2'b0	pull up/down control of pin 18
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[24]	RW	1'b0	wake up polarity select of pin 18
			0: active POSEDGE
			1: active NEGEDGE
[23:22]	RW	2'b0	pull up/down control of pin 17
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[21]	RW	1'b0	wake up polarity select of pin 17
			0: active POSEDGE
			1: active NEGEDGE
[20:19]	RW	2'b0	pull up/down control of pin 16
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[18]	RW	1'b0	wake up polarity select of pin 16
			0: active POSEDGE
			1: active NEGEDGE
[17:16]	RW	2'b0	pull up/down control of pin 15
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up

			11: pull down
[15]	RW	1'b0	wake up polarity select of pin 15
			0: active POSEDGE
			1: active NEGEDGE
[14:13]	RW	2'b0	pull up/down control of pin 14
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[12]	RW	1'b0	wake up polarity select of pin 14
			0: active POSEDGE
			1: active NEGEDGE
[11:10]	RW	2'b0	pull up/down control of pin 13
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[9]	RW	1'b0	wake up polarity select of pin 13
			0: active POSEDGE
			1: active NEGEDGE
[8:7]	RW	2'b0	pull up/down control of pin 12
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[6]	RW	1'b0	wake up polarity select of pin 12
			0: active POSEDGE
			1: active NEGEDGE
[5:4]	RW	2'b0	pull up/down control of pin 11
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[3]	RW	1'b0	wake up polarity select of pin 11
			0: active POSEDGE
			1: active NEGEDGE
[2:1]	RW	2'b0	pull up/down control of pin 10
			00: floating, no pull up and pull down
			01: weak pull up

			10: strong pull up 11: pull down
[0]	RW	1'b0	wake up polarity select of pin 10 0: active POSEDGE 1: active NEGEDGE
0xF010			**IOCTL2**
[31:30]	RW	2'd0	
[29:28]	RW	2'b0	pull up/down control of pin 29 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[27]	RW	1'b0	wake up polarity select of pin 29 0: active POSEDGE 1: active NEGEDGE
[26:25]	RW	2'b0	pull up/down control of pin 28 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[24]	RW	1'b0	wake up polarity select of pin 28 0: active POSEDGE 1: active NEGEDGE
[23:22]	RW	2'b0	pull up/down control of pin 27 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[21]	RW	1'b0	wake up polarity select of pin 27 0: active POSEDGE 1: active NEGEDGE
[20:19]	RW	2'b0	pull up/down control of pin 26 00: floating, no pull up and pull down 01: weak pull up 10: strong pull up 11: pull down
[18]	RW	1'b0	wake up polarity select of pin 26 0: active POSEDGE

			1: active NEGEDGE
[17:16]	RW	2'b11	pull up/down control of pin 25
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[15]	RW	1'b0	wake up polarity select of pin 25
			0: active POSEDGE
			1: active NEGEDGE
[14:13]	RW	2'b11	pull up/down control of pin 24
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[12]	RW	1'b0	wake up polarity select of pin 24
			0: active POSEDGE
			1: active NEGEDGE
[11:10]	RW	2'b0	pull up/down control of pin 23
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[9]	RW	1'b0	wake up polarity select of pin 23
			0: active POSEDGE
			1: active NEGEDGE
[8:7]	RW	2'b0	pull up/down control of pin 22
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[6]	RW	1'b0	wake up polarity select of pin 22
			0: active POSEDGE
			1: active NEGEDGE
[5:4]	RW	2'b0	pull up/down control of pin 21
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[3]	RW	1'b0	wake up polarity select of pin 21

			0: active POSEDGE
			1: active NEGEDGE
[2:1]	RW	2'b0	pull up/down control of pin 20
			00: floating, no pull up and pull down
			01: weak pull up
			10: strong pull up
			11: pull down
[0]	RW	1'b0	wake up polarity select of pin 20
			0: active POSEDGE
			1: active NEGEDGE

Table 10: Physical IO PAD control registers

2.10 GPIO

The General Purpose I/Os are a type of peripheral that can be mapped to physical I/O pads and programmed by software. The flexible GPIO are organized as two PORTs. Among them, PortA has bi-direction 18 bit lines, e.g., GPIO_PORTA[17:0], while PortB has 17 bi-directional bit lines, e.g., PIO_PORTB[16:0]. With default setting, physical pads: P00-P17 are connected to PortA; Pads P18-34 are connected to PortB, when all GPIOs are enabled, as described in the IOMUX table in IOMUX section.

All PortA and PortB pins can be configured as bi-directional serial interface, by selecting as input or output direction, and their corresponding data can be either read from or written to registers. All PortA and PortB pins support wake-up, but only 18 PortA pins support interrupt. Also only PortA pins support debounce function.
Each GPIO pins can be pulled up to AVDD33 or pulled down to ground by adding pull up or pull down resistors to have default functions/states.

For more detailed info, please refer to "QMS7926 GPIO Application Notes", in software SDK document folder.

#	GPIO	Default MODE	Default IN_OUT	IRQ	Wakeup	ANA_IO
0	GPIO_P00	jtag_dout	OUT	√	√	
1	GPIO_P01	jtag_din	IN	√	√	
2	GPIO_P02	jtag_tm	IN	√	√	
3	GPIO_P03	jtag_clk	IN	√	√	
4	GPIO_P04	GPIO	IN	√	√	
5	GPIO_P05	GPIO	IN	√	√	
6	GPIO_P06	GPIO	IN	√	√	

7	GPIO_P07	GPIO	IN	√	√	
8	TEST_MODE					
9	GPIO_P09	GPIO	IN	√	√	
10	GPIO_P10	GPIO	IN	√	√	
11	GPIO_P11	GPIO	IN	√	√	ADC_CH1N_P11
12	GPIO_P12	GPIO	IN	√	√	ADC_CH1P_P12
13	GPIO_P13	GPIO	IN	√	√	ADC_CH2N_P13
14	GPIO_P14	GPIO	IN	√	√	ADC_CH2P_P14
15	GPIO_P15	GPIO	IN	√	√	ADC_CH3N_P15
16	GPIO_P16	XTALI(ANA)	ANA	√	√	
17	GPIO_P17	XTALO(ANA)	ANA	√	√	
18	GPIO_P18	GPIO	IN		√	
19	GPIO_P19	GPIO	IN		√	
20	GPIO_P20	GPIO	IN		√	ADC_CH3P_P20
21	GPIO_P21	GPIO	IN		√	
22	GPIO_P23	GPIO	IN		√	
23	GPIO_P25	GPIO	IN		√	
24	GPIO_P26	GPIO	IN		√	
25	GPIO_P28	GPIO	IN		√	
26	GPIO_P31	GPIO	IN		√	
27	GPIO_P32	GPIO	IN		√	
28	GPIO_P33	GPIO	OUT		√	
29	GPIO_P34	GPIO	IN		√	

Table 11: QMS7926 GPIO Application Notes

2.10.1 Register table

Blow table are the Registers related to GPIOs.

Base address: 0x4000_8000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00			**gpio_swporta_dr**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RW	18'b0	Port A Data Register	Values written to this register are output on the I/O signals for Port A
0x04			**gpio_swporta_ddr**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RW	18'b0	Port A Data Direction Register	Values written to this register independently control the direction of the corresponding data bit in Port A
				1'b0: Input

				1'b1: Output
0x08			**gpio_swporta_ctl**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Port A Data Source	The data and control source for a signal can come from either software or hardware
				1'b0: Software mode
				1'b1: Hardware mode
0x0c			**gpio_swportb_dr**	
[31:15]	RO	15'b0	Reserved	Reserved
[16:0]	RW	17'b0	Port B Data Register	Values written to this register are output on the I/O signals for Port B
0x10			gpio_swportb_ddr	
[31:15]	RO	15'b0	Reserved	Reserved
[16:0]	RW	17'b0	Port B Data Direction Register	Values written to this register independently control the direction of the corresponding data bit in Port B
				1'b0: Input
				1'b1: Output
0x14			**gpio_swportb_ctl**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Port B Data Source	The data and control source for a signal can come from either software or hardware
				1'b0: Software mode
				1'b1: Hardware mode
0x30			**gpio_inten**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RW	18'b0	Interrupt enable	Allows each bit of Port A to be configured for interrupts
				1'b0: Configure Port A bit as normal GPIO signal
				1'b1: Configure Port A bit as interrupt
0x34			**gpio_intmask**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RW	18'b0	Interrupt mask	Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it
				1'b0: Interrupt bits are unmasked
				1'b1: Mask interrupt
0x38			**gpio_inttype_level**	
[31:18]	RO	14'b0	Reserved	Reserved

[17:0]	RW	18'b0	Interrupt level	Controls the type of interrupt that can occur on Port A 1'b0: Level-sensitive 1'b1: Edge-sensitive
0x3c			**gpio_int_polarity**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RW	18'b0	Interrupt polarity	Controls the polarity of edge or level sensitivity that can occur on input of Port A 1'b0: Active-low or falling-edge 1'b1: Active-high or rising-edge
0x40			**gpio_intstatus**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RO	18'b0	Interrupt status	Interrupt status of Port A
0x44			**gpio_raw_intstatus**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RO	18'b0	Raw interrupt status	Raw interrupt of status of Port A
0x48			**gpio_debounce**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RW	18'b0	Debounce enable	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches 1'b0: No debounce 1'b1: Enable debounce
0x4c			**gpio_porta_eoi**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	WO	18'b0	Clear interrupt	Controls the clearing of edge type interrupts from Port A 1'b0: No interrupt clear 1'b1: Clear interrupt
0x50			**gpio_ext_porta**	
[31:18]	RO	14'b0	Reserved	Reserved
[17:0]	RO	18'b0	External Port A	When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A
x54			**gpio_ext_portb**	
[31:17]	RO	15'b0	Reserved	Reserved

[16:0]	RO	17'b0	External Port B	When Port B is configured as Input, then reading this location reads the values on the signal. When the data direction of Port B is set as Output, reading this location reads the data register for Port B
0x60			**gpio_ls_sync**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Synchronization level	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr
				1'b0: No synchronization to pclk_intr
				1'b1: Synchronize to pclk_intr
0x64			**gpio_id_code**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RO	16'b0	GPIO ID code	This is a user-specified code that a system can read. It can be used for chip identification, and so on
0x6c			**gpio_ver_id_code**	
[31:0]	RO	32'b0	GPIO Component Version	ASCII value for each number in the version
0x74			**gpio_config_reg1**	
[31:21]	RO	11'b0	Reserved	Reserved
[20:16]	RO	5'b0x0f	ENCODED_ID_WIDTH	The value of this register is equal to GPIO_ID_WIDTH-1
[15]	RO	1'b0	GPIO_ID	The value of this register is derived from the GPIO_ID configuration parameter
				1'b0: Exclude
				1'b1: Include
[14]	RO	1'b0	ADD_ENCODED_PARAMS	The value of this register is derived from the GPIO_ADD_ENCODED_PARAMS configuration parameter
				1'b0: False
				1'b1: True
[13]	RO	1'b0	DEBOUNCE	The value of this register is derived from the GPIO_DEBOUNCE configuration parameter
				1'b0: Exclude
				1'b1: Include
[12]	RO	1'b0	PORTA_INTR	The value of this register is derived from the GPIO_PORTA_INTR configuration parameter
				1'b0: Exclude

				1'b1: Include
[11]	RO	1'b0	Reserved	Reserved
[10]	RO	1'b0	Reserved	Reserved
[9]	RO	1'b0	HW_PORTB	The value of this register is derived from the GPIO_HW_PORTB configuration parameter
				1'b0: Exclude
				1'b1: Include
[8]	RO	1'b0	HW_PORTA	The value of this register is derived from the GPIO_HW_PORTA configuration parameter
				1'b0: Exclude
				1'b1: Include
[7]	RO	1'b0	Reserved	Reserved
[6]	RO	1'b0	Reserved	Reserved
[5]	RO	1'b0	PORTB_SINGLE_CTL	The value of this register is derived from the GPIO_PORTB_SINGLE_CTL configuration parameter
				1'b0: False
				1'b1: True
[4]	RO	1'b0	PORTA_SINGLE_CTL	The value of this register is derived from the GPIO_PORTA_SINGLE_CTL configuration parameter
				1'b0: False
				1'b1: True
[3:2]	RO	2'b0x 2	NUM_PORTS	The value of this register is derived from the GPIO_NUM_PORT configuration parameter
				2'b00 1
				2'b01 2
				2'b10 3
				2'b11 4
[1:0]	RO	2'b0x 2	APB_DATA_WIDTH	The value of this register is derived from the GPIO_APB_DATA_WIDTH configuration parameter
				2'b00 8 bits
				2'b01 16 bits
				2'b10 32 bits
				2'b11 Reserved
0x70			**gpio_config_reg2**	
[31:10]	RO	22'b0	Reserved	Reserved

[9:5]	RO	5'b0x 0f	ENCODED_ID_PWIDTH_B	The value of this register is equal to GPIO_PWIDTH_B-1
[4:0]	RO	5'b0x 11	ENCODED_ID_PWIDTH_A	The value of this register is equal to GPIO_PWIDTH_A-1

Table 12: GPIOs registers

2.10.2 DC Characteristics

TA=25℃, VDD=3 V

PARAMETER	TEST CONDITIONS	Min.	TYP	Max.	Unit
Logic-0 input voltage				0.5	V
Logic-1 input voltage		2.4			V
Logic-0 input current	Input equals 0 V	-50		50	nA
Logic-1 input current	Input equals VDD	-50		50	nA
Logic-0 output voltage, 10-mA pins	Output load 10 mA			0.5	V
Logic-1 output voltage, 10-mA pins	Output load 10 mA	2.5			V

Table 13: DC Characteristics

3 Peripheral Blocks

3.1 2.4GHz Radio

The 2.4 GHz RF transceiver is designed to operate in the worldwide ISM frequency band at 2.4 to 2.4835 GHz. Radio modulation modes and configurable packet structure make the transceiver interoperable with *Bluetooth*® low energy (BLE) protocol implementations.

- General modulation format
  - FSK (configurable modulation index) with configurable Gaussian Filter Shaping
  - OQPSK with half-sine shaping
  - On-air data rates
  - 125kbps/250kbps/500kbps/1Mbps/2Mbps
- Transmitter with programmable output power of -20dBm to +10dBm, in 3dB steps
- RSSI function (1 dB resolution, ± 2 dB accuracy)
- Receiver sensitivity
  - -103dBm@125Kbps GFSK
  - -98dBm@500Kbps GFSK
  - -97dBm@1Mbps BLE
  - -94dBm@2Mbps BLE
- Embedded RF balun

- Integrated frac-N synthesizer with phase modulation

3.2 Timer/Counters (TIMER)

The implementation can include a 24-bit SysTick system timer, that extends the functionality of both the processor and the NVIC. When present, the NVIC part of the extension provides:

- **A 24-bit system timer (SysTick)**
- **Additional configurable priority SysTick interrupt.**
- **See the ARMv7-M ARM for more information.**

General purpose timers are included in the design. This timer is Synopsys DW_apb_timer. With the input clock running at 4Mhz.

3.2.1 Register table

The timer related registers are listed below, and there are two sets of identical timers.

