MT6260D GSM/GPRS/EDGE-RX SOC Processor Data Sheet

Version: 1.2
Release date: 2013-07-16

Document Revision History

Revision	Date	Author	Description
1.0	2013-02-01	Alisa Huang	1st release
1.1	2013-03-19	Alisa Huang	Update pin description
1.2	2013-07-16	Alisa Huang	Update multimedia features / Update pin description

Table of Contents

Lists of Tables and Figures

Preface

Acronyms for register types

R/W	For both read and write access
RO	Read only
RC	Read only. After the register bank is read, every bit that is HIGH(1) will be cleared to LOW(0) automatically.
WO	Write only
W1S	Write only. When data bits are written to the register bank, every bit that is HIGH(1) will cause the corresponding bit to be set to 1. Data bits that are LOW(0) have no effects on the corresponding bit.
W1C	Write only. When data bits are written to the register bank, every bit that is HIGH(1) will cause the corresponding bit to be cleared to 0. Data bits that are LOW(0) have no effects on the corresponding bit.

1 System Overview

MT6260D is a monolithic chip integrating leading edge power management unit, analog baseband and radio circuitry based on the low-power CMOS process.

MT6260D is a feature-rich and extremely powerful single-chip solution for high-end GSM/GPRS and EDGE-Rx capability. Based on the 32-bit ARM7EJ-S™ RISC processor, MT6260D's superb processing power, along with high bandwidth architecture and dedicated hardware support, provides a platform for high-performance GPRS/EDGE-Rx Class 12 MODEM application and leading-edge multimedia applications.

MT6260D also features:

- A highly integrated Bluetooth transceiver which is fully compliant with Bluetooth specification v3.0.
- A FM receiver supporting both audio broadcast de-modulation and RDS/RBDS data decoding.

Typical application diagram is shown in Figure 1.

Platform

MT6260D is capable of running the ARM7EJ-S™ RISC processor, which provides the best trade-off between system performance and power consumption.

For large amounts of data transfer, high-performance DMA (Direct Memory Access) with hardware flow control is implemented, which greatly enhances the data movement speed while reducing the MCU processing load.

Targeted as a media-rich platform for mobile applications, MT6260D also provides hardware security digital rights management for copyright protection. For further safeguard and to protect the manufacturer's development investment, hardware flash content protection is provided to prevent unauthorized porting of the software load.

Multimedia

The MT6260D multimedia subsystem provides conventional parallel interface and 2-bit serial interface for CMOS sensors. The camera resolution is up to 2M pixels. The software-based codec can be used to process various video types. Besides, MT6260D provides fancy UI capabilities through its hardware 2D accelerator. The 2D accelerator performs high-speed linear transformations with filtering. To take advantage of the high MCU performance, GIF and PNG decoders are implemented by the software.

In addition, MT6260D is implemented with a high-performance audio synthesis technology, as well as a high-quality audio amplifier to provide superior audio experiences.

Connectivity and storage

MT6260D supports UART, USB 1.1 FS/LS , SDIO, HIF interface and MMC/SD storage systems. These interfaces provide MT6260D users with the highest level of flexibility in implementing high-end solutions.

To achieve a complete user interface, MT6260D also brings together all the necessary peripheral blocks for a multimedia GSM/GPRS/EDGE-RX phone. The peripheral blocks include the keypad scanner with the capability to detect multiple key presses, SIM controller, real-time clock, PWM, serial/parallel LCD controller and general-purpose programmable I/Os.

Audio

Using a highly integrated mixed-signal audio front-end, the MT6260D architecture provides easy audio interfacing with direct connection to the audio transducers. The audio interface

integrates A/D converters for voice band, as well as high-resolution stereo D/A converters for both audio and voice band.

MT6260D supports AMR codec to adaptively optimize the quality of speech and audio. Moreover, HE-AAC codec is implemented to deliver CD-quality audio at low bit rates.

In addition, a 1.2W audio amplifier is also embedded to save the BOM cost of adopting external amplifiers.

GSM/GPRS/EDGE-Rx radio

MT6260D integrates a mixed-signal baseband front-end in order to provide a well-organized radio interface with flexibility for efficient customization. The front-end contains gain and offset calibration mechanisms and filters with programmable coefficients for comprehensive compatibility control on RF modules. MT6260D achieves outstanding MODEM performance by utilizing a highly dynamic range ADC in the RF downlink path.

MT6260D embeds a high-performance and completely integrated single-ended SAW-less RF transceiver for multi-band GSM cellular system. In this RF transceiver, a quad-band receiving feature with high sensitivity is supported utilizing one RF receiver and a fully integrated channel filter. With ultra-high dynamic range, the off-chip balun and SAW filters on the receiving path can be removed for BOM cost reduction. In addition, the minimum component count is guaranteed by realizing a highly integrated transmitter, low-spur frequency synthesizer and a Digitally-Controlled Crystal Oscillator (DCXO).

Bluetooth radio

MT6260D offers a highly integrated Bluetooth radio and baseband processor. Only a minimum of external components are required. MT6260D provides superior sensitivity and class 1 output power and thus ensures the quality of the connection with a wide range of Bluetooth devices.

MT6260D is fully compliant with Bluetooth v3.0 and offers enhanced data rates of up to 3Mbps. It also provides the coexistence protocol with 802.11 system.

MT6260D supports rich Bluetooth profiles, enabling diversified applications that are widely used on the handset with excellent interoperability.

FM radio

The FM radio subsystem provides a completely integrated FM Rx receiver supporting 87.5 ~ 108MHz FM bands with 50kHz tuning step. It also performs fast channel seek/scan algorithm to validate 200 carrier frequencies in 6 seconds. In addition to receiving FM audio broadcasting, the digital RDS/RBDS data system is supported as well. The integrated FM transceiver utilizes state-of-the-art digital demodulation/modulation techniques to achieve excellent performance.

In order to achieve high SINAD, good sensitivity and excellent noise suppression, the FM receiver adopts adaptive demodulation scheme to optimize Rx system performance in all ranges of signal quality by reference of a very sophisticated channel quality index (CQI). When the received signal quality is poor, the design not only enhances the ACI rejection capability but also uses a very ingenious skill to soft mute annoying noise so as to provide good perception quality.

The FM radio subsystem supports both long antenna, which is usually an earphone, and auto-calibrated short antenna, which is usually a FPC short antenna or shared antenna with GSM for different application scenarios.

Debugging function

The JTAG interface enables in-circuit debugging of the software program with the ARM7EJ-S™ core. With this standardized debugging interface, MT6260D provides developers with a wide set of options in choosing ARM development kits from different third party vendors.

Power management

A power management is embedded in MT6260D to provide rich features a high-end feature phone supports, including Li-ion battery charger, high performance and low quiescent current LDOs, and drivers for LED and backlight.

MT6260D offers various low-power features to help reduce the system power consumption. MT6260D is also fabricated in an advanced low-power CMOS process, hence providing an overall ultra-low leakage solution.

Package

The MT6260D device is offered in a 9.6mm×8.6mm, 199-ball, 0.5mm pitch, TFBGA package.

Figure 1. Typical application of MT6260D

1.1 Platform Features

General

- Integrated voice-band, audio-band and base-band analog front-end
- Integrated full-featured power management unit

MCU subsystem

- ARM7EJ-S™ 32-bit RISC processor
- Java hardware acceleration for fast Java-based games and applets
- High-performance multi-layer AHB bus
- Dedicated DMA bus with 15 DMA channels
- On-chip boot ROM for factory flash programming
- Watchdog timer for system crash recovery
- 4 sets of general-purpose timers
- Circuit switch data coprocessor
- Division coprocessor

User interfaces

- 5-row x 5-column keypad controller with hardware scanner
- Supports multiple key presses for gaming
- Dual SIM/USIM controller with hardware T = 0/T = 1 protocol control
- Real-time clock (RTC) operating with a low-quiescent-current power supply
- General-purpose I/Os (GPIOs) available for auxiliary applications
- 1 sets of Pulse Width Modulation (PWM) output
- 16 external interrupt lines
- 1 external channel auxiliary 10-bit A/D converter

Security

- Supports security key and chip random ID

Connectivity

- 2 UARTs with hardware flow control and supports baud rate up to 921,600 bps
- FS/LS USB 1.1 device controller
- Multimedia card, secure digital Memory Card, host controller with flexible I/O voltage power
- Supports SDIO interface for SDIO peripherals as well as WIFI connectivity
- DAI/PCM and I2S interface for audio applications
- I2C master interface for peripheral management including image sensors
- SPI master interface for peripheral management.

Power management

- Li-ion battery charger
- 14 LDOs for the power supply of memory card, camera, Bluetooth, RF, SIM card and other diversified usage
- 4 open-drain output switches to supply/control the LED
- LDO type vibrator
- One NMOS switch to control keypad LED
- Thermal overload protection
- Under-voltage lock-out protection
- Over-voltage protection
- Different levels of sleep modes with sophisticated software control enables excellent power saving performance.

Test and debugging

- Built-in digital and analog loop back modes for both audio and baseband front-end
- DAI port complies with GSM Rec.11.10.
- JTAG port for debugging embedded MCU

1.2 MODEM Features

Radio interface and baseband front-end

- Digital PM data path with baseband front-end
- High dynamic range delta-sigma ADC converts the downlink analog I and Q signals to digital baseband.
- 10-bit D/A converter for Automatic Power Control (APC)
- Programmable radio Rx filter with adaptive gain control
- Dedicated Rx filter for FB acquisition
- 4-pin Baseband Parallel Interface (BPI) with programmable driving strength
- Supports multi-band

Voice and modem CODEC

- Dial tone generation
- Voice memo
- Noise reduction
- Echo suppression
- Advanced sidetone oscillation reduction
- Digital sidetone generator with programmable gain
- Two programmable acoustic compensation filters
- Supports GSM/GPRS/EDGE-Rx modem
- GSM quad vocoders for adaptive multirate (AMR), enhanced full rate (EFR), full rate (FR) and half rate (HR)
- GSM channel coding, equalization and A5/1, A5/2 and A5/3 ciphering
- GPRS/EDGE-Rx GEA1, GEA2 and GEA3 ciphering
- GPRS packet switched data with CS1/CS2/CS3/CS4 coding schemes
- EDGE-Rx with MCS1-9 receiver coding schemes
- GSM circuit switch data
- GPRS/EDGE-Rx Class 12
- Supports SAIC (single antenna interference cancellation) technology
- Supports VAMOS(Voice services over Adaptive Multi-user channels on One Slot) technology in R9 spec.

Voice interface and voice front-end

- Two microphone inputs share one low-noise amplifier with programmable gain and Automatic Gain Control (AGC) mechanisms
- Voice power amplifier with programmable gain
- 2^{nd} order Sigma-Delta A/D converter for voice uplink path
- Shares D/A converter with audio playback path
- Supports full-duplex hands-free operation
- Compliant with GSM 03.50

1.3 GSM/GPRS/EDGE RF Features

Receiver

- Dual single-ended input LNAs support Quad bandQuadrature RF mixer
- Fully integrated channel filter
- High dynamic range ADC
- 12dB PGA gain with 6dB gain step

Transmitter

- Transmitter outputs support quad bands.
- Highly precise and low noise RF transmitter for GSM/GPRS applications

Frequency synthesizer

- Programmable fractional-N synthesizer
- Integrated wide range RFVCO
- Integrated loop filter
- Fast settling time suitable for multi-slot GPRS/EDGE-Rx applications

Digitally-Controlled Crystal Oscillator (DCXO)

- Two-pin 26MHz crystal oscillator
- On-chip programmable capacitor array for coarse-tuning
- On-chip programmable capacitor array for fine-tuning
- Low power mode supports 32K crystal removal

1.4 Multimedia Features

LCD/ WiFi interface

- Dedicated parallel interface supports 3 external devices with 8-bit for WiFi interface and 8-/9-bit for parallel LCD interface.

LCD controller

- Supports simultaneous connection to 2 parallel and 2 serial LCD modules
- LCM formats supported: RGB565, RGB666, RGB888
- Supports LCD module with maximum resolution up to 480x320
- Per pixel alpha channel
- True color engine
- Supports hardware display rotation
- Capable of combining display memories with up to 4 blending layers

Camera interface

- YUV422 format image input
- Capable of processing image of size up to 0.3M pixels (Mediatek serial interface) and 2M pixels (w/o compression)

JPEG decoder

- Baseline JPEG decoding
- Supports various YUV formats, DC/AC Huffman tables and quantization tables

JPEG encoder

- ISO/IEC 10918-1 JPEG baseline mode
- ISO/IEC 10918-2 compliance
- Supports YUV420 and grayscale formats
- Supports EXIF/JFIF
- Standard DC and AC Huffman tables
- Provides 5 levels of encode quality
- Supports zeros shutter delay

MJPEG

- Decode spec: CIF@30fps
- Encode spec: QVGA@15fps

Image data processing

- Supports 4x digital zoom
- High throughput hardware scaler. Capable of tailoring an image to an arbitrary size.
- Horizontal scaling with bilinear interpolation
- Vertical scaling with bilinear interpolation
- YUV and RGB color space conversion
- RGB/YCbCr format thumbnail output

MPEG-4/H.263 CODEC

- Software-based MPEG4 encoder
- Software-based MPEG4 decoder
- ISO/IEC 14496-2 simple profile:
  - Decode spec: 704x576@25fps
  - Encode spec: QVGA@15fps
- ISO/IEC 14496-2 advanced simple profile:
  - Decode @ level 0/1/2/3
  - ITU-T H.263 profile 0 @ level 40
- Supports visual tools for decoder: I-VOP, P-VOP, B-VOP, AC/DC prediction, 4-MV, unrestricted MV, error resilience, short header, global motion compensation, method 1/2 quantization, quarter-pel motion compensation.
- Error resilience for decoder: Slice resynchronization, data partitioning, reversible VLC
- Supports visual tools for encoder: I-VOP, P-VOP, Half-Pel, DC prediction, unrestricted MV, short header

H.264

- ISO/IEC 14496-10 baseline profile
  - Decode spec: QCIF@30fps

2D accelerator

- Supports 32-bpp ARGB8888, 24-bpp RGB888, 16-bpp RGB565, 24-bpp ARGB6666.
- 4 layers overlay with individual color format, window size, source key, constant alpha and rotation
- Rectangle fill with constant
- BitBlt: Capable with 7 rotation types

- Alpha blending with 7 rotation types, per-pixel alpha and pre-multiplied alpha
- Font drawing: Normal font and anti-aliasing font
- Linear transformation: Supports perspective transform, truncate/nearest/bi-linear sample filter.

Audio CODEC

- Supports AAC codec decoding
- Wavetable synthesis with up to 64 tones
- Advanced wavetable synthesizer capable of generating simulated stereo
- Wavetable including GM full set of 128 instruments and 47 sets of percussions
- PCM playback and record
- Digital audio playback

Audio interface and audio front-end

- Supports I2S interface
- High-resolution D/A converters for stereo audio playback
- Voice band A/D converter support
- Stereo to mono conversion

1.5 Bluetooth Features

Radio features

- Fully compliant with Bluetooth specification 3.0 + EDR
- Low out-of-band spurious emissions support simultaneous operation with GPS and GSM/GPRS worldwide radio systems
- Low-IF architecture with high degree of linearity and high order channel filter
- Integrated T/R switch and Balun
- Fully integrated PA provides 10dBm output power
- -95dBm sensitivity with excellent interference rejection performance
- Hardware AGC dynamically adjusts receiver performance in changing environments

Baseband features

- Up to 4 simultaneous active ACL links
- Up to 1 simultaneous SCO or eSCO link with CVSD coding
- Supports eSCO
- Scatternet support: Up to 4 piconets simultaneously with background inquiry/page scan
- Supports sniff mode
- AFH and PTA collaborative support for WLAN/BT coexistence
- Idle mode and sleep mode enables ultra-low power consumption.
- Supports PCM interface and built-in programmable transcoders for linear voice with re-transmission
- Built-in hardware modem engine for access code correlation, header error correction, forward error correction, CRC, whitening and encryption
- Channel quality driven data rate adaptation
- Channel assessment for AFH

Platform features

- Embedded processor for Bluetooth protocol stack with built-in memory system
- Fully verified ROM based system with code patch for feature enhancement

1.6 FM Features

- 76-108MHz worldwide FM bands with 50kHz tuning step
- Supports RDS/RBDS radio data system
- Supports long/short antenna
- 40ms seek time per channel, and 9sec search time for all channels (87.5 ~ 108MHz)
- Superior stereo noise reduction
- Soft mute volume control
- Supports short antenna, auto calibration for different FM channels
- 60dB SINAD with 22.5kHz FM deviation
- 3dBuVemf FM RX sensitivity with superior interference rejection
- 20dBuVemf RDS sensitivity (dev: 2kHz)
- More than 55dBc rejection capability against -200kHz ACI

1.7 General Descriptions

Figure 2 is the block diagram of MT6260D. Based on a multi-processor architecture, MT6260D integrates an ARM7EJ-S™ core, the main processor running high-level GSM/EDGE-Rx protocol software as well as multimedia applications, single digital signal processor core, which manages the low-level MODEM and advanced audio functions, an embedded processor running Bluetooth baseband and link control protocol and the Bluetooth radio control.

MT6260D consists of the following subsystems:

- Microcontroller Unit (MCU) subsystem: Includes an ARM7EJ-S™ RISC processor and its accompanying memory management and interrupt handling logics
- Digital Signal Processor (DSP) subsystem: Includes a DSP and its accompanying memory, memory controller and interrupt controller
- MCU/DSP interface: Junction at which the MCU and the DSP exchange hardware and software information
- Microcontroller peripherals: Include all user interface modules and RF control interface modules
- Microcontroller coprocessors: Run computing-intensive processes in place of the microcontroller
- DSP peripherals: Hardware accelerators for GSM/GPRS/EDGE-Rx channel codec
- Multimedia subsystem: Integrates several advanced accelerators to support multimedia applications
- Voice front-end: Data path for converting analog speech to and from digital speech
- Audio front-end: Data path for converting stereo audio from an audio source
- Baseband front-end: Data path for converting a digital signal to and from an analog signal from the RF modules
- Timing generator: Generates the control signals related to the TDMA frame timing
- Power, reset and clock subsystem: Manage the power, reset and clock distribution inside MT6260D.
- Bluetooth subsystem: Includes an embedded processor with embedded ROM/RAM system, baseband processor, and a high-performance radio block
- Power management unit: Self-contained power supply source which also controls the charging and system startup circuitry.

Details of the individual subsystems and blocks are described in the following chapters.

Figure 2. MT6260D block diagram

2 Product Descriptions

2.1 Pin Description

2.1.1 Ball Diagram

For MT6260D, an TFBGA 9.6mm*8.6mm, 199-ball, 0.5mm pitch package is offered. Pin-outs and the top view are illustrated in **Figure 3** for this package.

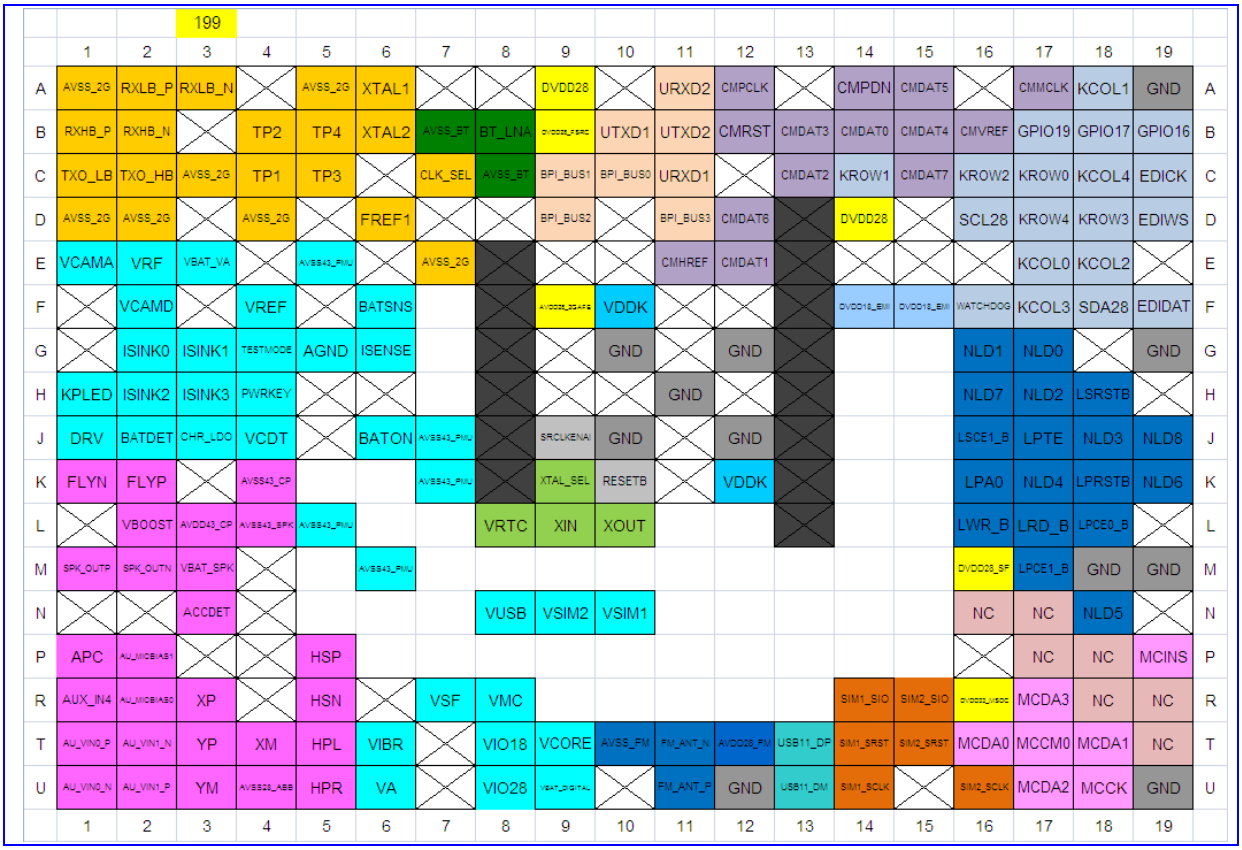


Figure 3. ***Ball diagram and top view***

2.1.2 Pin Coordination

Table 1. Pin coordinates

Pin#	Net name	Pin#	Net name	Pin#	Net name
A1	AVSS_2G	E5	AVSS43_PMU	M17	LPCE1_B
A11	URXD2	E7	AVSS_2G	M18	GND
A12	CMPCLK	F10	VDDK	M19	GND
A14	CMPDN	F14	DVDD18_EMI	M2	SPK_OUTN
A15	CMDAT5	F15	DVDD18_EMI	M3	VBAT_SPK
A17	CMMCLK	F16	WATCHDOG	M6	AVSS43_PMU
A18	KCOL1	F17	KCOL3	N10	VSIM1

Pin#	Net name	Pin#	Net name	Pin#	Net name
A19	GND	F18	SDA28	N16	NC
A2	RXLB_P	F19	EDIDAT	N17	NC
A3	RXLB_N	F2	VCAMD	N18	NLD5
A5	AVSS_2G	F4	VREF	N3	ACCDET
A6	XTAL1	F6	BATSNS	N8	VUSB
A9	DVDD28	F9	AVDD28_2GAFE	N9	VSIM2
B1	RXHB_P	G10	GND	P1	APC
B10	UTXD1	G12	GND	P17	NC
B11	UTXD2	G16	NLD1	P18	NC
B12	CMRST	G17	NLD0	P19	MCINS
B13	CMDAT3	G19	GND	P2	AU_MICBIAS1
B14	CMDAT0	G2	ISINK0	P5	HSP
B15	CMDAT4	G3	ISINK1	R1	AUX_IN4
B16	CMVREF	G4	TESTMODE	R14	SIM1_SIO
B17	GPIO19	G5	AGND	R15	SIM2_SIO
B18	GPIO17	G6	ISENSE	R16	DVDD33_MSDC
B19	GPIO16	H1	KPLED	R17	MCDA3
B2	RXHB_N	H11	GND	R18	NC
B4	TP2	H16	NLD7	R19	NC
B5	TP4	H17	NLD2	R2	AU_MICBIAS0
B6	XTAL2	H18	LSRSTB	R3	XP
B7	AVSS_BT	H2	ISINK2	R5	HSN
B8	BT_LNA	H3	ISINK3	R7	VSF
B9	DVDD28_FSRC	H4	PWRKEY	R8	VMC
C1	TXO_LB	J1	DRV	T1	AU_VIN0_P
C10	BPI_BUS0	J10	GND	T10	AVSS_FM
C11	URXD1	J12	GND	T11	FM_ANT_N
C13	CMDAT2	J16	LSCE1_B	T12	AVDD28_FM
C14	KROW1	J17	LPTE	T13	USB11_DP
C15	CMDAT7	J18	NLD3	T14	SIM1_SRST
C16	KROW2	J19	NLD8	T15	SIM2_SRST
C17	KROW0	J2	BATDET	T16	MCDA0
C18	KCOL4	J3	CHR_LDO	T17	MCCM0
C19	EDICK	J4	VCDT	T18	MCDA1
C2	TXO_HB	J6	BATON	T19	NC
C3	AVSS_2G	J7	AVSS43_PMU	T2	AU_VIN1_N
C4	TP1	J9	SRCLKENAI	T3	YP
C5	TP3	K1	FLYN	T4	XM
C7	CLK_SEL	K10	RESETB	T5	HPL
C8	AVSS_BT	K12	VDDK	T6	VIBR
C9	BPI_BUS1	K16	LPA0	T8	VIO18
D1	AVSS_2G	K17	NLD4	T9	VCORE
D11	BPI_BUS3	K18	LPRSTB	U1	AU_VIN0_N
D12	CMDAT6	K19	NLD6	U11	FM_ANT_P
D14	DVDD28	K2	FLYP	U12	GND
D16	SCL28	K4	AVSS43_CP	U13	USB11_DM

Pin#	Net name	Pin#	Net name	Pin#	Net name
D17	KROW4	K7	AVSS43_PMU	U14	SIM1_SCLK
D18	KROW3	K9	XTAL_SEL	U16	SIM2_SCLK
D19	EDIWS	L10	XOUT	U17	MCDA2
D2	AVSS_2G	L16	LWR_B	U18	MCCK
D4	AVSS_2G	L17	LRD_B	U19	GND
D6	FREF1	L18	LPCE0_B	U2	AU_VIN1_P
D9	BPI_BUS2	L2	VBOOST	U3	YM
E1	VCAMA	L3	AVDD43_CP	U4	AVSS28_ABB
E11	CMHREF	L4	AVSS43_SPK	U5	HPR
E12	CMDAT1	L5	AVSS43_PMU	U6	VA
E17	KCOL0	L8	VRTC	U8	VIO28
E18	KCOL2	L9	XIN	U9	VBAT_DIGITAL
E2	VRF	M1	SPK_OUTP		
E3	VBAT_VA	M16	DVDD28_SF		

2.1.3 Detailed Pin Description

Table 2. Acronym for pin types

Abbreviation	Description
AI	Analog input
AO	Analog output
AIO	Analog bi-direction
DI	Digital input
DO	Digital output
DIO	Digital bi-direction
P	Power
G	Ground

Table 3. PIN function description and power domain

Pin name	Type	Description	Power domain
System			
RESETB	DIO	System reset	DVDD18_EMI
SRCLKENAI	DIO	26MHz clock request by external devices	DVDD18_EMI
GPIO16	DIO	General purpose input /output 16	DVDD28
GPIO17	DIO	General purpose input /output 17	DVDD28
GPIO19	DIO	General purpose input /output 19	DVDD28
EDI interface			
EDICK	DIO	I2S clock	DVDD28
EDIDAT	DIO	I2S data	DVDD28
EDIWS	DIO	I2S word sync	DVDD28

Pin name	Type	Description	Power domain
RF control circuitro			
BPI_BUS0	DIO	RF hard-wire control bus bit 0	DVDD28
BPI_BUS1	DIO	RF hard-wire control bus bit 1	DVDD28
BPI_BUS2	DIO	RF hard-wire control bus bit 2	DVDD28
BPI_BUS3	DIO	RF hard-wire control bus bit 3	DVDD28
UART interface			
URXD1	DIO	UART1 receive data	DVDD28
UTXD1	DIO	UART1 transmit data	DVDD28
URXD2	DIO	UART2 receive data	DVDD28
UTXD2	DIO	UART2 transmit data	DVDD28
Keypad interface			
KCOL0	DIO	Keypad column 0	DVDD28
KCOL1	DIO	Keypad column 1	DVDD28
KCOL2	DIO	Keypad column 2	DVDD28
KCOL3	DIO	Keypad column 3	DVDD28
KCOL4	DIO	Keypad column 4	DVDD28
KROW0	DIO	Keypad row 0	DVDD28
KROW1	DIO	Keypad row 1	DVDD28
KROW2	DIO	Keypad row 2	DVDD28
KROW3	DIO	Keypad row 3	DVDD28
KROW4	DIO	Keypad row 4	DVDD28
Camera interface			
CMRST	DIO	CMOS sensor reset signal output	DVDD28
CMPDN	DIO	CMOS sensor power down control	DVDD28
CMVREF	DIO	CMOS sensor vertical reference signal input	DVDD28
CMHREF	DIO	CMOS sensor horizontal reference signal input	DVDD28
CMDAT0	DIO	CMOS sensor data input 0	DVDD28
CMDAT1	DIO	CMOS sensor data input 1	DVDD28
CMDAT2	DIO	CMOS sensor data input 2	DVDD28
CMDAT3	DIO	CMOS sensor data input 3	DVDD28
CMDAT4	DIO	CMOS sensor data input 4	DVDD28
CMDAT5	DIO	CMOS sensor data input 5	DVDD28
CMDAT6	DIO	CMOS sensor data input 6	DVDD28
CMDAT7	DIO	CMOS sensor data input 7	DVDD28
CMPCLK	DIO	CMOS sensor pixel clock output	DVDD28
CMMCLK	DIO	CMOS sensor pixel clock input	DVDD28
MS/SD card interface			
MCINS	DIO	SD card detect Input	DVDD18_EMI
MCDA0	DIO	SD serial data IO 0/memory stick serial data IO	DVDD33_MSDC

Pin name	Type	Description	Power domain
MCDA1	DIO	SD serial data IO 1/memory stick serial data IO	DVDD33_MSDC
MCDA2	DIO	SD serial data IO 2/memory stick serial data IO	DVDD33_MSDC
MCDA3	DIO	SD serial data IO 3/memory stick serial data IO	DVDD33_MSDC
MCCK	DIO	SD serial clock/memory stick serial clock	DVDD33_MSDC
MCCM0	DIO	SD command output/memory stick bus state output	DVDD33_MSDC
SIM card interface			
SIM1_SIO	DIO	SIM1 data input/outputs	VSIM1
SIM1_SRST	DIO	SIM1 card reset output	VSIM1
SIM1_SCLK	DIO	SIM1 card clock output	VSIM1
SIM2_SIO	DIO	SIM2 data input/outputs	VSIM2
SIM2_SRST	DIO	SIM2 card reset output	VSIM2
SIM2_SCLK	DIO	SIM2 card clock output	VSIM2
I2C interface			
SCL28	DIO	I2C clock 2.8v power domain	DVDD28
SDA28	DIO	I2C data 2.8v power domain	DVDD28
LCD interface			
LSRSTB	DIO	Serial display interface reset signal	DVDD18_EMI
LSCE1_B	DIO	Serial display interface chip select 1 output	DVDD18_EMI
LPCE1_B	DIO	Parallel display interface chip select 1 output	DVDD18_EMI
LPCE0_B	DIO	Parallel display interface chip select 0 output	DVDD18_EMI
LPTE	DIO	Parallel display interface tearing effect	DVDD18_EMI
LPRSTB	DIO	Parallel display interface reset signal	DVDD18_EMI
LRD_B	DIO	Parallel display interface read strobe	DVDD18_EMI
LPA0	DIO	Parallel display interface address output	DVDD18_EMI
LWR_B	DIO	Parallel display interface write strobe	DVDD18_EMI
NLD8	DIO	Parallel LCD data 8	DVDD18_EMI
NLD7	DIO	Parallel LCD data 7	DVDD18_EMI
NLD6	DIO	Parallel LCD data 6	DVDD18_EMI
NLD5	DIO	Parallel LCD data 5	DVDD18_EMI
NLD4	DIO	Parallel LCD data 4	DVDD18_EMI
NLD3	DIO	Parallel LCD data 3	DVDD18_EMI
NLD2	DIO	Parallel LCD data 2	DVDD18_EMI
NLD1	DIO	Parallel LCD data 1	DVDD18_EMI
NLD0	DIO	Parallel LCD data 0	DVDD18_EMI
Watchdog reset			

Pin name	Type	Description	Power domain
WATCHDOG	DIO	Reset external memory device	DVDD18_EMI
Not Connected			
NC	NC	Not Connected	NC
FM			
FM_ANT_P	AI	FM input from antenna	AVDD28_FM
FM_ANT_N	AI	FM input from antenna	AVDD28_FM
Bluetooth			
BT_LNA	AIO	Bluetooth RF single-ended input	DVDD28
2G RF			
RXHB_P	AI	RF input for highband Rx (DCS/PCS)	VRF
RXHB_N	AI	RF input for highband Rx (DCS/PCS)	VRF
RXLB_P	AI	RF input for lowband Rx (GSM900/GSM850)	VRF
RXLB_N	AI	RF input for lowband Rx (GSM900/GSM850)	VRF
TXO_HB	AO	RF output for highband Tx (DCS/PCS)	VRF
TXO_LB	AO	RF output pin for lowband Tx (GSM900/GSM850)	VRF
FREF1	AO	DCXO reference clock output	VRF
XTAL1	AIO	Input 1 for DCXO crystal	VRF
XTAL2	AIO	Input 2 for DCXO crystal	VRF
TP1	DI	GPI82	VRF
TP2	DI	GPI81	VRF
TP3	DI	GPI80	VRF
TP4	DO	Test pin	VRF
CLK_SEL	AIO	DCXO mode selection	VRF
USB			
USB11_DM	AIO	D- data input/output	-
USB11_DP	AIO	D+ data input/output	-
Analog baseband			
HPR	AIO	Audio head phone output (R channel)	AVDD28_ABB
HPL	AIO	Audio head phone output (L channel)	AVDD28_ABB
HSP	AIO	Voice handset output (positive)	AVDD28_ABB
HSN	AIO	Voice handset output (negative)	AVDD28_ABB
AU_VIN0_P	AIO	Microphone 0 input (positive)	AVDD28_ABB
AU_VIN0_N	AIO	Microphone 0 input (negative)	AVDD28_ABB
AU_VIN1_P	AIO	Microphone 1 input (positive)	AVDD28_ABB
AU_VIN1_N	AIO	Microphone 1 input (negative)	AVDD28_ABB
AUX_IN4	AIO	Auxiliary ADC input	AVDD28_ABB
SPK_OUTP	AIO	Speaker positive output	VBAT_SPK
SPK_OUTN	AIO	Speaker negative output	VBAT_SPK
APC	AIO	Automatic power control DAC output	AVDD28_ABB

Pin name	Type	Description	Power domain
XP	AI	Touch panel X-axis positive input	AVDD28_ABB
XM	AI	Touch panel X-axis negative input	AVDD28_ABB
YP	AI	Touch panel Y-axis positive input	AVDD28_ABB
YM	AI	Touch panel Y-axis negative input	AVDD28_ABB
AU_MICBIAS0	AIO	Microphone bias source 0	AVDD28_ABB
AU_MICBIAS1	AIO	Microphone bias source 1	AVDD28_ABB
ACCDET	AIO	Accessory detection	AVDD28_ABB
Real-time clock			
XIN	AIO	Input pin for 32K crystal	VRTC
XOUT	AIO	Input pin for 32K crystal	VRTC
XTAL_SEL	DIO	Pin option for external 32K crystal	VRTC
Power management unit			
VA	AIO	LDO output for ABB - VA	VBAT_ANALOG
VCAMA	AIO	LDO output for sensor – VCAMA	VBAT_VA
VCAMD	AIO	LDO output for sensor - VCAMD	VBAT_VA
VIBR	AIO	LDO output for vibrator - VIBR	VBAT_DIGITAL
VIO18	AIO	LDO output for 1.8V power - VIO18	VBAT_DIGITAL
VIO28	AIO	LDO output for 2.8V power - VIO28	VBAT_DIGITAL
VMC	AIO	LDO output for memory card - VMC	VBAT_DIGITAL
VSF	AIO	LDO output - VSF	VBAT_DIGITAL
VRF	AIO	LDO output for GSMRF - VRF	VBAT_VA
VRTC	AIO	LDO output for RTC - VRTC	VBAT_DIGITAL
VSIM1	AIO	LDO output for 1st SIM - VSIM	VBAT_DIGITAL
VSIM2	AIO	LDO output for 2nd SIM - VSIM2	VBAT_DIGITAL
VUSB	AIO	LDO output for USB - VUSB	VBAT_DIGITAL
VCORE	AIO	LDO output for core circuit - Vcore	VBAT_DIGITAL
VREF	AIO	Band gap reference	BATSNS
VCDT	AIO	Charger-In level sense pin	BATSNS
DRV	AIO	IDAC current output open-drain pin	BATSNS
BATON	AIO	Battery Pack, NTC connected pin	BATSNS
ISENSE	AIO	Top node of current sensing 0.2ohm Rsense resistor	BATSNS
CHR_LDO	AIO	2.8V shunt-regulator output	BATSNS
BATDET	AIO	Battery detection pin	BATSNS
ISINK0	AIO	Backlight driver channel 0	VBAT_VA
ISINK1	AIO	Backlight driver channel 1	VBAT_VA
ISINK2	AIO	Backlight driver channel 2	VBAT_VA
ISINK3	AIO	Backlight driver channel 3	VBAT_VA
KPLED	AIO	Keypad led driver	VBAT_VA
FLYN	AIO	Charge pump flying cap negative terminal	AVDD43_CP

Pin name	Type	Description	Power domain
FLYP	AIO	Charge pump flying cap positive terminal	AVDD43_CP
TESTMODE	AIO	Test mode	BATSNS
PWRKEY	AIO	PWR key	BATSNS
Analog power			
AVDD28_FM	P	FM power	-
AVDD28_2GAFE	P	2.8V power supply for 2G AFE	-
AVDD43_CP	P	VBAT input for charge pump	
VBAT_DIGITAL	P	Digital LDOs used battery voltage input	-
VBAT_VA	P	Analog LDOs used battery voltage input	-
VBAT_SPK	P	VBAT input for loud speaker driver	-
BATSNS	P	Battery node of battery pack	-
Analog ground			
AVSS28_ABB	G	ABB 2.8V ground	-
AVSS_BT	G	BT ground	-
AVSS_2G	G	2G RF ground	-
AVSS_FM	G	FM ground	-
AVSS43_PMU	G	PMU ground	-
AVSS43_SPK	G	SPK ground	-
AGND	G	GND for VREF	-
AVSS43_CP	G	Charge pump GND	
Digital power			
DVDD28	P	2.8V power supply for digital macros in transceiver	-
DVDD28_FSRC	P	E-FUSE blowing power control	-
DVDD33_MSDC	P	3.3V memory card power	-
DVDD18_EMI	P	1.8V EMI IO power	-
DVDD28_SF	P	2.8V IO power	-
VDDK	P	1.2V core power	
Digital ground			
GND	G	Ground	-

Table 4. Acronym for state of pins

Abbreviation	Description
I	Input
LO	Low output
HO	High output
XO	Low or high output
PU	Pull-up
PD	Pull-down
-	No PU/PD

Abbreviation	Description
0~N	Aux. function number
X	Delicate function pin

Table 5. State of pins

Name	Reset			Output drivability	Termination when not used	IO type
	State[1]	Aux[2]	PU/PD[3]			
System						
RESETB	HO	1	-	DIOH6/DIOL6	No need	IO Type 6
SRCLKENAI	I	7	PD	DIOH6/DIOL6	No need	IO Type 6
GPIO16	I	1	PD	DIOH2/DIOL2	No need	IO Type 1
GPIO17	I	1	PD	DIOH2/DIOL2	No need	IO Type 1
GPIO19	I	1	PD	DIOH2/DIOL2	No need	IO Type 1
EDI interface						
EDICK	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
EDIDAT	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
EDIWS	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
RF control circuitry						
BPI_BUS0	LO	1	-	DIOH1/DIOL1	No need	IO Type 1
BPI_BUS1	I	1	PD	DIOH1/DIOL1	No need	IO Type 1
BPI_BUS2	LO	1	-	DIOH1/DIOL1	No need	IO Type 1
BPI_BUS3	I	1	PD	DIOH1/DIOL1	No need	IO Type 1
UART interface						
URXD1	I	1	PU	DIOH3/DIOL3	No need	IO Type 3
UTXD1	HO	1	-	DIOH1/DIOL1	No need	IO Type 1
URXD2	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
UTXD2	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
Keypad Interface						
KCOL0	I	0	PU	DIOH4/DIOL4	No need	IO Type 4
KCOL1	I	0	PD	DIOH4/DIOL4	No need	IO Type 4
KCOL2	I	0	PD	DIOH4/DIOL4	No need	IO Type 4
KCOL3	I	0	PD	DIOH4/DIOL4	No need	IO Type 4
KCOL4	I	0	PD	DIOH4/DIOL4	No need	IO Type 4
KROW0	LO	0	-	DIOH5/DIOL5	No need	IO Type 5
KROW1	I	0	PD	DIOH5/DIOL5	No need	IO Type 5
KROW2	I	0	PD	DIOH5/DIOL5	No need	IO Type 5
KROW3	I	0	PD	DIOH5/DIOL5	No need	IO Type 5

[1] The column “State” of “Reset” shows the pin state during reset. (Input, High Output, Low Output, etc)

[2] The column “Aux” for “Reset” means the default aux function number, shown in the table “Pin Multiplexing, Capability and Settings”.

[3] The column “PU/PD” for “Reset” means if there is internal pull-up or pull-down when the pin is input in the reset state.

Name	Reset			Output drivability	Termination when not used	IO type
	State[1]	Aux[2]	PU/PD[3]			
KROW4	I	0	PD	DIOH5/DIOL5	No need	IO Type 5
Camera interface						
CMRST	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMPDN	HO	0	-	DIOH1/DIOL1	No need	IO Type 1
CMVREF	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
GMHREF	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT0	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT1	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT2	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT3	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT4	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT5	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT6	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMDAT7	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMPCLK	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
CMMCLK	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
MS/SD card interface						
MCINS	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
MCDA0	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCDA1	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCDA2	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCDA3	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCCM	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
MCMM0	I	0	PD	DIOH3/DIOL3	No need	IO Type 3
I2C interface						
SCL28	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
SDA28	I	0	PD	DIOH1/DIOL1	No need	IO Type 1
LCD interface						
LSRSTB	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
LSCE1B	HO	1	-	DIOH2/DIOL2	No need	IO Type 2
LPCE0B	HO	1	-	DIOH2/DIOL2	No need	IO Type 7
LPCE1B	HO	1	-	DIOH2/DIOL2	No need	IO Type 2
LPTE	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
LPRSTB	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
LRD_B	I	0	PD	DIOH2/DIOL2	No need	IO Type 7
LPA0	I	0	PD	DIOH2/DIOL2	No need	IO Type 7
LWR_B	I	0	PD	DIOH2/DIOL2	No need	IO Type 7
NLD8	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
NLD7	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
NLD6	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
NLD5	I	0	PD	DIOH2/DIOL2	No need	IO Type 2

Name	Reset			Output drivability	Termination when not used	IO type
	State[1]	Aux[2]	PU/PD[3]			
NLD4	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
NLD3	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
NLD2	I	0	PD	DIOH2/DIOL2	No need	IO Type 2
NLD1	I	0	PD	DIOH2/DIOL2	No need	IO Type 7
NLD0	I	0	PD	DIOH2/DIOL2	No need	IO Type 7
Watchdog reset						
WATCHDOG	XO	1	-	DIOH1/DIOL1	No need	IO Type 1
General purpose I/O interface						
TP1	I	0	PU	N/A	No need	IO Type 8
TP2	I	0	PU	N/A	No need	IO Type 8
TP3	I	0	PU	N/A	No need	IO Type 8
Touch panel						
XP	I	1	PU	N/A	No need	IO Type 9
XM	I	1	-	N/A	No need	IO Type 9
YP	I	1	-	N/A	No need	IO Type 9
YM	I	1	PD	N/A	No need	IO Type 9

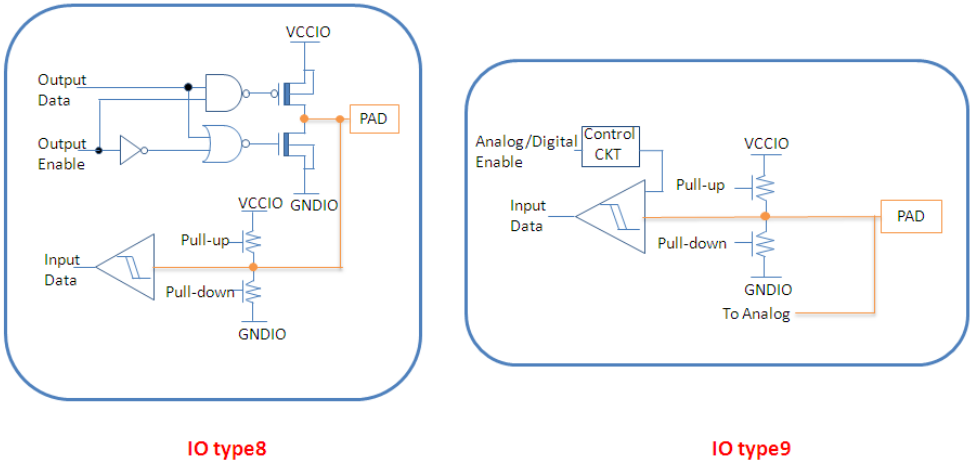


Figure 4. ***IO types in state of pins***

2.1.4 Pin Multiplexing, Capability and Settings

Table 6. Acronym for pull-up and pull-down types

Abbreviation	Description
PU	Pull-up, not controllable
PD	Pull-down, not controllable
CU	Pull-up, controllable
CD	Pull-down, controllable
X	Cannot pull-up or pull-down

Table 7. Capability of PU/PD, driving and Schmitt trigger

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
EDICK	0	GPIO0	IO	CU, CD	4, 8, 12, 16mA	0
	1	EDICK	O	-	4, 8, 12, 16mA	0
	2	PWM	O	-	4, 8, 12, 16mA	0
	3	EINT0	I	-	4, 8, 12, 16mA	0
EDIDAT	0	GPIO61	IO	CU, CD	4, 8, 12, 16mA	0
	1	EDIDAT	IO	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
	2	PWM	O	-	4, 8, 12, 16mA	0
	3	EINT8	I	-	4, 8, 12, 16mA	0
EDIWS	0	GPIO18	IO	CU, CD	4, 8, 12, 16mA	0
	1	EDIWS	O	-	4, 8, 12, 16mA	0
GPIO16	0	GPIO16	IO	CU, CD	4, 8, 12, 16mA	0
	1	GPSFSYNC	O	-	4, 8, 12, 16mA	0
GPIO17	0	GPIO17	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPIBUS5	O	-	4, 8, 12, 16mA	0
GPIO19	0	GPIO19	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPIBUS4	O	-	4, 8, 12, 16mA	0
SCL28	0	GPIO1	IO	CU, CD	4, 8, 12, 16mA	0
	1	SCL	IO	CU, CD	4, 8, 12, 16mA	0
	4	SPISCK0	O	-	4, 8, 12, 16mA	0
	5	D1_ICK	I	PD	4, 8, 12, 16mA	0
SDA28	0	GPIO2	IO	CU, CD	4, 8, 12, 16mA	0
	1	SDA	IO	CU, CD	4, 8, 12, 16mA	0
	4	SPICS0	O	-	4, 8, 12, 16mA	0
	5	D1_IMS	I	CU, CD	4, 8, 12, 16mA	0
KCLO4	0	GPIO5	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL4	IO	CU, CD	4, 8, 12, 16mA	0
	3	EINT1	I	CU, CD	4, 8, 12, 16mA	0
	4	FMJTDI	I	CU, CD	4, 8, 12, 16mA	0
	5	JDI	I	CU, CD	4, 8, 12, 16mA	0
	6	BTJDI	I	CU, CD	4, 8, 12, 16mA	0
KCOL3	0	GPIO6	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL3	IO	CU, CD	4, 8, 12, 16mA	0
	2	EDI2CK	O	CU, CD	4, 8, 12, 16mA	0
	4	FMJTMS	I	CU, CD	4, 8, 12, 16mA	0
	5	JTMS	I	CU, CD	4, 8, 12, 16mA	0
	6	BTJTMS	I	CU, CD	4, 8, 12, 16mA	0
KCOL2	0	GPIO7	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL2	IO	CU, CD	4, 8, 12, 16mA	0
	4	FMJTCK	I	CU, CD	4, 8, 12, 16mA	0
	5	JTCK	I	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
	6	BTJTCK	I	CU, CD	4, 8, 12, 16mA	0
KCOL1	0	GPIO8	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCOL1	IO	CU, CD	4, 8, 12, 16mA	0
KCOL0	0	GPIO9	IO	CU, CD	4, 8, 12, 16mA	0
	1	KCLO0	IO	CU, CD	4, 8, 12, 16mA	0
KROW4	0	GPIO11	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW4	IO	CU, CD	4, 8, 12, 16mA	0
	2	EDI2WS	O	-	4, 8, 12, 16mA	0
	3	EINT3	I	CU, CD	4, 8, 12, 16mA	0
	4	FMJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
	5	JTRSTB	I	CU, CD	4, 8, 12, 16mA	0
	6	BTJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
KROW3	0	GPIO12	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW3	IO	CU, CD	4, 8, 12, 16mA	0
	2	EDI2DAT	O	-	4, 8, 12, 16mA	0
	4	FMJTDO	O	-	4, 8, 12, 16mA	0
	5	JTDO	O	-	4, 8, 12, 16mA	0
	6	BTJTDO	O	-	4, 8, 12, 16mA	0
KROW2	0	GPIO13	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW2	IO	CU, CD	4, 8, 12, 16mA	0
	5	JTRCK	O	-	4, 8, 12, 16mA	0
	6	BTDBGACKN	O	-	4, 8, 12, 16mA	0
KROW1	0	GPIO14	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW1	IO	CU, CD	4, 8, 12, 16mA	0
	5	DI_IDA	IO	PU, CD	4, 8, 12, 16mA	0
	6	BTDBGIN	I	CU, CD	4, 8, 12, 16mA	0
KROW0	0	GPIO15	IO	CU, CD	4, 8, 12, 16mA	0
	1	KROW0	IO	CU, CD	4, 8, 12, 16mA	0
URXD2	0	GPIO20	IO	CU, CD	4, 8, 12, 16mA	0
	1	U2RXD	I	CU, CD	4, 8, 12, 16mA	0
	2	U1RTS	O	-	4, 8, 12, 16mA	0
	3	BTPRI	IO	CU, CD	4, 8, 12, 16mA	0
	4	SPIMISI0	O	-	4, 8, 12, 16mA	0
UTXD2	0	GPIO21	IO	CU, CD	4, 8, 12, 16mA	0
	1	U2TXD	O	-	4, 8, 12, 16mA	0
	2	U1CTS	I	PU	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
	3	CLKO0	O	-	4, 8, 12, 16mA	0
	4	SPIMISO0	I	PD	4, 8, 12, 16mA	0
URXD1	0	GPIO22	IO	CU, CD	4, 8, 12, 16mA	0
	1	U1RXD	I	PU	4, 8, 12, 16mA	0
	4	EINT4	I	CU, CD	4, 8, 12, 16mA	0
UTXD1	0	GPIO23	IO	CU, CD	4, 8, 12, 16mA	0
	1	U1TXD	O	-	4, 8, 12, 16mA	0
MCINS	0	GPIO24	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	EINT5	I	CU, CD	2,4,6,8,10,12 ,14,16mA	0
MCCK	0	GPIO25	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCCK	IO	CU, CD	4, 8, 12, 16mA	0
	5	JRTCK	O	-	4, 8, 12, 16mA	0
MCDA0	0	GPIO26	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA0	IO	CU, CD	4, 8, 12, 16mA	0
	4	FMJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
	5	JTRST_B	I	CU, CD	4, 8, 12, 16mA	0
MCDA1	0	GPIO3	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA1	IO	CU, CD	4, 8, 12, 16mA	0
	4	FMJTDI	I	CU, CD	4, 8, 12, 16mA	0
	5	JTDI	I	CU, CD	4, 8, 12, 16mA	0
MCDA2	0	GPIO4	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA2	IO	CU, CD	4, 8, 12, 16mA	0
	2	WIFITOBT	I	CU, CD	4, 8, 12, 16mA	0
	4	FMJTDI	I	CU, CD	4, 8, 12, 16mA	0
	5	JTDI	I	CU, CD	4, 8, 12, 16mA	0
MCDA3	0	GPIO10	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCDA3	IO	CU, CD	4, 8, 12, 16mA	0
	3	EINT2	I	CU, CD	4, 8, 12, 16mA	0
	4	FMJTCK	I	CU, CD	4, 8, 12, 16mA	0
	5	JTCK	I	CU, CD	4, 8, 12, 16mA	0
MCCM0	0	GPIO27	IO	CU, CD	4, 8, 12, 16mA	0
	1	MCCM0	IO	CU, CD	4, 8, 12, 16mA	0
	4	FMJTDO	O	-	4, 8, 12, 16mA	0
	5	JTDO	O	-	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
NLD8	0	GPIO28	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD8	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
	5	CMMCLK	O	-	2,4,6,8,10,12 ,14,16mA	0
NLD7	0	GPIO29	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD7	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
	5	CMCSK	I	CU, CD	2,4,6,8,10,12 ,14,16mA	0
NLD6	0	GPIO30	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD6	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
	5	CMRST	O	-	2,4,6,8,10,12 ,14,16mA	0
NLD5	0	GPIO31	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD5	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
	5	CMCSD0	I	PD	2,4,6,8,10,12 ,14,16mA	0
NLD4	0	GPIO32	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD4	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
	5	CMPDN	O	-	2,4,6,8,10,12 ,14,16mA	0
NLD3	0	GPIO33	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD3	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
	2	LSDI3	I	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	5	CMCSD1	I	CU,CD	2,4,6,8,10,12 ,14,16mA	0
NLD2	0	GPIO34	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD2	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
NLD1	0	GPIO35	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD1	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
NLD0	0	GPIO36	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	NLD0	IO	PU, CD	2,4,6,8,10,12 ,14,16mA	0
	2	LSA0DA0	O	-	2,4,6,8,10,12 ,14,16mA	0
LWR_B	0	GPIO37	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LWRB	O	-	2,4,6,8,10,12 ,14,16mA	0
	2	LSCK0	O	-	2,4,6,8,10,12 ,14,16mA	0
LRD_B	0	GPIO38	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LRDB	O	-	2,4,6,8,10,12 ,14,16mA	0
	2	LSDA0	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
LPA0	0	GPIO39	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LPA0	O	-	2,4,6,8,10,12 ,14,16mA	0
	2	LSDI0	I	PD	2,4,6,8,10,12 ,14,16mA	0
LPCE0_B	0	GPIO40	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
	1	LPCE0B	O	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	2	LSCE0B1	O	CU, CD	2,4,6,8,10,12 ,14,16mA	0
LSCE1_B	0	GPIO41	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LSCE1B	O	-	2,4,6,8,10,12 ,14,16mA	0
	2	LPCE2B	O	-	2,4,6,8,10,12 ,14,16mA	0
	3	SCL	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
LPCE1_B	0	GPIO42	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LPCE1B	O	-	2,4,6,8,10,12 ,14,16mA	0
	2	LPTE1	I	CU,CD	2,4,6,8,10,12 ,14,16mA	0
	3	SDA	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
LSRSTB	0	GPIO43	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LSRSTB	O	-	2,4,6,8,10,12 ,14,16mA	0
WATCHDOG	0	GPIO44	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	WATCHDOG	O	-	2,4,6,8,10,12 ,14,16mA	0
	2	LPCE2B	O	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	3	EINT6	I	CU, CD	2,4,6,8,10,12 ,14,16mA	0
LPTE	0	GPIO45	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LPTE1	I	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	3	CLKO1	O	-	2,4,6,8,10,12 ,14,16mA	0
LPRSTB	0	GPIO46	IO	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	1	LPRSTB	O	-	2,4,6,8,10,12 ,14,16mA	0
	2	LSRSTB	O	-	2,4,6,8,10,12 ,14,16mA	0
	4	LPTE1	I	CU, CD	2,4,6,8,10,12 ,14,16mA	0
	5	LPCE3B	O	-	2,4,6,8,10,12 ,14,16mA	0
CMDAT0	0	GPIO47	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT0	I	CU, CD	4, 8, 12, 16mA	0
	2	CMCSD0	I	CU, CD	4, 8, 12, 16mA	0
	3	U2RTS	O	CU, CD	4, 8, 12, 16mA	0
	4	FMJTDI	I	CU, CD	4, 8, 12, 16mA	0
	5	JTDI	I	CU, CD	4, 8, 12, 16mA	0
	6	BTJTDI	I	CU, CD	4, 8, 12, 16mA	0
CMDAT1	0	GPIO48	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT1	I	CU, CD	4, 8, 12, 16mA	0
	2	CMCSD1	I	CU, CD	4, 8, 12, 16mA	0
	3	U2CTS	O	CU, CD	4, 8, 12, 16mA	0
	4	FMJTMS	I	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
	5	JTMS	I	CU, CD	4, 8, 12, 16mA	0
	6	BTJTMS	I	CU, CD	4, 8, 12, 16mA	0
CMDAT2	0	GPIO49	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT2	I	CU, CD	4, 8, 12, 16mA	0
	2	SCL	IO	CU, CD	4, 8, 12, 16mA	0
	4	FMJTCK	I	CU, CD	4, 8, 12, 16mA	0
	5	JTCK	I	CU, CD	4, 8, 12, 16mA	0
	6	BTJTCK	I	CU, CD	4, 8, 12, 16mA	0
CMDAT3	0	GPIO50	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT3	I	CU, CD	4, 8, 12, 16mA	0
	2	LRSTB	O	-	4, 8, 12, 16mA	0
	3	MC3CM0	IO	CU, CD	4, 8, 12, 16mA	0
	4	DAICLK	O	-	4, 8, 12, 16mA	0
	5	JTRCK	O	-	4, 8, 12, 16mA	0
	6	BTDBGACKN	O	-	4, 8, 12, 16mA	0
CMDAT4	0	GPIO51	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT4	I	CU, CD	4, 8, 12, 16mA	0
	2	LSA0DA1	O	-	4, 8, 12, 16mA	0
	3	MC3DA0	IO	CU, CD	4, 8, 12, 16mA	0
	4	FMJTRSTB	I	CU, CD	4, 8, 12, 16mA	0
	5	JTRST_B	I	CU, CD	4, 8, 12, 16mA	0
	6	BTJTRST_B	I	CU, CD	4, 8, 12, 16mA	0
CMDAT5	0	GPIO52	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT5	I	CU, CD	4, 8, 12, 16mA	0
	2	LSCK5	O	-	4, 8, 12, 16mA	0
	3	MC3DA1	IO	CU, CD	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
	4	U2RXD	I	CU, CD	4, 8, 12, 16mA	0
	7	EGND48	I	PD	4, 8, 12, 16mA	0
CMDAT6	0	GPIO53	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT6	I	CU, CD	4, 8, 12, 16mA	0
	2	LSDA1	IO	PU, CD	4, 8, 12, 16mA	0
	3	MC2DA2	O	-	4, 8, 12, 16mA	0
	4	DAIPCMIN	I	-	4, 8, 12, 16mA	0
CMDAT7	0	GPIO54	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMDAT7	I	CU, CD	4, 8, 12, 16mA	0
	2	LSDI1	I	CU, CD	4, 8, 12, 16mA	0
	3	MC3DA3	O	-	4, 8, 12, 16mA	0
	4	DAIPCMOUT	I	CU, CD	4, 8, 12, 16mA	0
CMHREF	0	GPIO55	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMHREF	I	CU, CD	4, 8, 12, 16mA	0
	2	LSCE0B1	I	CU, CD	4, 8, 12, 16mA	0
	3	MC3CK	I	CU, CD	4, 8, 12, 16mA	0
	4	DAISYNC	O	-	4, 8, 12, 16mA	0
CMVREF	0	GPIO56	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMVREF	I	CU, CD	4, 8, 12, 16mA	0
	2	SDA	IO	CU, CD	4, 8, 12, 16mA	0
	3	EINT7	I	PD	4, 8, 12, 16mA	0
	4	DAIRST	I	PD	4, 8, 12, 16mA	0
	5	LPTE0	I	PD	4, 8, 12, 16mA	0
CMPDN	0	GPIO57	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMPDN	O	-	4, 8, 12, 16mA	0
CMMCLK	0	GPIO58	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMMCLK	O	-	4, 8, 12, 16mA	0
CMPCLK	0	GPIO59	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMPCLK	I	CU, CD	4, 8, 12, 16mA	0
	2	CMCSK	I	CU, CD	4, 8, 12, 16mA	0
CMRST	0	GPIO60	IO	CU, CD	4, 8, 12, 16mA	0
	1	CMRST	O	-	4, 8, 12, 16mA	0