Base address: Timer setA: 4000_1000, timer_setB: 4002_1000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00			**Timer1LoadCount**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:0]	RW	24'b0	Timer1 Load Count Register	Value to be loaded into Timer1
0x04			**Timer1CurrentValue**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:0]	RO	24'b0	Timer1 Current Value Register	Current Value of Timer1
0x08			**Timer1ControlReg**	
[31:3]	RO	29'b0	Reserved	Reserved
[2]	RW	1'b0	Timer Interrupt Mask	Timer interrupt mask for Timer1 1'b0: not masked 1'b1: masked
[1]	RW	1'b0	Timer Mode	Timer mode for Timer1 1'b0: free-running mode 1'b1: user-defined count mode
[0]	RW	1'b0	Timer Enable	Timer enable bit for Timer1 1'b0: disable 1'b1: enable
0x0c			**Timer1EOI**	

[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer1 End of-Interrupt Register	Reading from this register returns all zeroes (0) and clears the interrupt from Timer1
0x10			**Timer1IntStatus**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer1 Interrupt Status Register	Contains the interrupt status for Timer1
0x14			**Timer2LoadCount**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:0]	RW	24'b0	Timer2 Load Count Register	Value to be loaded into Timer2
0x18			**Timer2CurrentValue**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:0]	RO	24'b0	Timer2 Current Value Register	Current Value of TimerN
0x1c			**Timer2ControlReg**	
[31:3]	RO	29'b0	Reserved	Reserved
[2]	RW	1'b0	Timer Interrupt Mask	Timer interrupt mask for Timer2 1'b0: not masked 1'b1: masked
[1]	RW	1'b0	Timer Mode	Timer mode for Timer2 1'b0: free-running mode 1'b1: user-defined count mode
[0]	RW	1'b0	Timer Enable	Timer enable bit for Timer2 1'b0: disable 1'b1: enable
0x20			**Timer2EOI**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer2 End of-Interrupt Register	Reading from this register returns all zeroes (0) and clears the interrupt from Timer2
0x24			**Timer2IntStatus**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer2 Interrupt Status Register	Contains the interrupt status for Timer2
0x28			**Timer3LoadCount**	
[31:24]	RO	8'b0	Reserved	Reserved

[23:0]	RW	24'b0	Timer3 Load Count Register	Value to be loaded into Timer3
0x2c			**Timer3CurrentValue**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:0]	RO	24'b0	Timer3 Current Value Register	Current Value of TimerN
0x30			**Timer3ControlReg**	
[31:3]	RO	29'b0	Reserved	Reserved
[2]	RW	1'b0	Timer Interrupt Mask	Timer interrupt mask for Timer3
				1'b0: not masked
				1'b1: masked
[1]	RW	1'b0	Timer Mode	Timer mode for Timer3
				1'b0: free-running mode
				1'b1: user-defined count mode
[0]	RW	1'b0	Timer Enable	Timer enable bit for Timer3
				1'b0: disable
				1'b1: enable
0x34			**Timer3EOI**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer3 End of-Interrupt Register	Reading from this register returns all zeroes (0) and clears the interrupt from Timer3
0x38			**Timer3IntStatus**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer3 Interrupt Status Register	Contains the interrupt status for Timer3
0x3c			**Timer4LoadCount**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:0]	RW	24'b0	Timer4 Load Count Register	Value to be loaded into Timer4
0x40			**Timer4CurrentValue**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:0]	RO	24'b0	Timer4 Current Value Register	Current Value of Timer4
0x44			**Timer4ControlReg**	
[31:3]	RO	29'b0	Reserved	Reserved
[2]	RW	1'b0	Timer Interrupt Mask	Timer interrupt mask for Timer4
				1'b0: not masked
				1'b1: masked

[1]	RW	1'b0	Timer Mode	Timer mode for Timer4 1'b0: free-running mode 1'b1: user-defined count mode
[0]	RW	1'b0	Timer Enable	Timer enable bit for Timer4 1'b0: disable 1'b1: enable
0x48			**Timer4EOI**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer4 End of-Interrupt Register	Reading from this register returns all zeroes (0) and clears the interrupt from Timer4
0x4c			**Timer4IntStatus**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	Timer4 Interrupt Status Register	Contains the interrupt status for Timer4
0xa0			**TimersIntStatus**	
[31:4]	RO	28'b0	Reserved	Reserved
[3:0]	RO	4'b0	Timers Interrupt Status Register	Contains the interrupt status of all timers in the component 0: either timer_intr or timer_intr_n is not active after masking 1: either timer_intr or timer_intr_n is active after masking
0xa4			**TimersEOI**	
[31:4]	RO	28'b0	Reserved	Reserved
[3:0]	RO	4'b0	Timers End of-Interrupt Register	Reading this register returns all zeroes (0) and clears all active interrupts
0xa8			**TimersRawIntStatus**	
[31:4]	RO	28'b0	Reserved	Reserved
[3:0]	RO	4'b0	Timers Raw Interrupt Status Register	The register contains the unmasked interrupt status of all timers in the component 0: either timer_intr or timer_intr_n is not active prior to masking 1: either timer_intr or timer_intr_n is active prior to masking
0xac			**TimersRawIntStatus**	
[31:0]	RO	32'b0	Timers Component Version	Current revision number of the DW_apb_timers component

Table 14: Timer registers

3.3 Real Time Counter (RTC)

The Real Time Counter (RTC) module provides a generic, low power timer on the low-frequency clock source (LFCLK). The RTC features a 24 bit COUNTER, 12 bit (1/X) prescaler, capture/compare registers, and a tick event generator for low power, tickless RTOS implementation.

3.3.1 Register table

RTC related registers are listed below.

Base address: 4000_F000

0xF024			**RTCCTL**
[31:24]	RW	8'h0	
[23]	RW	1'b0	Counter overflow event enable.
			1'b0: disable
			1'b1: enable
[22]	RW	1'b0	Comparator 2 event enable.
			1'b0: disable
			1'b1: enable
[21]	RW	1'b0	Comparator 1 event enable.
			1'b0: disable
			1'b1: enable
[20]	RW	1'b0	Comparator 0 event enable.
			1'b0: disable
			1'b1: enable
[19]	RW	1'b0	RTC tick event enable.
			1'b0: disable
			1'b1: enable
[18]	RW	1'b0	Counter overflow interrupt enable.
			1'b0: disable
			1'b1: enable
[17]	RW	1'b0	Comparator 2 interrupt enable.
			1'b0: disable
			1'b1: enable
[16]	RW	1'b0	Comparator 1 interrupt enable.
			1'b0: disable
			1'b1: enable
[15]	RW	1'b0	Comparator 0 interrupt enable.
			1'b0: disable
			1'b1: enable

[14]	RW	1'b0	RTC tick interrupt enable. 1'b0: disable 1'b1: enable
[13:2]	RW	12'h0	12bit prescaler for RTC counter frequency (32768/(PRESCALER+1)).Can be written only when RTC is stopped.
[1]	RW	1'b0	RTC counter clear bit. Write 1'b1 will clear RTC counter and after one clock this bit will return to 1'b0.
[0]	RW	1'b0	RTC run/stop control. 1'b0: stop 1'b1: run
0xF028			**RTCCNT**
[31:24]	RO	8'h0	
[23:0]	RO	24'h0	Writing32'h5A5AA5A5 can trigger the overflow task that sets the RTC counter value to 24'hFFFFF0 to allow SW test of the overflow condition. Reading can read the value of RTC counter (low 24 bits).
0xF02C			**RTCCC0**
[31:24]	RW	8'h0	
[23:0]	RW	24'h0	Compare value of comparator 0
0xF030			**RTCCC1**
[31:24]	RW	8'h0	
[23:0]	RW	24'h0	Compare value of comparator 1
0xF034			**RTCCC2**
[31:24]	RW	8'h0	
[23:0]	RW	24'h0	Compare value of comparator 2
0xF038			**RTCFLAG**
[31:4]	R	28'h0	
[3]	RO	1'b0	Overflow result flag.
[2]	RO	1'b0	Compare result flag of comparator 2.
[1]	RO	1'b0	Compare result flag of comparator 1.
[0]	RO	1'b0	Compare result flag of comparator 0.

Table 15: RTC registers

3.4 AES-ECB Encryption (ECB)

The ECB encryption block supports 128 bit AES encryption. It can be used for a range of cryptographic functions like hash generation, digital signatures, and keystream generation for data encryption/decryption.

3.4.1 Register table

AES-ECB related registers are listed below.

Base address:4004_0000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00				**AES layer enable register**
[31:1]	—	31'b0	reserved	
[0]	RW	1'b0	Enable	Setting this bit to "1" will enable AES to do TX/RX
0x04				**AES layer control register**
[31:17]	—	15'b0	reserved	
[16]	RW	1'b0	FIFO out/in (PDU)	if pdu is little-endian set 0;if pdu is big-endian set 1
[15:12]	—	4'b0	reserved	
[11:8]	RW	4'b0	Enginne revert	[11]:data out:
				if it is little-endian set 0
				if it is big-endian set 1
				[10]:xor data :1
				[9]: key :
				if it is little-endian set 0
				if it is big-endian set 1
				[8]:data
				if it is little-endian set 0
				if it is big-endian set 1
[7:5]	—	3'b0	reserved	
[4]	RW	1'b0	AES_single mode	AES single mode
[3]	RW	1'b0	Code_mode	Encript /decript
[2:0]	—	3'b0	reserved	
0x08				**AES reserved register**
[31:0]	—	32'b0	reserved	
0x0c				**AES plen & aad register**
[31:16]	—	16'b0	reserved	
[15:8]	RW	8'b0	plen	Packet length
[7:0]	RW	8'b0	aad	aad
0x10				**AES interrupt mask register**
[31:4]	—	28'b0	reserved	
[3:0]	RW	4'b0	AES interrupt enable	[0]: encript done;[1]: decript failed;[2[: decript ok;[3] single mode done
0x14				**AES interrupt status register**
[31:4]	—	28'b0	reserved	

[3:0]	RO	4'b0	AES interrupt status	[0]: encript done;[1]: decript failed;[2[: decript ok;[3] single mode done
0x18				**AES reserved register**
[31:0]	—	32'b0	reserved	
0x1C				**AES reserved register**
[31:0]	—	32'b0	reserved	
0x20				**AES key0 register**
[31:0]	RW	32'b0	Key0[31:0]	Key[31:0]
0x24				**AES key1 register**
[31:0]	RW	32'b0	Key1[31:0]	Key[63:32]
0x28				**AES key2 register**
[31:0]	RW	32'b0	Key2[31:0]	Key[95:64]
0x2C				**AES key3 register**
[31:0]	RW	32'b0	Key3[31:0]	Key[127:96]
0x30				**AES nonce0 register**
[31:0]	RW	32'b0	Nonce0[31:0]	Single mode:data_in[31;0];ECB-CCM:Nonce[31:0]
0x34				**AES nonce1 register**
[31:0]	RW	32'b0	Nonce1[31:0]	Single mode:data_in[31;0];ECB-CCM:Nonce[63:32]
0x38				**AES nonce2 register**
[31:0]	RW	32'b0	Nonce2[31:0]	Single mode:data_in[31;0];ECB-CCM:Nonce[95:64]
0x3C				**AES nonce3 register**
[31:0]	RW	32'b0	Nonce3[31:0]	Single mode:data_in[31;0];ECB-CCM:Nonce[127:96]
0x50				**AES data out 0(single mode) register**
[31:0]	RO	32'b0	Data_o0[31:0]	Data_out[31:0]
0x54				**AES data out 1(single mode) register**
[31:0]	RO	32'b0	Data_o1[31:0]	Data_out[63:32]
0x58				**AES data out 2(single mode) register**
[31:0]	RO	32'b0	Data_o2[31:0]	Data_out[95:64]
0x5C				**AES data out 3(single mode) register**
[31:0]	RO	32'b0	Data_o3[31:0]	Data_out[127:96]
0x100				**AES memory (0x0100~0x01FC)**
[31:0]	RW	32'b0	memory write	Writing offset address 0x100~0x1FC will write data into AES memory

Table 16: AES-ECB registers

3.5 Random Number Generator (RNG)

The Random Number Generator (RNG) generates true non-deterministic random numbers based on internal thermal noise. These random numbers are suitable for cryptographic purposes. The RNG does not require a seed value.

3.6 Watchdog Timer (WDT)

A count down watchdog timer using the low-frequency clock source (LFCLK) offers configurable and robust protection against application lock-up. The watchdog can be paused during long CPU sleep periods for low power applications and when the debugger has halted the CPU.

3.7 SPI (SPI)

The SPI interface supports 3 serial synchronous protocols which are SPI, SSP and Microwire serial protocols. SPI wrapper contains one SPI master and one SPI slave. They are logically exclusive. Only one block is alive at a time. The operation mode for master mode and slave mode is controlled by PERI_MASTER_SELECT Register in COM block.

bit	Reset value	Definition
1	0	SPI1 is master mode when set
0	0	SPI0 is master mode when set

Table 17: PERI_MASTER_SELECT Register bit definition (base address = 0x4000_302C)

3.7.1 Register table

SPI0 and SPI1 configuration registers are listed below.

Base address: SPI0: 4000_6000; SPI1: 4000_7000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00			**CTRLR0**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:12]	RW	4'b0	CFS	Control Frame Size. Selects the length of the control word for the Microwire frame format
[11]	RW	1'b0	SRL	Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input
				1'b0: Normal Mode Operation
				1'b1: Test Mode Operation
[10]	RW	1'b0	SLV_OE	Slave Output Enable

				1'b0: Slave txd is enabled
				1'b1: Slave txd is disabled
[9:8]	RW	2'b0	TMOD	Transfer Mode. Selects the mode of transfer for serial communication.
				2'b00: Transmit & Receive
				2'b01: Transmit Only
				2'b10: Receive Only
				2'b11: EEPROM Read
[7]	RW	1'b0	SCPOL	Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock
				1'b0: Inactive state of serial clock is low
				1'b1: Inactive state of serial clock is high
[6]	RW	1'b0	SCPH	Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal
				1'b0: Serial clock toggles in middle of first data bit
				1'b1: Serial clock toggles at start of first data bit
[5:4]	RW	2'b0	FRF	Frame Format. Selects which serial protocol transfers the data
				2'b00: Motorola SPI
				2'b01: Texas Instruments SSP
				2'b10: National Semiconductors Microwire
				2'b11: Reserved
[3:0]	RW	4'b0x7	DFS	Data Frame Size. Selects the data frame length
0x04			**CTRLR1**	**DW_apb_ssi is configured as a master device**
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	NDF	Number of Data Frames
0x08			**SSIENR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	SSI_EN	This register enables and disables the DW_apb_ssi
				1'b0: disable
				1'b1: enable
0x0c			**MWCR**	

[31:3]	RO	29'b0	Reserved	Reserved
[2]	RW	1'b0	MHS	Microwire Handshaking
				1'b0: disabled
				1'b1: enabled
[1]	RW	1'b0	MDD	Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used
[0]	RW	1'b0	MWMOD	Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential
				1'b0: non-sequential transfer
				1'b1: sequential transfer
0x10			**SER**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	SER	Slave Select Enable Flag
				1'b0: non-sequential transfer
				1'b1: sequential transfer
0x14			**BAUDR**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	SCKDV	SSI Clock Divider
0x18			**TXFTLR**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b0	TFT	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt
0x1c			**RXFTLR**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b0	RFT	Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt
0x20			**TXFLR**	
[31:4]	RO	28'b0	Reserved	Reserved
[3:0]	RO	4'b0	TXTFL	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO
0x24			**RXFLR**	
[31:4]	RO	28'b0	Reserved	Reserved
[3:0]	RO	4'b0	RXTFL	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO
0x28			**SR**	

[31:7]	RO	25'b0	Reserved	Reserved
[6]	RO	1'b0	DCOL	Data Collision Error 1'b0: No error 1'b1: Transmit data collision error
[5]	RO	1'b0	TXE	Transmission Error. Set if the transmit FIFO is empty when a transfer is started 1'b0: No error 1'b1: Transmission error
[4]	RO	1'b0	RFF	Receive FIFO Full 1'b0: not full 1'b1: full
[3]	RO	1'b0	RFNE	Receive FIFO Not Empty 1'b0: empty 1'b1: not empty
[2]	RO	1'b1	TFE	Transmit FIFO Empty 1'b0: not empty 1'b1: empty
[1]	RO	1'b1	TFNF	Transmit FIFO Not Full 1'b0: full 1'b1: not full
[0]	RO	1'b0	BUSY	SSI Busy Flag 1'b0: DW_apb_ssi is idle or disabled 1'b1: DW_apb_ssi is actively transferring data
0x2c			**IMR**	
[31:6]	RO	26'b0	Reserved	Reserved
[5]	RW	1'b1	MSTIM	Multi-Master Contention Interrupt Mask 1'b0: masked 1'b1: not masked
[4]	RW	1'b1	RXFIM	Receive FIFO Full Interrupt Mask 1'b0: masked 1'b1: not masked
[3]	RW	1'b1	RXOIM	Receive FIFO Overflow Interrupt Mask 1'b0: masked 1'b1: not masked
[2]	RW	1'b1	RXUIM	Receive FIFO Underflow Interrupt Mask 1'b0: masked 1'b1: not masked
[1]	RW	1'b1	TXOIM	Transmit FIFO Overflow Interrupt Mask

				1'b0: masked 1'b1: not masked
[0]	RW	1'b1	TXEIM	Transmit FIFO Empty Interrupt Mask 1'b0: masked 1'b1: not masked
0x30			**ISR**	
[31:6]	RO	26'b0	Reserved	Reserved
[5]	RO	1'b0	MSTIS	Multi-Master Contention Interrupt Status 1'b0: not active 1'b1: active
[4]	RO	1'b0	RXFIS	Receive FIFO Full Interrupt Status 1'b0: not active 1'b1: active
[3]	RO	1'b0	RXOIS	Receive FIFO Overflow Interrupt Status 1'b0: not active 1'b1: active
[2]	RO	1'b0	RXUIS	Receive FIFO Underflow Interrupt Status 1'b0: not active 1'b1: active
[1]	RO	1'b0	TXOIS	Transmit FIFO Overflow Interrupt Status 1'b0: not active 1'b1: active
[0]	RO	1'b0	TXEIS	Transmit FIFO Empty Interrupt Status 1'b0: not active 1'b1: active
0x34			**RISR**	
[31:6]	RO	26'b0	Reserved	Reserved
[5]	RO	1'b0	MSTIR	Multi-Master Contention Raw Interrupt Status 1'b0: not active 1'b1: active
[4]	RO	1'b0	RXFIR	Receive FIFO Full Raw Interrupt Status 1'b0: not active 1'b1: active
[3]	RO	1'b0	RXOIR	Receive FIFO Overflow Raw Interrupt Status 1'b0: not active 1'b1: active
[2]	RO	1'b0	RXUIR	Receive FIFO Underflow Raw Interrupt Status 1'b0: not active

				1'b1: active
[1]	RO	1'b0	TXOIR	Transmit FIFO Overflow Raw Interrupt Status
				1'b0: not active
				1'b1: active
[0]	RO	1'b0	TXEIR	Transmit FIFO Empty Raw Interrupt Status
				1'b0: not active
				1'b1: active
0x38			**TXOICR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	TXOICR	Clear Transmit FIFO Overflow Interrupt
0x3c			**RXOICR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	RXOICR	Clear Receive FIFO Overflow Interrupt
0x40			**RXUICR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	RXUICR	Clear Receive FIFO Underflow Interrupt
0x44			**MSTICR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	MSTICR	Clear Multi-Master Contention Interrupt
0x48			**ICR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	ICR	Clear Interrupts
0x4c			**DMACR**	
[31:2]	RO	30'b0	Reserved	Reserved
[1]	RW	1'b0	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel
				1'b0: disable
				1'b1: enable
[0]	RW	1'b0	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel
				1'b0: disable
				1'b1: enable
0x50			**DMATDLR**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b0	DMATDL	Transmit Data Level
0x54			**DMARDLR**	
[31:3]	RO	29'b0	Reserved	Reserved

[2:0]	RW	3'b0	DMARDL	Receive Data Level
0x58			**IDR**	
[31:0]	RO	32'b0	IDCODE	Identification Code
0x5c			**SSI_COMP_VERSION**	
[31:0]	RO	32'b0	SSI_COMP_VERSION	Contains the hex representation of the Synopsys component version
0x60~0x9c			**DR**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	DR	Data Register Read: Receive FIFO buffer Write: Transmit FIFO buffer
0xf4			**RSVD_0**	
[31:0]	RW	32'b0	Reserved	Reserved location for future use
0xf8			**RSVD_1**	
[31:0]	RW	32'b0	Reserved	Reserved location for future use
0xfc			**RX_SAMPLE_DLY**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0	RSD	Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input signal

Table 18: SPI0 and SPI1 configuration registers

3.8 I2C (I2c0, I2c1 Two Independent Instances)

This I2C block support 100Khz, and 400Khz modes. It also supports 7-bit address and 10-bit address. It has built-in configurable spike suppression function for both lines.

3.8.1 Register table

I2C registers are listed below.

Base address: I2C0: 4000_5000, I2C1: 4000_5800

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00			**I2C Control Register**	
[31:7]	RO	25'b0	Reserved	Reserved
[6]	RW	1'b0	IC_SLAVE_DISABLE	This bit controls whether I2C has its slave disabled 1'b0: slave is enabled 1'b1: slave is disabled
[5]	RW	1'b1	IC_RESTART_EN	Determines whether RESTART conditions may be sent when acting

				as a master 1'b0: disable 1'b1: enable
[4]	RW	1'b1	IC_10BITADDR_MASTER	Controls whether the DW_apb_i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master 1'b0: 7-bit addressing 1'b1: 10-bit addressing
[3]	RW	1'b1	IC_10BITADDR_SLAVE	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses 1'b0: 7-bit addressing 1'b1: 10-bit addressing
[2:1]	RW	2'b11	SPEED	These bits control at which speed the DW_apb_i2c operates 2'b01: standard mode 2'b10: fast mode 2'b11: high speed mode
[0]	RW	1'b0	MASTER_MODE	This bit controls whether the DW_apb_i2c master is enabled 1'b0: enable 1'b1: disable
0x04			**I2C Target Address Register**	
[31:13]	RO	19'b0	Reserved	Reserved
[12]	RW	1'b1	IC_10BITADDR_MASTER	This bit controls whether the DW_apb_i2c starts its transfers in 7-or 10-bit addressing mode when acting as a master 1'b0: 7-bit addressing 1'b1: 10-bit addressing
[11]	RW	1'b0	SPECIAL	This bit indicates whether software performs a General Call or START BYTE command 1'b0: ignore bit 10 GC_OR_START and use IC_TAR normally 1'b1: perform special I2C command as specified in GC_OR_START bit
[10]	RW	1'b0	GC_OR_START	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General

				Call or START byte command is to be performed by the DW_apb_i2c 1'b0: General Call Address 1'b1: START BYTE
[9:0]	RW	10'b0x055	IC_TAR	This is the target address for any master transaction
0x08			**IC_SAR**	
[31:10]	RO	22'b0	Reserved	Reserved
[9:0]	RW	10'b0x055	IC_SAR	The IC_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR[6:0] is used
0x0c			**IC_HS_MADDR**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b1	IC_HS_MAR	This bit field holds the value of the I2C HS mode master code
0x10			**IC_DATA_CMD**	
[31:11]	RO	21'b0	Reserved	Reserved
[10]	WO	1'b0	RESTART	This bit controls whether a RESTART is issued before the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1
[9]	WO	1'b0	STOP	This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1
[8]	WO	1'b0	CMD	This bit controls whether a read or a write is performed 1'b0: Read 1'b1: Write
[7:0]	RW	8'b0	DAT	This register contains the data to be transmitted or received on the I2C bus
0x14			**IC_SS_SCL_HCNT**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	IC_SS_SCL_HCNT	This register must be set before any I2C bus transaction can take place

				to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed
0x18			**IC_SS_SCL_LCNT**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	IC_SS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed
0x1c			**IC_FS_SCL_HCNT**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	IC_FS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed
0x20			**IC_SS_SCL_LCNT**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	IC_FS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low-period count for fast speed
0x24			**IC_HS_SCL_HCNT**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	IC_HS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period count for high speed
0x28			**IC_HS_SCL_LCNT**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b0	IC_HS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for high speed
0x2c			**IC_INTR_STAT**	
[31:12]	RO	20'b0	Reserved	Reserved

[11]	RO	1'b0	R_GEN_CALL	Set only when a General Call address is received and it is acknowledged
[10]	RO	1'b0	R_START_DET	Indicates whether a START or RESTART condition has occurred on the I2C interface
[9]	RO	1'b0	R_STOP_DET	Indicates whether a STOP condition has occurred on the I2C interface
[8]	RO	1'b0	R_ACTIVITY	This bit captures DW_apb_i2c activity and stays set until it is cleared
[7]	RO	1'b0	R_RX_DONE	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte
[6]	RO	1'b0	R_TX_ABRT	This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO
[5]	RO	1'b0	R_RD_REQ	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c
[4]	RO	1'b0	R_TX_EMPTY	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register
[3]	RO	1'b0	R_TX_OVER	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register
[2]	RO	1'b0	R_RX_FULL	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register
[1]	RO	1'b0	R_RX_OVER	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an

				additional byte is received from an external I2C device
[0]	RO	1'b0	R_RX_UNDER	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register
0x30			**IC_INTR_MASK**	
[31:12]	RW	20'b0	Reserved	Reserved
[11]	RW	1'b1	R_GEN_CALL	mask R_GEN_CALL interrupt status bits
[10]	RW	1'b0	R_START_DET	mask R_START_DET interrupt status bits
[9]	RW	1'b0	R_STOP_DET	mask R_STOP_DET interrupt status bits
[8]	RW	1'b0	R_ACTIVITY	mask R_ACTIVITY interrupt status bits
[7]	RW	1'b1	R_RX_DONE	mask R_RX_DONE interrupt status bits
[6]	RW	1'b1	R_TX_ABRT	mask R_TX_ABRT interrupt status bits
[5]	RW	1'b1	R_RD_REQ	mask R_RD_REQ interrupt status bits
[4]	RW	1'b1	R_TX_EMPTY	mask R_TX_EMPTY interrupt status bits
[3]	RW	1'b1	R_TX_OVER	mask R_TX_OVER interrupt status bits
[2]	RW	1'b1	R_RX_FULL	mask R_RX_FULL interrupt status bits
[1]	RW	1'b1	R_RX_OVER	mask R_RX_OVER interrupt status bits
[0]	RW	1'b1	R_RX_UNDER	mask R_RX_UNDER interrupt status bits
0x34			**IC_RAW_INTR_STAT**	
[31:12]	RO	20'b0	Reserved	Reserved
[11]	RO	1'b0	GEN_CALL	Set only when a General Call address is received and it is acknowledged
[10]	RO	1'b0	START_DET	Indicates whether a START or RESTART condition has occurred on the I2C interface

[9]	RO	1'b0	STOP_DET	Indicates whether a STOP condition has occurred on the I2C interface
[8]	RO	1'b0	ACTIVITY	This bit captures DW_apb_i2c activity and stays set until it is cleared
[7]	RO	1'b0	RX_DONE	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte
[6]	RO	1'b0	TX_ABRT	This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO
[5]	RO	1'b0	RD_REQ	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c
[4]	RO	1'b0	TX_EMPTY	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register
[3]	RO	1'b0	TX_OVER	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register
[2]	RO	1'b0	RX_FULL	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register
[1]	RO	1'b0	RX_OVER	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device
[0]	RO	1'b0	RX_UNDER	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register
0x38			**IC_RX_TL**	

[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0	RX_TL	Receive FIFO Threshold Level
0x3c			**IC_TX_TL**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0	TX_TL	Transmit FIFO Threshold Level
0x40			**IC_CLR_INTR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_INTR	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register
0x44			**IC_CLR_RX_UNDER**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_RX_UNDER	Read this register to clear the RX_UNDER interrupt (bit 0) of the IC_RAW_INTR_STAT register
0x48			**IC_CLR_RX_OVER**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_RX_OVER	Read this register to clear the RX_OVER interrupt (bit 1) of the IC_RAW_INTR_STAT register
0x4c			**IC_CLR_TX_OVER**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_TX_OVER	Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register
0x50			**IC_CLR_RD_REQ**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_RD_REQ	Read this register to clear the RD_REQ interrupt (bit 5) of the IC_RAW_INTR_STAT register
0x54			**IC_CLR_TX_ABRT**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_TX_ABRT	Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the IC_TX_ABRT_SOURCE register
0x58			**IC_CLR_RX_DONE**	
[31:1]	RO	31'b0	Reserved	Reserved

[0]	RO	1'b0	CLR_RX_DONE	Read this register to clear the RX_DONE interrupt (bit 7) of the IC_RAW_INTR_STAT register
0x5c			**IC_CLR_ACTIVITY**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_ACTIVITY	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set
0x60			**IC_CLR_STOP_DET**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_STOP_DET	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register
0x64			**IC_CLR_START_DET**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_START_DET	Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register
0x68			**IC_CLR_GEN_CALL**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b0	CLR_GEN_CALL	Read this register to clear the GEN_CALL interrupt (bit 11) of IC_RAW_INTR_STAT register
0x6c			**IC_ENABLE**	
[31:2]	RO	30'b0	Reserved	Reserved
[1]	RW	1'b0	ABORT	When set, the controller initiates the transfer abort 1'b0: ABORT not initiated or ABORT done 1'b1: ABORT operation in progress
[0]	RW	1'b0	ENABLE	Controls whether the DW_apb_i2c is enabled 1'b0: disable 1'b1: enable
0x70			**IC_STATUS**	
[31:7]	RO	25'b0	Reserved	Reserved

[6]	RO	1'b0	SLV_ACTIVITY	Slave FSM Activity Status 1'b0: in IDLE state 1'b1: not in IDLE state
[5]	RO	1'b0	MST_ACTIVITY	Master FSM Activity Status 1'b0: in IDLE state 1'b1: not in IDLE state
[4]	RO	1'b0	RFF	Receive FIFO Completely Full 1'b0: not full 1'b1: full
[3]	RO	1'b0	RFNE	Receive FIFO Not Empty 1'b0: empty 1'b1: not empty
[2]	RO	1'b1	TFE	Transmit FIFO Completely Empty 1'b0: not empty 1'b1: empty
[1]	RO	1'b1	TFNF	Transmit FIFO Not Full 1'b0: full 1'b1: not full
[0]	RO	1'b0	ACTIVITY	I2C Activity Status
0x74			**IC_TXFLR**	
[31:4]	RO	28'b0	Reserved	Reserved
[3:0]	RO	4'b0	TXFLR	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO
0x78			**IC_RXFLR**	
[31:4]	RO	28'b0	Reserved	Reserved
[3:0]	RO	4'b0	RXFLR	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO
0x7c			**IC_RXFLR**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RW	16'b1	IC_SDA_HOLD	Sets the required SDA hold time in units of ic_clk period
0x80			**IC_TX_ABRT_SOURCE**	
[31:24]	RO	8'b0	TX_FLUSH_CNT	This field preserves the TXFLR value prior to the last TX_ABRT event
[23:17]	RO	7'b0	Reserved	Reserved
[16]	RO	1'b0	ABRT_USER_ABRT	This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1])

[15]	RO	1'b0	ABRT_SLVRD_INTX	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 IC_DATA_CMD register
[14]	RO	1'b0	ABRT_SLV_ARBLOST	Slave lost the bus while transmitting data to a remote master
[13]	RO	1'b0	ABRT_SLVFLUSH_TXFIFO	Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO
[12]	RO	1'b0	ARB_LOST	Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration
[11]	RO	1'b0	ABRT_MASTER_DIS	User tries to initiate a Master operation with the Master mode disabled
[10]	RO	1'b0	ABRT_10B_RD_NORSTRT	The restart is disabled(IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode
[9]	RO	1'b0	ABRT_SBYTE_NORSTRT	The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte
[8]	RO	1'b0	ABRT_HS_NORSTRT	The restart is disabled(IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode
[7]	RO	1'b0	ABRT_SBYTE_ACKDET	Master has sent a START Byte and the START Byte was acknowledged (wrong behavior)
[6]	RO	1'b0	ABRT_HS_ACKDET	Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior)

[5]	RO	1'b0	ABRT_GCALL_READ	DW_apb_i2c in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1)
[4]	RO	1'b0	ABRT_GCALL_NOACK	DW_apb_i2c in master mode sent a General Call and no slave on the bus acknowledged the General Call
[3]	RO	1'b0	ABRT_TXDATA_NOACK	This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s)
[2]	RO	1'b0	ABRT_10ADDR2_NOACK	Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave
[1]	RO	1'b0	ABRT_10ADDR1_NOACK	Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave
[0]	RO	1'b0	ABRT_7B_ADDR_NOACK	Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave
0x84			**IC_SLV_DATA_NACK_ONLY**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	NACK	Generate NACK 1'b0: generate NACK after data byte received 1'b1: generate NACK/ACK normally
0x88			**IC_DMA_CR**	
[31:2]	RO	30'b0	Reserved	Reserved
[1]	RW	1'b0	TDMAE	Transmit DMA Enable 1'b0: disable 1'b1: enable
[0]	RW	1'b0	RDMAE	Receive DMA Enable 1'b0: disable 1'b1: enable
0x8c			**IC_DMA_TDLR**	

[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b0	DMATDL	Transmit Data Level
0x90			**IC_DMA_RDLR**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b0	DMARDL	Receive Data Level
0x94			**IC_SDA_SETUP**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0x64	SDA_SETUP	SDA Setup
0x98			**IC_ACK_GENERAL_CALL**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b1	ACK_GEN_CALL	ACK General Call
0x9c			**IC_ENABLE_STATUS**	
[31:3]	RO	31'b0	Reserved	Reserved
[2]	RO	1'b0	SLV_RX_DATA_LOST	Slave Received Data Lost
[1]	RO	1'b0	SLV_DISABLED_WHILE_BUSY	Slave Disabled While Busy (Transmit, Receive)
[0]	RO	1'b0	IC_EN	ic_en Status 1'b0: DW_apb_i2c is deemed completely inactive 1'b1: DW_apb_i2c is deemed to be in an enabled state
0xa0			**IC_FS_SPKLEN**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0xff	IC_FS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation
0xa4			**IC_HS_SPKLEN**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0xff	IC_HS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation
0xf4			**IC_COMP_PARAM_1**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:16]	RO	8'b0	TX_BUFFER_DEPTH	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter 8'b0x00: Reserved 8'b0x01: 2 8'b0x02: 3

				… 8'b0xff: 256
[15:8]	RO	8'b0	RX_BUFFER_DEPTH	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. For a description of this parameter 8'b0x00: Reserved 8'b0x01: 2 8'b0x02: 3 … 8'b0xff: 256
[7]	RO	1'b0	ADD_ENCODED_PARAMS	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. 1'b0: False 1'b1: True
[6]	RO	1'b0	HAS_DMA	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter 1'b0: False 1'b1: True
[5]	RO	1'b0	INTR_IO	The value of this register is derived from the IC_INTR_IO coreConsultant parameter 1'b0: Individual 1'b1: Combined
[4]	RO	1'b0	HC_COUNT_VALUES	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter 1'b0: False 1'b1: True
[3:2]	RO	2'b0	MAX_SPEED_MODE	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter 2'b00: Reserved 2'b01: Standard 2'b10: Fast 2'b11: High

[1:0]	RO	2'b0	APB_DATA_WIDTH	The value of this register is derived from the APB_DATA_WIDTH coreConsultant parameter 2'b00: 8 bits 2'b01: 16 bits 2'b10: 32 bits 2'b11: Reserved
0xf8			**IC_COMP_VERSION**	
[31:0]	RO	32'b0	IC_COMP_VERSION	Specific values for this register are described in the Releases Table in the AMBA 2 release notes
0xfc			**IC_COMP_TYPE**	
[31:0]	RO	32'b0	IC_COMP_TYPE	This assigned unique hex value is constant and is derived from the two ASCII letters “DW” followed by a 16-bit unsigned number

Table 19: I2C registers

3.9 I2S

I2S wrapper contains one I2S master and one I2S slave. They are logically exclusive. Only one block is alive at a time. The operation mode for master mode and slave mode is controlled by PERI_MASTER_SELECT Register in COM block.

bit	Reset value	Definition
3	0	I2S1 is master mode when set
2	0	I2S0 is master mode when set

Table 20: PERI_MASTER_SELECT Register bit definition
(base address = 0x4002_302C)

3.9.1 Register table

I2S registers are listed below.