Name	Aux. function	Aux. name	Aux. type	PU/PD/ CU/CD	Driving	SMT
EDIDAT	0	GPIO61	IO	CU, CD	4, 8, 12, 16mA	0
	1	EDIDAT	IO	CU,CD	4, 8, 12, 16mA	0
	2	PWM	O	-	4, 8, 12, 16mA	0
	3	EINT8	I	CU,CD	4, 8, 12, 16mA	0
BPI_BUS3	0	GPIO62	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPIBUS3	O	-	4, 8, 12, 16mA	0
BPI_BUS2	0	GPIO63	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPIBUS2	O	-	4, 8, 12, 16mA	0
BPI_BUS1	0	GPIO64	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPIBUS1	O	-	4, 8, 12, 16mA	0
BPI_BUS0	0	GPIO65	IO	CU, CD	4, 8, 12, 16mA	0
	1	BPIBUS0	O	-	4, 8, 12, 16mA	0
SRCLKENAI	0	GPIO73	IO	CU, CD	4, 8, 12, 16mA	0
	1	SRCLKENAI	I	CU, CD	4, 8, 12, 16mA	0
RESETB	0	GPIO74	IO	CU, CD	4, 8, 12, 16mA	0
	1	RESETB	IO	CU, CD	4, 8, 12, 16mA	0
XP	0	AGPI75	I	CU, CD	N/A	0
	2	EINT30	I	CU, CD	N/A	0
XM	0	AGPI76	I	CU, CD	N/A	0
	2	EINT31	I	CU, CD	N/A	0
YP	0	AGPI77	I	CU, CD	N/A	0
	2	EINT32	I	CU, CD	N/A	0
YM	0	AGPI78	I	CU, CD	N/A	0
	2	EINT33	I	CU, CD	N/A	0
TP3	0	AGPI80	I	PU	N/A	0
	1	EINT34	I	PU	N/A	0
TP2	0	AGPI81	I	PU	N/A	0
	1	EINT35	I	PU	N/A	0
TP1	0	AGPI82	I	PU	N/A	0
	1	EINT36	I	PU	N/A	0

2.2 Electrical Characteristics

2.2.1 Absolute Maximum Ratings

Table 8. ***Absolute maximum ratings for power supply***

Symbol or pin name	Description	Min.	Max.	Unit
VBAT_DIGITAL	Digital used battery voltage input	-0.3	+4.3	V
VBAT_ANALOG	Analog used battery voltage input	-0.3	+4.3	V
VBAT_SPK	VBAT input for loud speaker driver	-0.3	+4.3	V
VBAT_RF	RF used battery voltage input	-0.3	+4.3	V
VUSB	LDO output for USB-VUSB	+3.0	+3.6	V
AVDD28_FM	2.8V power supply for FM	+2.52	+3.08	V
AVDD28_2GAFE	2.8V power supply for 2G AFE	+2.52	+3.08	V
AVDD28_ABB	2.8V power supply for ABB	+2.52	+3.08	V
DVDD28	2.8V power supply for digital macros in transceiver	+2.52	+3.08	V
DVDD18	1.8V power supply for digital macros in transceiver	+1.62	+1.98	V
DVDD28_SF	2.8V IO power	+2.7	+3.6	V
	1.8V IO power	+1.7	+1.98	V
DVDD33_MSDC	3.3V memory card power	+3.0	+3.6	V
DVDD18_EMI	1.8V EMI IO power	+1.62	+1.98	V
DVDD28_FSRC	E-FUSE blowing power control	+2.52	+3.08	V
VDDK	1.3v core power	+1.17	+1.43	V

Table 9. ***Absolute maximum ratings for voltage input***

Symbol or pin name	Description	Min.	Max.	Unit
VIN1	Digital input voltage for IO Type 1	-0.3	3.08	V
VIN2	Digital input voltage for IO Type 2	-0.3	3.08	V
VIN3	Digital input voltage for IO Type 3	-0.3	3.08	V
VIN4	Digital input voltage for IO Type 4	-0.3	3.08	V
VIN5	Digital input voltage for IO Type 5	-0.3	3.08	V
VIN6	Digital input voltage for IO Type 6	-0.3	3.08	V

Table 10. Absolute maximum ratings for storage temperature

Symbol or pin name	Description	Min.	Max.	Unit
Tstg	Storage temperature	-55	125	^{o}C

2.2.2 Recommended Operating Conditions

Table 11. ***Recommended operating conditions for power supply***

Symbol or pin name	Description	Min.	Typ.	Max.	Unit
VBAT_DIGITAL	Digital used battery voltage input	3.4	3.8	4.2	V
VBAT_ANALOG	Analog used battery voltage input	3.4	3.8	4.2	V

Symbol or pin name	Description	Min.	Typ.	Max.	Unit
VBAT_SPK	VBAT input for loud speaker driver	3.4	3.8	4.2	V
VBAT_RF	RF used battery voltage input	3.4	3.8	4.2	V
VUSB	LDO output for USB-VUSB	3.0	3.3	3.6	V
AVDD28_FM	2.8V power supply for FM	2.6	2.8	3.0	V
AVDD28_2GAFE	2.8V power supply for 2G AFE	2.65	2.8	2.95	V
AVDD28_ABB	2.8V power supply for ABB	2.6	2.8	3.0	V
DVDD28	2.8V power supply for digital macros in transceiver	2.7	2.8	2.9	V
DVDD18	1.8V power supply for digital macros in transceiver	1.62	1.8	1.98	V
DVDD28_SF	2.8V IO power	2.7	3.3	3.6	V
	1.8V IO power	1.7	1.8	1.98	
DVDD33_MSDC	3.3V memory card power	3.0	3.3	3.6	V
DVDD18_EMI	1.8V EMI IO power	1.62	1.8	1.98	V
DVDD28_FSRC	E-FUSE blowing powr control	2.7	2.8	3.08	V
VDDK	1.2v core power	1.17	1.3	1.43	VDDK

Table 12. ***Recommended operating conditions for voltage input***

Symbol or pin name	Description	Min.	Typ.	Max.	Unit
VIN1	Digital input voltage for IO Type 1	-0.3	-	DVDIO+0.3	V
VIN2	Digital input voltage for IO Type 2	-0.3	-	DVDIO+0.3	V
VIN3	Digital input voltage for IO Type 3	-0.3	-	DVDIO+0.3	V
VIN4	Digital input voltage for IO Type 4	-0.3	-	DVDIO+0.3	V
VIN5	Digital input voltage for IO Type 5	-0.3	-	DVDIO+0.3	V
VIN6	Digital input voltage for IO Type 6	-0.3	-	DVDIO+0.3	V
VIN7	Digital input voltage for IO Type 7	-0.3	-	DVDIO+0.3	V

*Table 13****. Recommended operating conditions for operating temperature***

Symbol or pin name	Description	Min.	Typ	Max.	Unit
Tc	Operating temperature	-20	-	85	^{o}C

2.2.18 Electrical Characteristics under Recommended Operating Conditions

Table 14. ***Electrical characteristics***

Symbol	Description	Condition	Min.	Typ.	Max.	Unit

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
DIIH1	Digital high input current for IO Type 1	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN1 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1<VIN1<3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN1 < 2.1	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN1 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN1 < 2.1	-0.8	-	35	
DIIL1	Digital low input current for IO Type 1	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN1 < 0.7	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN1 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN1 < 0.7	-12.5	-	22.5	
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN1 < 0.45	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN1 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN1 < 0.45	-9.3	-	11.4	
DIOH1	Digital high output current for IO Type 1	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL1	Digital low output current for IO Type 1	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU1	Digital I/O pull-up	DVDIO = 2.8V	40	85	190	kΩ

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	resistance for IO Type 1	DVDIO = 1.8V	70	150	320	kΩ
DRPD1	Digital I/O pull-down resistance for IO Type 1	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DVOH1	Digital output high voltage for IO Type 1	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL1	Digital output low voltage for IO Type 1	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V
DIIH2	Digital high input current for IO Type 2	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN2 < 3.1	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN2 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1<VIN2<3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN2 < 2.1	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN2 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN2 < 2.1	-0.8	-	35	
DIIL2	Digital low input current for IO Type 2	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN2 < 0.7	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN2 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN2 < 0.7	-12.5	-	22.5	
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN2 < 0.45	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN2 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN2 < 0.45	-9.3	-	11.4	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
DIOH2	Digital high output current for IO Type 2	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL2	Digital low output current for IO Type 2	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU2	Digital I/O pull-up resistance for IO Type 2	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPD2	Digital I/O pull-down resistance for IO Type 2	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DVOH2	Digital output high voltage for IO Type 2	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL2	Digital output low voltage for IO Type 2	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V
DIIH3	Digital high input current for IO Type 3	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN3 < 3.1	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN3 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN3 < 3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN3 < 2.1	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN3 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN3 < 2.1	-0.8	-	35	
DIIL3	Digital low input current for IO Type 3	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN3 < 0.7	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN3 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN3 < 0.7	-12.5	-	22.5	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN3 < 0.45	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN3 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN3 < 0.45	-9.3	-	11.4	
DIOH3	Digital high output current for IO Type 3	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL3	Digital low output current for IO Type 3	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU3	Digital I/O pull-up resistance for IO Type 3	DVDIO = 2.8V	10	47	100	kΩ
		DVDIO = 1.8V	10	47	100	kΩ
DRPD3	Digital I/O pull-down resistance for IO Type 3	DVDIO = 2.8V	10	47	100	kΩ
		DVDIO = 1.8V	10	47	100	kΩ
DVOH3	Digital output high voltage for IO Type 3	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL3	Digital output low voltage for IO Type 3	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V
DIIH4	Digital high input current for IO Type 4	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN4 < 3.1	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN4 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN4 < 3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN4 < 2.1	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN4 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN4 < 2.1	-0.8	-	35	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
DIIL4	Digital low input current for IO Type 4	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN4 < 0.7	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN4 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN4 < 0.7	-12.5	-	22.5	
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN4 < 0.45	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN4 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN4 < 0.45	-9.3	-	11.4	
DIOH4	Digital high output current for IO Type 4	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL4	Digital low output current for IO Type 4	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU4	Digital I/O pull-up resistance for IO Type 4 (GPIO mode)	DVDIO = 2.8V	40	75	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPD4	Digital I/O pull-down resistance for IO Type 4 (GPIO mode)	DVDIO = 2.8V	40	75	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPU4 200K	Digital I/O pull-up resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	200	-	380	kΩ
DRPD4 200K	Digital I/O pull-down resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	200	-	380	kΩ
DVOH4	Digital output high voltage for IO Type 4	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL4	Digital output low voltage for IO Type 4	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V
DIIH5	Digital high input current for IO Type 5	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN5 < 3.1	-5	-	5	µA

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		PU enabled, DVDIO = 2.8V, 2.1 < VIN5 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN5 < 3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN5 < 2.1	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN5 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN5 < 2.1	-0.8	-	35	
DIIL5	Digital low input current for IO Type 5	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN5 < 0.7	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN5 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN5 < 0.7	-12.5	-	22.5	
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN5 < 0.45	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN5 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN5 < 0.45	-9.3	-	11.4	
DIOH5	Digital high output current for IO Type 5	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL5	Digital low output current for IO Type 5	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU5	Digital I/O pull-up resistance for IO Type 5 (GPIO mode)	DVDIO = 2.8V	40	75	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPD5	Digital I/O pull-down resistance for IO Type 5 (GPIO mode)	DVDIO = 2.8V	40	75	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
DRPU52K	Digital I/O pull-up resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	1	-	2	kΩ
DRPD52K	Digital I/O pull-down resistance for IO Type 4 (Key PAD mode)	DVDIO = 2.8V	1	-	2	kΩ
DVOH5	Digital output high voltage for IO Type 5	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL5	Digital output low voltage for IO Type 5	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V
DIIH6	Digital high input current for IO Type 6	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN6 < 3.1	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN6 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1 < VIN6 < 3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN6 < 2.1	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN6 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN6 < 2.1	-0.8	-	35	
DIIL6	Digital low input current for IO Type 6	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN6 < 0.7	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN6 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN6 < 0.7	-12.5	-	22.5	
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN6 < 0.45	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN6 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN6 < 0.45	-9.3	-	11.4	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
DIOH6	Digital high output current for IO Type 6	DVOH > 2.38V, DVDIO = 2.8V	-8	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-6	-	-	mA
DIOL6	Digital low output current for IO Type 6	DVOL < 0.42V, DVDIO = 2.8V	-	-	8	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	6	mA
DRPU6	Digital I/O pull-up resistance for IO Type 6	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPD6	Digital I/O pull-down resistance for IO Type 6	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DVOH6	Digital output high voltage for IO Type 6	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL6	Digital output low voltage for IO Type 6	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V
DIIH7	Digital high input current for IO Type 7	PU/PD disabled, DVDIO = 2.8V, 2.1 < VIN7 < 3.1	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, 2.1 < VIN7 < 3.1	-22.5	-	12.5	
		PD enabled, DVDIO = 2.8V, 2.1<VIN7<3.1	6.1	-	82.5	
		PU/PD disabled, DVDIO = 1.8V, 1.35 < VIN7 < 2.1	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, 1.35 < VIN7 < 2.1	-11.4	-	9.3	
		PD enabled, DVDIO = 1.8V, 1.35 < VIN7 < 2.1	-0.8	-	35	
DIIL7	Digital low input current for IO Type 7	PU/PD disabled, DVDIO = 2.8V, -0.3 < VIN7 < 0.7	-5	-	5	µA
		PU enabled, DVDIO = 2.8V, -0.3 < VIN7 < 0.7	-82.5	-	-6.1	
		PD enabled, DVDIO = 2.8V, -0.3 < VIN7 < 0.7	-12.5	-	22.5	

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
		PU/PD disabled, DVDIO = 1.8V, -0.3 < VIN7 < 0.45	-5	-	5	µA
		PU enabled, DVDIO = 1.8V, -0.3 < VIN7 < 0.45	-35	-	0.8	
		PD enabled, DVDIO = 1.8V, -0.3 < VIN7 < 0.45	-9.3	-	11.4	
DIOH7	Digital high output current for IO Type 7	DVOH > 2.38V, DVDIO = 2.8V	-16	-	-	mA
		DVOH > 1.53V, DVDIO = 1.8V	-12	-	-	mA
DIOL7	Digital low output current for IO Type 7	DVOL < 0.42V, DVDIO = 2.8V	-	-	16	mA
		DVOL < 0.27V, DVDIO = 1.8V	-	-	12	mA
DRPU7	Digital I/O pull-up resistance for IO Type 7	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DRPD7	Digital I/O pull-down resistance for IO Type 7	DVDIO = 2.8V	40	85	190	kΩ
		DVDIO = 1.8V	70	150	320	kΩ
DVOH7	Digital output high voltage for IO Type 7	DVDIO = 2.8V	2.38			V
		DVDIO = 1.8V	1.53			V
DVOL7	Digital output low voltage for IO Type 7	DVDIO = 2.8V			0.42	V
		DVDIO = 1.8V			0.27	V

2.3 System Configuration

2.3.1 Strapping Resistors

Table 15. Strapping table

Pin name	Description	Trapping condition
CMPDN	Pull-up/down with 75K resister	Power-on reset
CMRST	Pull-up with 10K resister (Default internal Pull-down with 75K resister)	Power-on reset/watchdog reset
BPI_BUS1	Pull-up with 10K resister (Default internal Pull-down with 75K resister)	Power-on reset
BPI_BUS3	Pull-up with 10K resister (Default internal	Power-on reset

Pin name	Description	Trapping condition
	Pull-down with 75K resister)	

2.3.2 Mode Selection

Table 16. Mode selection of chip

Pin name	Description
XTAL_SEL	GND: Uses external 32K crystal as RTC clock source VRTC: Uses internal 32K as RTC clock source
CMPDN	GND: Uses 1.8V device DVDD28: Uses 3.3V device
KCOL0	GND: Boots ROM to enter USB download mode DVDD28: Normal boot-up mode
CMRST	GND: Disables USB download sub-feature to support virtual 2 USB com port DVDD28: Enables USB download sub-feature to support virtual 2 USB com port
{BPI_BUS1,BPI_BUS3}	{GND, GND}: No JTAG {GND, DVDD28}: JTAG at keypad pins {DVDD28, GND}: JTAG at memory card pins {DVDD28, DVDD28}: JTAG at camera pins

2.3.3 Constant Tied Pins

Table 17. Constant tied pin of chip

Pin name	Description
TESTMODE	Tied to GND

2.4 Power-on Sequence and Protection Logic

MT6260D provides 32K crystal removal feature. The XOSC32_ENB state tells if MT6260D provides this feature or not. VRF will be turned on at the same time with VRTC when XOSC32_ENB = 1. The power-on/off sequence controlled by “Control” and “Reset Generator” is shown as the figure below.

Figure 5. Power-on/off control sequence by pressing PWRKEY and XOSC32_ENB = 0

Figure 6. Power-on/off control sequence by pressing PWRKEY and XOSC32_ENB = 1

Note that each of the above figures only shows one power-on/off condition when XOSC32_ENB = 0 or XOSC32_ENB = 1. MT6260D handles the power-on and off of the handset. The following three methods can switch on the handset (when leaving UVLO): XOSC32_ENB = 0

1. Push PWRKEY (Pull the PWRKEY pin to the low level.)
   Pulling PWRKEY low is the typical way to turn on the handset. The turn-on sequence is VCORE → VIO18 → VIO28 → VSF, VMC → VA → VUSB → VRF

The supplies for the baseband are ready, and the system reset ends at the moment when the above LDOs are fully turned on to ensure correct timing and function. After that, the baseband will send the PWRBB signal back to the PMU for acknowledgement. To successfully power on the handset, PWRKEY should be kept low until PMU receives PWRBB from the baseband.

2. RTC module generates PWRBB to wake up the system.
   If the RTC module is scheduled to wake up the handset at a certain time, the PWRBB signal will be directly sent to the PMU. In this case, PWRBB will become high at specific moment and allow the PMU to be powered on as the sequence described above. This is called the RTC alarm.

3. Valid charger plug-in (CHRIN voltage is within the valid range.)
   The charger plug-in will also turn on the handset if the charger is valid (no OVP takes place). However, if the battery voltage is too low to power on the handset (UVLO state), the system will not be turned on by any of the three methods. In this case, the charger will charge the battery first and the handset will be powered on automatically as long as the battery voltage is high enough.

Under-voltage lockout (UVLO)

The UVLO state in the PMU prevents startup if the initial voltage of the main battery is below the 3.2V threshold. It ensures that the handset is powered on with the battery in good condition. The UVLO function is performed by a hysteretic comparator which ensures a smooth power-on sequence. In addition, when the battery voltage is getting lower, it will enter the UVLO state, and the PMU will be turned off by itself, except for VRTC LDO, to prevent further discharging. Once the PMU enters the UVLO state, it will draw low quiescent current. The RTC LDO will still be working until the DDLO disables it.

Deep discharge lockout (DDLO)

The PMU will enter the deep discharge lockout (DDLO) state when the battery voltage drops below 2.5V. In this state, the VRTC LDO will be shut down. Otherwise, it will draw very low quiescent current to prevent further discharging or damage to the cells.

Reset

The PMU contains a reset control circuit which takes effect at both power-up and power-down. The RESETB pin is held low in the beginning of power-up and returns to high after the pre-determined delay time. The delay time is controlled by a large counter which uses the clock from internal ring-oscillator. At power-off, the RESETB pin will return to low immediately without any delay.

Over-temperature protection

If the die temperature of PMU exceeds 150°C, the PMU will automatically disable all the LDOs except for VRTC. Once the over-temperature state is resolved, a new power-on sequence will be required to enable the LDOs.

2.5 Analog Baseband

To communicate with analog blocks, a common control interface for all analog blocks is implemented. In addition, there are some dedicated interfaces for data transfer. The common control interface translates the APB bus write and read cycle for specific addresses related to analog front-end control.

During the writing or reading of any of these control registers, there is a latency associated with the transfer of data to or from the analog front-end. Dedicated data interface of each analog block is implemented in the corresponding digital block. An analog block includes the following analog functions for the complete GSM/GPRS baseband signal processing:

1. RF control: DAC for automatic power control (APC) is included, and its output is provided to external RF power amplifier respectively.
2. Auxiliary ADC: Provides an ADC for the battery and other auxiliary analog functions monitoring
3. Audio mixed-signal block: Provides complete analog voice signal processing including microphone amplification, A/D conversion, D/A conversion, earphone driver, etc. Dedicated stereo D/A conversion and amplification for audio signals are also included.
4. Clock generation: Includes a clock squarer for shaping the system clock, and PLL providing clock signals to DSP, MCU and USB unit
5. XOSC32: A 32-kHz crystal oscillator circuit for RTC applications on analog blocks

2.5.1 APC-DAC

2.5.1.1 Block Description

APC-DAC is a 10-bit DAC with output buffer aiming at automatic power control. See the tables below for its analog pin assignment and functional specifications. It is an event-driven scheme for power saving purpose.

2.5.1.2 Functional Specifications

Table 18. APC-DAC specifications

Symbol	Parameter	Min.	Typ.	Max.	Unit
N	Resolution		10		Bit
FS	Sampling rate			1.0833	MSPS
SINAD	Signal to noise and distortion ratio (10-kHz sine with 1.0V swing & 100-kHz BW)	47			dB
	99% settling time (full swing on maximal capacitance)			5	µS
	Output swing	0		AVDD	V
	Output capacitance		200	2200	pF
	Output resistance	0.47	10		KΩ
DNL	Differential nonlinearity for code 20 to 970		± 1		LSB
INL	Integral nonlinearity for code 20 to 970		± 1		LSB
DVDD	Digital power supply	1.1	1.2	1.3	V
AVDD	Analog power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	°C
	Current consumption Power-up Power-down		 400 1		 µA µA

2.5.2 Auxiliary ADC

2.5.2.1 Block Description

The auxiliary ADC includes the following functional blocks:

1. Analog multiplexer: Selects signal from one of the seven auxiliary input pins. Real-world messages to be monitored, e.g. temperature, should be transferred to the voltage domain.
2. 10-bit A/D converter: Converts the multiplexed input signal to 10-bit digital data.

Channel	Application	Input range [V]
0	BATSNS	3.2 ~ 4.2
1	ISENSE	3.2 ~ 4.2
2	VCDT	Decided by application circuit
3	BATON	0 ~ AVDD28
4	AUXIN4	0 ~ AVDD28
others	Internal use	N/A

2.5.2.2 Functional Specifications

The functional specifications of the auxiliary ADC are listed in the following table.

Table 19. Functional specifications of auxiliary ADC

Symbol	Parameter	Min.	Typ.	Max.	Unit
N	Resolution		10		Bit
FC	Clock rate		1.08		MHz
FS	Sampling rate @ N-Bit		1.08/(N+1)		MSPS
	Input swing	0		AVDD	V
CIN	Input capacitance Unselected channel Selected channel			 50 4	 fF pF
RIN	Input resistance Unselected channel Selected channel	 400 1			 MΩ MΩ
	Clock latency		N+1		1/FC
DNL	Differential nonlinearity		± 1		LSB
INL	Integral nonlinearity		± 1		LSB
OE	Offset error		± 10		mV
FSE	Full swing error		± 10		mV
SINAD	Signal to noise and distortion ratio (10-kHz full swing input & 1.0833-MHz clock rate)		50		dB
DVDD	Digital power supply	1.1	1.2	1.3	V
AVDD	Analog power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	°C

Symbol	Parameter	Min.	Typ.	Max.	Unit
	Current consumption Power-up Power-down		 280 1		 μA μA

2.5.3 Audio Mixed-Signal Blocks

2.5.3.1 Block Description

Audio mixed-signal blocks (AMB) integrate complete voice uplink/downlink and audio playback functions. As shown in the figure below, it includes three parts. The first consists of stereo audio DACs and audio amplifiers for audio playback. The second part is the voice downlink path, including voice-band DACs (left channel audio DAC) and voice amplifier, which produces voice signals to earphones or other auxiliary output devices. Amplifiers in the two blocks are equipped with multiplexers to accept signals from the internal audio/voice. Moreover, a ClassK amplifier is embedded to support continuous >1W output power with an on-chip charge-pump even under low battery scenario. The last part is the voice uplink path, which is the interface between the microphone (or other auxiliary input device) input and MT6260D DSP. A set of bias voltage is provided for the external electric microphone.

Figure 7. ***Block diagram of audio mixed-signal blocks***

2.5.3.2 Functional Specifications

See the table below for the functional specifications of voice-band uplink/downlink blocks.

Table 20. Functional specifications of analog voice blocks

Symbol	Parameter	Min.	Typ.	Max.	Unit
FS	Sampling rate		6,500		kHz
DVDD	Digital power supply	1.1	1.2	1.3	V
AVDD	Analog power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	°C
VMIC	Microphone biasing voltage		1.9	2.2	V
IMIC	Current draw from microphone bias pins			2	mA
Uplink path[4]					
IDC	Current consumption for one channel		1.5		mA
SINAD	Signal to noise and distortion ratio Input level: -40 dbm0 Input level: 0 dbm0	 29	 69		 dB dB
RIN	Input impedance (differential)	13	20	27	KΩ
ICN	Idle channel noise			-67	dBm0
Downlink path					
IDC	Current consumption		4		mA
SINAD	Signal to noise and distortion ratio Input level: -40 dBm0 Input level: 0 dBm0	 29	 69		 dB dB
RLOAD	Output resistor load (differential)	16	32		Ω
CLOAD	Output capacitor load			250	pF
ICN	Idle channel noise of transmit path			-67	dBm0
XT	Crosstalk level on transmit path			-66	dBm0

See the table below for the functional specifications of audio blocks.

Table 21. Functional specifications of analog audio blocks

Symbol	Parameter	Min.	Typ.	Max.	Unit
FCK	Clock frequency		6.5		MHz
Fs	Sampling rate	32	44.1	48	kHz
AVDD	Power supply	2.6	2.8	3.0	V
T	Operating temperature	-20		80	°C
IDC	Current consumption		4		mA
PSNR	Peak signal to noise ratio		88		dB
DR	Dynamic range		88		dB
VOUT	Output swing for 0dBFS input level @ -1dB		0.78		Vrms

[4] For uplink-path, not all gain settings of VUPG meet the specifications listed in the table, especially for several the lowest gains. The minimum gain that meets the specifications is to be determined.

Symbol	Parameter	Min.	Typ.	Max.	Unit
	headphone gain				
$VOUT_{MAX}$	Maximum output swing		2.4		Vpp
THD	Total harmonic distortion 10mW at 64Ω load			-70	dB
RLOAD	Output resistor load (single-ended)	64			Ω
CLOAD	Output capacitor load			250	pF
XT	L-R channel cross talk	70			dB

2.5.4 32-kHz Crystal Oscillator

2.5.4.1 Block Description

The low-power 32-kHz crystal oscillator XOSC32 is designed to work with an external piezoelectric 32.768 kHz crystal and a load composed of two functional capacitors. See the figure below.

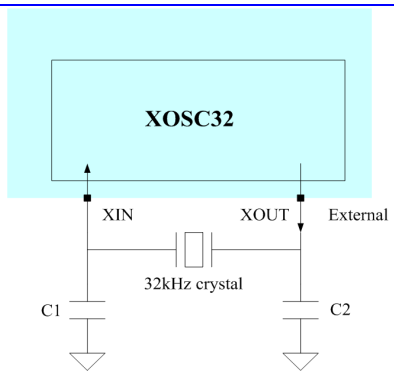


Figure 8. ***Block diagram of XOSC32***

2.5.4.2 Functional Specifications

See the table below for the functional specifications of XOSC32.

Table 22. Functional specifications of XOSC32

Symbol	Parameter	Min.	Typ.	Max.	Unit
AVDDRTC	Analog power supply	1	2.8		V
Tosc	Start-up time			1	sec
Dcyc	Duty cycle	30	50	70	%
	Current consumption			5	μA
T	Operating temperature	-25		70	°C

See the table below for recommendations on crystal parameters to use with XOSC32.