Base address: 4000_9000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00			**IER**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	IEN	DW_apb_i2s enable
				1'b0: disable
				1'b1: enable
0x04			**IRER**	
[31:1]	RO	31'b0	Reserved	Reserved

[0]	RW	1'b0	RXEN	Receiver block enable 1'b0: disable 1'b1: enable
0x08			**ITER**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	TXEN	Transmitter block enable 1'b0: disable 1'b1: enable
0x0c			**CER**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	CLKEN	Clock generation enable/disable 1'b0: disable 1'b1: enable
0x10			**CCR**	
[31:5]	RO	27'b0	Reserved	Reserved
[4:3]	RW	2'b00	WSS	These bits are used to program the number of sclk cycles for which the world select line(ws_out) stays in the left or right sample mode 2'b00: 16 clock cycles 2'b01: 24 clock cycles 2'b10: 32 clock cycles
[2:0]	RW	3'b0	SCLKG	These bits are used to program the gating of sclk 3'b000: No clock gating 3'b001: Gate after 12 clock cycles 3'b010: Gate after 16 clock cycles 3'b011: Gate after 20 clock cycles 3'b100: Gate after 24 clock cycles
0x14			**RXFFR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	WO	1'b0	RXFFR	Receiver FIFO Reset;Receiver Block must be disabled prior to writing this bit
0x18			**TXFFR**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	WO	1'b0	TXFFR	Transimitter FIFO Reset;Transimitter Block must be disabled prior to writing this bit
0x20			**LRBR0**	

[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RO	16'b0	LRBR0	The left stereo data received serially from the receive channel input is read through this register
0x20			**LTHR0**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	WO	16'b0	LTHR0	The left stereo to be transmitted serially through the transmit channel output is written through this register
0x24			**RRBR0**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	RO	16'b0	RRBR0	The right stereo data received serially from the receive channel input is read through this register
0x24			**RTHR0**	
[31:16]	RO	16'b0	Reserved	Reserved
[15:0]	WO	16'b0	RTHR0	The right stereo to be transmitted serially through the transmit channel output is written through this register
0x28			**RER0**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b1	RXCHEN0	Receive channel enable 1'b0: disable 1'b1: enable
0x2c			**TER0**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b1	TXCHEN0	Transimit channel enable 1'b0: disable 1'b1: enable
0x30			**RCR0**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b010	WLEN	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 LRBR0(or RRBE0) register 3'b000: Ignore world length 3'b001: 12 bit resolution 3'b010: 16 bit resolution

				3'b011: 20 bit resolution
				3'b100: 24 bit resolution
				3'b101: 32 bit resolution
0x34			**TCR0**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b010	WLEN	These bits are used to program the data resolution of the transmitter and ensure the MSB of the data is transmitted first
				3'b000: Ignore world length
				3'b001: 12 bit resolution
				3'b010: 16 bit resolution
				3'b011: 20 bit resolution
				3'b100: 24 bit resolution
				3'b101: 32 bit resolution
0x38			**ISR0**	
[31:6]	RO	26'b0	Reserved	Reserved
[5]	RO	1'b0	TXFO	Status of Data Overrun interrupt for the TX channel
				1'b0: TX FIFO write valid
				1'b1: TX FIFO write overrun
[4]	RO	1'b1	TXFE	Status of Transmit Empty Trigger interrupt
				1'b0: trigger level not reached
				1'b1: trigger level reached
[3:2]	RO	2'b0	Reserved	Reserved
[1]	RO	1'b0	RXFO	Status of Data Overrun interrupt for the RX channel
				1'b0: RX FIFO write valid
				1'b1: RX FIFO write overrun
[0]	RO	1'b0	RXDA	Status of Receive Data Available interrupt
				1'b0: trigger level not reached
				1'b1: trigger level reached
0x3c			**IMR0**	
[31:6]	RO	26'b0	Reserved	Reserved
[5]	RW	1'b1	TXFOM	Masks TX FIFO Overrun interrupt
				1'b0: unmasks interrupt
				1'b1: masks interrupt
[4]	RW	1'b1	TXFEM	Masks TX FIFO Empty interrupt
				1'b0: unmasks interrupt

				1'b1: masks interrupt
[3:2]	RO	2'b0	Reserved	Reserved
[1]	RW	1'b1	RXFOM	Masks RX FIFO Overrun interrupt
				1'b0: unmasks interrupt
				1'b1: masks interrupt
[0]	RW	1'b1	RXDAM	Masks RX FIFO Data Available interrupt
				1'b0: unmasks interrupt
				1'b1: masks interrupt
0x40			**ROR0**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b1	RXCHO	Read this bit to clear the RX FIFO Data Overrun interrupt
				1'b0: RX FIFO write valid
				1'b1: RX FIFO write overrun
0x44			**TOR0**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RO	1'b1	TXCHO	Read this bit to clear the TX FIFO Data Overrun interrupt
				1'b0: TX FIFO write valid
				1'b1: TX FIFO write overrun
0x48			**RFCR0**	
[31:4]	RO	29'b0	Reserved	Reserved
[3:0]	RW	3'b011	RXCHDT	These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated
0x4c			**TFCR0**	
[31:4]	RO	29'b0	Reserved	Reserved
[3:0]	RW	3'b011	TXCHET	Transmit Channel Empty Trigger; These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached interrupt is generated
0x50			**RFF0**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	WO	1'b0	RXCHFR	Receive Channel FIFO Reset; Writing a 1 to this register flushes an individual RX FIFO, Rx channel or block must be disabled prior to writing this bit
0x54			**TFF0**	
[31:1]	RO	31'b0	Reserved	Reserved

[0]	WO	1'b0	TXCHFR	Transmit Channel FIFO Reset; Writing a 1 to this register flushes channel's TX FIFO,Tx channel or block must be disabled prior to writing this bit
0x1c0			**RXDMA**	
[31:0]	RO	32'b0	RXDMA	Receiver Block DMA Register. Used to cycle repeatedly through the enabled receive channels(from lowest numbered to highest),reading stereo data pairs
0x1c4			**RRXDMA**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	WO	1'b0	RRXDMA	Reset Receiver Block DMA Register. Writing a 1 to this self-clearing register reset the RXDMA register mid-cycle to point to the lowest enabled Receive channel
0x1c8			**TXDMA**	
[31:0]	RO	32'b0	TXDMA	Transmitter Block DMA Register. Used to cycle repeatedly through the enabled receive channels(from lowest numbered to highest),reading stereo data pairs
0x1cc			**RTXDMA**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	WO	1'b0	RTXDMA	Reset Transmitter Block DMA Register. Writing a 1 to this self-clearing register reset the TXDMA register mid-cycle to point to the lowest enabled Receive channel
0x1f0			**I2S_COMP_PARAM_2**	
[31:13]	RO	19'b0	Reserved	Reserved
[12:10]	RO	3'b0	I2S_RX_WORDSIZE_3	3'b000: 12 bit resolution
				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution
				3'b100: 32 bit resolution
				3'b101~111: Reserved
[9:7]	RO	3'b0	I2S_RX_WORDSIZE_2	3'b000: 12 bit resolution
				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution

				3'b100: 32 bit resolution
				3'b101~111: Reserved
[6]	RO	1'b0	Reserved	Reserved
[5:3]	RO	3'b0	I2S_RX_WORDSIZE_1	3'b000: 12 bit resolution
				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution
				3'b100: 32 bit resolution
				3'b101~111: Reserved
[2:0]	RO	3'b0	I2S_RX_WORDSIZE_0	3'b000: 12 bit resolution
				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution
				3'b100: 32 bit resolution
				3'b101~111: Reserved
0x1f4			**I2S_COMP_PARAM_1**	
[31:28]	RO	4'b0	Reserved	Reserved
[27:25]	RO	3'b0	I2S_TX_WORDSIZE_3	3'b000: 12 bit resolution
				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution
				3'b100: 32 bit resolution
				3'b101~111: Reserved
[24:22]	RO	3'b0	I2S_TX_WORDSIZE_2	3'b000: 12 bit resolution
				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution
				3'b100: 32 bit resolution
				3'b101~111: Reserved
[21:19]	RO	3'b0	I2S_TX_WORDSIZE_1	3'b000: 12 bit resolution
				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution
				3'b100: 32 bit resolution
				3'b101~111: Reserved
[18:16]	RO	3'b0	I2S_TX_WORDSIZE_0	3'b000: 12 bit resolution

				3'b001: 16 bit resolution
				3'b010: 20 bit resolution
				3'b011: 24 bit resolution
				3'b100: 32 bit resolution
				3'b101~111: Reserved
[15:11]	RO	5'b0	Reserved	Reserved
[10:9]	RO	2'b0	I2S_TX_CHANNELS	2'b00: 1 channel
				2'b01: 2 channels
				2'b10: 3 channels
				2'b11: 4 channels
[8:7]	RO	2'b0	I2S_RX_CHANNELS	2'b00: 1 channel
				2'b01: 2 channels
				2'b10: 3 channels
				2'b11: 4 channels
[6]	RO	1'b0	I2S_RECEIVER_BLOCK	1'b0: FALSE
				1'b1: TRUE
[5]	RO	1'b0	I2S_TRANSIMITER_BLOCK	1'b0: FALSE
				1'b1: TRUE
[4]	RO	1'b0	I2S_MODE_EN	1'b0: FALSE
				1'b1: TRUE
[3:2]	RO	2'b0	I2S_FIFO_DEPTH_GLONAL	2'b00: 2
				2'b01: 4
				2'b10: 8
				2'b11: 16
[1:0]	RO	2'b0	APB_DATA_WIDTH	2'b00: 8
				2'b01: 16
				2'b10: 32
				2'b11: Reserved
0x1f8			**I2S_COMP_VERSION**	
[31:28]	RO	32'b0	I2S_COMP_VERSION	Specific values for this register are described in the Releases Table in the AMBA 2 release notes
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b0	DMATDL	Transmit Data Level
0x90			**IC_DMA_RDLR**	
[31:3]	RO	29'b0	Reserved	Reserved
[2:0]	RW	3'b0	DMARDL	Receive Data Level
0x94			**IC_SDA_SETUP**	
[31:8]	RO	24'b0	Reserved	Reserved

[7:0]	RW	8'b0x64	SDA_SETUP	SDA Setup
0x98			**IC_ACK_GENERAL_CALL**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b1	ACK_GEN_CALL	ACK General Call
0x9c			**IC_ENABLE_STATUS**	
[31:3]	RO	31'b0	Reserved	Reserved
[2]	RO	1'b0	SLV_RX_DATA_LOST	Slave Received Data Lost
[1]	RO	1'b0	SLV_DISABLED_WHILE_BUSY	Slave Disabled While Busy (Transmit, Receive)
[0]	RO	1'b0	IC_EN	ic_en Status 1'b0: DW_apb_i2c is deemed completely inactive 1'b1: DW_apb_i2c is deemed to be in an enabled state
0xa0			**IC_FS_SPKLEN**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0xff	IC_FS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation
0xa4			**IC_HS_SPKLEN**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0xff	IC_HS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation
0xf4			**IC_COMP_PARAM_1**	
[31:24]	RO	8'b0	Reserved	Reserved
[23:16]	RO	8'b0	TX_BUFFER_DEPTH	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter 8'b0x00: Reserved 8'b0x01: 2 8'b0x02: 3 ... 8'b0xff: 256
[15:8]	RO	8'b0	RX_BUFFER_DEPTH	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. For a description of this parameter 8'b0x00: Reserved

				8'b0x01: 2
				8'b0x02: 3
				…
				8'b0xff: 256
[7]	RO	1'b0	ADD_ENCODED_PARAMS	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter.
				1'b0: False
				1'b1: True
[6]	RO	1'b0	HAS_DMA	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter
				1'b0: False
				1'b1: True
[5]	RO	1'b0	INTR_IO	The value of this register is derived from the IC_INTR_IO coreConsultant parameter
				1'b0: Individual
				1'b1: Combined
[4]	RO	1'b0	HC_COUNT_VALUES	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter
				1'b0: False
				1'b1: True
[3:2]	RO	2'b0	MAX_SPEED_MODE	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter
				2'b00: Reserved
				2'b01: Standard
				2'b10: Fast
				2'b11: High
[1:0]	RO	2'b0	APB_DATA_WIDTH	The value of this register is derived from the APB_DATA_WIDTH coreConsultant parameter
				2'b00: 8 bits
				2'b01: 16 bits
				2'b10: 32 bits
				2'b11: Reserved
0xf8			**IC_COMP_VERSION**	

[31:0]	RO	32'b0	IC_COMP_VERSION	Specific values for this register are described in the Releases Table in the AMBA 2 release notes
0xfc			**IC_COMP_TYPE**	
[31:0]	RO	32'b0	IC_COMP_TYPE	This assigned unique hex value is constant and is derived from the two ASCII letters “DW” followed by a 16-bit unsigned number

Table 21: I2S registers

3.10 UART (UART)

The Universal Asynchronous Receiver/Transmitter offers fast, full-duplex, asynchronous serial communication with built-in flow control (CTS, RTS) support in HW up to 1Mbps baud. Parity checking and generation for the 9th data bit are supported.

The GPIOs used for each UART interface line can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pin out and enables efficient use of board space and signal routing.

3.10.1 Register table

UART registers are listed below.

Base address: 4000_4000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00			**RBR(Receive Buffer Register)**	**LCR[7] bit = 0**
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RO	8'b0	Receive Buffer Register	LSR[0] bit = 1,The data in this register is valid
0x00			**THR(Transmit Holding Register)**	**LCR[7] bit = 0**
[31:8]	WO	24'b0	Reserved	Reserved
[7:0]	WO	8'b0	Transmit Holding Register	LSR[5] bit = 1,The data should only be written to the THR
0x00			**DLL(Divisor Latch Low)**	**1.When UART_16550 == YES, Then LCR[7] bit = 1**
				2.When UART_16550 == NO, Then LCR[7] bit = 1,USR[0] = 0
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0	Divisor Latch (low)	baud rate = (serial clock freq) / (16 * divisor)

0x04			**DLH(Divisor Latch High)**	**1.When UART_16550 == YES, Then LCR[7] bit = 1** **2.When UART_16550 == NO, Then LCR[7] bit = 1,USR[0] = 0**
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0	Divisor Latch (high)	baud rate = (serial clock freq) / (16 * divisor)
0x04			**IER(Interrupt Enable Register)**	**LCR[7] bit = 0**
[31:8]	RO	24'b0	Reserved	Reserved
[7]	RW	1'b0	PTIME	This is used to enable/disable the generation of THRE Interrupt 1'b0: disable 1'b1: enable
[6:4]	RO	3'b0	Reserved	Reserved
[3]	RW	1'b0	EDSSI	This is used to enable/disable the generation of Modem Status Interrupt 1'b0: disable 1'b1: enable
[2]	RW	1'b0	ELSI	This is used to enable/disable the generation of Receiver Line Status Interrupt 1'b0: disable 1'b1: enable
[1]	RW	1'b0	ETBEI	This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt 1'b0: disable 1'b1: enable
[0]	RW	1'b0	ERBFI	This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled) 1'b0: disable 1'b1: enable
0x08			**IIR(Interrupt Identity Register)**	
[31:8]	RO	24'b0	Reserved	Reserved

[7:6]	RO	2'b0	FIFOSE	This is used to indicate whether the FIFOs are enabled or disabled
				2'b00: disable
				2'b11: enable
[5:4]	RO	2'b0	Reserved	Reserved
[3:0]	RO	4'b0001	IID	This is used to indicates the highest priority pending interrupt which can be one of the following types
				4'b0000: modem status
				4'b0001: no interrupt pending
				4'b0010: THR empty
				4'b0100: received data available
				4'b0110: receiver line status
				4'b0111: busy detect
				4'b1100: character timeout
0x08			**FCR(FIFO Control Register)**	**FIFO_MODE != NONE**
[31:8]	RO	24'b0	Reserved	Reserved
[7:6]	WO	2'b0	RT	This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated, The following trigger levels are supported:
				2'b00: 1 character in the FIFO
				2'b01: FIFO ¼ full
				2'b10: FIFO ½ full
				2'b11: FIFO 2 less than full
[5:4]	WO	2'b0	TET	This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active, The following trigger levels are supported:
				2'b00: FIFO empty
				2'b01: 2 characters in the FIFO
				2'b10: FIFO ¼ full
				2'b11: FIFO ½ full
[3]	WO	1'b0	DMAM	This determines the DMA signalling mode
				1'b0: mode 0

				1'b1: mode 1
[2]	WO	1'b0	XFIFOR	This resets the control portion of the transmit FIFO and treats the FIFO as empty
[1]	WO	1'b0	RFIFOR	This resets the control portion of the receive FIFO and treats the FIFO as empty
[0]	WO	1'b0	FIFOE	This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs
				1'b0: disable
				1'b1: enable
0x0C			**LCR(Line Control Register)**	
[31:8]	RO	24'b0	Reserved	Reserved
[7]	RW	1'b0	DLAB	USR[0]=0,the bit is writeable; This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART
				1'b0: disable
				1'b1: enable
[6]	RW	1'b0	Break	This is used to cause a break condition to be transmitted to the receiving device
[5]	RO	1'b0	Reserved	Reserved
[4]	RW	1'b0	EPS	USR[0]=0,the bit is writeable; This is used to select between even and odd parity, when parity is enabled (PEN set to one)
				1'b0: an odd number of logic 1s is transmitted or checked
				1'b1: an even number of logic 1s is transmitted or checked
[3]	RW	1'b0	PEN	USR[0]=0,the bit is writeable; enable and disable parity generation and detection in transmitted and received serial character respectively
				1'b0: disable
				1'b1: enable