Table 23. Recommended parameters of 32kHz crystal

Symbol	Parameter	Min.	Typ.	Max.	Unit
F	Frequency range		32,768		Hz
GL	Drive level			1	uW
Δf/f	Frequency tolerance		± 20		ppm
ESR	Series resistance		50	70	KΩ
C0	Static capacitance		0.9		pF
CL	Load capacitance		12.5		pF

2.6 Power Management Unit Blocks

The power management unit (PMU) manages the power supply of the entire chip, such as baseband, processor, memory, SIM cards, camera, vibrator, etc. The digital part of PMU is integrated into the analog part (see the figure below). PMU includes the following analog functions for signal processing:

- LDO: Regulates battery voltage to lower voltage level
- Charge pump BOOST (CP-BOOST): Boosts battery voltage to target voltage for Class-AB audio amplifier
- Keypad LED driver (KPLED) and current sink (ISINK) switches: Sink current for keypad LED and LCM module
- Start-up (STRUP): Generates power-on/off control sequence of start-up circuits
- Pulse charger (PCHR): Controls battery charging

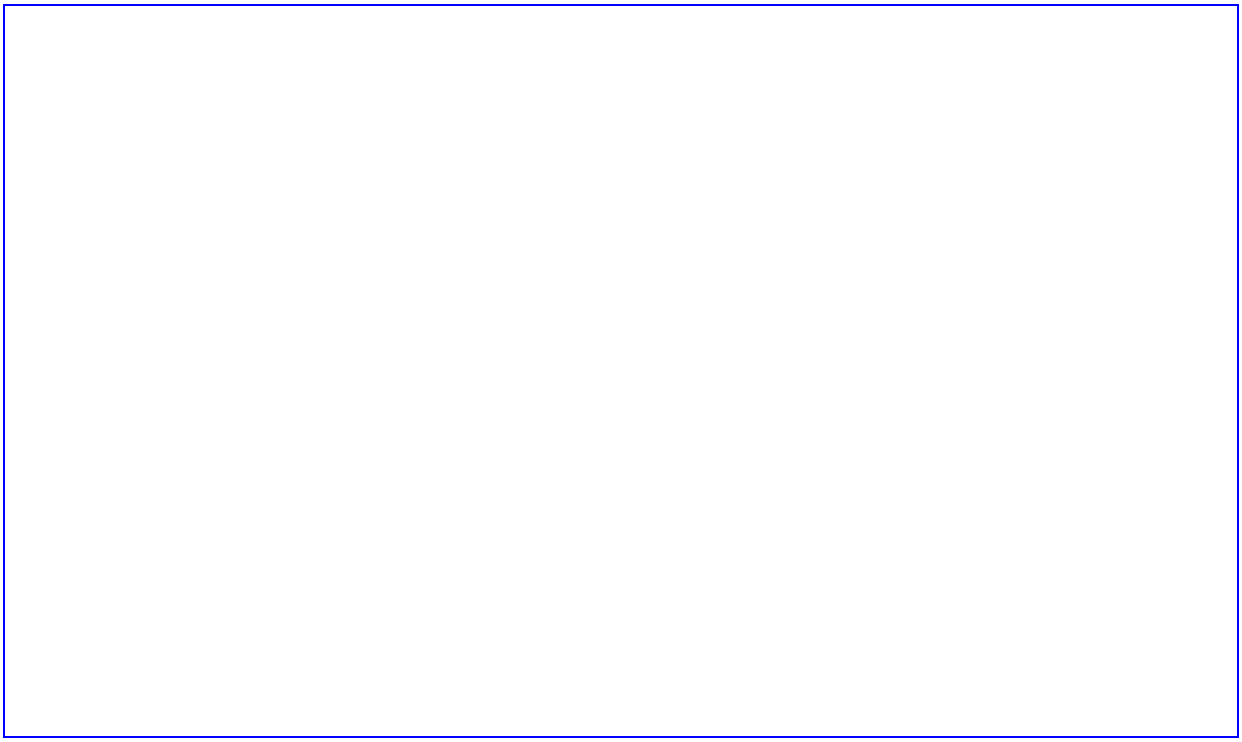


Figure 9. ***PMU system block diagram.***

2.6.1 LDO

PMU integrates 14 general low dropout regulators (LDO) optimized for their given functions by balancing the quiescent current, dropout voltage, line/load regulation, ripple rejection and output noise.

Figure 10. ***Power domain.***

2.6.1.1 LDO

The low-dropout regulator (LDO) is capable of maintaining its specified output voltage over a wide range of load current and input voltage, down to very small differences between input and output voltages.

There are several features in the design of LDO, including discharge control, soft start and current limit. Before LDO is enabled, the output pin of LDO should be discharged first to avoid voltage accumulation on the capacitance. The soft-start limits inrush current and controls output-voltage rising time during power-up. The current limit is the current protection to limit the LDO's output current and power dissipation.

There are three types of LDOs in PMU of MT6260D. The analog LDO is optimized for low-frequency ripple rejection in order to reject the ripples coming from the burst of RF power amplifier. The digital IO LDO is a linear regulator optimized for very low quiescent current. The single-step RTC LDO is a linear regulator that can charge up a capacitor-type backup coin cell, which also supplies the RTC module even at the absence of the main battery. The single-step LDO features reverse current protection and is optimized for ultra low quiescent current while sustaining the RTC function as long as possible.

2.6.1.1.1 Block Description

Figure 11. *LDO block diagram.*

2.6.1.1.2 LDO Types

Table 24. LDO types and brief specifications

Type	LDO name	Vout (Volt)	Imax(mA)	Description
ALDO	VRF	2.8	150	RF chip and 26MHz reference clock
ALDO	VA	2.8	150	Analog baseband
ALDO	VCAMA	2.8	150	Camera sensor
DLDO	VIO28	2.8	100	Digital IO and Blue tooth
DLDO	VSIM	1.8/3.0	30	SIM card
DLDO	VSIM2	1.8/3.0	30	SIM card
DLDO	VUSB	3.3	50	USB
DLDO	VIO18	1.8	100	Digital IO
DLDO	VCORE	0.75~1.3	250	Digital baseband
DLDO	VCAMD	1.8/2.8	100	Camera sensor
DLDO	VIBR	1.8/2.8/3.0	150	Vibrator, drop out <400mV at 150mA loading
DLDO	VMC	2.8/3.0/3.3	200	Memory card, drop out <400mV at 200mA loading

Type	LDO name	Vout (Volt)	Imax(mA)	Description
DLDO	VSF	1.86/3.3	50	General purpose LDO
RTCLDO	VRTC	2.8	2	Real-time clock

2.6.1.1.3 Functional Specifications

Table 25. Analog LDO specification.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Load capacitor			1		µF
	Vout	Includes load regulation, line regulation, and temperature coefficient	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Transient response	Slew: 15mA/us	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Temperature coefficient				100	ppm/C
	PSRR	Iout < 0.5*Imax 10 < f < 3 kHz	65			dB
		Iout < 0.5*Imax 3K < f< 30 kHz	45			dB
	Output noise	With A-weighted filter			90	uVrms
	Quiescent current	Iout = 0		55		µA
	Turn-on overshoot	Iout = 0			Max. (+10%, +0.1V)	V
	Turn-on settling time	Iout = 0			240	µsec

Table 26. Digital LDO specification.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Load capacitor			1[5]		µF
	Vout	Includes load regulation, line regulation, and temperature coefficient	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Transient response	Slew: 15mA/us	Max. (-5%, -0.1V)		Max. (+5%, +0.1V)	V
	Temperature coefficient				100	ppm/C
	Quiescent current	Iout = 0		15		µA
	Turn-on overshoot	Iout = 0			Max.	V

[5] VCORE loading capacitor typical value is 2.2uF. Other LDOs are 1uF.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
					(+10%, +0.1V)	
	Turn-on settling time	Iout = 0			240	µs

Table 27. RTC LDO specification.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Load capacitor			1		µF
	Vout	Includes load regulation, line regulation, and temperature coefficient	2	2.8	3	V
	Temperature coefficient				100	ppm/C
	Quiescent current	Iout = 0		15		µA

2.6.2 Charge Pump BOOST

2.6.2.1 Functional Specifications

Table 28 . Charge pump BOOST specification

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Cfly Chold			4.7 4.7		µF
	Vout			4.2		V
	Ripple	Vin=3.4~4.05V Cfly/Cout=4.7uF/4.7uF @ Iload=250mA			200	mV
	Switching frequency			600		KHz
	Quiescent current	Iout = 0		4	6	mA

2.6.3 ISINK and KPLED Switches

One built-in open-drain output switch drives the keypad LED (KPLED) in the handset. The switch is controlled by the baseband with enabling registers. The switch of keypad LED can sink as much as 60mA current, and the output is high impedance when disabled. The value of the sink current decides the brightness of the LED.

Four current controlled open drain drivers (ISINK0 ~ 3) are also implemented to drive the LCM backlight module, and each channel provides 6-current-level steps of up to 24mA. The maximum current-level can be extended to 48mA by doubling the current-level through register settings.

2.6.3.1 Block Description

Figure 12. *ISINKs and KPLED switches bock diagram.*

2.6.3.2 Functional Specifications

Table 29. ISINKs and KPLED Switches Specification.

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Sink current of keypad LED driver	Von > 0.5V, 100% dimming duty	60			mA
	Sink current of ISINK0 ~ 3	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 000		4		mA
	Sink current of ISINK0 ~ 3	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 001		8		mA
	Sink current of ISINK0 ~ 3	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 010		12		mA
	Sink current of ISINK0 ~ 3	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 011		16		mA
	Sink current of ISINK0 ~ 3	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 100		20		mA
	Sink current of ISINK0 ~ 3	Von > 0.15V, 100% dimming duty, ISINKS_CHx_STEP = 101		24		mA
	Current mismatch	Von > 0.15V, 100% dimming	-5		5	%

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	between the 4 channels	duty				

2.6.4 STRUP

PMU handles the power-on and off of the handset. If the battery voltage is neither in the UVLO state (VBAT ≥ 3.4V) nor in the thermal condition, there are three methods to power on the handset system: pulling PWRKEY low (the user pushes PWRKEY), pulling PWRBB high (baseband BB_WakeUp) or valid charger plug-in.

According to different battery voltage (VBAT) and phone states, control signals and regulators will have different responses.

2.6.5 PCHR

The charger controller senses the charger input voltage from either a standard AC-DC adaptor or an USB charger. When the charger input voltage is within a pre-determined range, the charging process will be activated. This detector can resist higher input voltage than other parts of the PMU(30V for max. rating voltage, 7V for max. operation range).

2.6.5.1 Block Description

Figure 13. *PCHR block diagram.*

2.6.5.1.1 Charger Detection

Whenever an invalid charging source is detected (> 7.0 V), the charger detector stops the charging process immediately to avoid burning out the chip or even the phone. In addition, if the charger-in level is not high enough (< 4.3V), the charger will also be disabled to avoid improper charging behavior.

2.6.5.1.2 Charging Control

When the charger is active, the charger controller manages the charging phase according to the battery status. During the charging period, the battery voltage is constantly monitored. The battery charger supports pre-charge mode (VBAT < 3.2V, PMU power-off state), CC mode (constant current mode or fast charging mode at the range 3.2V < VBAT < 4.2V) and CV mode (constant voltage mode) to optimize the charging procedure for Li-ion battery. The charging states diagram is shown in the figure below.

Figure 14*. Charging states diagram*

Pre-charge mode

When the battery voltage is in the UVLO state, the charger operates in the pre-charge mode. There are two steps in this mode. When the battery voltage is deeply discharged below 2.2V, PRECC0 trickle charging current will be applied to the battery.

The PRECC1 trickle charging current is about 550ms pulse 70mA current when VBAT is under 2.2V.

When the battery voltage exceeds 2.2V, called the PRECC2 stage, the closed-loop pre-charge is enabled. The voltage drop across the external RSENSE is kept around 40mV (AC charger) or 14mV (USB host). The closed-loop pre-charge current can be calculated:

Constant current mode

As the battery is charged up and over 3.4V, it can switch to the CC mode. (CHR_EN should be high) In the CC mode, several charging currents can be set by programming registers or the external RSENSE resistor. The charging current can be determined by CS_VTH/RSENSE, where CS_VTH is programmed by registers. For example, if RSENSE is selected as 0.2ohm, the CC mode charging current can be set from 70 to 800mA. It can accommodate the battery charger to various charger inputs with different current capabilities.

Constant voltage mode and over-voltage protection (OV)

While the battery voltage reaches about 4.2V, a constant voltage is used for charging. This is called the full-voltage charging mode or constant-voltage charging mode in correspondence to a linear charger. While the battery voltage actually reaches 4.2V, the charging current is gradually decreased step-by-step, the end-of-charging process starts. It may prolong the charging and detecting period for acquiring optimized full charging volume. The charging process is completed once the current reaches zero automatically and this mechanism is optimized for different battery

BC1.1 Dead-Battery Support of China Standard

MT6260D supports dead-battery condition from China standard (called BC1.1). The specification protects dead-battery charging by timer and trickle current. Once the battery voltage is below 2.2V, a period (TUNIT) of trickle current (IUNIT) will be applied to the battery.

If the battery voltage is still below 2.2V after applying trickle current, the charger will be disabled. On the other hand, if the battery voltage is raised to above 2.2V, the charger will enter the PRECC2 stage, and the charging current will be 70mA or 200mA depending on the type of charging port.

Under the condition of battery voltage from 2.2V to 3.3V, the charger will charge the battery with the PRECC2 current.

A dedicated 5mins (T1) timer will be timed out and disable the charger if the battery voltage is always below 2.7V under charging. Another 35mins (T2) timer will also be timed out and disable the charger if the battery voltage is always kept between 2.7V and 3.3V under charging.

The trickle current (IUNIT) and two dedicated timers protect the charging action if the battery is dead.

2.6.5.2 Functional Specifications

Table 30. Charger detection specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Charger detect-on range		4.3		7	V

Table 31. Pre-charge specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	IUNIT with 500ms pulse	VBAT < 2.2V	28	56	84	mA
	Pre-charging current	VBAT < 2.2V (500ms pulse)	28	56	84	mA
		VBAT ≥ 2.2V (USB host)	$7/R_{sense}$	$14/R_{sense}$	$20/R_{sense}$	mA
		VBAT ≥ 2.2V (AC adapter < 7V)	$30/R_{sense}$	$40/R_{sense}$	$50/R_{sense}$	mA
		VBAT ≥ 2.2V (AC adapter > 7V)	$7/R_{sense}$	$14/R_{sense}$	$20/R_{sense}$	mA
	Pre-charging off threshold	CHR_EN = L		3.3		V
	Pre-charging off hysteresis			0.4		V

- *Table 32.* Constant current specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	CC mode charging current (CS_VTH)	CS_VTH [2:0] = 000		$160/R_{sense}$		mA
		CS_VTH [2:0] = 001		$140/R_{sense}$		mA
		CS_VTH [2:0] = 010		$130/R_{sense}$		mA
		CS_VTH [2:0] = 011		$110/R_{sense}$		mA
		CS_VTH [2:0] = 100		$90/R_{sense}$		mA
		CS_VTH [2:0] = 101		$60/R_{sense}$		mA
		CS_VTH [2:0] = 110		$40/R_{sense}$		mA
		CS_VTH [2:0] = 111		$14/R_{sense}$		mA
	Current sensing resistor	RSENSE		0.2		ohm

Table 33. ***Constant voltage and over-voltage protection specifications***

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Charging complete threshold		4.15	4.2	4.25	V
	Battery over-voltage protection threshold (OV)			4.3		V

Table 34. BC1.1 specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
IUNIT	BC1.1 trickle current	VBAT < 2.2V		56	100	mA
IPRECC1 (USB host)	PRECC1 current	2.2 < VBAT < 3.3V		70	100	mA
IPRECC1 (AC adapter)	PRECC1 current	2.2 < VBAT < 3.3V		200	250	mA
T1	5 minute dedicated timer	2.2 < VBAT < 2.7V		5	7	min.
T2	35 minute dedicated timer	2.7 < VBAT < 3.3V		35	36	min.
TUNIT	BC1.1 trickle current period				1	sec.

USBDL without battery

The flash tool can download firmware from PC to cell phone without plugging in a valid battery (i.e. > UVLO, 3.2V) through this feature named “USBDL w/o battery”. This feature provides the needed power from the adaptor to the system. MT6260D provides two methods for downloading without battery: Auto-Power-On (APO) and pressing download key (DLKEY). APO is activated automatically when no battery is inserted and an valid adaptor is plugged in. APO can source a typical current level of 450mA from the adaptor, and the current can be confined within a maximum level determined by a fixed and pre-defined setting and the external power device’s current gain (i.e. β of the BJT). You can also enable USBDL w/o battery by pressing DLKEY. This auxiliary DL function supports a system with a battery that has no NTC pin. For this specific scenario, the BATON pin of MT6260D is recommended to be tied to the ground. On top of that, pressing DLKEY is the only way to conduct USBDL. Moreover, the current level is the same as that of APO. The current lasts for 8 seconds and can be programmed once SW can control the system. During DL, VBAT is limited by a default OV level of 4.2V to avoid over-stressing on the internal components.

2.7 GSM/GPRS/EDGE-Rx RF

2.7.1 General Description

2G RFSYS which is built in MT6260D SOC is a highly integrated RF transceiver for multi-band GSM, GPRS and EDGE-Rx cellular systems.

The features include:

Receiver

- Single-end saw-less Rx
- Quadrature RF mixer
- Fully integrated channel filter

- High dynamic range ADC
- 12dB PGA gain with 6dB gain step

Transmitter

- High accurate transmitter modulator for GSM/GPRS application
- Built-in calibration of SX loop filter and loop gain

Frequency synthesizer

- Programmable fractional-N synthesizer
- Integrated wide range RFVCO
- Integrated loop filter
- Fast settling time suitable for multi-slot GSM/GPRS/EDGE-Rx applications

Digitally-Controlled Crystal Oscillator (DCXO)

- Two-pin26 MHz crystal oscillator
- On-chip programmable capacitor array for coarse-tuning
- On-chip programmable capacitor array for fine-tuning
- Supports 32K XTAL-less operation

2.7.2 Functional Block Diagram

Figure 15. ***Diagram of MT6260D 2G RFSYS***

2.7.3 Electrical Characteristics

Table 35. DC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

RFSYS mode	VRF	AVDD28_2GAFE	RFSYS total	Unit
BCM_Deep sleep (DCXO is off)	17	1	18	uA
BCM_Sleep (DCXO is on)	2.8	0.26	3.1	mA
Low power mode	65	1	66	uA
Full power mode	2.8	0.26	3.1	mA
RX (GSM850/EGSM)	64	5	69	mA
RX (DCS/PCS)	74	5	79	mA
TX (GSM850/EGSM)	52	2	54	mA
TX (DCS/PCS)	47	2	49	mA

Table 36. Rx AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
Input frequency	F_{RX}	GSM850		869		894	MHz
		GSM900		925		960	MHz
		DCS1800		1,805		1,880	MHz
		PCS1900		1,930		1,990	MHz
Voltage gain 1	G_1	GSM850	LNA = High gain	52[1]	55		dB
		GSM900	PGA = Middle high gain	52[2]	55		dB
		DCS1800	LNA = High gain	52[3]	55		dB
		PCS1900	PGA = Middle high gain	52[4]	55		dB
Voltage gain 2	G_2	GSM850	LNA = Middle high gain		51		dB
		GSM900	PGA = Middle high gain		51		dB
		DCS1800	LNA = Middle high gain		53		dB
		PCS1900	PGA = Middle high gain		53		dB
Voltage gain 3	G_3	GSM850	LNA = Middle gain		49		dB
		GSM900	PGA = Middle high gain		49		dB
		DCS1800	LNA = Middle gain		50		dB
		PCS1900	PGA = Middle high gain		50		dB
Voltage gain 4	G_4	GSM850	LNA = Low gain		26.5		dB
		GSM900	PGA = Middle high gain		26.5		dB
		DCS1800	LNA = Low gain		28		dB
		PCS1900	PGA = Middle high gain		28		dB
Noise figure at 25°C	NF_{25}	GSM850	G_1		4	6[1]	dB
		GSM900			4	6[2]	dB
		DCS1800			4	6[3]	dB
		PCS1900			4	6[4]	dB
Noise figure at 85°C	NF_{85}	GSM850	G_1		5		dB
		GSM900			5		dB
		DCS1800			5		dB
		PCS1900			5		dB
2^{nd}-order input intercept point	IIP2	GSM850	G2	31[1]	43		dBm
		GSM900		31[2]	43		dBm
		DCS1800		31[3]	43		dBm
		PCS1900		31[4]	43		dBm
3^{rd}-order input intercept point	IIP3	GSM850	G2	-14[1]	-3		dBm
		GSM900		-14[2]	-3		dBm
		DCS1800		-14[3]	-3		dBm

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
		PCS1900		-14[4]	-3		dBm
3rd-order input intercept point @ -20°C	$IIP3_{-20}$	GSM850	G2		-5		dBm
		GSM900			-5		dBm
		DCS1800			-5		dBm
		PCS1900			-5		dBm
Receiver S/N with 3MHz blocker	SN_{3M}	GSM850	G2	8[1]	12		dB
		GSM900	Blocker = -23dBm	8[2]	12		dB
		DCS1800	G2	8[3]	12		dB
		PCS1900	Blocker = -26dBm	8[4]	12		dB
Receiver S/N with OBB	SN_{OOB}	GSM850	G3	6[5]	8		dB
		GSM900	Blocker = 2dBm, offset 20MHz	6[5]	8		dB
		DCS1800	G3	6[5]	8		dB
		PCS1900	Blocker = 2dBm, offset 80MHz	6[5]	8		dB
Image rejection ratio	IRR	ALL	G2	32[1,2,3,4]	40		dB
Receiver channel response attenuation		ALL	@3MHz offset		20		dB
			@6MHz offset		35		dB
Receiver filtering 3-dB bandwidth		ALL	For all gain settings		900		kHz
PGA gain linearity		ALL	INL		0.2	1[5]	dBΩ
			DNL		0.1	0.5[5]	dBΩ
PGA gain step		ALL			6		dBΩ
PGA dynamic range		ALL			12		dBΩ
I/Q common-mode output voltage		ALL	G1	1.1[5]	1.2	1.3[5]	V
Output static dc offset		ALL	G1		100	200	mV

Table 37. Tx GMSK AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
Frequency	F_{TX}	GSM850		824		849	MHz
		GSM900		880		915	MHz
		DCS1800		1,710		1,785	MHz
		PCS1900		1,850		1,910	MHz

Item	Symbol	Band	Test condition	Min.	Typ.	Max.	Unit
RMS phase error	PE_{rms}	GSM850 GSM900			1.5	2.5[1,2]	degree
		DCS1800 PCS1900			1.5	2.5[3,4]	degree
Output modulation spectrum	ORFS	GSM850 GSM900	400kHz offset (RBW = 30kHz bandwidth)		-66	-64[1,2]	dBc
		DCS1800 PCS1900			-66	-64[3,4]	dBc
		GSM850 GSM900	1.8MHz offset (RBW = 30kHz bandwidth)			-75[5]	dBc
		DCS1800 PCS1900				-75[5]	dBc
Tx noise in Rx band		GSM850	20MHz offset		-165	-164[5]	dBc/Hz
			35MHz offset		-168	-167[5]	dBc/Hz
		GSM900	20MHz offset		-165	-164[5]	dBc/Hz
			35MHz offset		-168	-167[5]	dBc/Hz
		DCS1800	20MHz offset		-160	-156[5]	dBc/Hz
		PCS1900	20MHz offset		-160	-156[5]	dBc/Hz
Output power level	P_{out}	GSM850 GSM900	PA driver amplifier R_{load} = 50Ω	1[1,2]	3	6[1,2]	dBm
		DCS1800 PCS1900		1[3,4]	3	6[3,4]	dBm
Output 3rd harmonics		ALL	PA driver amplifier		-10		dBc

Table 38. SX AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Test condition	Min.	Typ.	Max.	Unit
Frequency range	F_{range}		3,296		3,980	MHz
Reference frequency	F_{ref}			26		MHz
Frequency step resolution	F_{res}			3		Hz
Phase noise	PN_{10k}	@ 10kHz offset		-83		dBc/Hz
	PN_{400k}	@ 400kHz offset		-116		dBc/Hz
	PN_{3M}	@ 3MHz offset		-136		dBc/Hz
Lock time of Rx burst	T_{lock_rx}	Frequency error < ± 0.1ppm		150	200[5]	us
Lock time of Tx burst	T_{lock_tx}	Frequency error < ± 0.1ppm		200	300[5]	us
Pushing figure		With internal RFVCO LDO		400		kHz/V

Table 39. DCXO AC characteristics (TA = 25oC, VDD = 2.8V unless otherwise stated)

Item	Symbol	Test condition	Min.	Typ.	Max.	Unit
Operating frequency	F_{ref}			26		MHz
Crystal C load	C_L			7.5		pF
Crystal tuning sensitivity	T_S		27.5	32.3		ppm/pF
Static range	SR	CDAC from 0 to 255	± 22	± 50		ppm
Dynamic range	DR	CAFC from 0 to 8191	36	50		ppm
AFC tuning step	$F_{res\text{-}AFC}$			0.006		ppm/DAC
AFC settling time	T_{AFC}	CAFC from 0 to 8191 CAFC from 8191 to 0 Frequency error < 0.1ppm		100	200[5]	us
Start-up time	T_{DCXO}	Frequency error < 1ppm Amplitude > 90 %			2[5]	ms
Pushing figure				0.2		ppm/V
Fref buffer output level	V_{Fref}	Max. loading = 19pF	0.8[5]			$V_{p\text{-}p}$
Fref buffer output phase noise		10kHz offset Jitter noise		-140		dBc/Hz

[1, 2]: Tested at E-GSM Tx channel 0 and GSM850 Rx channel 190.

[3, 4]: Tested at PCS Tx channel 601 and DCS Rx channel 636.

[5]: Not subject to production test – verified by characterization and design.

2.8 Bluetooth

2.8.1 Block Description

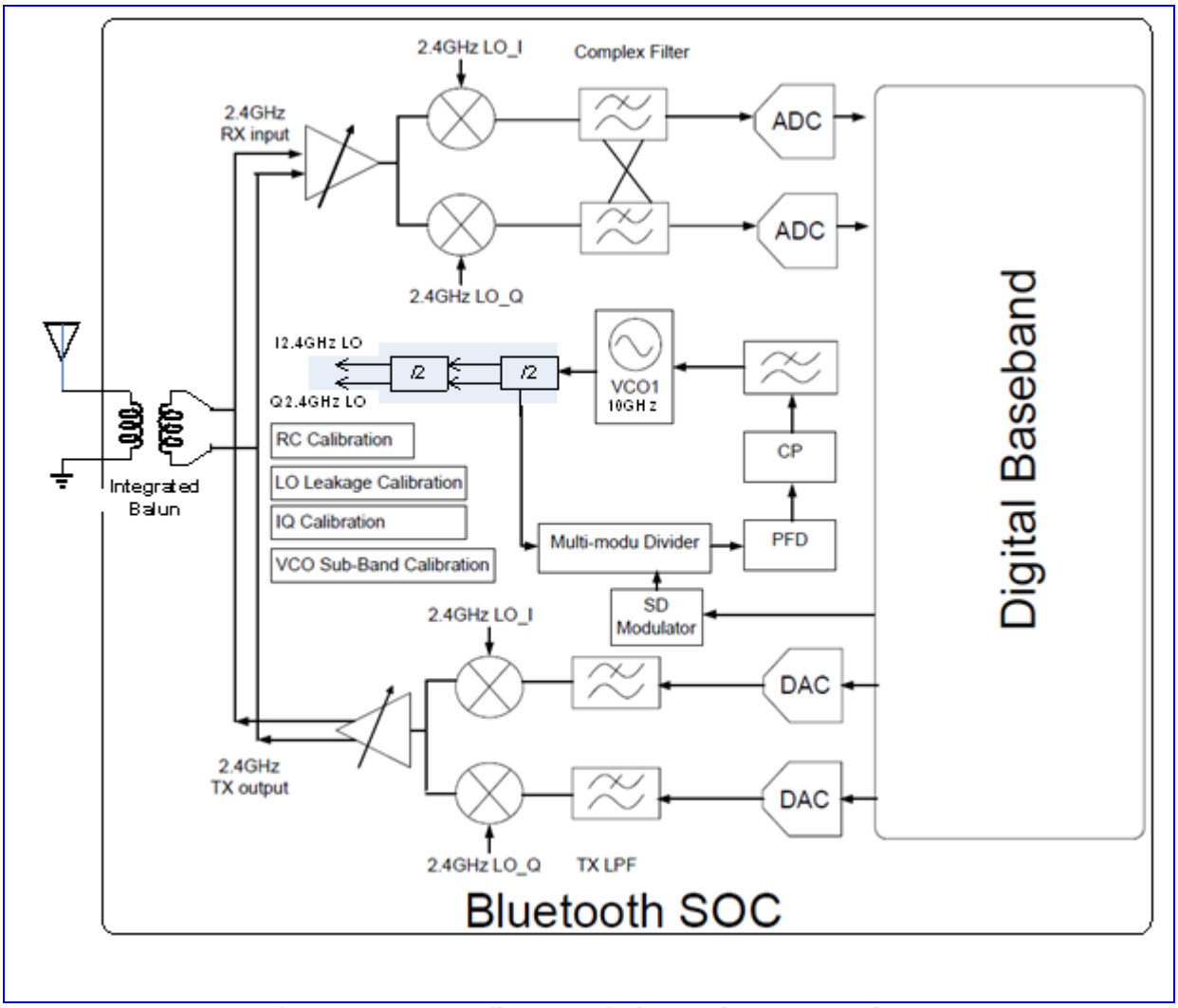


Figure 16. ***System diagram of Bluetooth RF transceiver***

The Bluetooth RF subsystem contains a fully integrated transceiver.

For TX path, the baseband transmit data are digitally modulated in the baseband processor then up-converted to 2.4GHz RF channels through the DA converter, filter, IQ up-converter and the power amplifier. The power amplifier is capable of transmitting 10dBm power for class-1.5 operation.

For RX path, MT6260D is a low IF receiver architecture. An image-reject mixer down-converts the RF signal to the IF with the LO from the synthesizer, which supports different clock frequencies as the reference clock. The mixer output is then converted to digital signal and down-converted to baseband for demodulation. A fast AGC enables the effective discovery of device within the dynamic range of the receiver.

MT6260D features self calibration schemes to compensate the process and temperature variation to maintain high performance. Those calibrations are performed automatically right after system boot-up.

2.8.2 Functional Specifications

2.8.2.1 Basic Data Rate – Receiver Specifications

Table 40. Basic data rate – receiver specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Receiver sensitivity	BER < 0.1%	-	-95	-	dBm
	Max. detectable input power	BER < 0.1%	-	0	-	dBm
	C/I co-channel selectivity	BER < 0.1%	-	4	-	dB
	C/I 1 MHz adj. channel selectivity	BER < 0.1%	-	-12	-	dB
	C/I 2 MHz adj. channel selectivity	BER < 0.1%	-	-47.5	-	dB
	C/I ≥ 3 MHz adj. channel selectivity	BER < 0.1%	-	-46	-	dB
	C/I image channel selectivity	BER < 0.1%	-	-24	-	dB
	C/I image 1 MHz adj. channel selectivity	BER < 0.1%	-	-45	-	dB
	Out-of-band blocking	30 to 2,000 MHz	-	-4	-	dBm
		2,000 to 2,399 MHz	-	-18	-	dBm
		2,498 to 3,000 MHz	-	-18	-	dBm
		3,000 MHz to 12.75 GHz	-	1	-	dBm
	Intermodulation		-	-22	-	dBm

2.8.2.2 Basic Data Rate – Transmitter Specification

Table 41. Basic data rate – transmitter specification

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Maximum transmit power		-	9.5	-	dBm
	Gain step		-	4	-	dB
	Δf1avg (00001111)		140	158	175	kHz
	Δf2max (10101010)		115	130	-	kHz
	Δf1avg/Δf2avg		0.8	0.9	-	kHz
	Initial carrier frequency drift		-75	5	75	kHz
	Frequency drift	DH1	-25	9	25	kHz
		DH3	-40	10	40	kHz
		DH5	-40	10	40	kHz
	Max. drift rate		-	100	400	Hz/µs
	BW_{20dB} of Tx output spectrum		-	920	1,000	kHz

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	In-band spurious emission	±2 MHz offset	-	-38	-	dBm
		±3 MHz offset	-	-43	-	dBm
		> ±3 MHz offset	-	-43	-	dBm
	Out-of-band spurious emission	30 MHz to 1 GHz	-	-36	-	dBm
		1 to 12.75 GHz	-	-30	-	dBm
		1.8 to 1.9 GHz	-	-47	-	dBm
		5.15 to 5.3 GHz	-	-47	-	dBm

2.8.2.3 Enhanced Data Rate – Receiver Specifications

Table 42. Enhanced data rate – receiver specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Receiver sensitivity	π/4 DQPSK, BER < 0.01%	-	-95	-	dBm
		8PSK, BER < 0.01%	-	-88	-	dBm
	Max. detectable input power	π/4 DQPSK, BER < 0.01%	-	-4.5	-	dBm
		8PSK, BER < 0.01%	-	-4.5	-	dBm
	C/I co-channel selectivity	π/4 DQPSK, BER < 0.01%	-	8	-	dB
		8PSK, BER < 0.01%	-	14.5	-	dB
	C/I 1MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-13	-	dB
		8PSK, BER < 0.01%	-	-7	-	dB
	C/I 2MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-47.5	-	dB
		8PSK, BER < 0.01%	-	--41.5	-	dB
	C/I ≥ 3MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-48	-	dB
		8PSK, BER < 0.01%	-	-44.5	-	dB
	C/I image channel selectivity	π/4 DQPSK, BER < 0.01%	-	-30	-	dB
		8PSK, BER < 0.01%	-	-23	-	dB
	C/I image 1 MHz adj. channel selectivity	π/4 DQPSK, BER < 0.01%	-	-47.5	-	dB
		8PSK, BER < 0.01%	-	-44.5	-	dB

2.8.2.4 Enhanced Data Rate – Transmitter Specifications

Table 43. Enhanced data rate – transmitter specifications

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Frequency range		2,402	-	2,480	MHz
	Max. transmit power	π/4 DQPSK	-	6.5	-	dBm
		8PSK	-	6.5	-	dBm
	Relative transmit power	π/4 DQPSK	-	-1.7	-	dB
		8PSK	-	-1.7	-	dB

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Freq. stability ω_0	π/4 DQPSK	-	1.5	-	kHz
		8PSK	-	1.5	-	kHz
	Freq. stability ω_1	π/4 DQPSK	-	3	-	kHz
		8PSK	-	3	-	kHz
	$\lvert \omega_0+\omega_1 \rvert$	π/4 DQPSK	-	2.8	-	kHz
		8PSK	-	2.8	-	kHz
	RMS DEVM	π/4 DQPSK	-	5.4	-	%
		8PSK	-	5.7	-	%
	99% DEVM	π/4 DQPSK	-	10	-	%
		8PSK	-	11	-	%
	Peak DEVM	π/4 DQPSK	-	18	-	%
		8PSK	-	18	-	%
	In-band spurious emission	π/4 DQPSK, ±1 MHz offset	-	-28	-	dBc
		8PSK, ±1 MHz offset	-	-28	-	dBc
		π/4 DQPSK, ±2 MHz offset	-	-25	-	dBm
		8PSK, ±2 MHz offset	-	-25	-	dBm
		π/4 DQPSK, ±3 MHz offset	-	-40.5	-	dBm
		8PSK, ±3 MHz offset	-	-40.5	-	dBm

Note: To meet this specification, use a front-end band-pass filter.

2.9 FM RF

2.9.1 Block Description

The connection between internal modules, as well as, external interfaces can be found in Figure 17. The FM receiver section incorporates the complete receiving path with wide tuning range. The FM baseband signal processor incorporates the digital demodulator and audio processing function which provides superior audio quality.

FM contains completely integrated FM audio receiver functions (RDS/RBDS may also be supported depending on the model number). The integrated receiver enables superior sensitivity, ACI performance and FM audio performances with minimum external BOM.

The FM subsystem supports either high performance stereo analog line out or digital audio output(I2S).

For models supporting RDS/RBDS, large dedicated internal data buffers are allocated to reduce the frequency of the interrupt to the host, so that the receiving host can enter low power states efficiently.

Figure 17. Block diagram of hardware top-level architecture

2.9.2 Functional Specifications

Table 1. FM receiver DC characteristics (TA=25°C, VDD=2.8V unless otherwise stated)

Operating mode	Current consumption	Unit
Idle	5	μA
FM receiver	12	mA

Unless otherwise stated, all receiver characteristics are applicable to both long and short antenna ports when operated under the recommended operating conditions. Typical specifications are for channel 98.7 MHz, default register settings and under recommended operating conditions. The minimum and maximum specifications are for extreme operating voltage and temperature conditions, unless otherwise stated.

Table 2. FM receiver AC characteristics

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	Input frequency range		65		108	MHz
	Sensitivity (long antenna)[1,3]	(S+N)/N=26dB, unmatched		3		dBμVemf
		(S+N)/N = 26dB, matched		2		dBμVemf
	RDS sensitivity (long antenna)	Δf=2kHz,BLER<5%, unmatched		18		dBμVemf
	Sensitivity (short antenna)[1,3]	(S+N)/N=26dB, unmatched		3		dBμVemf
	RDS sensitivity (short antenna)	Δf=2kHz,BLER<5%, unmatched		18		dBμVemf
	LNA input resistance[4]			2.4k		Ohm

Symbol	Description	Condition	Min.	Typ.	Max.	Unit
	LNA input capacitance[4]			8		pF
	AM suppression[1,4]	M = 0.3		58		dB
	Adjacent channel selectivity[1,4]	±200 kHz		53		dB
	Alternate channel selectivity[1,4]	±400 kHz		65		dB
	Spurious response rejection[4]	In-band		55		dB
	Maximum input level			117		dBμVemf
	Audio mono (S+N+D)/(N+D)[1,3,4]			60		dB
	Audio stereo (S+N+D)/(N+D)[2,3,4]			52		dB
	Audio stereo separation[4]	Δf = 22.5 kHz		45		dB
	Audio output load resistance	Single-ended at AFR/AFL outputs		10k		Ohm
	Audio output load capacitance	Single-ended at AFR/AFL outputs		12.5		pF
	Audio output voltage[1,4]	At AFR/AFL outputs		80		mVrms
	Audio output THD[1,4]			0.05	0.1	%
	Audio output frequency range	3dB corner frequency	30		15k	Hz

[1] Δf = 22.5kHz, fm = 1kHz, 50μs de-emphasis, mono, L = R

[2] Δf = 22.5kHz, fm = 1kHz, 50us de-emphasis, stereo

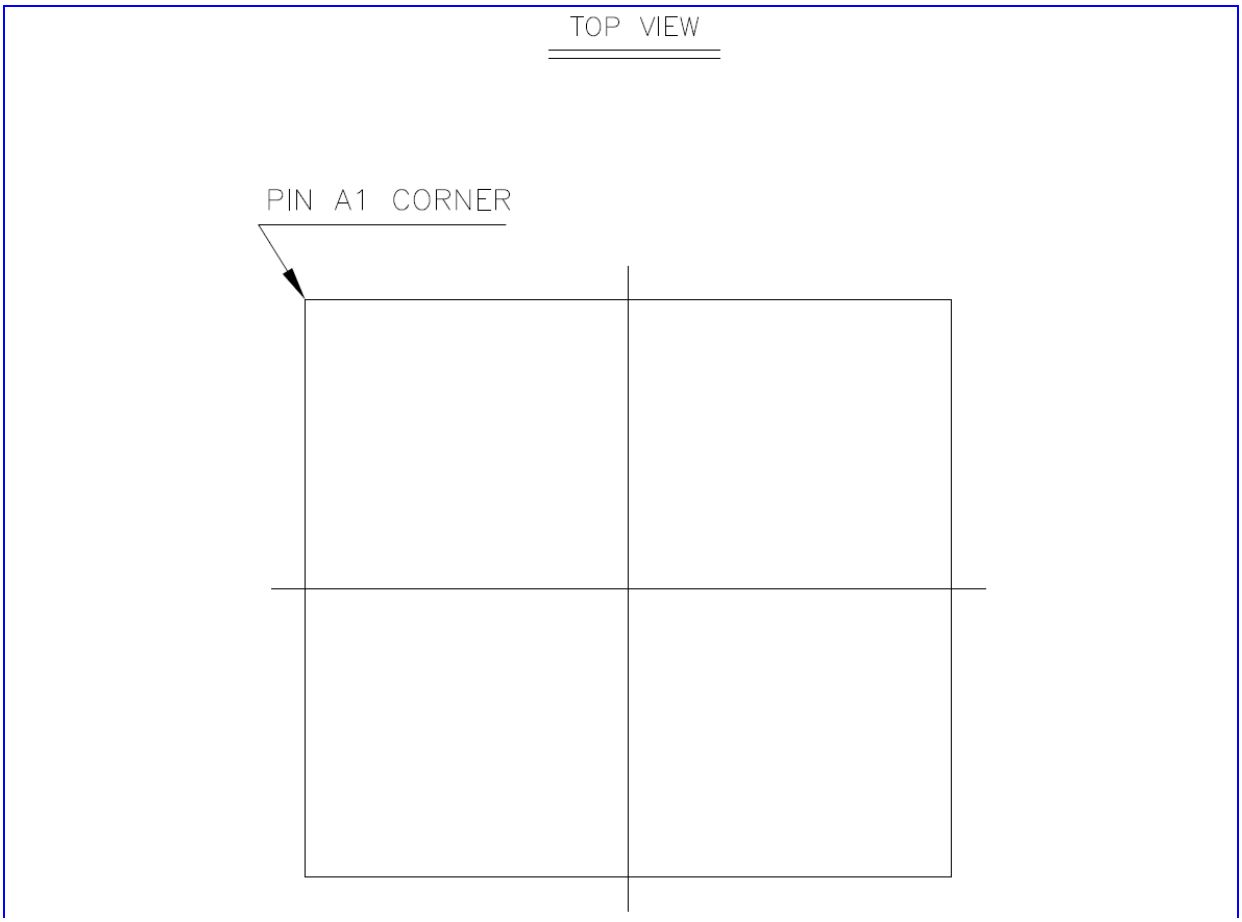
[3] A-weighting, BW = 300 Hz to 15 kHz

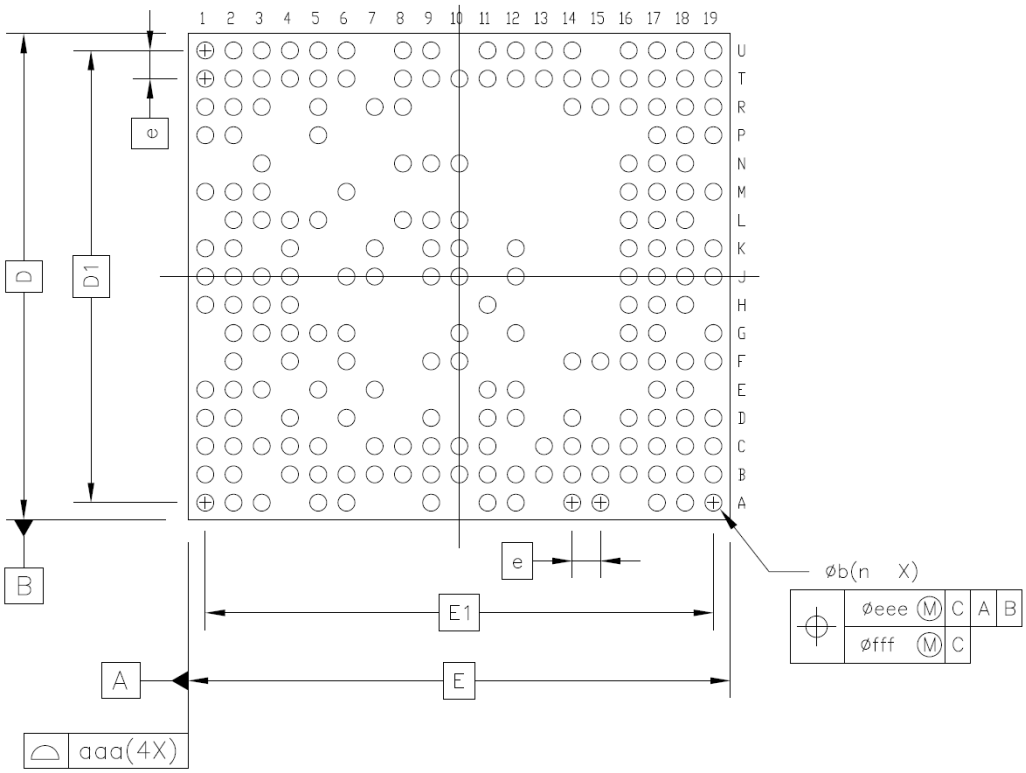
[4] Vin = 60dBμVemf

[5] Reference clock accuracy assumes ideal FM source. If the input FM source has less frequency error, then it is recommended to use a reference clock of accuracy within ±100ppm so as not to affect the channel scan quality.

2.10 Package Information

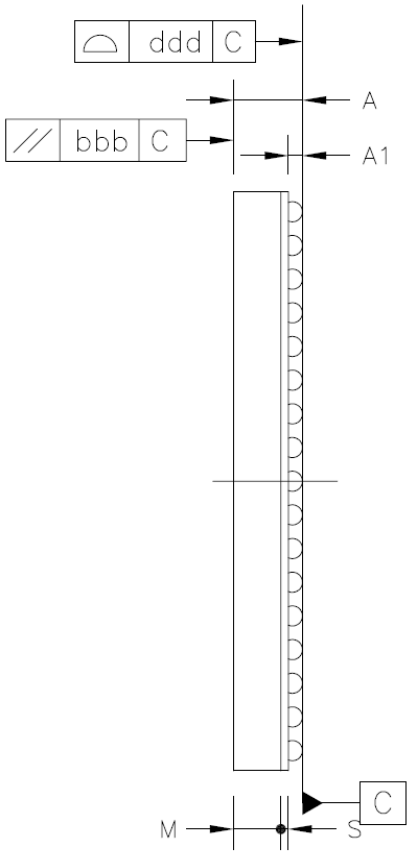
2.10.1 Package Outlines

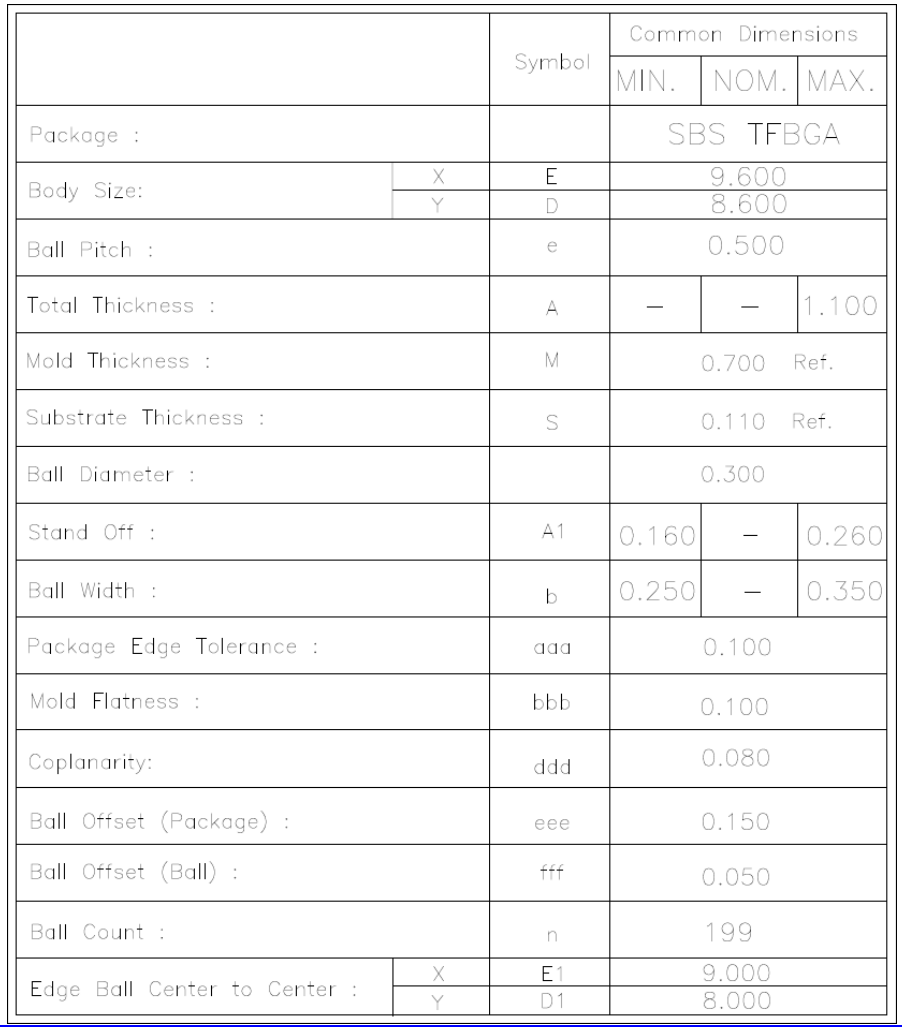




BOTTOM VIEW

SIDE VIEW





		Symbol	Common Dimensions		
			MIN.	NOM.	MAX.
Package :			SBS TFBGA		
Body Size:	X	E	9.600		
	Y	D	8.600		
Ball Pitch :		e	0.500		
Total Thickness :		A	—	—	1.100
Mold Thickness :		M	0.700 Ref.		
Substrate Thickness :		S	0.110 Ref.		
Ball Diameter :			0.300		
Stand Off :		A1	0.160	—	0.260
Ball Width :		b	0.250	—	0.350
Package Edge Tolerance :		aaa	0.100		
Mold Flatness :		bbb	0.100		
Coplanarity:		ddd	0.080		
Ball Offset (Package) :		eee	0.150		
Ball Offset (Ball) :		fff	0.050		
Ball Count :		n	199		
Edge Ball Center to Center :	X	E1	9.000		
	Y	D1	8.000		

Figure 18. ***Outlines and dimension of TFBGA 9.6mm*8.6mm, 199-ball, 0.5 mm pitch package***

2.10.2 Thermal Operating Specifications

Symbol	Description	Value	Unit	Notes
	Thermal resistance from device junction to package case	48	C/W	
	Maximum package temperature	65	Deg C	
	Maximum power dissipation	1.28	W	

2.10.3 Lead-free Packaging

MT6260D is provided in a lead-free package and meets RoHS requirements

2.11 Ordering Information

2.11.1 Top Marking Definition

Figure 19. ***Mass production top marking of MT6260D***

Part number	Package	Description
MT6260DA/A	TFBGA	8.6mm*9.6mm, 199-ball, 0.5 mm pitch package, non-security version

3 Micro-Controller Unit Peripherals

3.1 Pulse-Width Modulation Outputs

3.1.1 General Description

1 generic pulse-width modulator is implemented to generate pulse sequences with programmable frequency and duty cycles for LCD backlight. As long as the internal counter value is bigger than or equal to the threshold value, the duration of the PWM output signal is LOW. The waveform is shown in Figure 20.

Figure 20. PWM waveform

The frequency and volume of the PWM output signal are determined by PWM1_COUNT, PWM1_THRES and PWM1_CON. The POWERDOWN (pdn1_pwm) signal is applied to power-down the PWM_1ch module. When PWM_1ch is deactivated (pwm1_pdn=1), the output is in the LOW state.

The output PWM frequency is determined by:

CLOCK_DIV = 1, when CLK[1:0] = 00b
CLOCK_DIV = 2, when CLK[1:0] = 01b
CLOCK_DIV = 4, when CLK[1:0] = 10b
CLOCK_DIV = 8, when CLK[1:0] = 11b

The output PWM duty cycle is determined by:

Note: PWM_THRES should be less than the PWM_COUNT. If this condition is not satisfied, the output pulse of the PWM will always be HIGH.

Figure 21 shows the PWM waveform with the indicated register values.

Figure 21. PWM waveform with register values

3.1.2 Register Definition

Module name: Pulse Width Modulation base address: (+A00E0000h)

Address	Name	Width	Register function
A00E0000	**PWM1_CTRL_ADDR**	16	PWM1 control register
A00E0004	**PWM1_COUNT_ADDR**	16	PWM1 max counter value register
A00E0008	**PWM1_THRESH_ADDR**	16	PWM1 threshold value register

A00E0000 PWM1_CTRL_ADDR PWM1 Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PWM1_CLK_SEL	PWM1_CLK_DIV	
Type														RW	RW	
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2	**PWM1_CLK_SEL**	CLK_SEL	**Selects source clock frequency of PWM1** 0: CLK = 13MHz 1: CLK = 32kHz
1:0	**PWM1_CLK_DIV**	CLK_DIV	**Selects clock prescaler scale of PWM1** 2'b00: f = fclk 2'b01: f = fclk/2 2'b10: f = fclk/4 2'b11: f = fclk/8

A00E0004 PWM1_COUNT_ADDR PWM1 Max. Counter Value Register 1FFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				PWM1_COUNT												
Type				RW												
Reset				1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
12:0	**PWM1_COUNT**	PWM1_COUNT	**PWM1 maximum counter value** This value is the initial value of the internal counter. Regardless of the operation mode, if PWM1_COUNT is written while the internal counter is counting backwards, the new initial value will not take effect until the internal counter counts down to 0, i.e. a complete period.

A00E0008 PWM1_THRESH_ADDR PWM1 Threshold Value Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				PWM1_THRES												
Type				RW												
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
12:0	**PWM1_THRES**	PWM1_THRES	**PWM1 threshold value** When the internal counter value is bigger than or equal to PWM1_THRES, the PWM1 output signal will be "0". When the internal counter is less than PWM1_THRES, the PWM1 output signal will be "1".

Module name: PWM_2CH base address: (+A0740000h)

Address	Name	Width	Register Function
A074000C	**PWM2_CTRL**	16	**PWM2 control register** Select CLK SRC and prescaler scale.
A0740014	**PWM2_THRES**	16	**PWM2 threshold value register** Controls the duty of waveform
A0740018	**PWM3_CTRL**	16	**PWM3 control register** Select CLK SRC and prescaler scale.
A074001C	**PWM3_COUNT**	16	**PWM3 max counter value register** Configures internal counter's max. value
A0740020	**PWM3_THRES**	16	**PWM3 threshold value register**

A074000C PWM2_CTRL PWM2 Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PWM2_CLK_SEL		
Type														RW		
Reset														1		

Bit(s)	Name	Description
2	PWM2_CLK_SEL	**Selects PWM2 CLK** 0: CLK = 13M CLK 1: CLK = 32k CLK

A0740014 PWM2_THRES PWM2 Threshold Value Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															PWM2_THRES	
Type															RW	
Reset															0	0

Bit(s)	Name	Description
1:0	PWM2_THRES	**PWM2 threshold value** 0: Duty = 0% 1: Duty = 50%

Bit(s)	Name	Description
		2: Duty = 100%

A0740018 PWM3_CTRL PWM3 Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													PWM3_ALWAYS_HIGH	PWM3_CLK_SEL	PWM3_CLK_DIV	
Type													RW	RW	RW	
Reset													0	0	0	0

Bit(s)	Name	Description
3	PWM3_ALWAYS_HIGH	**When pwm3_thresh is set to be bigger than pwm3_width, which means the PWM output is always high, the driver should set this register to 1.** It is specially used by ISINK. 0: Duty! = 100% 1: Duty = 100%
2	PWM3_CLK_SEL	**Selects PWM3 CLK** 0: CLK = 13M CLK 1: CLK = 32k CLK
1:0	PWM3_CLK_DIV	**PWM3 CLK division** 2'b0: f = fclk 2'b1: f = fclk/2 2'b2: f = fclk/4 2'b3: f = fclk/8

A074001C PWM3_COUNT PWM3 Max Counter Value Register 1FFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				PWM3_COUNT												
Type				RW												
Reset				1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Name	Description
12:0	PWM3_COUNT	**PWM3 maximum counter value** This value is the initial value of the internal counter. Regardless of the operation mode, if PWM、3_COUNT is written while the internal counter is counting backwards, the new initial value will not take effect until the internal counter counts down to 0, i.e. a complete period.

A0740020 PWM3_THRES PWM3 Threshold Value Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				PWM3_THRES												
Type				RW												
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
12:0	PWM3_THRES	**PWM3 threshold value** When the internal counter value is bigger than or equal to PWM3_THRES, the PWM3 output signal will be 0. When the internal counter is less than

Bit(s)	Name	Description
		PWM3_THRES, the PWM3 output signal will be 1.

3.2 SIM interface

MT6260D contains two dedicated smart card interfaces to allow the MCU to access two SIM cards. Each interface can operate via 5 terminals. See Figure *22* , SIMVCC, SIMSEL, SIMRST, SIMCLK and SIMDATA are for one SIM interface, and SIM2VCC, SIM2SEL, SIM2RST, SIM2CLK and SIM2DATA are for the other SIM interface.

Figure 22 : Block diagram of SIM interface

The functions of the two SIM interfaces are identical; therefore, only the first SIM interface will be described in this document. SIMVCC is used to control the external voltage supply to the SIM card, and SIMSEL determines the regulated smart card supply voltage. SIMRST is used as the SIM card reset signal. Besides, SIMDATA and SIMCLK are used for data exchange.

The SIM interface is a half duplex asynchronous communication port, and its data format is composed of ten consecutive bits: a start bit in state “low”, eight information bits and a tenth bit used for parity checking. The data format can be divided into two modes as follows:

- Direct convention mode (ODD = SDIR = SINV = 0)

  SB D0 D1 D2 D3 D4 D5 D6 D7 PB
  SB: Start bit (in state “low”)
  Dx: Data byte (LSB is the first and logic level ONE is in state “high”)

PB: Even parity check bit

- Inverse convention mode (ODD = SDIR = SINV = 1)

  SB N7 N6 N5 N4 N3 N2 N1 N0 PB
  SB: Start bit (in state “low”)
  Nx**:** Data byte (MSB is the first and logic level ONE is in state “low”)
  PB: Odd parity check bit

If the receiver obtains a wrong parity bit, it will respond by pulling the SIMDATA “low” to inform the transmitter, and the transmitter will retransmit the character.

If the receiver is an SIM card, the error response will start 0.5 bit after the PB and may last for 1 ~ 2-bit period. If the receiver is an SIM interface, the error response will start 0.5 bit after the PB and last for 1.5-bit period.

If the SIM interface is a transmitter, it will take totally 14 bits guard period wherever the error response appears. If the receiver shows the error response, the SIM interface will retransmit the previous character again, or it will transmit the next character.

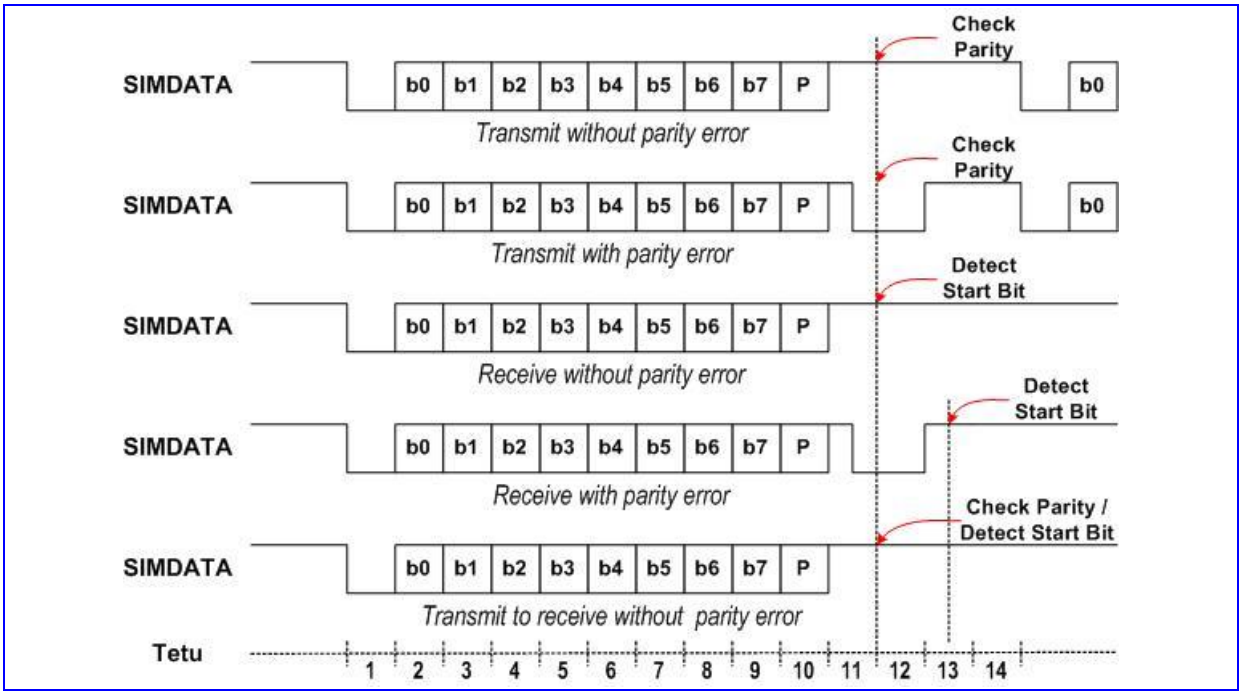


Figure 23: Timing diagram of SIM interface

3.2.1 Definitions of registers

When the MCU controls two SIM card interfaces, all registers will be duplicated to two copies but with different base address. n = “0” is for the 1st SIM card interface; n=1 is for the 2nd SIM card interface. For example, address SIMIF0+0000h is mapped to the SIMIF0_SIM_CTRL register while address SIMIF1+0000h is mapped to the SIMIF1_SIM_CTRL register.