[2]	RW	1'b0	STOP	USR[0]=0,the bit is writeable; select the number of stop bits per character that the peripheral transmits and receives 1'b0: 1 stop bit 1'b1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit
[1:0]	RW	2'b0	DLS	USR[0]=0,the bit is writeable; This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 2'b00: 5 bits 2'b01: 6 bits 2'b10: 7 bits 2'b11: 8 bits
0x10			**MCR(Modem Control Register)**	
[31:7]	RO	25'b0	Reserved	Reserved
[6]	RW	1'b0	SIRE	SIR_MODE == Enabled, the bit is writeable; enable/disable the IrDA SIR Mode 1'b0: disable 1'b1: enable
[5]	RW	1'b0	AFCE	AFCE_MODE == Enabled, the bit is writeable; enable/disable the Auto Flow Control 1'b0: disable 1'b1: enable
[4]	RW	1'b0	LoopBack	This is used to put the UART into a diagnostic mode for test purposes
[3]	RW	1'b0	OUT2	This is used to directly control the user-designated Output2 (out2_n) output 1'b0: de-asserted (logic 1) 1'b1: asserted (logic 0)
[2]	RW	1'b0	OUT1	This is used to directly control the user-designated Output1 (out1_n) output

				1'b0: de-asserted (logic 1) 1'b1: asserted (logic 0)
[1]	RW	1'b0	RTS	Request to Send. This is used to directly control the Request to Send (rts_n) output 1'b0: de-asserted (logic 1) 1'b1: asserted (logic 0)
[0]	RW	1'b0	DTR	This is used to directly control the Data Terminal Ready (dtr_n) output 1'b0: de-asserted (logic 1) 1'b1: asserted (logic 0)
0x14			**LSR(Line Status Register)**	
[31:8]	RO	24'b0	Reserved	Reserved
[7]	RO	1'b0	RFE	FIFO_MODE != NONE and FCR[0] = 1,the bit is relevant; This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO 1'b0: no error 1'b1: error
[6]	RO	1'b1	TEMT	Transmitter Empty bit; FIFO_MODE != NONE and FCR[0] = 1,this bit is set whenever the Transmitter Shift Register and the FIFO are both empty FIFO_MODE == NONE and FCR[0] = 0,this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty
[5]	RO	1'b1	THRE	Transmit Holding Register Empty bit
[4]	RO	1'b0	BI	This is used to indicate the detection of a break sequence on the serial input data.
[3]	RO	1'b0	FE	This is used to indicate the occurrence of a framing error in the receiver 1'b0: no framing error

				1'b1: framing error
[2]	RO	1'b0	PE	This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set
				1'b0: no parity error
				1'b1: parity error
[1]	RO	1'b0	OE	This is used to indicate the occurrence of an overrun error
				1'b0: no overrun error
				1'b1: overrun error
[0]	RO	1'b0	DR	This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO
				1'b0: no data ready
				1'b1: data ready
0x18			**MSR(Modem Status Register)**	
[31:8]	RO	24'b0	Reserved	Reserved
[7]	RO	1'b0	DCD	This is used to indicate the current state of the modem control line dcd_n
				1'b0: de-asserted (logic 1)
				1'b1: asserted (logic 0)
[6]	RO	1'b0	RI	This is used to indicate the current state of the modem control line ri_n
				1'b0: de-asserted (logic 1)
				1'b1: asserted (logic 0)
[5]	RO	1'b0	DSR	This is used to indicate the current state of the modem control line dsr_n
				1'b0: de-asserted (logic 1)
				1'b1: asserted (logic 0)
[4]	RO	1'b0	CTS	This is used to indicate the current state of the modem control line cts_n
				1'b0: de-asserted (logic 1)
				1'b1: asserted (logic 0)

[3]	RO	1'b0	DDCD	This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read
				1'b0: no change on dcd_n since last read of MSR
				1'b1: change on dcd_n since last read of MSR
[2]	RO	1'b0	TERI	This is used to indicate that a change on the input ri_n has occurred since the last time the MSR was read
				1'b0: no change on ri_n since last read of MSR
				1'b1: change on ri_n since last read of MSR
[1]	RO	1'b0	DDSR	This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read
				1'b0: no change on dsr_n since last read of MSR
				1'b1: change on dsr_n since last read of MSR
[0]	RO	1'b0	DCTS	This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read
				1'b0: no change on ctsdsr_n since last read of MSR
				1'b1: change on ctsdsr_n since last read of MSR
0x1C			**SCR(Scratchpad Register)**	
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0	Scratchpad Register	This register is for programmers to use as a temporary storage space
0x20			**LPDLL(Low Power Divisor Latch Low Register)**	**SIR_LP_RX == Yes**
[31:8]	RO	24'b0	Reserved	Reserved

[7:0]	RW	8'b0	LPDLL	This register makes up the lower 8-bits of a 16-bit,this register that contains the baud rate divisor for the UART
0x24			**LPDLH(Low Power Divisor Latch High Register)**	**SIR_LP_RX == Yes**
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RW	8'b0	LPDLH	This register makes up the upper 8-bits of a 16-bit,this register that contains the baud rate divisor for the UART
0x30~0x6c			**SRBR(Shadow Receive Buffer Register)**	**SHADOW == YES and LCR[7] bit = 0**
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RO	8'b0	Shadow Receive Buffer Register	This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master
0x30~0x6c			**STHR(Shadow Transmit Holding Register)**	**SHADOW == YES and LCR[7] bit = 0**
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	WO	8'b0	Shadow Transmit Holding Register	This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master
0x70			**FAR(FIFO Access Register)**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	FIFO Access Register	Writes have no effect when FIFO_ACCESS == No, always readable, This register is use to enable a FIFO access mode for testing 1’b0: disable 1’b1: enable
0x74			**TFR(Transmit FIFO Read)**	**FIFO_ACCESS == YES**
[31:8]	RO	24'b0	Reserved	Reserved
[7:0]	RO	8'b0	Transmit FIFO Read	FAR[0] = 1,the bit is valid;Reading this register gives the data at the

				top of the transmit FIFO or the data in the THR
0x78			**RFW(Receive FIFO Write)**	**FIFO_ACCESS == YES**
[31:10]	RO	22'b0	Reserved	Reserved
[9]	WO	1'b0	RFFE	FAR[0] = 1,the bit is valid; This bit is used to write framing error detection information to the receive FIFO or the RBR
[8]	WO	1'b0	RFPE	FAR[0] = 1,the bit is valid; This bit is used to write parity error detection information to the receive FIFO or the RBR
[7:0]	WO	8'b0	RFWD	FAR[0] = 1,the bit is valid; This bit of the data that is written to the RFWD is pushed into the receive FIFO or the RBR
0x7C			**USR(UART Status Register)**	
[31:5]	RO	27'b0	Reserved	Reserved
[4]	RO	1'b0	RFF	FIFO_STAT == YES, the bit is valid; This is used to indicate that the receive FIFO is completely full
				1'b0: not full
				1'b1: full
[3]	RO	1'b0	RFNE	FIFO_STAT == YES, the bit is valid; This is used to indicate that the receive FIFO contains one or more entries
				1'b0: empty
				1'b1: not empty
[2]	RO	1'b1	TFE	FIFO_STAT == YES, the bit is valid; This is used to indicate that the transmit FIFO is completely empty
				1'b0: not empty
				1'b1: empty
[1]	RO	1'b1	TFNF	FIFO_STAT == YES,the bit is vlid;This is used to indicate that the transmit FIFO in not full
				1'b0: full
				1'b1: not full

[0]	RO	1'b0	BUSY	This is indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive
				1'b0: idle or inactive
				1'b1: busy (actively transferring data)
0x80			**TFL(Transmit FIFO Level)**	**FIFO_STAT == YES;FIFO_ADDR_WIDTH=4**
[31:5]	RO	27'b0	Reserved	Reserved
[4:0]	RO	5'b0	Transmit FIFO Level	This is indicates the number of data entries in the transmit FIFO
0x84			**RFL(Receive FIFO Level)**	**FIFO_STAT == YES;FIFO_ADDR_WIDTH=4**
[31:5]	RO	27'b0	Reserved	Reserved
[4:0]	RO	5'b0	Receive FIFO Level	This is indicates the number of data entries in the receive FIFO
0x88			**SRR(Software Reset Register)**	**SHADOW == YES**
[31:3]	RO	29'b0	Reserved	Reserved
[2]	WO	1'b0	XFR	FIFO_MODE == None, the written have no effect; XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit(FCR[2])
[1]	WO	1'b0	RFR	FIFO_MODE == None, the written have no effect; RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit(FCR[1])
[0]	WO	1'b0	UR	This asynchronously resets the DW_apb_uart and synchronously removes the reset assertion
0x8C			**SRTS(Shadow Request to Send)**	**SHADOW == YES**
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Shadow Request to Send	This is a shadow register for the RTS bit(MCR[1])
0x90			**SBCR(Shadow Break Control Register)**	**SHADOW == YES**
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Shadow Break Control Register	This is a shadow register for the Break bit(LCR[6])

0x94			**SDMAM(Shadow DMA Mode)**	**FIFO_MODE != None and SHADOW == YES**
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Shadow DMA Mode	This is a shadow register for the DMA mode bit(FCR[3])
				1'b0: mode 0
				1'b1: mode 1
0x98			**SFE(Shadow FIFO Enable)**	**FIFO_MODE != None and SHADOW == YES**
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Shadow FIFO Enable	This is a shadow register for the FIFO enable bit(FCR[0])
				1'b0: disable
				1'b1: enable
0x9C			**SRT(Shadow RCVR Trigger)**	**FIFO_MODE != None and SHADOW == YES**
[31:2]	RO	30'b0	Reserved	Reserved
[1:0]	RW	2'b0	Shadow RCVR Trigger	This is a shadow register for the RCVR trigger bits(FCR[7:6])
				2'b00: 1 character in the FIFO
				2'b01: FIFO ¼ full
				2'b10: FIFO ½ full
				2'b11: FIFO 2 less than full
0xA0			**STET(Shadow TX Empty Trigger)**	**FIFO_MODE != None and THRE_MODE_USER == Enabled and SHADOW == YES**
[31:2]	RO	30'b0	Reserved	Reserved
[1:0]	RW	2'b0	Shadow TX Empty Trigger	THRE_MODE_USER == Disabled, the written have no effect; This is a shadow register for the TX empty trigger bits (FCR[5:4])
				2'b00: FIFO empty
				2'b01: 2 characters in the FIFO
				2'b10: FIFO ¼ full
				2'b11: FIFO ½ full
0xA4			**HTX(Halt TX)**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	RW	1'b0	Halt TX	FIFO_MODE == None, the written have no effect; This register is use to halt transmissions for testing

				1'b0: disable
				1'b1: enable
0xA8			**DMASA(DMA Software Acknowledge)**	
[31:1]	RO	31'b0	Reserved	Reserved
[0]	WO	1'b0	DMA Software Acknowledge	DMA_EXTRA == No, the written have no effect; This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition
0xF4			**CPR(Component Parameter Register)**	**UART_ADD_ENCODED_PARAMS == YES**
[31:24]	RO	8'b0	Reserved	Reserved
[23:16]	RO	8'b0	FIFO_MODE	8'b0x00: 0
				8'b0x01: 16
				8'b0x02: 32
				……
				8'b0x80: 2048
				8'b0x81- 0xff: reserved
[15:14]	RO	2'b0	Reserved	Reserved
[13]	RO	1'b0	DMA_EXTRA	1'b0: FALSE
				1'b1: TRUE
[12]	RO	1'b0	UART_ADD_ENCODED_PARAMS	1'b0: FALSE
				1'b1: TRUE
[11]	RO	1'b0	SHADOW	1'b0: FALSE
				1'b1: TRUE
[10]	RO	1'b0	FIFO_STAT	1'b0: FALSE
				1'b1: TRUE
[9]	RO	1'b0	FIFO_ACCESS	1'b0: FALSE
				1'b1: TRUE
[8]	RO	1'b0	ADDITIONAL_FEAT	1'b0: FALSE
				1'b1: TRUE
[7]	RO	1'b0	SIR_LP_MODE	1'b0: FALSE
				1'b1: TRUE
[6]	RO	1'b0	SIR_MODE	1'b0: FALSE
				1'b1: TRUE
[5]	RO	1'b0	THRE_MODE	1'b0: FALSE
				1'b1: TRUE

[4]	RO	1'b0	AFCE_MODE	1'b0: FALSE
				1'b1: TRUE
[3:2]	RO	2'b0	Reserved	Reserved
[1:0]	RO	2'b0	APB_DATA_WIDTH	2'b00: 8 bits
				2'b01: 16 bits
				2'b10: 32 bits
				2'b11: reserved
0xF8			**UCV(UART Component Version)**	**ADDITIONAL_FEATURES == YES**
[31:0]	RO	32'b0	UART Component Version	ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*
0xFC			**CTR(Component Type Register)**	**ADDITIONAL_FEATURES == YES**
[31:0]	RO	32'b0x44570110	Peripheral ID	This register contains the peripherals identification code

Table 22: UART registers

3.11 Pulse Width Modulation (PWM)

QMS7926 supports 6 channels of Pulse Width Modulation (PWM) outputs. PWM outputs generate waveforms with variable duty cycle or pulse width programmed by registers. And each of the 6 PWM outputs can be individually programmed. Their duty cycles are controlled by programming individual counters associated with each channel.

The master clock is 16MHz. For each PWM outputs, first there is a prescaler (pre-divider) with division ratio of 2 to 128 (only 2^N division ratios are supported), followed by another 16bit counter with programmable max count, denoted as top_count. When the 16bit counter counts from 0 to top_count, it resets back to 0. So the frequency of the PWM is given by:

Freq_PWM = 16MHz / (N_prescaler * N_top_count);

A threshold counter number can be programmed, when the 16bit counter reaches the threshold, PWM output toggles. So the duty cycle is:

Duty_cycle_PWM = N_threshold/N_top_count;

The polarity of the PWM can also be programmed, which indicates output 1 or 0 when counter is below/above the threshold. A PWM waveform vs counter values are illustrated in the following Figure 13, where the polarity is positive. Also in this case the counter ramps up and then resets, we call it “up mode”.

There is also a “up and down mode”, where the counter ramps up to count_top and then ramps down, instead of reset.

As discussed above, the key register bits for one PWM channel are: 16bit top_count, 16bit threshold count, 3bit prescaler count, PWM polarity, PWM mode (up or up/down), PWM enable, and PWM load enable (load new settings). All 6 PWM channels can be individually programmed by registers with addresses from 0x4000_E004 to 0x4000_E044. In addition, one should enable registers 0x4000_E000<0><4> to allow all PWM channels can be programmed. For details please refer to documents of QMS7926 register tables.

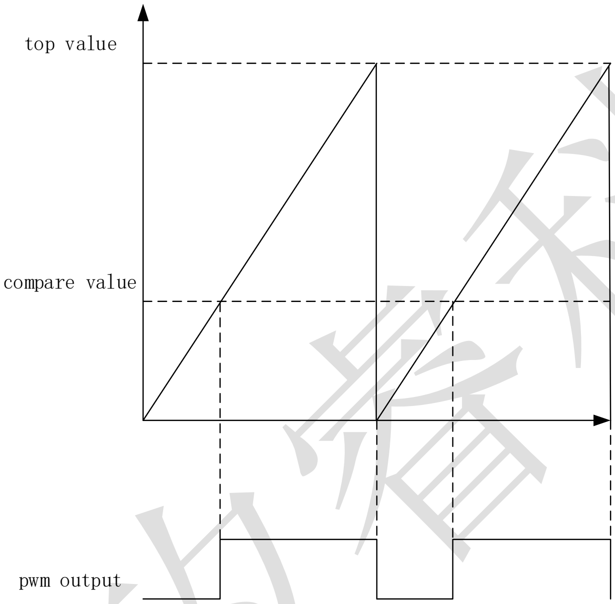


Figure 10: PWM operation

3.11.1 Register table

PWM related registers are listed below.