3.2.1.1 Overview of registers

MCU register address (hex)	Acronym	Description
1st SIM card interface		
SIMIF0+0000h	SIMIF0_SIM_CTRL	Control register
SIMIF0+0004h	SIMIF0_SIM_CONF	Configuration register
SIMIF0+0008h	SIMIF0_SIM_BRR	Baudrate register
SIMIF0+0010h	SIMIF0_SIM_IRQEN	Interrupt enabling register
SIMIF0+0014h	SIMIF0_SIM_STS	Status register
SIMIF0+0020h	SIMIF0_SIM_RETRY	Retry limit register
SIMIF0+0024h	SIMIF0_SIM_TIDE	FIFO tide mark register
SIMIF0+0030h	SIMIF0_SIM_DATA	TX/RX data register
SIMIF0+0034h	SIMIF0_SIM_COUNT	FIFO count register
SIMIF0+0040h	SIMIF0_SIM_ATIME	Activation time register
SIMIF0+0044h	SIMIF0_SIM_DTIME	Deactivation time register
SIMIF0+0048h	SIMIF0_SIM_TOUT	Character to character waiting time register
SIMIF0+004Ch	SIMIF0_SIM_GTIME	Block to block guard time register
SIMIF0+0050h	SIMIF0_SIM_ETIME	Block to error signal time register
SIMIF0+0054h	SIMIF0_SIM_EXT_TIME	Extend data I/O state switch time register
SIMIF0+0058h	SIMIF0_SIM_CGTIME	Character to character guard time register
SIMIF0+0060h	SIMIF0_SIM_INS	Command header register : INS
SIMIF0+0064h	SIMIF0_SIM_IMP3	Command header register : P3
SIMIF0+0068h	SIMIF0_SIM_SW1	Procedure byte register : SW1
SIMIF0+006Ch	SIMIF0_SIM_SW2	Procedure byte register : SW2
SIMIF0+0070h	SIMIF0_SIM_ATRSTA	ATR state register
SIMIF0+0074h	SIMIF0_SIM_STATUS	Protocol state register
SIMIF0+0080h	SIMIF0_SIM_DMADATA	TX/RX data register for DMA
SIMIF0+0090h	SIMIF0_SIM_DBG	Debug register
SIMIF0+0094h	SIMIF0_SIM_DBGDATA	FIFO data debug register
SIMIF0+00A0h	SIMIF0_SIM_SCLK	SCLK PAD control register
SIMIF0+00A4h	SIMIF0_SIM_SRST	SRST PAD control register
SIMIF0+00A8h	SIMIF0_SIM_SIO	SIO PAD control register
SIMIF0+00ACh	SIMIF0_SIM_MON	PAD monitor register
SIMIF0+00B0h	SIMIF0_SIM_SEL	Testing output select
2nd SIM card interface		
SIMIF1+0000h	SIMIF1_SIM_CTRL	Control register
SIMIF1+0004h	SIMIF1_SIM_CONF	Configuration register
SIMIF1+0008h	SIMIF1_SIM_BRR	Baudrate register
SIMIF1+0010h	SIMIF1_SIM_IRQEN	Interrupt enabling register
SIMIF1+0014h	SIMIF1_SIM_STS	Status register
SIMIF1+0020h	SIMIF1_SIM_RETRY	Retry limit register
SIMIF1+0024h	SIMIF1_SIM_TIDE	FIFO tide mark register
SIMIF1+0030h	SIMIF1_SIM_DATA	TX/RX data register
SIMIF1+0034h	SIMIF1_SIM_COUNT	FIFO count register

SIMIF1+0040h	SIMIF1_SIM_ATIME	Activation time register
SIMIF1+0044h	SIMIF1_SIM_DTIME	Deactivation time register
SIMIF1+0048h	SIMIF1_SIM_TOUT	Character to character waiting time register
SIMIF1+004Ch	SIMIF1_SIM_GTIME	Block to block guard time register
SIMIF1+0050h	SIMIF1_SIM_ETIME	Block to error signal time register
SIMIF1+0054h	SIMIF1_SIM_EXT_TIME	Extend data I/O state switch time register
SIMIF1+0058h	SIMIF1_SIM_CGTIME	Character to character guard time register
SIMIF1+0060h	SIMIF1_SIM_INS	Command header register : INS
SIMIF1+0064h	SIMIF1_SIM_IMP3	Command header register : P3
SIMIF1+0068h	SIMIF1_SIM_SW1	Procedure byte register : SW1
SIMIF1+006Ch	SIMIF1_SIM_SW2	Procedure byte register : SW2
SIMIF1+0070h	SIMIF1_SIM_ATRSTA	ATR state register
SIMIF1+0074h	SIMIF1_SIM_STATUS	Protocol state register
SIMIF1+0080h	SIMIF1_SIM_DMADATA	TX/RX data register for DMA
SIMIF1+0090h	SIMIF1_SIM_DBGD	Debug register
SIMIF1+0094h	SIMIF1_SIM_DBGDATA	FIFO data debug register
SIMIF1+00A0h	SIMIF1_SIM_SCLK	SCLK PAD control register
SIMIF1+00A4h	SIMIF1_SIM_SRST	SRST PAD control register
SIMIF1+00A8h	SIMIF1_SIM_SIO	SIO PAD control register
SIMIF1+00ACh	SIMIF1_SIM_MON	PAD monitor register
SIMIF1+00B0h	SIMIF1_SIM_SEL	Testing output select

3.2.1.2 Register description

SIMn+0000h SIM module control register SIMIFN_SIM_CTRL

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										VCCCTRL	VCCLV	RSTCTRL	RSTLV	WRST	CSTOP	SIMON
Type										R/W	R/W	R/W	R/W	W	R/W	R/W
Reset										0	0	0	0	0	0	0

SIMON SIM card power-up/power-down control

- **0** A 1-to-0 change will start the card deactivation sequence
- **1** A 0-to-1 change will start the card activation sequence

CSTOP Enabling clock stop mode. Together with CPOL in the SIM_CONF register, it determines the polarity of SIMCLK in this mode.

- **0** Enable SIMCLK output.
- **1** Disable SIMCLK output

WRST Controlling SIM card warm reset

RSTLV Controlling SIMRST parking level in SIMRST direct control mode

RSTCTRL Enabling SIMRST direct control mode

VCCLV Controlling SIMVCC parking level in SIMVCC direct control mode

VCCCTRL Enabling SIMVCC direct control mode

SIMn+0004h SIM module configuration register SIMIFN_SIM_CONF

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			T1TX2RXEN	TXRDIS	RXRDIS	HFEN	T0EN	T1EN	TOUT	SIMSEL	ODD	SDIR	SINV	CPOL	TXACK	RXACK
Type			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0	0	0	0	0	0	0	0	0

RXACK Handshaking control of SIM card reception error
- **0** Disable character receipt handshaking
- **1** Enable character receipt handshaking

TXACK Handshaking control of SIM card transmission error
- **0** Disable character transmission handshaking
- **1** Enable character transmission handshaking

CPOL SIMCLK polarity control in clock stop mode
- **0** Make SIMCLK stop in “low” level
- **1** Make SIMCLK stop in “high” level

SINV Data invert mode
- **0** Not invert the transmitted and received data; data logic ONE is in “high” state
- **1** Invert the transmitted and received data; data logic ONE is in “low” state

SDIR Direction of data transfer
- **0** LSB is transmitted and received first
- **1** MSB is transmitted and received first

ODD Selecting odd or even parity
- **0** Even parity
- **1** Odd parity

SIMSEL Selecting SIM card supply voltage (please also configure SIMSEL in PMU register)
- **0** SIMSEL pin is set to “low” level, 1.8V
- **1** SIMSEL pin is set to “high” level, 3V

TOUT Controlling SIM work waiting time counter
- **0** Disable time-out counter
- **1** Enable time-out counter

T1EN Controlling T = 1 protocol controller
- **0** Disable T = 1 protocol controller
- **1** Enable T = 1 protocol controller

T0EN Controlling T = 0 protocol controller
- **0** Disable T = 0 protocol controller
- **1** Enable T = 0 protocol controller

HFEN Controlling hardware flow
- **0** Disable hardware flow control
- **1** Enable hardware flow control

RXRDIS Disabling RX DMA request
- **0** Enable RX DMA request (default)
  RXRDIS must be set to 0 for protocol T = 1
- **1** Disable RX DMA request
  During TX transmission and not protocol T = 1, the recommended setting of RXRDIS is 1

TXRDIS Disabling TX DMA request disable
- **0** Enable TX DMA request (default)
  TXRDIS must be set to 0 for protocol T = 1

1 Disable TX DMA request

During RX transmission and not protocol T = 1, the recommended setting of TXRDIS is 1

T1TX2RXEN Enabling DMA type auto switch for protocol T = 1 (this function is not supported in MT6260D)

0 Disable DMA type auto switch function

If the current block is TX transmission and the next block is also TX transmission, disabling this bit is recommended

1 Enable DMA type auto switch function

If the current block is TX transmission and the next block is RX transmission, enabling this bit is recommended to improve transmission quality

SIMn +0008h SIM baudrate register SIMIFN_SIM_BRR

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						ETU[8:0]									SIMCLK[1:0]	
Type						R/W									R/W	
Reset						372d									01	

SIMCLK Setting up SIMCLK frequency

00 Reserved

01 13/4 MHz

10 13/8 MHz

11 13/12 MHz

ETU Determinign the duration of elementary time unit in SIMCLK unit

The minimum valid setting of ETU is 8

SIMn +0010h SIM interrupt enable register SIMIFN_SIM_IRQEN

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					UDRUN	EDCERR	T1END	RXERR	T0END	SIMOFF	ATRERR	TXERR	TOUT	OVRUN	RXTIDE	TXTIDE
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0	0	0	0	0	0	0	0	0	0	0	0

For all the bits

0 Interrupt is disabled

1 Interrupt is enabled

SIMn +0014h SIM module status register SIMIFN_SIM_STS

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					UDRUN	EDCERR	T1END	RXERR	T0END	SIMOFF	ATRERR	TXERR	TOUT	OVRUN	RXTIDE	TXTIDE
Type					R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R/C	R	R
Reset				-	-	-	-	-	-	-	-	-	-	-	-	-

TXTIDE The interrupt occurs when the number of transmitted data in the FIFO is less than the transmitted tide.

RXTIDE The interrupt occurs when the number of received data in the FIFO is less than the received tide.

OVRUN Receiving FIFO overflow interrupt occurrs.

TOUT Between characters time-out interrupt occurrs.

TXERR Character transmission error interrupt occurrs.
ATRERR ATR start time-out interrupt occurrs.
SIMOFF Card deactivation completed interrupt occurrs.
T0END Data transfer handled by T = 0 controller completed interrupt occurrs.
RXERR Character reception error interrupt occurrs.
T1END Data transfer handled by T = 1 controller completed interrupt occurrs.
EDCERR T = 1 controller CRC error occurrs.
UDRUN FIFO underflow interrupt occurrs (still reading FIFO when FIFO is empty).

SIMn +0020h SIM retry limit register SIMIFN_SIM_RETRY

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						TXRETRY[2:0]								RXRETRY[2:0]		
Type						R/W								R/W		
Reset						3h								3h		

RXRETRY Specifying the maximum number of receive retries allowed when parity error occurrs.
TXRETRY Specifying the maximum number of transmit retries allowed when parity error occurrs.

SIMn +0024h SIM FIFO tide mark register SIMIFN_SIM_TIDE

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					TXTIDE[3:0]								RXTIDE[3:0]			
Type					R/W								R/W			
Reset					0h								0h			

RXTIDE Trigger point of RXTIDE interrupt
TXTIDE Trigger point of TXTIDE interrupt

SIMn +0030h Data register used as Tx/Rx data register SIMIFN_SIM_DATA

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA[7:0]							
Type									R/W							
Reset									-							

DATA Eight data digits, corresponding to the character being read or written

SIMn +0034h SIM FIFO count register SIMIFN_SIM_COUNT

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												COUNT[4:0]				
Type												R/W				
Reset												0h				

COUNT The number of characters in the SIM FIFO when read and flushes when written.

SIMn +0040h SIM activation time register SIMIFN_SIM_ATIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							ATIME[9:0]									
Type							R/W									
Reset							2BEh									

ATIME The register defines the duration, in 64 SIM clock cycles, of the time taken for each of the three stages of the card activation process, from SIMON transiting to “high” to turning on VCC, from turning on VCC to pull data “high” and then from pulling data “high” to turning on CLK.

SIMn +0044h SIM deactivation time register SIMIFN_SIM_DTIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											DTIME[5:0]					
Type											R/W					
Reset											Fh					

DTIME The register defines the duration, in 64 13 MHz clock cycles, of the time taken for each of the three stages of the card deactivation sequence, from pulling RST “low” to turning off CLK, from turning off CLK to pulling data “low”, from pulling data “low” to turning off VCC.

SIMn +0048h Character to character waiting time register SIMIFN_SIM_TOUT

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name											WTIME[21:0]					
Type											R/W					
Reset											260h					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WTIME[21:0]															
Type	R/W															
Reset	260h															

WTIME Maximum interval between the leading edge of two consecutive characters in 16 ETU units

SIMn +004Ch Block to block guard time register SIMIFN_SIM_GTIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													GTIME[3:0]			
Type													R/W			
Reset													10d			

GTIME Minimum interval between the leading edge of two consecutive characters sent in opposite directions in ETU unit

SIMn +0050h Block to error signal time register SIMN_SIM_ETIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											ETIME[5:0]					
Type											R/W					
Reset											15d					

ETIME The register defines the interval, in 1/16 ETU unit, between the end of the transmitted parity bit and the time to check the parity error signal sent from SIM card.

SIMn +0054h Active high period control register SIMIFN_SIM_EXT_TIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													EXT_TIME[3:0]			
Type													R/W			

Reset													1d

EXT_TIME The register defines the interval, in 1/16 ETU unit, between the end of the transmitted parity bit and the time to switch SIO to input mode. This value should be smaller than ETIME.

SIMn +0058h Character to character guard time register SIMIFN_SIM_CGTIME

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CGTIME[7:0]							
Type									R/W							
Reset									2h							

CGTIME The register defines the minimum interval between the leading edges of two consecutive characters in ETU unit.
In the same transmission direction, the minimum interval is (12 + CGTIME) ETU. In opposite transmission direction, the minimum interval is (12 + CGTIME + GTIME) ETU.

SIMn +0060h SIM command header register: INS SIMIFN_SIM_INS

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								INSD	SIMINS[7:0]							
Type								R/W	R/W							
Reset								0h	0h							

SIMINS This field should be identical to the INS instruction code. When writing to this register, the T = 0 controller will be activated and data transfer initiated.

INSD Instruction direction

- **0** T = 0 controller receives data from the SIM card.
- **1** T = 0 controller sends data to the SIM card.

SIMn +0064h SIM command header register: P3 SIMIFN_SIM_IMP3 (ICC_LEN)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								SIMP3[8]	SIMP3[7:0]							
Type								R	R/W							
Reset								0h	0h							

SIMP3 This field should be identical to the P3 instruction code. It should be written prior to the SIM_INS register. When the data transfer is being conducted, this field will show the number of the remaining data to be sent or to be received.

SIMn +0068h SIM procedure byte register: SW1 SIMIFN_SIM_SW1 (ICC_LEN)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SIMSW1[7:0]							
Type									R							
Reset									0h							

SIMSW1 This field holds the last received procedure byte for debugging. When the T0END interrupt occurrs, it will keep the SW1 procedure byte.

SIMn +006Ch SIM procedure byte register: SW2

SIMIFN_SIM_SW2 (ICC_EDC)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SIMSW2[7:0]							
Type									R							
Reset									0h							

SIMSW2 This field holds the SW2 procedure byte

SIMn +0070h SIM ATR state register

SIMIFN_SIM_ATRSTA

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								AL	IR							OFF
Type								R	R							R
Reset								0h	0h							1h

The SIM card is initially turned off. After configuring SIMON of SIMn_SIM_CTRL and ATR procedure, SIMn_SIM_ATRSTA will set IR or AL to 1 to indicate the card's feature.

OFF Indicating On/Off of the SIM card
IR SIM card is IR (internal reset) card
AL SIM card is AL (active low reset) card

SIMn +0074h SIM protocol state register

SIMIFN_SIM_STATUS

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											ALL	ONE				IDLE
Type											R	R				R
Reset											0h	0h				1h

T0 or T1 protocol of the SIM card is initially turned off. When T0 or T1 protocol is turned on, SIMn_SIM_T0STA will transit between ONE or ALL according to the procedure byte of the SIM card.

IDLE SIM card's T0 or T1 protocol is active or idle.
ONE SIM card will send the next byte
ALL SIM card will send all the remaining bytes.

SIMn +0080h Data register used as Tx/Rx data register

SIMIFN_SIM_DMADATA

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA[7:0]							
Type									R/W							
Reset									-							

DATA Eight data digits, corresponding to the character being read or written

SIMn +0090h SIM module debug register

SIMIFN_SIM_DBG

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					DBG7	DBG6	DBG5	DBG4				DBG3[4:0]				
Type					R	R	R	R				R				
Reset					0h	0h	0h	0h				0h				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				DBG2[4:0]								DBG1[4:0]				

Type				R				R
Reset				0h				0h

DBG1 Debug register 1
DBG2 Debug register 2
DBG3 Debug register 3
DBG4 Debug register 4
DBG5 Debug register 5
DBG6 Debug register 6
DBG7 Debug register 7

SIMn +0094h SIM FIFO data debug register SIMIFN_SIM_DBGDATA

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					DBGRPTR[3:0]				DBGDATA[7:0]							
Type					R				R							
Reset					0h				-							

DBGDATA FIFO data debug register
There is no influence on data transmission when reading this register
DBGRPTR FIFO read pointer related to DBGDATA
Automatically increase by 1 after reading this register

SIMn +00A0h SIM SCLK PAD control register SIMIFN_SIM_SCLK

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															DEBUG	ACD_FUNC
Type															R/W	R/W
Reset															0h	0h
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IES_CTRL	IES_LV			TDSEL[1:0]		RDSEL[1:0]		R1	R0	PUPD	SMT	E4	E2	SR[1:0]	
Type	R/W	R/W			R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0h	0h			0h		0h		0h	0h	0h	1h	0h	1h	3h	

SR Output slew rate control
High asserted. SR = 1, slower slew. SR = 0, no slew rate control.
For SIM card mode, SR[1:0] = [1 1] is the recommended setting to eliminate overshooting/undershooting. For non-SIM card mode, SR[1:0] = [0 0] is set for best speed.
E2 TX driving strength control
For SIM card mode, E2 = [1] is the recommended setting for SCLK/SRST/SIO. (SIO/SRST can use [0])
E4 TX driving strength control
For SIM card mode, E4 = [0] is the recommended setting for SCLK/SRST/SIO.
SMT RX input buffer schmit trigger hysteresis control enable
High asserted. SMT = 1, schmit trigger enable.
For SIM card mode, SMT = [1] is the recommended setting.
PUPD Weak pull-up/pull-down control
0 Pull-up
1 Pull-down

R0 Weak pull-up/pull-down resistance select
Please check the table in register “R1”.

R1 Weak pull-up/pull-down resistance select
Please check the following table.

E	PUPD	R1	R0	R Value
0	0	0	0	High – Z
0	0	0	1	PU – 20k
0	0	1	0	PU – 5k
0	0	1	1	PU – 4k
0	1	0	0	High – Z
0	1	0	1	PD – 75k
0	1	1	0	PD – 75k
0	1	1	1	PD – 37.5k
1	x	x	x	High - Z

RDSEL RX duty select
RDSEL[0] : input buffer duty high when asserted. (high pulse width adjustment)
RDSEL[1] : input buffer duty low when asserted. (low pulse width adjustment)
For SIM card mode, RDSEL = [0 0] is the recommended setting.

TDSEL TX duty select
TDSEL[0] : output level shifter duty high when asserted. (high pulse width adjustment)
TDSEL[1] : output level shifter duty low when asserted. (low pulse width adjustment)
For SIM card mode, TDSEL = [0 0] is the recommended setting.

IES_LV Controlling IES (RX input buffer enable) parking level in IES direct control mode
High asserted. Datapath : from IO to O. IES = 0, O = 0.
In quiescent mode, IES = 0 is suggested for power saving.

IES_CTRL Enabling IES direct control mode

ACD_FUNC ACD function mode for analog designer

DEBUG Output PAD related signals for monitoring

0 Disable

1 Enable

SIMn +00A4h SIM SRST PAD control register SIMIFN_SIM_SRST

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IES_CTRL	IES_LV			TDSEL[1:0]		RDSEL[1:0]		R1	R0	PUPD	SMT	E4	E2	SR[1:0]	
Type	R/W	R/W			R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0h	0h			0h		0h		0h	0h	0h	1h	0h	1h	3h	

SR Output slew rate control
High asserted. SR = 1, slower slew. SR = 0, no slew rate control.
For SIM card mode, SR[1:0] = [1 1] is the recommended setting to eliminate overshooting/undershooting. For non-SIM card mode, SR[1:0] = [0 0] is set for best speed.

E2 TX driving strength control
For SIM card mode, E2 = [1] is the recommended setting for SCLK/SRST/SIO. (SIO/SRST can use [0])

E4 TX driving strength control

For SIM card mode, E4 = [0] is the recommended setting for SCLK/SRST/SIO.

SMT RX input buffer schmit trigger hysteresis control enable

High asserted. SMT = 1, schmit trigger enable.

For SIM card mode, SMT = [1] is the recommended setting.

PUPD Weak pull-up/pull-down control

0 Pull-up

1 Pull-down

R0 Weak pull-up/pull-down resistance select

Please check the table in register “R1”.

R1 Weak pull-up/pull-down resistance select

Please check the following table.

E	PUPD	R1	R0	R Value
0	0	0	0	High – Z
0	0	0	1	PU – 20k
0	0	1	0	PU – 5k
0	0	1	1	PU – 4k
0	1	0	0	High – Z
0	1	0	1	PD – 75k
0	1	1	0	PD – 75k
0	1	1	1	PD – 37.5k
1	x	x	x	High - Z

RDSEL RX duty select

RDSEL[0] : input buffer duty high when asserted. (high pulse width adjustment)

RDSEL[1] : input buffer duty low when asserted. (low pulse width adjustment)

For SIM card mode, RDSEL = [0 0] is the recommended setting.

TDSEL TX duty select

TDSEL[0] : output level shifter duty high when asserted. (high pulse width adjustment)

TDSEL[1] : output level shifter duty low when asserted. (low pulse width adjustment)

For SIM card mode, TDSEL = [0 0] is the recommended setting.

IES_LV Controlling IES (RX input buffer enable) parking level in IES direct control mode

High asserted. Datapath : from IO to O. IES = 0, O = 0.

In quiescent mode, IES = 0 is suggested for power saving.

IES_CTRL Enabling IES direct control mode

SIMn +00A8h SIM SIO PAD control register SIMIFN_SIM_SIO

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IES_CTRL	IES_LV			TDSEL[1:0]		RDSEL[1:0]		R1	R0	PUPD	SMT	E4	E2	SR[1:0]	
Type	R/W	R/W			R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0h	0h			0h		0h		1h	0h	0h	1h	0h	1h	3h	

SR Output slew rate control

High asserted. SR = 1, slower slew. SR = 0, no slew rate control.

For SIM card mode, SR[1:0] = [1 1] is the recommended setting to eliminate overshooting/undershooting. For non-SIM card mode, SR[1:0] = [0 0] is set for best speed.

E2 TX driving strength control

For SIM card mode, E2 = [1] is the recommended setting for SCLK/SRST/SIO. (SIO/SRST can use [0])

E4 TX driving strength control

For SIM card mode, E4 = [0] is the recommended setting for SCLK/SRST/SIO.

SMT RX input buffer schmit trigger hysteresis control enable

High asserted. SMT = 1, schmit trigger enable.

For SIM card mode, SMT = [1] is the recommended setting.

PUPD Weak pull-up/pull-down control

0 Pull-up

1 Pull-down

R0 Weak pull-up/pull-down resistance select

Please check the table in register “R1”.

R1 Weak pull-up/pull-down resistance select

Please check the following table.

For SIO, [R1 R0] = [1 0] is the recommended setting for 5k weak pull-up. In 4 SIM application and SIO is connected to external SIM switch, please disable pull-up resistance. ([R1 R0] = [0 0])

E	PUPD	R1	R0	R Value
0	0	0	0	High – Z
0	0	0	1	PU – 20k
0	0	1	0	PU – 5k
0	0	1	1	PU – 4k
0	1	0	0	High – Z
0	1	0	1	PD – 75k
0	1	1	0	PD – 75k
0	1	1	1	PD – 37.5k
1	x	x	x	High - Z

RDSEL RX duty select

RDSEL[0] : input buffer duty high when asserted. (high pulse width adjustment)

RDSEL[1] : input buffer duty low when asserted. (low pulse width adjustment)

For SIM card mode, RDSEL = [0 0] is the recommended setting.

TDSEL TX duty select

TDSEL[0] : output level shifter duty high when asserted. (high pulse width adjustment)

TDSEL[1] : output level shifter duty low when asserted. (low pulse width adjustment)

For SIM card mode, TDSEL = [0 0] is the recommended setting.

IES_LV Controlling IES (RX input buffer enable) parking level in IES direct control mode

High asserted. Datapath : from IO to O. IES = 0, O = 0.

In quiescent mode, IES = 0 is suggested for power saving.

IES_CTRL Enabling IES direct control mode

SIMn +00ACh SIM monitor register **SIMIFN_SIM_MON**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		MON12	MON11	MON10		MON9	MON8	MON7		MON6	MON5	MON4		MON3	MON2	MON1
Type		R	R	R		R	R	R		R	R	R		R	R	R
Reset		0h	0h	0h		0h	0h	0h		0h	0h	0h		0h	0h	0h

MON1 Monitor signal 1
MON2 Monitor signal 2
MON3 Monitor signal 3
MON4 Monitor signal 4
MON5 Monitor signal 5
MON6 Monitor signal 6
MON7 Monitor signal 7
MON8 Monitor signal 8
MON9 Monitor signal 9
MON10 Monitor signal 10
MON11 Monitor signal 11
MON12 Monitor signal 12

SIMn+00B0h SIM Test Select **SIMIFN_SIM_SEL**

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														SIMSEL		
Type														R/W		
Reset														3'b001		

SIMSEL Monitor SIMRST, SIMCLK, SIMIO input signal select
001 SIMRST input is monitor
010 SIMCLK input is monitor
100 SIMSIO input is monitor
Other value has no meaning

3.2.2 SIM card insertion and removal

The detection of physical connection to the SIM card and card removal can be done by the external interrupt controller or by GPIO.

3.2.3 Card activation and deactivation

The card activation and deactivation sequence are both controlled by H/W. The MCU initiates the activation sequence by writing "1" to bit 0 of the SIM_CTRL register, and then the interface performs the following activation sequence:

- Assert SIMRST "low"
- Set SIMVCC at "high" level and SIMDATA in the reception mode
- Enable SIMCLK clock
- De-assert SIMRST "high" (required if it belongs to active low reset SIM card)

The final step in a typical card session is contacting deactivation in case the card will be electrically damaged. The deactivation sequence is initiated by writing “0” to bit 0 of the SIM_CTRL register, and the interface will perform the following deactivation sequence:

- Assert SIMRST ”low”
- Set SCIMCLK at “low” level
- Set SIMDATA at “low” level
- Set SIMVCC at “low” level

3.2.4 Answering to reset sequence

After the card is activated, a reset operation will result in an answer from the card consisting of the initial character TS, followed by maximum 32 characters. The initial character TS provides a bit synchronization sequence and defines the conventions to interpret data bytes in all subsequent characters.

On reception of the first character, TS, the MCU should read this character, establish the respective required convention and re-program the related registers. These processes should be completed prior to the completion of reception of the next character. Next, the remainder of the ATR sequence will be received, read via the SIM_DATA in the selected convention and interpreted by the S/W.

The timing requirement and procedures for ATR sequence are handled by H/W and shall meet the requirement of ISO 7816-3, as shown in Figure *24*.

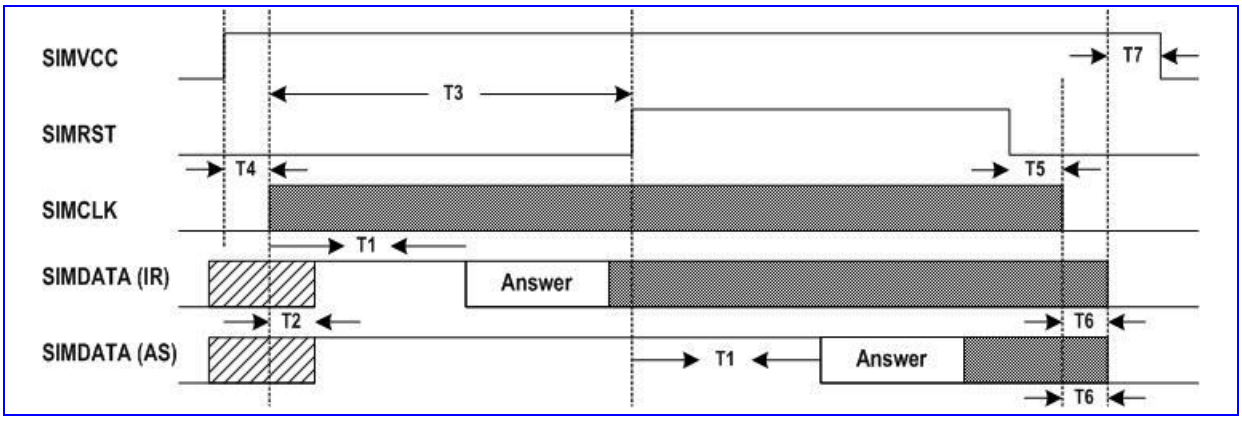


Figure 24 Answering to reset sequence

Time	Value	Comment
T1	> 400 SIMCLK	SIMCLK start to ATR appears
T2	< 200 SIMCLK	SIMCLK start to SIMDATA in reception mode
T3	> 40,000 SIMCLK	SIMCLK start to SIMRST ”high”
T4	-	SIMVCC “high” to SIMCLK start
T5	-	SIMRST “low” to SIMCLK stop
T6	-	SIMCLK stop to SIMDATA ”low”
T7	-	SIMDATA “low” to SIMVCC ”low”

Table 44 : Time-out condition for **answering to reset sequence**

3.2.5 SIM data transfer

There are two transfer modes provided, in software controlled byte by byte fashion or in a block fashion using T=0 controller and DMA controller. In both modes, the time-out counter can be enabled to monitor the elapsed time between two consecutive bytes.

3.2.5.1 Byte transfer mode

This mode is used during ATR and PPS procedure. In this mode, the SIM interface only ensures error free character transmission and reception.

Receiving characters

Upon detection of the start-bit sent by SIM card, the interface transforms into reception mode and the following bits are shifted into an internal register. If no parity error is detected or the character-received handshaking is disabled, the received-character will be written into the SIM FIFO and the SIM_COUNT register increased by one. Otherwise, the SIMDATA line will be held “low” at 0.5 etu after detecting the parity error for 1.5 etus, and the character will be re-received. If a character fails to be received correctly for the RXRETRY times, the receive-handshaking will be aborted, the last-received character written into the SIM FIFO, the SIM_COUNT increased by one and the RXERR interrupt generated.

When the number of characters held in the received FIFO exceeds the level defined in the SIM_TIDE register, a RXTIDE interrupt will be generated. The number of characters held in the SIM FIFO can be determined by reading the SIM_COUNT register, and writing to this register will flush the SIM FIFO.

Sending characters

Characters that are to be sent to the card are first written into the SIM FIFO and then automatically transmitted to the card at timed intervals. If character-transmitted handshaking is enabled, the SIMDATA line will be sampled at 1 etu after the parity bit. If the card indicates that it does not receive the character correctly, the character will be re-transmitted for maximum of TXRETRY times before a TXERR interrupt is generated and the transmission is aborted. Otherwise, the succeeding byte in the SIM FIFO will be transmitted.

If a character fails to be transmitted and a TXERR interrupt is generated, the interface will need to be reset by flushing the SIM FIFO before any subsequent transmission or reception operation.

When the number of characters held in the SIM FIFO falls below the level defined in the SIM_TIDE register, a TXTIDE interrupt will be generated. The number of characters held in the SIM FIFO can be determined by reading the SIM_COUNT register, and writing to this register will flush the SIM FIFO.

3.2.5.2 Block transfer mode

Basically, the SIM interface is designed to work in conjunction with the T=0 protocol controller and the DMA controller during non-ATR and non-PPS phase, though it is still possible for software to service the data transfer manually as in the byte transfer mode if necessary. Thus the T=0 protocol should be controlled by software.

The T=0 controller can be accessed via four registers representing the instruction header bytes INS and P3, and the procedure bytes SW1 and SW2. The registers are:

- SIM_INS, SIM_P3
- SIM_SW1, SIM_SW2

During the character transfer, SIM_P3 holds the number of characters to be sent or to be received, and SIM_SW1 holds the last received procedure byte including NULL, ACK, NACK and SW1 for debugging.

Data receiving instruction

Data receiving instructions receive data from the SIM card. See the following instantiated procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in the SIM_CONF register.
2. Program the SIM_TIDE register to 0x0000 (TXTIDE = 0, RXTIDE = 0).
3. Program the SIM_IRQEN to 0x019C (enable RXERR, TXERR, T0END, TOUT and OVRUN interrupts).
4. Write CLA, INS, P1, P2 and P3 into SIM FIFO.
5. Program the DMA controller:
   - DMA*n*_MSBSRC and DMA*n*_LSBSRC: Address of the SIM_DATA register
   - DMA*n*_MSBDST and DMA*n*_LSBDST: Memory address reserved to store the received characters
   - DMA*n*_COUNT: Identical to P3 or 256 (if P3 = 0)
   - DMA*n*_CON: 0x0078
6. Write P3 into the SIM_P3 register and then INS into SIM_INS register. (Data transfer is initiated now.)
7. Enable the time-out counter by setting the TOUT bit to 1 in the SIM_CONF register.
8. Start the DMA controller by writing 0x8000 into the DMA*n*_START register.

Upon completion of the data receiving instruction, T0END interrupt will be generated and the time-out counter should be disabled by setting the TOUT bit to 0 in the SIM_CONF register.

If error occurs during the data transfer (RXERR, TXERR, OVRUN or TOUT interrupt is generated), the SIM card should be deactivated first and then activate the prior subsequent operations.

Data sending instruction

Data sending instructions send data to the SIM card. See the following instantiated procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in the SIM_CONF register.
2. Program the SIM_TIDE register to 0x0100 (TXTIDE = 1, RXTIDE = 0)

3. Program the SIM_IRQEN to 0x019C (enable RXERR, TXERR, T0END, TOUT and OVRUN interrupts)
4. Write CLA, INS, P1, P2 and P3 into SIM FIFO
5. Program the DMA controller:
   - DMA*n*_MSBSRC and DMA*n*_LSBSRC: Memory address reserved to store the transmitted characters
   - DMA*n*_MSBDST and DMA*n*_LSBDST: Address of the SIM_DATA register
   - DMA*n*_COUNT: Identical to P3
   - DMA*n*_CON: 0x0074
6. Write P3 into the SIM_P3 register and then (0x0100 | INS) into SIM_INS register. (Data transfer is initiated now.)
7. Enable the time-out counter by setting the TOUT bit to 1 in the SIM_CONF register.
8. Start the DMA controller by writing 0x8000 into the DMA*n*_START register.

Upon completion of the data sending instruction, T0END interrupt will be generated and the time-out counter should be disabled by setting the TOUT bit back to 0 in the SIM_CONF register.

If error occurs during the data transfer (RXERR, TXERR, OVRUN or TOUT interrupt is generated), the SIM card should be deactivated first and then activate the prior subsequent operations.

3.3 Keypad Scanner

3.3.1 General Description

The keypad can support two kinds of keypad: 5*5 single keypad and 5*5 double keypad.
The 5*5keypad can be divided into two parts: one is the keypad interface including 5 columns and 5 rows with one dedicated power-key, as shown in Fig. 1; the other is the key detection block which provides key pressed, key released and de-bounce mechanisms.
Each time the key is pressed or released, i.e. something different in the 5 x 5 matrix or power-key, the key detection block senses the change and recognizes if a key has been pressed or released. Whenever the key status changes and is stable, a KEYPAD IRQ is issued. The MCU can then read the key(s) pressed directly in KP_MEM1, KP_MEM2, KP_MEM3, KP_MEM4 and KP_MEM5 registers. To ensure that the key pressed information is not missed, the status register in keypad is not read-cleared by APB read command. The status register can only be changed by the key-pressed detection FSM.
This keypad can detect one or two key-pressed simultaneously with any combination. Fig. 2 shows one key pressed condition. Fig. 3(a) and Fig. 3 (b) illustrate two keys pressed cases. Since the key press detection depends on the HIGH or LOW level of the external keypad interface, if keys are pressed at the same time and there exists a

key that is on the same column and the same row with the other keys, the pressed key cannot be correctly decoded. For example, if there are three key presses: key1 = (x1, y1), key2 = (x2, y2), and key3 = (x1, y2), then both key3 and key4 = (x2, y1) are detected, and therefore they cannot be distinguished correctly. Hence, the keypad can detect only one or two keys pressed simultaneously at any combination. More than two keys pressed simultaneously in a specific pattern retrieve the wrong information.

The 5*5 double keypad can support a 5*5*2 = 50 keys matrix. The 50 keys are divided into 25 sub groups and each group consists 2 keys and a 20 ohm resistor. Besides the limitation of 5*5 single keypad, 5*5 double keypad has another limitation that it cannot detect two keys pressed simultaneously when the two keys are in one group, ie. 5*5 keypad cannot detect key0 and key1 pressed simultaneously or key16 and key17 pressed simultaneously.

Figure 25 ***5x5 single keys matrix (25 keys)***

Figure 26 ***5x5 double keypad matrix (50 keys)***

Figure 27 ***single keys matrix scan waveform***

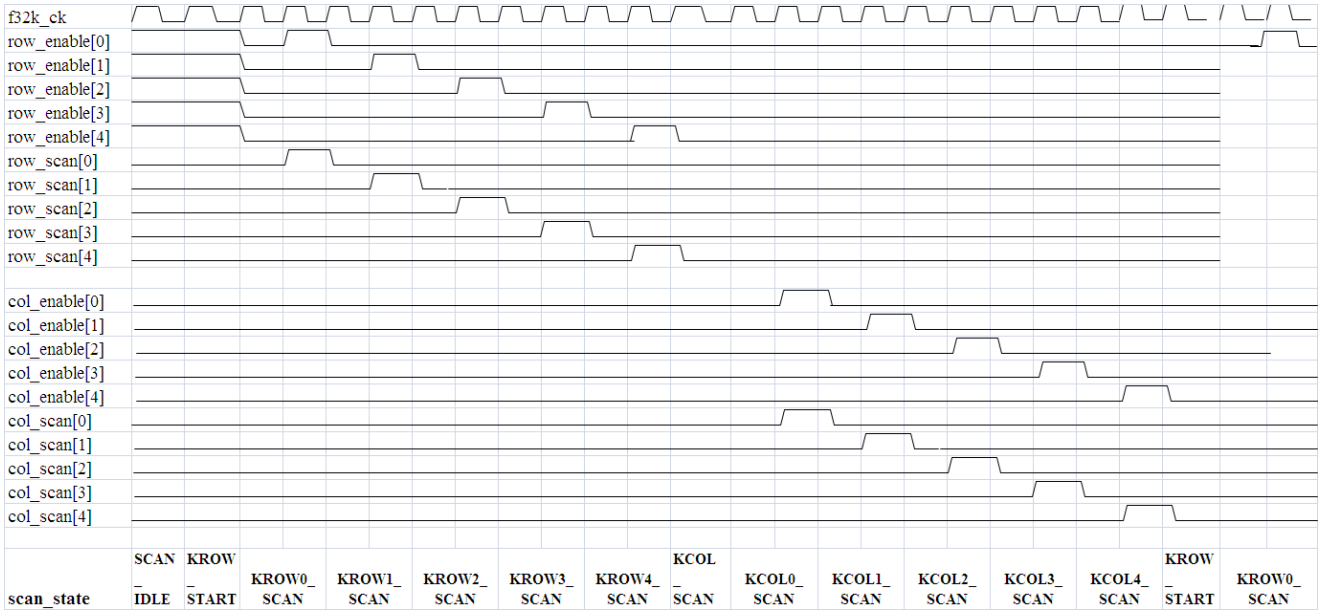


Figure 28 ***5*5 double keypad scan waveform***

Figure 29 ***One key pressed with de-bounce mechanism denoted***

Figure 30 ***(a) Two keys pressed, case 1 (b) Two keys pressed, case 2***

3.3.2 Register Definitions

Module name: KeypadScanner Base address: (+A00D0000h)

Address	Name	Width	Register Function
A00D0000	KP_STA	16	**Keypad status**
A00D0004	KP_MEM1	16	**Keypad scanning output Register** The register shows up the key-press status of key0(LSB)~key15.
A00D0008	KP_MEM2	16	**Keypad scanning output Register** The register shows up the key-press status of key16(LSB)~key31.
A00D000C	KP_MEM3	16	**Keypad scanning output Register** The register shows up the key-press status of key32(LSB)~key47
A00D0010	KP_MEM4	16	**Keypad scanning output Register** The register shows up the key-press status of key48(LSB)~key63.
A00D0014	KP_MEM5	16	**Keypad scanning output Register** The register shows up the key-press status of key64(LSB)~key71.
A00D0018	KP_DEBOUNCE	16	**De-bounce period setting** This register defines the waiting period before key press or release events are considered stale. If debounce setting is too small, keypad will be too sensitive and detect too many unexpected key press. The suitable debounce time setting must be adjust for the user's habit.
A00D001C	KP_SCAN_TIMING	16	**Keypad Scan Timing Adjustment Register** This register is used to set the double keypad scan timing. Mention: ROW_SCAN_DIV > ROW_INTERVAL_DIV and COL_SCAN_DIV > COL_INTERVAL_DIV. ROW_INTERVAL_DIV/COL_INTERVAL_DIV are used to lower power consumption for it decreases the actual scan number in the debounce time.
A00D0020	KP_SEL	16	**Keypad Selection Register** This register is used to select the single keypad or double keypad
A00D0024	KP_EN	16	**Keypad Enable Register** This register is used to enable/disable keypad.

A00D0000 KP_STA Keypad status 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																STA
Type																RO
Reset																0

Overview

Bit(s)	Mnemonic	Name	Description
0	STA	STA	**This register indicates the keypad status. The register is not cleared by the read operation.** 0: No key pressed 1: Key pressed

A00D0004 KP_MEM1 Keypad scanning output Register FFFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY15	KEY14	KEY13	KEY12	KEY11	KEY10	KEY9	KEY8	KEY7	KEY6	KEY5	KEY4	KEY3	KEY2	KEY1	KEY0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview The register shows up the key-press status of key0(LSB)~key15.

Bit(s)	Mnemonic	Name	Description
15	KEY15	KEY15	
14	KEY14	KEY14	
13	KEY13	KEY13	
12	KEY12	KEY12	
11	KEY11	KEY11	
10	KEY10	KEY10	
9	KEY9	KEY9	
8	KEY8	KEY8	
7	KEY7	KEY7	
6	KEY6	KEY6	
5	KEY5	KEY5	
4	KEY4	KEY4	
3	KEY3	KEY3	
2	KEY2	KEY2	
1	KEY1	KEY1	
0	KEY0	KEY0	

A00D0008 KP_MEM2 Keypad scanning output Register FFFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY31	KEY30	KEY29	KEY28	KEY27	KEY26	KEY25	KEY24	KEY23	KEY22	KEY21	KEY20	KEY19	KEY18	KEY17	KEY16
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview The register shows up the key-press status of key16(LSB)~key31.

Bit(s)	Mnemonic	Name	Description
15	KEY31	KEY31	
14	KEY30	KEY30	
13	KEY29	KEY29	
12	KEY28	KEY28	
11	KEY27	KEY27	
10	KEY26	KEY26	
9	KEY25	KEY25	
8	KEY24	KEY24	
7	KEY23	KEY23	
6	KEY22	KEY22	
5	KEY21	KEY21	
4	KEY20	KEY20	
3	KEY19	KEY19	
2	KEY18	KEY18	
1	KEY17	KEY17	
0	KEY16	KEY16	

A00D000C KP_MEM3 Keypad scanning output Register FFFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY47	KEY46	KEY45	KEY44	KEY43	KEY42	KEY41	KEY40	KEY39	KEY38	KEY37	KEY36	KEY35	KEY34	KEY33	KEY32

	7															2
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview The register shows up the key-press status of key32(LSB)~key47

Bit(s)	Mnemonic	Name	Description
15	KEY47	KEY47	
14	KEY46	KEY46	
13	KEY45	KEY45	
12	KEY44	KEY44	
11	KEY43	KEY43	
10	KEY42	KEY42	
9	KEY41	KEY41	
8	KEY40	KEY40	
7	KEY39	KEY39	
6	KEY38	KEY38	
5	KEY37	KEY37	
4	KEY36	KEY36	
3	KEY35	KEY35	
2	KEY34	KEY34	
1	KEY33	KEY33	
0	KEY32	KEY32	

A00D0010 KP_MEM4 Keypad scanning output Register FFFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	KEY63	KEY62	KEY61	KEY60	KEY59	KEY58	KEY57	KEY56	KEY55	KEY54	KEY53	KEY52	KEY51	KEY50	KEY49	KEY48
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview The register shows up the key-press status of key48(LSB)~key63.

Bit(s)	Mnemonic	Name	Description
15	KEY63	KEY63	**Not used at MT6260D**
14	KEY62	KEY62	**Not used at MT6260D**
13	KEY61	KEY61	**Not used at MT6260D**
12	KEY60	KEY60	**Not used at MT6260D**
11	KEY59	KEY59	
10	KEY58	KEY58	
9	KEY57	KEY57	
8	KEY56	KEY56	
7	KEY55	KEY55	
6	KEY54	KEY54	
5	KEY53	KEY53	
4	KEY52	KEY52	
3	KEY51	KEY51	
2	KEY50	KEY50	
1	KEY49	KEY49	
0	KEY48	KEY48	

A00D0014 KP_MEM5 Keypad scanning output Register 00FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									KEY71	KEY70	KEY69	KEY68	KEY67	KEY66	KEY65	KEY64
Type									RO	RO	RO	RO	RO	RO	RO	RO
Reset									1	1	1	1	1	1	1	1

Overview The register shows up the key-press status of key64(LSB)~key71.

Bit(s)	Mnemonic	Name	Description
7	KEY71	KEY71	**Not used at MT6260D**
6	KEY70	KEY70	**Not used at MT6260D**
5	KEY69	KEY69	**Not used at MT6260D**
4	KEY68	KEY68	**Not used at MT6260D**
3	KEY67	KEY67	**Not used at MT6260D**
2	KEY66	KEY66	**Not used at MT6260D**
1	KEY65	KEY65	**Not used at MT6260D**
0	KEY64	KEY64	**Not used at MT6260D**

\These five registers list the status of 72 keys on the keypad but KEY[8], KEY[17], KEY[26], KEY[35], KEY[44] , KEY[53] , KEY[62] , KEY[71] is dedicated for power key. For 5*5 keypad, KP_MEM1~4 registers list the status of 50 keys on the keypad. When the MCU receives the KEYPAD IRQ, both two registers must be read. If any key is pressed, the relative bit is set to 0.

In order to work normally, the corresponding pull-up/down setting must be programmed correctly. If some keys can be use because their COL or ROW is use as GPIO, these corresponding bit will be tie to high.

KEYS Status list of the 72 keys.

A00D0018 KP_DEBOUNCE De-bounce period setting 0400

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			DEBOUNCE													
Type			RW													
Reset			0	0	0	1	0	0	0	0	0	0	0	0	0	0

Overview This register defines the waiting period before key press or release events are considered stale. If debounce setting is too small, keypad will be too sensitive and detect too many unexpected key press. The suitable debounce time setting must be adjust for the user's habit.

Bit(s)	Mnemonic	Name	Description
13:0	DEBOUNCE	DEBOUNCE	**De-bounce time = KP_DEBOUNCE/32 ms.**

A00D001C KP_SCAN_TIMING Keypad Scan Timing Adjustment Register 0011

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	COL_INTERVAL_DIV				ROW_INTERVAL_DIV				COL_SCAN_DIV				ROW_SCAN_DIV			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Overview This register is used to set the double keypad scan timing. Mention: ROW_SCAN_DIV > ROW_INTERVAL_DIV and COL_SCAN_DIV > COL_INTERVAL_DIV.
ROW_INTERVAL_DIV/COL_INTERVAL_DIV are used to lower power consumption for it decreases the actual

scan number in the debounce time.

Bit(s)	Mnemonic	Name	Description
15:12	COL_INTERVAL_DIV	COL_INTERVAL_DIV	**used to set the COL SCAN interval cycle, that is , the cycles between two scans, default 0 means there is 1 cycle between two high scan pulse.**
11:8	ROW_INTERVAL_DIV	ROW_INTERVAL_DIV	**used to set the ROW SCAN interval cycle, that is , the cycles between two scans, default 0 means there is 1 cycle between two high scan pulse.**
7:4	COL_SCAN_DIV	COL_SCAN_DIV	**used to set the COL SCAN cycle which include COL_INTERVAL_DIV and the high pulse period, default 1 means there is 2 cycles for each scan, including 1 cycle high pulse and 1 cycle interval.**
3:0	ROW_SCAN_DIV	ROW_SCAN_DIV	**used to set the ROW SCAN cycle which include ROW_INTERVAL_DIV and the high pulse period, default 1 means there is 2 cycles for each scan, including 1 cycle high pulse and 1 cycle interval.**

Figure 31 ***kp timing register***

A00D0020 KP_SEL Keypad Selection Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																KP_SEL
Type																RW
Reset																0

Overview This register is used to select the single keypad or double keypad

Bit(s)	Mnemonic	Name	Description
0	KP_SEL	KP_SEL	0: use the single keypad 1: use the double keypad

A00D0024 KP_EN Keypad Enable Register 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																KP_EN
Type																RW
Reset																1

Overview This register is used to enable/disable keypad.

Bit(s)	Mnemonic	Name	Description
0	KP_EN	KP_EN	0: disable keypad (both of single key and double key will not work) 1: enable keypad (only one of single key and double key can work)

	COL0	COL1	COL2	COL3	COL4	COL5	COL6	COL7	PWRKEY
ROW7	63	64	65	66	67	68	69	70	71
ROW6	54	55	56	57	58	59	60	61	62
ROW5	45	46	47	48	49	50	51	52	53
ROW4	36	37	38	39	40	41	42	43	44
ROW3	27	28	29	30	31	32	33	34	35
ROW2	18	19	20	21	22	23	24	25	26
ROW1	9	10	11	12	13	14	15	16	17
ROW0	0	1	2	3	4	5	6	7	8

Table 45 5*5 single KEY's order number in COL/ROW matrix

	COL0	COL1	COL2	COL3	COL4	PWRKEY
ROW4	44/45	46/47	48/49	50/51	52/53	54
ROW3	33/34	35/36	37/38	39/40	41/42	43
ROW2	22/23	24/25	26/27	28/29	30/31	32
ROW1	11/12	13/14	15/16	17/18	19/20	21
ROW0	0/1	2/3	4/5	6/7	8/9	10

Table 46 5*5 double KEY's order number in COL/ROW matrix

3.4 General Purpose Inputs/Outputs

MT6260D offers 82 general purpose I/O pins. By setting up the control registers, the MCU software can control the direction, the output value, and read the input values on these pins. These GPIOs and GPOs are multiplexed with other functions to reduce the pin count. In addition, all GPO pins are removed. To facilitate application use, the software can configure which clock to send outside the chip. There are 2 clock-out ports embedded in 82 GPIO pins, and each clock-out can be programmed to

output appropriate clock source. Besides, when 2 GPIO function for the same peripheral IP, the smaller GPIO serial numbers have higher priority than larger numbers.

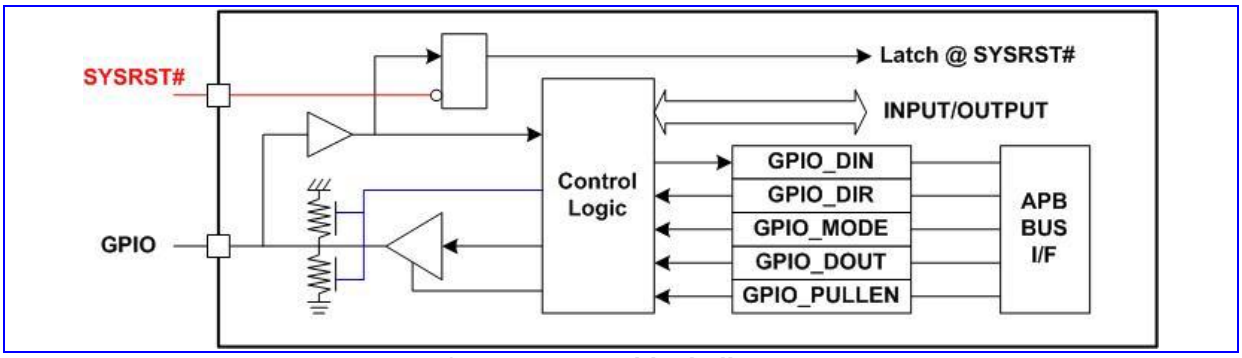


Figure 32: ***GPIO block diagram***

Register address	Register name	Synonym
0xA002_0000h	GPIO direction control register 0	GPIO_DIR0
0xA002_0010h	GPIO direction control register 1	GPIO_DIR1
0xA002_0020h	GPIO direction control register 2	GPIO_DIR2
0xA002_0030h	GPIO direction control register 3	GPIO_DIR3
0xA002_0040h	GPIO direction control register 4	GPIO_DIR4
0xA002_0100h	GPIO pull-up/down enable 0 register	GPIO_PULLEN0
0xA002_0110h	GPIO pull-up/down enable 1 register	GPIO_PULLEN1
0xA002_0120h	GPIO pull-up/down enable 2 register	GPIO_PULLEN2
0xA002_0130h	GPIO pull-up/down enable 3 register	GPIO_PULLEN3
0xA002_0140h	GPIO pull-up/down enable 4 register	GPIO_PULLEN4
0xA002_0200h	GPIO data inversion control register 0	GPIO_DINV0
0xA002_0210h	GPIO data inversion control register 1	GPIO_DINV1
0xA002_0220h	GPIO data inversion control register 2	GPIO_DINV2
0xA002_0230h	GPIO data inversion control register 3	GPIO_DINV3
0xA002_0240h	GPIO data inversion control register 4	GPIO_DINV4
0xA002_0300h	GPIO data output register 0	GPIO_DOUT0
0xA002_0310h	GPIO data output register 1	GPIO_DOUT1
0xA002_0320h	GPIO data output register 2	GPIO_DOUT2
0xA002_0330h	GPIO data output register 3	GPIO_DOUT3
0xA002_0340h	GPIO data output register 4	GPIO_DOUT4
0xA002_0400h	GPIO data input register 0	GPIO_DIN0
0xA002_0410h	GPIO data input register 1	GPIO_DIN1
0xA002_0420h	GPIO data input register 2	GPIO_DIN2
0xA002_0430h	GPIO data input register 3	GPIO_DIN3
0xA002_0440h	GPIO data input register 4	GPIO_DIN4
0xA002_0500h	GPIO pull-up/down select 0	GPIO_PULLSEL0
0xA002_0510h	GPIO pull-up/down select 1	GPIO_PULLSEL1
0xA002_0520h	GPIO pull-up/down select 2	GPIO_PULLSEL2

Register address	Register name	Synonym
0xA002_0530h	GPIO pull-up/down select 3	GPIO_PULLSEL3
0xA002_0540h	GPIO pull-up/down select 4	GPIO_PULLSEL4
0xA002_0600h	GPIO Schmit trigger control register 0	GPIO_SMT0
0xA002_0610h	GPIO Schmit trigger control register 1	GPIO_SMT1
0xA002_0620h	GPIO Schmit trigger control register 2	GPIO_SMT2
0xA002_0700h	GPIO slew rate control register 0	GPIO_SR0
0xA002_0710h	GPIO slew rate control register 1	GPIO_SR1
0xA002_0720h	GPIO slew rate control register 2	GPIO_SR2
0xA002_0730h	GPIO slew rate control register 3	GPIO_SR3
0xA002_0800h	GPIO driving control register 0	GPIO_DRV0
0xA002_0810h	GPIO driving control register 1	GPIO_DRV1
0xA002_0820h	GPIO driving control register 2	GPIO_DRV2
0xA002_0900h	GPIO Rx input buffer control register 0	GPIO_IES0
0xA002_0910h	GPIO Rx input buffer control register 1	GPIO_IES1
0xA002_0920h	GPIO Rx input buffer control register 2	GPIO_IES2
0xA002_0a00h	GPIO PUPD enable control register 0	GPIO_PUPDEN0
0xA002_0a10h	GPIO PUPD enable control register 1	GPIO_PUPDEN1
0xA002_0a20h	GPIO PUPD enable control register 2	GPIO_PUPDEN2
0xA002_0b00h	GPIO resistance enable control register 0	GPIO_RESEN0
0xA002_0b10h	GPIO resistance enable control register 1	GPIO_RESEN1
0xA002_0b20h	GPIO resistance enable control register 2	GPIO_RESEN2
0xA002_0b30h	GPIO resistance enable control register 3	GPIO_RESEN3
0xA002_0b40h	GPIO resistance enable control register 4	GPIO_RESEN4
0xA002_0b50h	GPIO resistance enable control register 5	GPIO_RESEN5
0xA002_0c00h	GPIO mode control register 0	GPIO_MODE0
0xA002_0c10h	GPIO mode control register 1	GPIO_MODE1
0xA002_0c20h	GPIO mode control register 2	GPIO_MODE2
0xA002_0c30h	GPIO mode control register 3	GPIO_MODE3
0xA002_0c40h	GPIO mode control register 4	GPIO_MODE4
0xA002_0c50h	GPIO mode control register 5	GPIO_MODE5
0xA002_0c60h	GPIO mode control register 6	GPIO_MODE6
0xA002_0c70h	GPIO mode control register 7	GPIO_MODE7
0xA002_0c80h	GPIO code control register 8	GPIO_MODE8
0xA002_0c90h	GPIO mode control register 9	GPIO_MODE9
0xA002_0d00h	GPIO Rx duty select control register	GPIO_RDSEL
0xA002_0d10h	GPIO Tx duty select control register	GPIO_TDSEL
0xA002_0e00h	CLK OUT0 setting	CLK_OUT0
0xA002_0e10h	CLK OUT1 setting	CLK_OUT1

Address	Name	Width	Register function
A0730104	**ACFG_PIN_PULLEN1**	16	Pin control 1