Base address: 4000_E000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00			**PWMEN**	**pwm enable**
[31:18]	RO	14'b0	reserved	Reserved
[17]	RW	1'b0	pwm_load_45	load parameter of PWM channel 4, 5. need to be conjunction with setting bit16 of PWMxCTL0 registers.
				1'b0: no load
				1'b1: load
[16]	RW	1'b0	pwm_en_45	enable of PWM channel 4, 5. need to be conjunction with setting bit0 of PWMxCTL0 registers.
				1'b0: disable

				1'b1: enable
[15]	RW	1'b0	pwm_load_23	load parameter of PWM channel 2, 3. need to be conjunction with setting bit16 of PWMxCTL0 registers.
				1'b0: no load
				1'b1: load
[14]	RW	1'b0	pwm_en_23	enable of PWM channel 2, 3. need to be conjunction with setting bit0 of PWMxCTL0 registers.
				1'b0: disable
				1'b1: enable
[13]	RW	1'b0	pwm_load_01	load parameter of PWM channel 0, 1. need to be conjunction with setting bit16 of PWMxCTL0 registers.
				1'b0: no load
				1'b1: load
[12]	RW	1'b0	pwm_en_01	enable of PWM channel 0, 1. need to be conjunction with setting bit0 of PWMxCTL0 registers.
				1'b0: disable
				1'b1: enable
[11]	RW	1'b0	pwm_load_345	load parameter of PWM channel 3, 4, 5. need to be conjunction with setting bit16 of PWMxCTL0 registers.
				1'b0: no load
				1'b1: load
[10]	RW	1'b0	pwm_en_345	enable of PWM channel 3, 4, 5. need to be conjunction with setting bit0 of PWMxCTL0 registers.
				1'b0: disable
				1'b1: enable
[9]	RW	1'b0	pwm_load_012	load parameter of PWM channel 0, 1, 2. need to be conjunction with setting bit16 of PWMxCTL0 registers.
				1'b0: no load
				1'b1: load
[8]	RW	1'b0	pwm_en_012	enable of PWM channel 0, 1, 2. need to be conjunction with setting bit0 of PWMxCTL0 registers.

				1'b0: disable
				1'b1: enable
[7:5]	RO	3'b0	reserved	Reserved
[4]	RW	1'b0	pwm_load_all	load parameter of all six PWM channels. need to be conjunction with setting bit16 of PWMxCTL0 registers.
				1'b0: no load
				1'b1: load
[3:1]	RO	3'b0	reserved	Reserved
[0]	RW	1'b0	pwm_en_all	enable of all six PWM channels. need to be conjunction with setting bit0 of PWMxCTL0 registers.
				1'b0: disable
				1'b1: enable
0x04			**PWM0CTL0**	**pwm channel 0 contrl reigister**
[31]	RW	1'b0	pwm0_load_instant	instant load parameter of PWM channel 0.
				1'b0: no load
				1'b1: instant load
[30:17]	RO	14'b0	reserved	Reserved
[16]	RW	1'b0	pwm0_load	load parameter of PWM channel 0.
				1'b0: no load
				1'b1: load
[15]	RO	1'b0	reserved	Reserved
[14:12]	RW	3'b0	pwm0_clk_div	clock prescaler of PWM channel 0.
				3'b000: pwm_clk is divided by 1 for count clock
				3'b001: pwm_clk is divided by 2 for count clock
				3'b010: pwm_clk is divided by 4 for count clock
				3'b011: pwm_clk is divided by 8 for count clock
				3'b100: pwm_clk is divided by 16 for count clock
				3'b101: pwm_clk is divided by 32 for count clock
				3'b110: pwm_clk is divided by 64 for count clock
				3'b111: pwm_clk is divided by 128 for count clock
[11:9]	RO	3'b0	reserved	Reserved
[8]	RW	1'b0	pwm0_cnt_mode	count mode of PWM channel 0.
				1'b0: up mode
				1'b1: up and down mode
[7:5]	RO	3'b0	reserved	Reserved
[4]	RW	1'b0	pwm0_polarity	output polarity setting of PWM channel 0.

				1'b0: rising edge. Second edge within the PWM period is rising
				1'b1: falling edge. Second edge within the PWM period is falling
[3:1]	RO	3'b0	reserved	Reserved
[0]	RW	1'b0	pwm0_en	enable of PWM channel 0.
				1'b0: disable
				1'b1: enable
0x08			**PWM0CTL1**	**pwm channel 0 conter value setting**
[31:16]	RW	16'b0	pwm0_cmp_val	the compare value of PWM channel 0
[15:0]	RW	16'b0	pwm0_cnt_top	the counter top value of PWM channel 0
0x10			**PWM1CTL0**	**pwm channel 1 contrl reigister**
[31]	RW	1'b0	pwm1_load_instant	instant load parameter of PWM channel 1.
				1'b0: no load
				1'b1: instant load
[30:17]	RO	14'b0	reserved	Reserved
[16]	RW	1'b0	pwm1_load	load parameter of PWM channel 1.
				1'b0: no load
				1'b1: load
[15]	RO	1'b0	reserved	Reserved
[14:12]	RW	3'b0	pwm1_clk_div	clock prescaler of PWM channel 1.
				3'b000: pwm_clk is divided by 1 for count clock
				3'b001: pwm_clk is divided by 2 for count clock
				3'b010: pwm_clk is divided by 4 for count clock
				3'b011: pwm_clk is divided by 8 for count clock
				3'b100: pwm_clk is divided by 16 for count clock
				3'b101: pwm_clk is divided by 32 for count clock
				3'b110: pwm_clk is divided by 64 for count clock
				3'b111: pwm_clk is divided by 128 for count clock
[11:9]	RO	3'b0	reserved	Reserved
[8]	RW	1'b0	pwm1_cnt_mode	count mode of PWM channel 1.
				1'b0: up mode
				1'b1: up and down mode
[7:5]	RO	3'b0	reserved	Reserved
[4]	RW	1'b0	pwm1_polarity	output polarity setting of PWM channel 1.
				1'b0: rising edge. Second edge within the PWM period is rising

				1'b1: falling edge. Second edge within the PWM period is falling
[3:1]	RO	3'b0	reserved	Reserved
[0]	RW	1'b0	pwm1_en	enable of PWM channel 1.
				1'b0: disable
				1'b1: enable
0x14			**PWM1CTL1**	**pwm channel 1 conter value setting**
[31:16]	RW	16'b0	pwm1_cmp_val	the compare value of PWM channel 1
[15:0]	RW	16'b0	pwm1_cnt_top	the counter top value of PWM channel 1
0x1C			**PWM2CTL0**	**pwm channel 2 contrl reigister**
[31]	RW	1'b0	pwm2_load_instant	instant load parameter of PWM channel 2.
				1'b0: no load
				1'b1: instant load
[30:17]	RO	14'b0	reserved	Reserved
[16]	RW	1'b0	pwm2_load	load parameter of PWM channel 2.
				1'b0: no load
				1'b1: load
[15]	RO	1'b0	reserved	Reserved
[14:12]	RW	3'b0	pwm2_clk_div	clock prescaler of PWM channel 2.
				3'b000: pwm_clk is divided by 1 for count clock
				3'b001: pwm_clk is divided by 2 for count clock
				3'b010: pwm_clk is divided by 4 for count clock
				3'b011: pwm_clk is divided by 8 for count clock
				3'b100: pwm_clk is divided by 16 for count clock
				3'b101: pwm_clk is divided by 32 for count clock
				3'b110: pwm_clk is divided by 64 for count clock
				3'b111: pwm_clk is divided by 128 for count clock
[11:9]	RO	3'b0	reserved	Reserved
[8]	RW	1'b0	pwm2_cnt_mode	count mode of PWM channel 2.
				1'b0: up mode
				1'b1: up and down mode
[7:5]	RO	3'b0	reserved	Reserved
[4]	RW	1'b0	pwm2_polarity	output polarity setting of PWM channel 2.
				1'b0: rising edge. Second edge within the PWM period is rising
				1'b1: falling edge. Second edge within the PWM period is falling
[3:1]	RO	3'b0	reserved	Reserved
[0]	RW	1'b0	pwm2_en	enable of PWM channel 2.

				1'b0: disable
				1'b1: enable
0x20			**PWM2CTL1**	**pwm channel 2 conter value setting**
[31:16]	RW	16'b0	pwm2_cmp_val	the compare value of PWM channel 2
[15:0]	RW	16'b0	pwm2_cnt_top	the counter top value of PWM channel 2
0x28			**PWM3CTL0**	**pwm channel 3 contrl reigister**
[31]	RW	1'b0	pwm3_load_instant	instant load parameter of PWM channel 3.
				1'b0: no load
				1'b1: instant load
[30:17]	RO	14'b0	reserved	Reserved
[16]	RW	1'b0	pwm3_load	load parameter of PWM channel 3.
				1'b0: no load
				1'b1: load
[15]	RO	1'b0	reserved	Reserved
[14:12]	RW	3'b0	pwm3_clk_div	clock prescaler of PWM channel 3.
				3'b000: pwm_clk is divided by 1 for count clock
				3'b001: pwm_clk is divided by 2 for count clock
				3'b010: pwm_clk is divided by 4 for count clock
				3'b011: pwm_clk is divided by 8 for count clock
				3'b100: pwm_clk is divided by 16 for count clock
				3'b101: pwm_clk is divided by 32 for count clock
				3'b110: pwm_clk is divided by 64 for count clock
				3'b111: pwm_clk is divided by 128 for count clock
[11:9]	RO	3'b0	reserved	Reserved
[8]	RW	1'b0	pwm3_cnt_mode	count mode of PWM channel 3.
				1'b0: up mode
				1'b1: up and down mode
[7:5]	RO	3'b0	reserved	Reserved
[4]	RW	1'b0	pwm3_polarity	output polarity setting of PWM channel 3.
				1'b0: rising edge. Second edge within the PWM period is rising
				1'b1: falling edge. Second edge within the PWM period is falling
[3:1]	RO	3'b0	reserved	Reserved
[0]	RW	1'b0	pwm3_en	enable of PWM channel 3.
				1'b0: disable
				1'b1: enable
0x2C			**PWM3CTL1**	**pwm channel 0 conter value setting**
[31:16]	RW	16'b0	pwm3_cmp_val	the compare value of PWM channel 3

[15:0]	RW	16'b0	pwm3_cnt_top	the counter top value of PWM channel 3
0x34			**PWM4CTL0**	**pwm channel 4 contrl reigister**
[31]	RW	1'b0	pwm4_load_instant	instant load parameter of PWM channel 4.
				1'b0: no load
				1'b1: instant load
[30:17]	RO	14'b0	reserved	Reserved
[16]	RW	1'b0	pwm4_load	load parameter of PWM channel 4.
				1'b0: no load
				1'b1: load
[15]	RO	1'b0	reserved	Reserved
[14:12]	RW	3'b0	pwm4_clk_div	clock prescaler of PWM channel 4.
				3'b000: pwm_clk is divided by 1 for count clock
				3'b001: pwm_clk is divided by 2 for count clock
				3'b010: pwm_clk is divided by 4 for count clock
				3'b011: pwm_clk is divided by 8 for count clock
				3'b100: pwm_clk is divided by 16 for count clock
				3'b101: pwm_clk is divided by 32 for count clock
				3'b110: pwm_clk is divided by 64 for count clock
				3'b111: pwm_clk is divided by 128 for count clock
[11:9]	RO	3'b0	reserved	Reserved
[8]	RW	1'b0	pwm4_cnt_mode	count mode of PWM channel 4.
				1'b0: up mode
				1'b1: up and down mode
[7:5]	RO	3'b0	reserved	Reserved
[4]	RW	1'b0	pwm4_polarity	output polarity setting of PWM channel 4.
				1'b0: rising edge. Second edge within the PWM period is rising
				1'b1: falling edge. Second edge within the PWM period is falling
[3:1]	RO	3'b0	reserved	Reserved
[0]	RW	1'b0	pwm4_en	enable of PWM channel 4.
				1'b0: disable
				1'b1: enable
0x38			**PWM4CTL1**	**pwm channel 4 conter value setting**
[31:16]	RW	16'b0	pwm4_cmp_val	the compare value of PWM channel 4
[15:0]	RW	16'b0	pwm4_cnt_top	the counter top value of PWM channel 4
0x40			**PWM5CTL0**	**pwm channel 5 contrl reigister**
[31]	RW	1'b0	pwm5_load_instant	instant load parameter of PWM channel 5.
				1'b0: no load

				1'b1: instant load
[30:17]	RO	14'b0	reserved	Reserved
[16]	RW	1'b0	pwm5_load	load parameter of PWM channel 5.
				1'b0: no load
				1'b1: load
[15]	RO	1'b0	reserved	Reserved
[14:12]	RW	3'b0	pwm5_clk_div	clock prescaler of PWM channel 5.
				3'b000: pwm_clk is divided by 1 for count clock
				3'b001: pwm_clk is divided by 2 for count clock
				3'b010: pwm_clk is divided by 4 for count clock
				3'b011: pwm_clk is divided by 8 for count clock
				3'b100: pwm_clk is divided by 16 for count clock
				3'b101: pwm_clk is divided by 32 for count clock
				3'b110: pwm_clk is divided by 64 for count clock
				3'b111: pwm_clk is divided by 128 for count clock
[11:9]	RO	3'b0	reserved	Reserved
[8]	RW	1'b0	pwm5_cnt_mode	count mode of PWM channel 5.
				1'b0: up mode
				1'b1: up and down mode
[7:5]	RO	3'b0	reserved	Reserved
[4]	RW	1'b0	pwm5_polarity	output polarity setting of PWM channel 5.
				1'b0: rising edge. Second edge within the PWM period is rising
				1'b1: falling edge. Second edge within the PWM period is falling
[3:1]	RO	3'b0	reserved	Reserved
[0]	RW	1'b0	pwm5_en	enable of PWM channel 5.
				1'b0: disable
				1'b1: enable
0x44			**PWM5CTL1**	**pwm channel 5 conter value setting**
[31:16]	RW	16'b0	pwm5_cmp_val	the compare value of PWM channel 5
[15:0]	RW	16'b0	pwm5_cnt_top	the counter top value of PWM channel 5

Table 23: PWM registers

3.12 Quadrature Decoder (QDEC)

The quadrature decoder provides buffered decoding of quadrature-encoded sensor signals with input debounce filters. It is suitable for mechanical and optical sensors. The sample period and accumulation are configurable to match application requirements. The quadrature decoder has three-axis capability and index channel support. It can be programmed as 4x/2x/1x count mode.

3.12.1 Register table

Quadrature decoder related registers are listed below.

Base address: 4000_B000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0x00				
[31:9]	—	23'b0	reserved	
[8]	RW	1'b0	chnz_en	enable channel z
[7:5]	—	3'b0	reserved	
[4]	RW	1'b0	chny_en	enable channel y
[3:1]	—	3'b0	reserved	
[0]	RW	1'b0	chnx_en	enable channel x
0x04			**int_enable**	
[31:30]	—	2'b0	reserved	
[29]	RW	1'b0	int_quaz_02f_en	enable interrupt, counter addition overflow (from 0 to F)
[28]	RW	1'b0	int_quaz_f20_en	enable interrupt, counter subtraction overflow (from F to 0)
[27]	RW	1'b0	int_quay_02f_en	
[26]	RW	1'b0	int_quay_f20_en	
[25]	RW	1'b0	int_quax_02f_en	
[24]	RW	1'b0	int_quax_f20_en	
[23]	—	1'b0	reserved	
[22]	RW	1'b0	incz_int_mode	index counter interrupt mode 0 index changes, 1 index equals hit
[21]	—	1'b0	reserved	
[20]	RW	1'b0	int_incz_en	enable index counter interrupt
[19]	—	1'b0	reserved	
[18]	RW	1'b0	quaz_int_mode	quadrature counter interrupt mode 0 index changes, 1 index equals hit
[17]	—	1'b0	reserved	
[16]	RW	1'b0	int_quaz_en	enable quadrature counter interrupt
[15]	—	1'b0	reserved	
[14]	RW	1'b0	incy_int_mode	
[13]	—	1'b0	reserved	
[12]	RW	1'b0	int_incy_en	

[11]	—	1'b0	reserved	
[10]	RW	1'b0	quay_int_mode	
[9]	—	1'b0	reserved	
[8]	RW	1'b0	int_quay_en	
[7]	—	1'b0	reserved	
[6]	RW	1'b0	incx_int_mode	
[5]	—	1'b0	reserved	
[4]	RW	1'b0	int_incx_mode	
[3]	—	1'b0	reserved	
[2]	RW	1'b0	quax_int_mode	
[1]	—	1'b0	reserved	
[0]	RW	1'b0	int_quax_en	
0x08			**int_clear**	
[31:30]	—	2'b0	reserved	
[29]	WC	1'b0	quaz_02f_clr	clear 0 to F interrupt
[28]	WC	1'b0	quaz_f20_clr	clear F to 0 interrupt
[27]	WC	1'b0	quay_02f_clr	
[26]	WC	1'b0	quay_f20_clr	
[25]	WC	1'b0	quax_02f_clr	
[24]	WC	1'b0	quax_f20_clr	
[23:21]	—	3'b0	reserved	
[20]	WC	1'b0	incz_clr	clear index counter interrupt
[19:17]	—	3'b0	reserved	
[16]	WC	1'b0	quaz_clr	clear quadrature counter interrupt
[15:13]	—	3'b0	reserved	
[12]	WC	1'b0	incy_clr	
[11:9]	—	3'b0	reserved	
[8]	WC	1'b0	quay_clr	
[7:5]	—	3'b0	reserved	
[4]	WC	1'b0	incx_clr	
[3:1]	—	3'b0	reserved	
[0]	WC	1'b0	quax_clr	
0x0C			**int_status**	
[31:30]	—	2'b0	reserved	
[29]	RO	1'b0	int_quaz_02f	0 to F interrupt status
[28]	RO	1'b0	int_quaz_f20	F to 0 interrupt status
[27]	RO	1'b0	int_quay_02f	
[26]	RO	1'b0	int_quay_f20	
[25]	RO	1'b0	int_quax_02f	

[24]	RO	1'b0	int_quax_f20	
[23:21]	—	3'b0	reserved	
[20]	RO	1'b0	int_inc_z	index counter interrupt status
[19:17]	—	3'b0	reserved	
[16]	RO	1'b0	int_qua_z	quadrature counter interrupt status
[15:13]	—	3'b0	reserved	
[12]	RO	1'b0	int_inc_y	
[11:9]	—	3'b0	reserved	
[8]	RO	1'b0	int_qua_y	
[7:5]	—	3'b0	reserved	
[4]	RO	1'b0	int_inc_x	
[3:1]	—	3'b0	reserved	
[0]	RO	1'b0	int_qua_x	
0x10				
[31:18]	—	14'b0	reserved	
[17:16]	RW	2'b0	incx_mode	index counter mode 00 high level 01 positive edge 10 negative edge 11 pos and neg edge
[15:2]	—	14'b0	reserved	
[1:0]	RW	2'b0	quax_mode	quadrature counter mode 01 mode 1x , 10 mode 2x, 11 mode 3x
0x14				
[31:0]	RW	32'b0	quax_hit	to compare with qua_cnt, trigger interrupt
0x18				
[31:0]	RW	32'b0	incx_hit	to compare with inc_cnt, trigger interrupt
0x1C				
[31:0]	RO	32'b0	quax_cnt	quadrature counter
0x20				
[31:0]	RO	32'b0	incx_cnt	index counter
0x24				
[31:18]	—	14'b0	reserved	
[17:16]	RW	2'b0	incy_mode	
[15:2]	—	14'b0	reserved	
[1:0]	RW	2'b0	quay_mode	
0x28				
[31:0]	RW	32'b0	quay_hit	
0x2C				
[31:0]	RW	32'b0	incy_hit	
0x30				

[31:0]	RO	32'b0	quay_cnt	
0x34				
[31:0]	RO	32'b0	incy_cnt	
0x38				
[31:18]	—	14'b0	reserved	
[17:16]	RW	2'b0	incz_mode	
[15:2]	—	14'b0	reserved	
[1:0]	RW	2'b0	quaz_mode	
0x3C				
[31:0]	RW	32'b0	quaz_hit	
0x40				
[31:0]	RW	32'b0	incz_hit	
0x44				
[31:0]	RO	32'b0	quaz_cnt	
0x48				
[31:0]	RO	32'b0	incz_cnt	
0x3FC				
[31:0]	RW	32'b0	dummy	
[15]	RO	1'b0	reserved	Reserved

Table 24: Quadrature decoder registers

3.13 Key Scan (KSCAN)

Keyscan supports key matrix with up to 16 rows by 18 columns. Each individual rows or columns can be enabled or disabled through register settings. GPIO pins can be configured to be used for key scan. A few key scan Parameters can be set through registers, including polarity (low or high indicating key pressed); support multi-key-press or only single-key-press; de-bounce time (the time duration a key press is deemed valid) from 0 to 128mS with 255us step.

A valid key press can trigger an interrupt when keyscan interrupt is enabled. After a keyscan interrupt is serviced, writing 1 to the interrupt state register bit can clear the state bit.

The keyscan has a manual mode and an auto mode. For manual mode, when a keyscan interrupt is received, it is upo the MCU/software to scan the keyscan output pins and check the input pins, to determine which keys have been pressed. Manual mode is relatively slow and need CPU to process. On the contrary, in automode keyscan will automatically scan the output/input pins, and store the row/column info corresponding to the key pressed into read only registers, then trigger an interrupt for software to retrieve key press information.

3.13.1 Register table

Key scan related registers are listed below.

Base address: 4002_4000

OFFSET	TYPE	RESET	NAME	DESCRIPTION
0xC0				
[31:24]	RW	8'h00	mkdi	key scan debounce interval, 0-255, unit: 512uS
[23]	RW	1'b0	mk_pol	key matrix polarity, 0: active scan high, active sense high; 1: active scan low, active sense low;
[22]	RO	1'b0	reserved	no use/as
[21]	RW	1'b0	asact	auto scan on activity: 0, no auto scan, 1, auto scan on activity
[20]	RW	1'b0	imkp	ignore multi key press
[19:2]	RW	18'h0	ms	matrix scan outputs enable: 1: enable, 0: disable
[1]	RW	1'b0	ks_ie	key scan interrupt enable
[0]	RW	1'b0	ks_en	key scan enable
0xC4				
[31:18]	RO	14'b0	reserved	
[17]	WC	1'b0	mkp	key pressed indicator, 0: no key press, 1: key pressed, write 1 to clear
[16:1]	RO	16'h0FFF	mr	key scan inputs states
[0]	WC	1'b0	mi	interrupt state, write 1 to clear interrupt, 0: no interrupt, 1: interrupt issued,
0xC8				
[31:13]	RO	19'b0	reserved	
[12]	RO	1'b0	so	scan on: 1: auto scan is ongoing, 0: scan off
[11:10]	RO	2'b0	mukp	multi key pressed, 00, no key press, 01: 1 key press, 10, more than 1 key pressed
[9:5]	RO	5'h1F	rp	row of key pressed, only for 1 key pressed case
[4:0]	RO	5'h1F	cp	column of key pressed, only for 1 key pressed case
0xCC				
[31:16]	RO	16'h0	mkc1	column 1 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc0	column 0 key pressed, for multi key pressed case
0xD0				
[31:16]	RO	16'h0	mkc3	column 3 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc2	column 2 key pressed, for multi key pressed case
0xD4				
[31:16]	RO	16'h0	mkc5	column 5 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc4	column 4 key pressed, for multi key pressed case

0xD8				
[31:16]	RO	16'h0	mkc7	column 7 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc6	column 6 key pressed, for multi key pressed case
0xDC				
[31:16]	RO	16'h0	mkc9	column 9 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc8	column 8 key pressed, for multi key pressed case
0xE0				
[31:16]	RO	16'h0	mkc11	column 11 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc10	column 10 key pressed, for multi key pressed case
0xE4				
[31:16]	RO	16'h0	mkc13	column 13 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc12	column 12 key pressed, for multi key pressed case
0xE8				
[31:16]	RO	16'h0	mkc15	column 15 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc14	column 14 key pressed, for multi key pressed case
0xEC				
[31:16]	RO	16'h0	mkc17	column 17 key pressed, for multi key pressed case
[15:0]	RO	16'h0	mkc16	column 16 key pressed, for multi key pressed case
0xF0				
[31:16]	RW	16'h0	reserved	
[15:0]	RW	16'h0FFF	mk_in_en	enable/disable key scan inputs: 0: disable, 1: enable
0xF4				
[31:2]	RW	30'h0	reserved	
[1:0]	RW	2'b0	ks_pena_i	
0xF8				
[31:0]	RW	32'h0	ks_iosel	

Table 25: Key scan related registers

3.14 Analog to Digital Converter (ADC) with Programmable Gain Amplifier (PGA)

The 12bit SAR ADC has total 10 inputs. Among them, there are two for PGA inputs, and two differential inputs for the on-chip temperature sensor. The other six inputs can be programmed to 3 pair differential inputs or six single-ended inputs. There is a manual mode with which the ADC can be configured to convert a specific input in single-ended or differential and with a specific ADC clock rate. There is also an auto sweep mode, namely all enabled input channels can be swept automatically in order by the ADC and the converted data will be stored at corresponding memory locations.

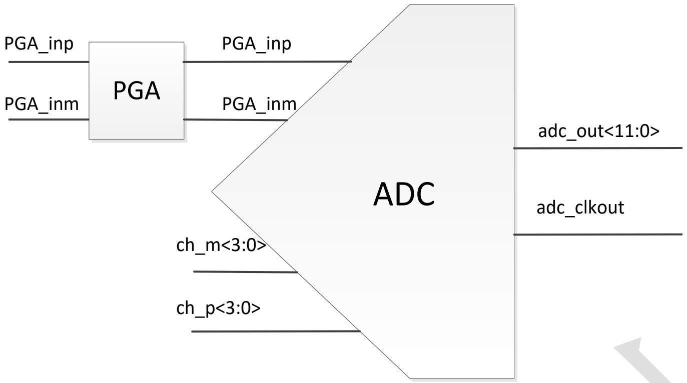


Figure 11: ADC

3.14.1 PGA Path

The PGA provides 42dB gain range from 0dB to 42dB in 3dB steps.

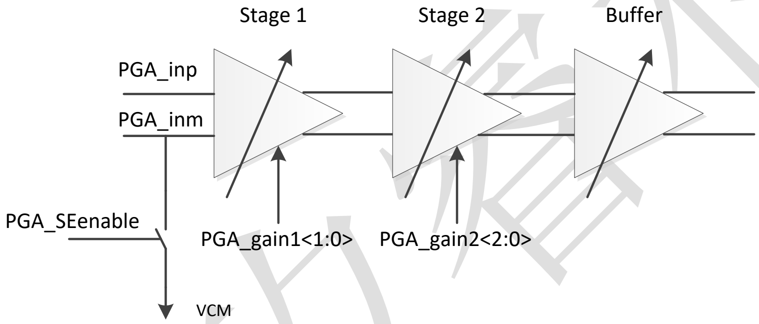


Figure 12: PGA path

pga_gain1 <1>	pga_gain1 <0>	Stage1 gain (dB)	pga_gain2<2>	pga_gain2<1>	pga_gain2<0>	Stage2 gain(dB)
0	0	0	0	0	0	0
0	1	12	0	0	1	3
1	0	24	0	1	0	6
			0	1	1	9
			1	0	0	12
			1	0	1	15
			1	1	0	18

Table 26: PGA gain

Set PGA_SEenable to “1”, PGA will be set to Single-ended mode by pulling the PGA into its Common-mode voltage.

3.14.2 ADC Path

By default the ADC is configured in manual mode. In this mode, the ADC clock rate can be configured to 80k/160k/320k sample per second. Select the pair of inputs and configure it to differential or singled-ended (positive or negative). By default it is differential. After enabling, the ADC will take samples with the configured clock rate and store the data to a channel dependent memory location. For each channel a memory size of 128Byte is allocated, when it is full an interrupt bit will be flagged. Each sample of 12bits takes 2 Byte memory space.

0x4000_F07C		**Register Description**
[4]	adc_ctrl_override	Set manual mode: 1: manual, 0: auto. Default 1
[3]	adc_tconv_sel	For auto mode only, adc conversion time sel: 0: 1.56us, 1: 2.34us
[2:1]	adc_clk_sel	For manual mode only, clksel: 00: 80k, 01: 160k, 10: 320k
[0]	max_rate_256k_320k	For auto mode only, max rate base: 0, 256k, 1, 320k
0x4000_F048		**Register Description**
[11]	adc12b_semode_enm	For manual mode only: 12 bit ADC single-ended mode negative side enable. Bit<11> Bit<8> cannot both be 1; 1: Enable single-ended mode 0: Differential mode
[8]	Adc12b_semode_epm	For manual mode only: 12 bit ADC single-ended mode positive side enable. Bit<8> Bit<11> cannot both be 1; 1: Enable single-ended mode 0: Differentail mode
[7:5]	Channel configure	For manual mode only: 12 bit ADC input channel select control bits. adc12_ctrl<3:1> Selected channel 000 PGA inputs, differential 001 Temperature sensing inputs, differential 010 input A, positive and negative 011 input B, positive and negative 100 input C, positive and negative
[3]	ADC enable	12b ADC power up control. 1: Power up ADC 0: Power down ADC
Memory start/end addresses		**ADC channels**
4005_0400 – 4005_047F		PAG inputs, differential
4005_0480 – 4005_04FF		Temperature sensing, differential

4005_0500 – 4005_057F		Input A, positive or differential
4005_0580 – 4005_05FF		Input A, negative
4005_0600 – 4005_067F		Input B, positive or differential
4005_0680 – 4005_06FF		Input B, negative
4005_0700 – 4005_077F		Input C, positive or differential
4005_0780 – 4005_07FF		Input C, negative
0x4005_003C	**ADC interrupt status**	**Register Description**
[7]		input C, negative
[6]		Input C, positive or differential
[5]		Input B, negative
[4]		Input B, positive or differential
[3]		Input A, negative
[2]		Input A, positive or differential
[1]		Temperature sensing, differential
[0]		PGA inputs, differential
0x4005_0038	**ADC interrupt write clear**	**Register Description**
[7]		input C, negative, write 1 to clear
[6]		Input C, positive or differential, write 1 to clear
[5]		Input B, negative, write 1 to clear
[4]		Input B, positive or differential, write 1 to clear
[3]		Input A, negative, write 1 to clear
[2]		Input A, positive or differential, write 1 to clear
[1]		Temperature sensing, differential, write 1 to clear
[0]		PGA inputs, differential, write 1 to clear

Table 27: ADC manual mode

ADC can also be configured into auto channel sweep mode by setting the “adc_ctrl_override” bit to 0, with which the enabled channels will be sampled in the configured order automatically. The ten ADC input channels can be configured by programming their corresponding registers. Their configurations include sampling time, enable/disable, differential/single-ended, and continuous sampling/single-shot, based on the following register table. The sampled data is stored in the corresponding memory locations as in manual mode.

0x4000_F06C	**ADC_CTL0**	**Register Description**
[31:16]	Temperature sensing, auto mode, differential	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only
[15:0]	PGA inputs, differential	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only
0x4000_F070	**ADC_CTL1**	**Register Description**
[31:16]	Inputs A, negative	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only
[15:0]	Input A, positive or differential	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only
0x4000_F074	**ADC_CTL2**	**Register Description**
[31:16]	Input B, negative	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only

[15:0]	Input B, positive or differential	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only
0x4000_F078	**ADC_CTL3**	**Register Description**
[31:16]	Input C, negative	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only
[15:0]	Input C, positive or differential	channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only

Table 28: ADC channel configurations

3.14.3 ADC Channel <3:0> Connectivity

PGA inputs	hardwired
temp sensing	hardwired
aio<0>	Input A negative
aio<1>	Input A positive
aio<2>	Input B negative
aio<3>	Input B positive
aio<4>	Input C negative
aio<9>	Input C positive

Table 29: ADC channel connectivity

Aio<9, 4:0> and PGA inputs(Aio<7:8>) can be selected through an analog Mux by programming aio_pass<7:0> or aio_attn<7:0>. For example, register 0x4000_F020<8><0> set to 01, then Aio<0> is connected to ADC input A positive node.

0x4000_F020		Register Description
[13:8]	Attenuation ctrl	attn[5:0]. analogIO control for {aio<9>, aio<4>, aio<3>, aio<2>, aio<1>, aio<0>}. {attn[x], pass[x]}: 00 switch off 01 pass 10 attenuate to 1/4 11 NC
[5:0]	pass ctrl	pass[5:0]. analogIO control for {aio<9>, aio<4>, aio<3>, aio<2>, aio<1>, aio<0>}. {attn[x], pass[x]}: 00 switch off 01 pass 10 attenuate to 1/4 11 NC note: analog IO sharing gpio<11>/aio<0> gpio<12>/aio<1> gpio<13>/aio<2> gpio<14>/aio<3> gpio<15>/aio<4> gpio<16>/aio<5>/32K XTAL input gpio<17>/aio<6>/32K XTAL output gpio<18>/aio<7>/pga in+ gpio<19>/aio<8>/pga in- gpio<20>/aio<9>/mic bias

Table 30: analog Mux

4 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which QMS7926 can be exposed without permanently damaging it. Exposure to absolute maximum ratings for prolonged periods of time may affect the reliability of the QMS7926. **Table 32** specifies the absolute maximum ratings for QMS7926.

Symbol	Parameter	Min.	Max.	Unit
Supply voltages				
VDD3		-0.3	+3.6	V
DEC			1.32	V
VSS			0	V
I/O pin voltage				
VIO		-0.3	VDD + 0.3	V
Environmental				
Storage temperature		-40	+125	°C
MSL	Moisture Sensitivity Level		2	
ESD HBM	Human Body Model Class 2		2	kV
ESD CDMQF	Charged Device Model		500	V
Flash memory				
Endurance			100 000	write/erase cycles
Retention			10 years at 40 °C	
Number of times an address can be written between erase cycles			2	times

Table 31: Absolute maximum ratings

5 Operating Conditions

The operating conditions are the physical Parameters that QMS7926 can operate within as defined in **Table 33**.

Symbol	Parameter	Min.	Typ.	Max.	Units
VDD3	Supply voltage, normal mode	1.8	3	3.6	V
tr_VDD	Supply rise time (0 V to 1.8 V)			100	ms
TA	Operating temperature	-40	27	125	°C

Table 32: Operating conditions

6 Radio Transceiver

6.1 Radio Current Consumption

Parameter	Description	MIN	TYP	MAX	UNIT
Tx only at 0dBm	with internal DC-DC @3V		8		mA
Rx Only	with internal DC-DC @3V		8		mA

Table 33: Radio current consumption

6.2 Transmitter Specification

Parameter	Description	MIN	TYP	MAX	UNIT
RF Max Output Power			10		dBm
RF Min Output Power			-20		dBm
OBW for BLE 1Mbps	20dB occupy-bandwidth for BLE modulation 1Mbps		1100		KHz
OBW for BLE 2Mbps	20dB occupy-bandwidth for BLE modulation 2Mbps		2300		KHz
OBW for GFSK 500Kbps	20dB occupy-bandwidth for GFSK modulation 2Mbps		1100		KHz
OBW for GFSK 125bps	20dB occupy-bandwidth for GFSK modulation 2Mbps		1100		KHz
Error Vector Measure	Offset EVM for OQPSK modulation		0.02		
FDEV for BLE 1Mbps	Frequency deviation for GFSK modulation 1Mbps	160		250	KHz
FDEV for BLE 2Mbps	Frequency deviation for GFSK modulation 2Mbps	320		500	KHz

Table 34: Transmitter specification

6.3 Receiver Specification

6.3.1 RX BLE 1Mbps GFSK

Parameter	Description	MIN	TYP	MAX	UNIT
Rx Sensitivity	Sensitivity test 1Mbps BLE ideal transmitter, 37 Byte BER=1E-3		-97		dBm

co-channel rejection	modulated interferer in channel, 37 Byte BER=1E-3		-6		I/C dB
Selectivity +-1MHz	Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3		7		I/C dB
Selectivity +-2MHz	Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3		45		I/C dB
Selectivity +-3MHz	Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3		50		I/C dB
Selectivity +-4MHz	Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3		50		I/C dB
Selectivity +-5MHz or More	Wanted signal at -67dBm, modulated interferer at >=+/- 5MHz, 37 Byte BER=1E-3		55		I/C dB
Selectivity Imag frequency	Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3		22		I/C dB
Intermodulation	Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte BER=1E-3		-20		dBm
Carrier Frequency Offset Tolerance			+-350		KHz
Sample Clock Offset Tolerance			+-120		ppm

Table 35: RX BLE 1Mbps GFSK

6.3.2 RX BLE 2Mbps GFSK

Parameter	Description	MIN	TYP	MAX	UNIT
Rx Sensitivity	Sensitivity test 2Mbps BLE ideal transmitter, 37 Byte BER=1E-3		-94		dBm
co-channel rejection	modulated interferer in channel, 37 Byte BER=1E-3		-6		I/C dB
Selectivity +-1MHz	Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3		-5		I/C dB
Selectivity +-2MHz	Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3		9		I/C dB
Selectivity +-3MHz	Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3		30		I/C dB
Selectivity +-4MHz	Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3		40		I/C dB
Selectivity +-5MHz or More	Wanted signal at -67dBm, modulated interferer at >=+/- 5MHz, 37 Byte BER=1E-3		55		I/C dB

Selectivity Imag frequency	Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3		22		I/C dB
Intermodulation	Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte BER=1E-3		-20		dBm
Carrier Frequency Offset Tolerance			+-350		KHz
Sample Clock Offset Tolerance			+-120		ppm

Table 36: RX BLE 2Mbps GFSK

6.3.3 RX 500Kbps GFSK

Parameter	Description	MIN	TYP	MAX	UNIT
Rx Sensitivity	Sensitivity test 500Kbps BLE ideal transmitter, 37 Byte BER=1E-3		-98		dBm
co-channel rejection	modulated interferer in channel, 37 Byte BER=1E-3		-4		I/C dB
Selectivity +-1MHz	Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3		10		I/C dB
Selectivity +-2MHz	Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3		45		I/C dB
Selectivity +-3MHz	Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3		50		I/C dB
Selectivity +-4MHz	Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3		50		I/C dB
Selectivity +-5MHz or More	Wanted signal at -67dBm, modulated interferer at >=+/- 5MHz, 37 Byte BER=1E-3		55		I/C dB
Selectivity Imag frequency	Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3		24		I/C dB
Intermodulation	Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte Ber=1E-3		-19		dBm
Carrier Frequency Offset Tolerance			+-350		KHz

Sample Clock Offset Tolerance			+- 120		ppm

Table 37: RX 500Kbps GFSK

6.3.4 RX 125Kbps GFSK

Parameter	Description	MIN	TYP	MAX	UNIT
Rx Sensitivity	Sensitivity test 125Kbps BLE ideal transmitter, 37 Byte BER=1E-3		-103		dBm
co-channel rejection	modulated interferer in channel, 37 Byte BER=1E-3		-1		I/C dB
Selectivity +-1MHz	Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3		-11		I/C dB
Selectivity +-2MHz	Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3		45		I/C dB
Selectivity +-3MHz	Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3		50		I/C dB
Selectivity +-4MHz	Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3		50		I/C dB
Selectivity +-5MHz or More	Wanted signal at -67dBm, modulated interferer at >=+/- 5MHz, 37 Byte BER=1E-3		55		I/C dB
Selectivity Imag frequency	Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3		28		I/C dB
Intermodulation	Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte BER=1E-3		-18		dBm
Carrier Frequency Offset Tolerance			+- 350		KHz
Sample Clock Offset Tolerance			+- 120		ppm

Table 38: RX 125Kbps GFSK

6.4 RSSI Specifications

Parameter	Description	MIN	TYP	MAX	UNIT
RSSI Dynamic Range			70		dB

RSSI Accuracy	RSSI Accuracy Valid in range -100 to -30dBm		+/-2		dB
RSSI Resolution	Totally 7bit, from 0 to 127		1		dB
RSSI Period			8		us

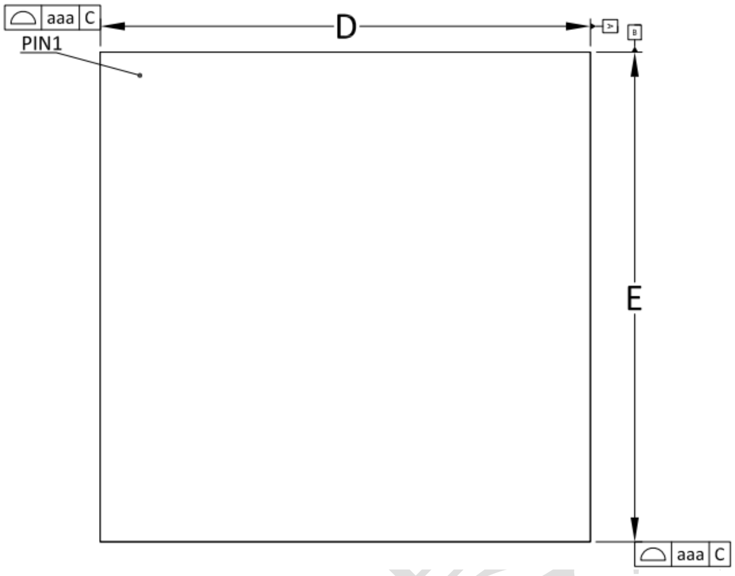
Table 39: RSSI specifications

7 Glossary

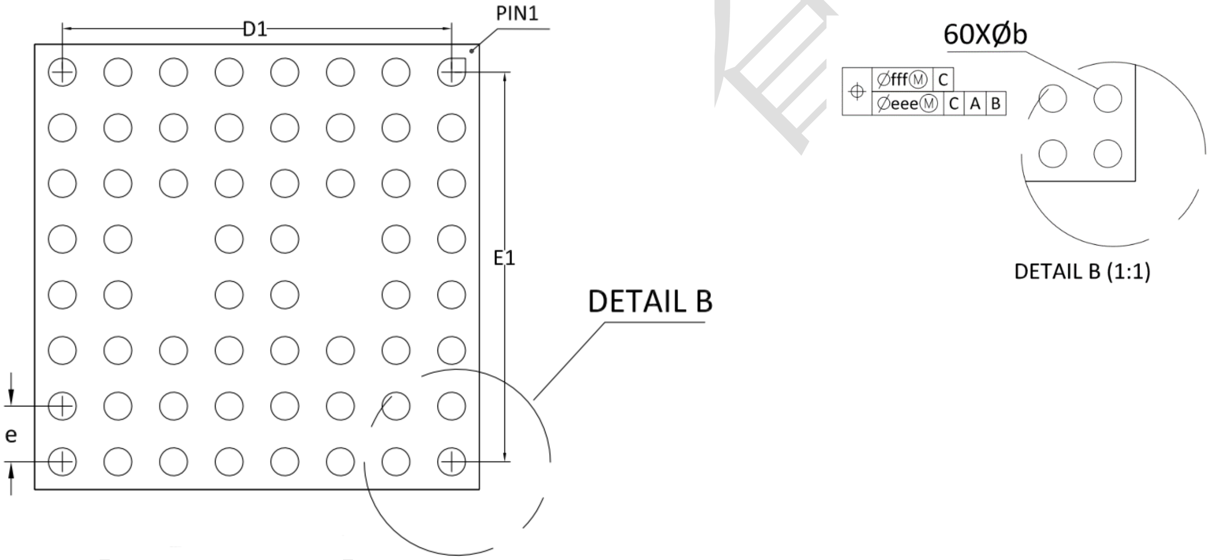
Term	Description
AHB	Advanced High-performance Bus (ARM bus standard)
AHB-AP	DAP AHB Port for debug component access thru AHB bus
AMBA	Advanced Microcontroller Bus Architecture
AON	Always-on power domain
APB	Advanced Peripheral Bus (ARM bus standard)
APB-AP	DAP APB Port for debug component access thru APB bus
BROM	Boot ROM
DAP	Debug Access Port (ARM bus standard)
ETM	Embedded trace module
FPU	Floating Point Unit
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound, Integrated Interchip Sound
ITM	Instrumentation Trace Macrocell Unit
JTAG	Joint Test Access Group (IEEE standard)
JTAG-AP	DAP’s JTAG Access Port to access debug components
JTAG-DP	DAP’s JTAG Debug Port used by external debugger
J&M	Jun and Marty LLC
MPU	Memory Protection Unit
NVIC	Nested vector Interrupt Controller
PCR	Power Clock Reset controller
POR	Power on reset, it is active low in this document
RFIF	APB peripheral to interface RF block
SWD	Serial Wire DAP (ARM bus standard)
SoC	System on chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access memory
TWI	Two-Wire Interface
UART	Universal Asynchronous Receiver and Transmitter
WDT	Watchdog Timer

Table 40: Glossary

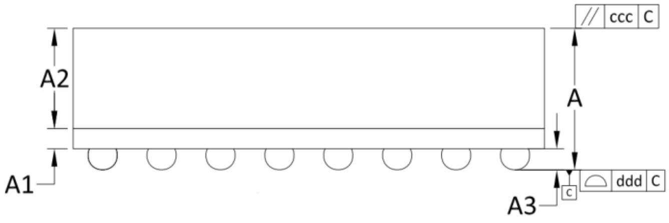
Package dimensions



TOP VIEW



BOTTOM VIEW



SIDE VIEW

Size specification

ALL DIMENSIONS ARE IN MILLIMETERS.			
SYMBOL	**MILLIMETER**		
	MIN	**NOR**	**MAX**
A	1.09	1.20	1.31
A1	0.14	0.17	0.20
A2	0.80	0.85	0.90
A3	0.15	0.18	0.21
D	3.90	4.00	4.10
D1	3.5 BASIC		
E	3.90	4.00	4.10
E1	3.5 BASIC		
e	0.50 BASIC		
b	---	0.25 TYP	---
aaa	0.10		
bbb	0.10		
ccc	0.10		
ddd	0.08		
eee	0.15		
fff	0.05		

Ordering information

Ordering Number	Temperature Range	Package	Packaging
QMS7926B	-40℃ to 85℃	BGA-60	Tray: 2940 pieces

Sample Application and Layout Guide

1.1 Sample Application

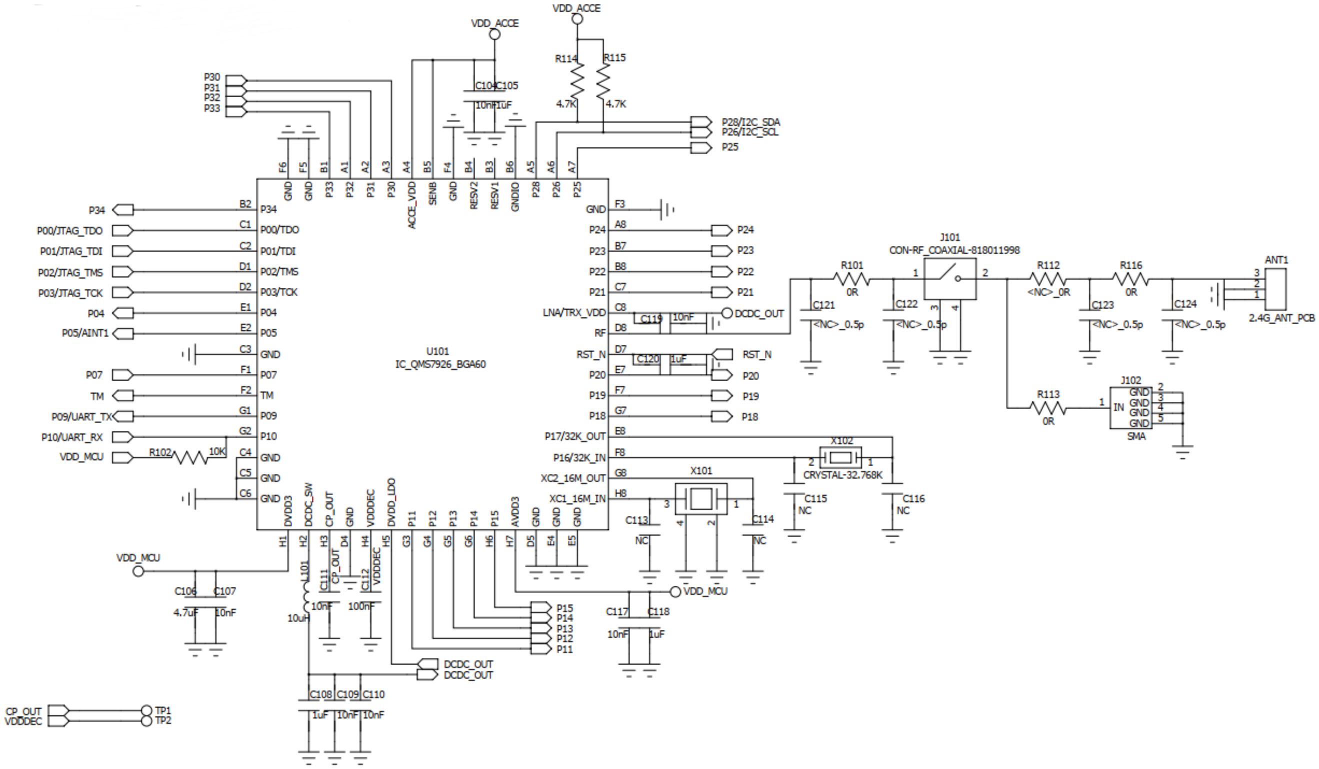


Figure10： Sample application

1.2 Layout Guide

1.2.1 Placement

1. RF matching/Loop filter leading to antenna should be isolated from any other AC/DC signal as much as possible;
2. XTAL/OSC clock is a noise source to other circuits, keep clock trace as short as possible and away from any important area;
3. LDO’s are sensitive and could be easily contaminated, care should be taken for the environment;
4. Antenna is the main RF radiation point, other important blocks should be shielded or away from this area.

RF traces

1. Define RF line width with given dielectric thickness (thickness of PCB dielectric layer to ground plain) to achieve 50ohm impedance; this is mainly for the RF line connecting to matching/loop filter and antenna.

2. Differential traces should be kept in the same length and component should be placed symmetrically;
3. Certain length of RF trace should be treated as part of RF matching.

1.2.2 Bypass Capacitor

1. Each VDD pin needs a bypass capacitor to release chip internal noise and block noise from power supply.
2. For power traces, bypass capacitors should be placed as close as possible to VDD pins.
3. Use one large and one small capacitor when the pin needs two capacitors. Typically the capacitance of the larger capacitor is about 100 times of that of the smaller one. The smaller capacitor usually has better quality factor than the larger one. Place the larger capacitor closer to the pin.
4. The capacitors of Loop filter need to have larger clearance to prevent EMC/EMI issue.
5. Ground via should be close to the Capacitor GND side, and away from strong signals.

1.2.3 Layer Definition

1. Normally 4 layer PCB is recommended.
2. RF trace must be on the surface layer, i.e. top layer or bottom.
3. The second layer of RF PCB must be “Ground ” layer , for both signal ground and RF reference ground , DO NOT put any other trace or plane on second layer, otherwise “antenna effect” will complicate debug process.
4. Power plane generally is on the 3rd layer.
5. Bottom layer is for “signal ” layer.
6. If two layer PCB is used, quality will degrade in general. More care needs to be taken. Try to maximize ground plane, avoid crossing of signal trace with other noise lines or VDD, shield critical signal line with ground plane, maximize bypass capacitor and number of ground vias.

1.2.4 Reference clock and trace

1. Oscillator signal trace is recommended to be on the 1st layer;
2. DO NOT have any trace around or across the reference clock (oscillator) trace.
3. Isolate the reference clock trace and oscillator by having more GND via around.
4. DO NOT have any other traces under the Oscillator.

1.2.5 Power line or plane

1. Whether to use power plain or power line depend on the required current, noise and layout condition. For RF chip, we generally suggest to use power line to bring power into IC pin. Line has parasitic inductance, which forms a low pass filter to reduce the noise traveling around PCB.
2. Add more conductive via on the current source, it will increase max current limit and reduce inductance of via.

3. Add some capacitor alone the power trace when power line travels a long distance.
4. DO NOT place power line or any plane under RF trace or oscillator and its clock trace , the strong clock or RF signal would travel with power line.

1.2.6 Ground Via

1. Ground Via must be as close to the ground pad of bypass capacitor as possible , too much distance between via and ground pad will reduce the effect of bypass capacitor.
2. Having as many ground via as possible.
3. Place ground via around RF trace, the RF trace should be shielded with via trail.