Address	Name	Width	Register function
A073010C	**ACFG_PIN_PULLSEL1**	16	Pin control 3
A0730118	**ACFG_PIN_DRV2**	16	Pin control 6
A0730120	**ACFG_PIN_SMT1**	16	Pin control 8
A0730128	**ACFG_PIN_IES1**	16	Pin control 10
A073012C	**ACFG_PIN_G**	16	Pin control 11
A0730130	**ACFG_PIN_DINV**	16	Pin control 12
A0730134	**ACFG_GPIO_OE**	16	GPIO OE control
A0730138	**ACFG_GPIO_DOUT**	16	GPIO DOUT control
A073013C	**ACFG_GPIO_DIN**	16	GPIO DIN status
A0730144	**ACFG_GPIO_MODE1**	16	GPIO mode control 1
A0730148	**ACFG_GPIO_MODE2**	16	GPIO mode control 2
A073014C	**ACFG_GPIO_MODE3**	16	GPIO mode control 3

0xA002_0000 GPIO Direction Control Register 0 GPIO_DIR0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO15**	**GPIO14**	**GPIO13**	**GPIO12**	**GPIO11**	**GPIO10**	**GPIO9**	**GPIO8**	**GPIO7**	**GPIO6**	**GPIO5**	**GPIO4**	**GPIO3**	**GPIO2**	**GPIO1**	**GPIO0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO15**	GPIO15_DIR	**GPIO 15 direction control** 0: Input 1: Output
14	**GPIO14**	GPIO14_DIR	**GPIO 14 direction control** 0: Input 1: Output
13	**GPIO13**	GPIO13_DIR	**GPIO 13 direction control** 0: Input 1: Output
12	**GPIO12**	GPIO12_DIR	**GPIO 12 direction control** 0: Input 1: Output
11	**GPIO11**	GPIO11_DIR	**GPIO 11 direction control** 0: Input 1: Output
10	**GPIO10**	GPIO10_DIR	**GPIO 10 direction control** 0: Input 1: Output
9	**GPIO9**	GPIO9_DIR	**GPIO 9 direction control** 0: Input 1: Output
8	**GPIO8**	GPIO8_DIR	**GPIO 8 direction control** 0: Input 1: Output
7	**GPIO7**	GPIO7_DIR	**GPIO 7 direction control**

Bit(s)	Mnemonic	Name	Description
			0: Input 1: Output
6	**GPIO6**	GPIO6_DIR	**GPIO 6 direction control** 0: Input 1: Output
5	**GPIO5**	GPIO5_DIR	**GPIO 5 direction control** 0: Input 1: Output
4	**GPIO4**	GPIO4_DIR	**GPIO 4 direction control** 0: Input 1: Output
3	**GPIO3**	GPIO3_DIR	**GPIO 3 direction control** 0: Input 1: Output
2	**GPIO2**	GPIO2_DIR	**GPIO 2 direction control** 0: Input 1: Output
1	**GPIO1**	GPIO1_DIR	**GPIO 1 direction control** 0: Input 1: Output
0	**GPIO0**	GPIO0_DIR	**GPIO 0 direction control** 0: Input 1: Output

0xA002_0010 GPIO Direction Control Register 1 GPIO_DIR1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO31**	GPIO31_DIR	**GPIO 31 direction control** 0: Input 1: Output
14	**GPIO30**	GPIO30_DIR	**GPIO 30 direction control** 0: Input 1: Output
13	**GPIO29**	GPIO29_DIR	**GPIO 29 direction control** 0: Input 1: Output
12	**GPIO28**	GPIO28_DIR	**GPIO 28 direction control** 0: Input 1: Output
11	**GPIO27**	GPIO27_DIR	**GPIO 27 direction control** 0: Input 1: Output
10	**GPIO26**	GPIO26_DIR	**GPIO 26 direction control** 0: Input 1: Output

Bit(s)	Mnemonic	Name	Description
9	**GPIO25**	GPIO25_DIR	**GPIO 25 direction control** 0: Input 1: Output
8	**GPIO24**	GPIO24_DIR	**GPIO 24 direction control** 0: Input 1: Output
7	**GPIO23**	GPIO23_DIR	**GPIO 23 direction control** 0: Input 1: Output
6	**GPIO22**	GPIO22_DIR	**GPIO 22 direction control** 0: Input 1: Output
5	**GPIO21**	GPIO21_DIR	**GPIO 21 direction control** 0: Input 1: Output
4	**GPIO20**	GPIO20_DIR	**GPIO 20 direction control** 0: Input 1: Output
3	**GPIO19**	GPIO19_DIR	**GPIO 19 direction control** 0: Input 1: Output
2	**GPIO18**	GPIO18_DIR	**GPIO 18 direction control** 0: Input 1: Output
1	**GPIO17**	GPIO17_DIR	**GPIO 17 direction control** 0: Input 1: Output
0	**GPIO16**	GPIO16_DIR	**GPIO 16 direction control** 0: Input 1: Output

0xA002_0020 GPIO Direction Control Register 2 GPIO_DIR2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO47**	GPIO47_DIR	**GPIO 47 direction control** 0: Input 1: Output
14	**GPIO46**	GPIO46_DIR	**GPIO 46 direction control** 0: Input 1: Output
13	**GPIO45**	GPIO45_DIR	**GPIO 45 direction control** 0: Input 1: Output
12	**GPIO44**	GPIO44_DIR	**GPIO 44 direction control** 0: Input

Bit(s)	Mnemonic	Name	Description
			1: Output
11	**GPIO43**	GPIO43_DIR	**GPIO 43 direction control** 0: Input 1: Output
10	**GPIO42**	GPIO42_DIR	**GPIO 42 direction control** 0: Input 1: Output
9	**GPIO41**	GPIO41_DIR	**GPIO 41 direction control** 0: Input 1: Output
8	**GPIO40**	GPIO40_DIR	**GPIO 40 direction control** 0: Input 1: Output
7	**GPIO39**	GPIO39_DIR	**GPIO 39 direction control** 0: Input 1: Output
6	**GPIO38**	GPIO38_DIR	**GPIO 38 direction control** 0: Input 1: Output
5	**GPIO37**	GPIO37_DIR	**GPIO 37 direction control** 0: Input 1: Output
4	**GPIO36**	GPIO36_DIR	**GPIO 36 direction control** 0: Input 1: Output
3	**GPIO35**	GPIO35_DIR	**GPIO 35 direction control** 0: Input 1: Output
2	**GPIO34**	GPIO34_DIR	**GPIO 34 direction control** 0: Input 1: Output
1	**GPIO33**	GPIO33_DIR	**GPIO 33 direction control** 0: Input 1: Output
0	**GPIO32**	GPIO32_DIR	**GPIO 32 direction control** 0: Input 1: Output

0xA002_0030 GPIO Direction Control Register 3 GPIO_DIR3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO63**	GPIO63_DIR	**GPIO 63 direction control** 0: Input 1: Output

Bit(s)	Mnemonic	Name	Description
14	**GPIO62**	GPIO62_DIR	**GPIO 62 direction control** 0: Input 1: Output
13	**GPIO61**	GPIO61_DIR	**GPIO 61 direction control** 0: Input 1: Output
12	**GPIO60**	GPIO60_DIR	**GPIO 60 direction control** 0: Input 1: Output
11	**GPIO59**	GPIO59_DIR	**GPIO 59 direction control** 0: Input 1: Output
10	**GPIO58**	GPIO58_DIR	**GPIO 58 direction control** 0: Input 1: Output
9	**GPIO57**	GPIO57_DIR	**GPIO 57 direction control** 0: Input 1: Output
8	**GPIO56**	GPIO56_DIR	**GPIO 56 direction control** 0: Input 1: Output
7	**GPIO55**	GPIO55_DIR	**GPIO 55 direction control** 0: Input 1: Output
6	**GPIO54**	GPIO54_DIR	**GPIO 54 direction control** 0: Input 1: Output
5	**GPIO53**	GPIO53_DIR	**GPIO 53 direction control** 0: Input 1: Output
4	**GPIO52**	GPIO52_DIR	**GPIO 52 direction control** 0: Input 1: Output
3	**GPIO51**	GPIO51_DIR	**GPIO 51 direction control** 0: Input 1: Output
2	**GPIO50**	GPIO50_DIR	**GPIO 50 direction control** 0: Input 1: Output
1	**GPIO49**	GPIO49_DIR	**GPIO 49 direction control** 0: Input 1: Output
0	**GPIO48**	GPIO48_DIR	**GPIO 48 direction control** 0: Input 1: Output

0xA002_0040 GPIO Direction Control Register 4 GPIO_DIR4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64

Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
8	**GPIO72**	GPIO72_DIR	**GPIO 72 direction control** 0: Input 1: Output
7	**GPIO71**	GPIO71_DIR	**GPIO 71 direction control** 0: Input 1: Output
6	**GPIO70**	GPIO70_DIR	**GPIO 70 direction control** 0: Input 1: Output
5	**GPIO69**	GPIO69_DIR	**GPIO 69 direction control** 0: Input 1: Output
4	**GPIO68**	GPIO68_DIR	**GPIO 68 direction control** 0: Input 1: Output
3	**GPIO67**	GPIO67_DIR	**GPIO 67 direction control** 0: Input 1: Output
2	**GPIO66**	GPIO66_DIR	**GPIO 66 direction control** 0: Input 1: Output
1	**GPIO65**	GPIO65_DIR	**GPIO 65 direction control** 0: Input 1: Output
0	**GPIO64**	GPIO64_DIR	**GPIO 64 direction control** 0: Input 1: Output

0xA002_0100 GPIO Pull-up/down Enable 0 — GPIO_PULLEN0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO15**	**GPIO14**	**GPIO13**	**GPIO12**	**GPIO11**	**GPIO10**	**GPIO9**	**GPIO8**	**GPIO7**	**GPIO6**	**GPIO5**	**GPIO4**	**GPIO3**	**GPIO2**	**GPIO1**	**GPIO0**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15	**GPIO15**	GPIO15_PULLEN	**Enables GPIO 15 pull-up/down** 0: Disable 1: Enable
14	**GPIO14**	GPIO14_PULLEN	**Enables GPIO 14 pull-up/down** 0: Disable 1: Enable
13	**GPIO13**	GPIO13_PULLEN	**Enables GPIO 13 pull-up/down** 0: Disable 1: Enable

Bit(s)	Mnemonic	Name	Description
12	**GPIO12**	GPIO12_PULLEN	**Enables GPIO 12 pull-up/down** 0: Disable 1: Enable
11	**GPIO11**	GPIO11_PULLEN	**Enables GPIO 11 pull-up/down** 0: Disable 1: Enable
10	**GPIO10**	GPIO10_PULLEN	**Enables GPIO 10 pull-up/down** 0: Disable 1: Enable
9	**GPIO9**	GPIO9_PULLEN	**Enables GPIO 9 pull-up/down** 0: Disable 1: Enable
8	**GPIO8**	GPIO8_PULLEN	**Enables GPIO 8 pull-up/down** 0: Disable 1: Enable
7	**GPIO7**	GPIO7_PULLEN	**Enables GPIO 7 pull-up/down** 0: Disable 1: Enable
6	**GPIO6**	GPIO6_PULLEN	**Enables GPIO 6 pull-up/down** 0: Disable 1: Enable
5	**GPIO5**	GPIO5_PULLEN	**Enables GPIO 5 pull-up/down** 0: Disable 1: Enable
4	**GPIO4**	GPIO4_PULLEN	**Enables GPIO 4 pull-up/down** 0: Disable 1: Enable
3	**GPIO3**	GPIO3_PULLEN	**Enables GPIO 3 pull-up/down** 0: Disable 1: Enable
2	**GPIO2**	GPIO2_PULLEN	**Enables GPIO 2 pull-up/down** 0: Disable 1: Enable
1	**GPIO1**	GPIO1_PULLEN	**Enables GPIO 1 pull-up/down** 0: Disable 1: Enable
0	**GPIO0**	GPIO0_PULLEN	**Enables GPIO 0 pull-up/down** 0: Disable 1: Enable

0xA002_0110 GPIO Pull-up/down Enable 1

GPIO_PULLEN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15	**GPIO31**	GPIO31_ PULLEN	**Enables GPIO 31 pull-up/down**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
14	**GPIO30**	GPIO30_ PULLEN	**Enables GPIO 30 pull-up/down** 0: Disable 1: Enable
13	**GPIO29**	GPIO29_ PULLEN	**Enables GPIO 29 pull-up/down** 0: Disable 1: Enable
12	**GPIO28**	GPIO28_ PULLEN	**Enables GPIO 28 pull-up/down** 0: Disable 1: Enable
11	**GPIO27**	GPIO27_ PULLEN	**Enables GPIO 27 pull-up/down** 0: Disable 1: Enable
10	**GPIO26**	GPIO26_ PULLEN	**Enables GPIO 26 pull-up/down** 0: Disable 1: Enable
9	**GPIO25**	GPIO25_ PULLEN	**Enables GPIO 25 pull-up/down** 0: Disable 1: Enable
8	**GPIO24**	GPIO24_ PULLEN	**Enables GPIO 24 pull-up/down** 0: Disable 1: Enable
7	**GPIO23**	GPIO23_ PULLEN	**Enables GPIO 23 pull-up/down** 0: Disable 1: Enable
6	**GPIO22**	GPIO22_ PULLEN	**Enables GPIO 22 pull-up/down** 0: Disable 1: Enable
5	**GPIO21**	GPIO21_ PULLEN	**Enables GPIO 21 pull-up/down** 0: Disable 1: Enable
4	**GPIO20**	GPIO20_ PULLEN	**Enables GPIO 20 pull-up/down** 0: Disable 1: Enable
3	**GPIO19**	GPIO19_ PULLEN	**Enables GPIO 19 pull-up/down** 0: Disable 1: Enable
2	**GPIO18**	GPIO18_ PULLEN	**Enables GPIO 18 pull-up/down** 0: Disable 1: Enable
1	**GPIO17**	GPIO17_ PULLEN	**Enables GPIO 17 pull-up/down** 0: Disable 1: Enable
0	**GPIO16**	GPIO16_PULLEN	**Enables GPIO 16 pull-up/down** 0: Disable 1: Enable

0xA002_0120 GPIO Pull-up/down Enable 2

GPIO_PULLEN2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	1	0	0	0	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15	**GPIO47**	GPIO47_ PULLEN	**Enables GPIO 47 pull-up/down** 0: Disable 1: Enable
14	**GPIO46**	GPIO46_ PULLEN	**Enables GPIO 46 pull-up/down** 0: Disable 1: Enable
13	**GPIO45**	GPIO45_ PULLEN	**Enables GPIO 45 pull-up/down** 0: Disable 1: Enable
12	**GPIO44**	GPIO44_ PULLEN	**Enables GPIO 44 pull-up/down** 0: Disable 1: Enable
11	**GPIO43**	GPIO43_ PULLEN	**Enables GPIO 43 pull-up/down** 0: Disable 1: Enable
10	**GPIO42**	GPIO42_ PULLEN	**Enables GPIO 42 pull-up/down** 0: Disable 1: Enable
9	**GPIO41**	GPIO41_ PULLEN	**Enables GPIO 41 pull-up/down** 0: Disable 1: Enable
8	**GPIO40**	GPIO40_ PULLEN	**Enables GPIO 40 pull-up/down** 0: Disable 1: Enable
7	**GPIO39**	GPIO39_ PULLEN	**Enables GPIO 39 pull-up/down** 0: Disable 1: Enable
6	**GPIO38**	GPIO38_ PULLEN	**Enables GPIO 38 pull-up/down** 0: Disable 1: Enable
5	**GPIO37**	GPIO37_ PULLEN	**Enables GPIO 37 pull-up/down** 0: Disable 1: Enable
4	**GPIO36**	GPIO36_ PULLEN	**Enables GPIO 36 pull-up/down** 0: Disable 1: Enable
3	**GPIO35**	GPIO35_ PULLEN	**Enables GPIO 35 pull-up/down** 0: Disable 1: Enable
2	**GPIO34**	GPIO34_ PULLEN	**Enables GPIO 34 pull-up/down** 0: Disable 1: Enable
1	**GPIO33**	GPIO33_ PULLEN	**Enables GPIO 33 pull-up/down**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
0	**GPIO32**	GPIO32_ PULLEN	**Enables GPIO 32 pull-up/down** 0: Disable 1: Enable

0xA002_0130 GPIO Pull-up/down Enable 3

GPIO_PULLEN 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15	**GPIO63**	GPIO63_ PULLEN	**Enables GPIO 63 pull-up/down** 0: Disable 1: Enable
14	**GPIO62**	GPIO62_ PULLEN	**Enables GPIO 62 pull-up/down** 0: Disable 1: Enable
13	**GPIO61**	GPIO61_ PULLEN	**Enables GPIO 61 pull-up/down** 0: Disable 1: Enable
12	**GPIO60**	GPIO60_ PULLEN	**Enables GPIO 60 pull-up/down** 0: Disable 1: Enable
11	**GPIO59**	GPIO59_ PULLEN	**Enables GPIO 59 pull-up/down** 0: Disable 1: Enable
10	**GPIO58**	GPIO58_ PULLEN	**Enables GPIO 58 pull-up/down** 0: Disable 1: Enable
9	**GPIO57**	GPIO57_ PULLEN	**Enables GPIO 57 pull-up/down** 0: Disable 1: Enable
8	**GPIO56**	GPIO56_ PULLEN	**Enables GPIO 56 pull-up/down** 0: Disable 1: Enable
7	**GPIO55**	GPIO55_ PULLEN	**Enables GPIO 55 pull-up/down** 0: Disable 1: Enable
6	**GPIO54**	GPIO54_ PULLEN	**Enables GPIO 54 pull-up/down** 0: Disable 1: Enable
5	**GPIO53**	GPIO53_ PULLEN	**Enables GPIO 53 pull-up/down** 0: Disable 1: Enable
4	**GPIO52**	GPIO52_ PULLEN	**Enables GPIO 52 pull-up/down**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
3	**GPIO51**	GPIO51_ PULLEN	**Enables GPIO 51 pull-up/down** 0: Disable 1: Enable
2	**GPIO50**	GPIO50_ PULLEN	**Enables GPIO 50 pull-up/down** 0: Disable 1: Enable
1	**GPIO49**	GPIO49_ PULLEN	**Enables GPIO 49 pull-up/down** 0: Disable 1: Enable
0	**GPIO48**	GPIO48_ PULLEN	**Enables GPIO 48 pull-up/down** 0: Disable 1: Enable

0xA002_0140 GPIO Pull-up/down Enable 4 GPIO_PULLEN 4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
8	**GPIO72**	GPIO72_ PULLEN	**Enables GPIO 72 pull-up/down** 0: Disable 1: Enable
7	**GPIO71**	GPIO71_ PULLEN	**Enables GPIO 71 pull-up/down** 0: Disable 1: Enable
6	**GPIO70**	GPIO70_ PULLEN	**Enables GPIO 70 pull-up/down** 0: Disable 1: Enable
5	**GPIO69**	GPIO69_ PULLEN	**Enables GPIO 69 pull-up/down** 0: Disable 1: Enable
4	**GPIO68**	GPIO68_ PULLEN	**Enables GPIO 68 pull-up/down** 0: Disable 1: Enable
3	**GPIO67**	GPIO67_ PULLEN	**Enables GPIO 67 pull-up/down** 0: Disable 1: Enable
2	**GPIO66**	GPIO66_ PULLEN	**Enables GPIO 66 pull-up/down** 0: Disable 1: Enable
1	**GPIO65**	GPIO65_ PULLEN	**Enables GPIO 65 pull-up/down** 0: Disable 1: Enable
0	**GPIO64**	GPIO64_ PULLEN	**Enables GPIO 64 pull-up/down**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable

0xA002_0200 GPIO Data Inversion Control Register 0 GPIO_DINV0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV15	INV14	INV13	INV12	INV11	INV10	INV9	INV8	INV7	INV6	INV5	INV4	INV3	INV2	INV1	INV0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**INV15**	GPIO15_DINV	**GPIO 15 inversion control** 0: Disable 1: Enable
14	**INV14**	GPIO14_DINV	**GPIO 14 inversion control** 0: Disable 1: Enable
13	**INV13**	GPIO13_DINV	**GPIO 13 inversion control** 0: Disable 1: Enable
12	**INV12**	GPIO12_DINV	**GPIO 12 inversion control** 0: Disable 1: Enable
11	**INV11**	GPIO11_DINV	**GPIO 11 inversion control** 0: Disable 1: Enable
10	**INV10**	GPIO10_DINV	**GPIO 10 inversion control** 0: Disable 1: Enable
9	**INV9**	GPIO9_DINV	**GPIO 9 inversion control** 0: Disable 1: Enable
8	**INV8**	GPIO8_DINV	**GPIO 8 inversion control** 0: Disable 1: Enable
7	**INV7**	GPIO7_DINV	**GPIO 7 inversion control** 0: Disable 1: Enable
6	**INV6**	GPIO6_DINV	**GPIO 6 inversion control** 0: Disable 1: Enable
5	**INV5**	GPIO5_DINV	**GPIO 5 inversion control** 0: Disable 1: Enable
4	**INV4**	GPIO4_DINV	**GPIO 4 inversion control** 0: Disable 1: Enable
3	**INV3**	GPIO3_DINV	**GPIO 3 inversion control** 0: Disable 1: Enable
2	**INV2**	GPIO2_DINV	**GPIO 2 inversion control** 0: Disable

Bit(s)	Mnemonic	Name	Description
			1: Enable
1	**INV1**	GPIO1_DINV	**GPIO 1 inversion control** 0: Disable 1: Enable
0	**INV0**	GPIO0_DINV	**GPIO 0 Inversion Control** 0: Disable 1: Enable

0xA002_0210 GPIO Data Inversion Control Register 1 GPIO_DINV1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV31	INV30	INV29	INV28	INV27	INV26	INV25	INV24	INV23	INV22	INV21	INV20	INV19	IVN18	INV17	INV16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**INV31**	GPIO31_DINV	**GPIO 31 inversion control** 0: Disable 1: Enable
14	**INV30**	GPIO30_DINV	**GPIO 30 inversion control** 0: Disable 1: Enable
13	**INV29**	GPIO29_DINV	**GPIO 29 inversion control** 0: Disable 1: Enable
12	**INV28**	GPIO28_DINV	**GPIO 28 inversion control** 0: Disable 1: Enable
11	**INV27**	GPIO27_DINV	**GPIO 27 inversion control** 0: Disable 1: Enable
10	**INV26**	GPIO26_DINV	**GPIO 26 inversion control** 0: Disable 1: Enable
9	**INV25**	GPIO25_DINV	**GPIO 25 inversion control** 0: Disable 1: Enable
8	**INV24**	GPIO24_DINV	**GPIO 24 inversion control** 0: Disable 1: Enable
7	**INV23**	GPIO23_DINV	**GPIO 23 inversion control** 0: Disable 1: Enable
6	**INV22**	GPIO22_DINV	**GPIO 22 inversion control** 0: Disable 1: Enable
5	**INV21**	GPIO21_DINV	**GPIO 21 inversion control** 0: Disable 1: Enable
4	**INV20**	GPIO20_DINV	**GPIO 20 inversion control**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
3	**INV19**	GPIO19_DINV	**GPIO 19 inversion control** 0: Disable 1: Enable
2	**INV18**	GPIO18_DINV	**GPIO 18 inversion control** 0: Disable 1: Enable
1	**INV17**	GPIO17_DINV	**GPIO 17 inversion control** 0: Disable 1: Enable
0	**INV16**	GPIO16_DINV	**GPIO 16 inversion control** 0: Disable 1: Enable

0xA002_0220 GPIO Data Inversion Control Register 2 GPIO_DINV2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV47	INV46	INV45	INV44	INV43	INV42	INV41	INV40	INV39	INV38	INV37	INV36	INV35	INV34	INV33	INV32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**INV47**	GPIO47_DINV	**GPIO 47 inversion control** 0: Disable 1: Enable
14	**INV46**	GPIO46_DINV	**GPIO 46 inversion control** 0: Disable 1: Enable
13	**INV45**	GPIO45_DINV	**GPIO 45 inversion control** 0: Disable 1: Enable
12	**INV44**	GPIO44_DINV	**GPIO 44 inversion control** 0: Disable 1: Enable
11	**INV43**	GPIO43_DINV	**GPIO 43 inversion control** 0: Disable 1: Enable
10	**INV42**	GPIO42_DINV	**GPIO 42 inversion control** 0: Disable 1: Enable
9	**INV41**	GPIO41_DINV	**GPIO 41 inversion control** 0: Disable 1: Enable
8	**INV40**	GPIO40_DINV	**GPIO 40 inversion control** 0: Disable 1: Enable
7	**INV39**	GPIO39_DINV	**GPIO 39 inversion control** 0: Disable 1: Enable

Bit(s)	Mnemonic	Name	Description
6	**INV38**	GPIO38_DINV	**GPIO 38 inversion control** 0: Disable 1: Enable
5	**INV37**	GPIO37_DINV	**GPIO 37 inversion control** 0: Disable 1: Enable
4	**INV36**	GPIO36_DINV	**GPIO 36 inversion control** 0: Disable 1: Enable
3	**INV35**	GPIO35_DINV	**GPIO 35 inversion control** 0: Disable 1: Enable
2	**INV34**	GPIO34_DINV	**GPIO 34 inversion control** 0: Disable 1: Enable
1	**INV33**	GPIO33_DINV	**GPIO 33 inversion control** 0: Disable 1: Enable
0	**INV32**	GPIO32_DINV	**GPIO 32 inversion control** 0: Disable 1: Enable

0xA002_0230 GPIO Data Inversion Control Register 3 GPIO_DINV3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INV63	INV62	INV61	INV60	INV59	INV58	INV57	INV56	INV55	INV54	INV53	INV52	INV51	INV50	INV49	INV48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**INV63**	GPIO63_DINV	**GPIO 63 inversion control** 0: Disable 1: Enable
14	**INV62**	GPIO62_DINV	**GPIO 62 inversion control** 0: Disable 1: Enable
13	**INV61**	GPIO61_DINV	**GPIO 61 inversion control** 0: Disable 1: Enable
12	**INV60**	GPIO60_DINV	**GPIO 60 inversion control** 0: Disable 1: Enable
11	**INV59**	GPIO59_DINV	**GPIO 59 inversion control** 0: Disable 1: Enable
10	**INV58**	GPIO58_DINV	**GPIO 58 inversion control** 0: Disable 1: Enable
9	**INV57**	GPIO57_DINV	**GPIO 57 inversion control** 0: Disable

Bit(s)	Mnemonic	Name	Description
			1: Enable
8	**INV56**	GPIO56_DINV	**GPIO 56 inversion control** 0: Disable 1: Enable
7	**INV55**	GPIO55_DINV	**GPIO 55 inversion control** 0: Disable 1: Enable
6	**INV54**	GPIO54_DINV	**GPIO 54 inversion control** 0: Disable 1: Enable
5	**INV53**	GPIO53_DINV	**GPIO 53 inversion control** 0: Disable 1: Enable
4	**INV52**	GPIO52_DINV	**GPIO 52 inversion control** 0: Disable 1: Enable
3	**INV51**	GPIO51_DINV	**GPIO 51 inversion control** 0: Disable 1: Enable
2	**INV50**	GPIO50_DINV	**GPIO 50 inversion control** 0: Disable 1: Enable
1	**INV49**	GPIO49_DINV	**GPIO 49 inversion control** 0: Disable 1: Enable
0	**INV48**	GPIO48_DINV	**GPIO 48 inversion control** 0: Disable 1: Enable

0xA002_0240 GPIO Data Inversion Control Register 4 GPIO_DINV4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								INV72	INV71	INV70	INV69	INV68	INV67	INV66	INV65	INV64
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
8	**INV72**	GPIO72_DINV	**GPIO 72 inversion control** 0: Disable 1: Enable
7	**INV71**	GPIO71_DINV	**GPIO 71 inversion control** 0: Disable 1: Enable
6	**INV70**	GPIO70_DINV	**GPIO 70 inversion control** 0: Disable 1: Enable
5	**INV69**	GPIO69_DINV	**GPIO 69 inversion control** 0: Disable 1: Enable
4	**INV68**	GPIO68_DINV	**GPIO 68 inversion control**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
3	**INV67**	GPIO67_DINV	**GPIO 67 inversion control** 0: Disable 1: Enable
2	**INV66**	GPIO66_DINV	**GPIO 66 inversion control** 0: Disable 1: Enable
1	**INV65**	GPIO65_DINV	**GPIO 65 inversion control** 0: Disable 1: Enable
0	**INV64**	GPIO64_DINV	**GPIO 64 inversion control** 0: Disable 1: Enable

0xA002_0300 GPIO Data Output Register 0 GPIO_DOUT0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPO15**	GPIO15_DOUT	**GPIO 15 data output value** 0: Output 0 1: Output 1
14	**GPO14**	GPIO14_DOUT	**GPIO 14 data output value** 0: Output 0 1: Output 1
13	**GPO13**	GPIO13_DOUT	**GPIO 13 data output value** 0: Output 0 1: Output 1
12	**GPO12**	GPIO12_DOUT	**GPIO 12 data output value** 0: Output 0 1: Output 1
11	**GPO11**	GPIO11_DOUT	**GPIO 11 data output value** 0: Output 0 1: Output 1
10	**GPO10**	GPIO10_DOUT	**GPIO 10 data output value** 0: Output 0 1: Output 1
9	**GPO9**	GPIO9_DOUT	**GPIO 9 data output value** 0: Output 0 1: Output 1
8	**GPO8**	GPIO8_DOUT	**GPIO 8 data output value** 0: Output 0 1: Output 1
7	**GPO7**	GPIO7_DOUT	**GPIO 7 data output value** 0: Output 0 1: Output 1

Bit(s)	Mnemonic	Name	Description
6	**GPO6**	GPIO6_DOUT	**GPIO 6 data output value** 0: Output 0 1: Output 1
5	**GPO5**	GPIO5_DOUT	**GPIO 5 data output value** 0: Output 0 1: Output 1
4	**GPO4**	GPIO4_DOUT	**GPIO 4 data output value** 0: Output 0 1: Output 1
3	**GPO3**	GPIO3_DOUT	**GPIO 3 data output value** 0: Output 0 1: Output 1
2	**GPO2**	GPIO2_DOUT	**GPIO 2 data output value** 0: Output 0 1: Output 1
1	**GPO1**	GPIO1_DOUT	**GPIO 1 data output value** 0: Output 0 1: Output 1
0	**GPO0**	GPIO0_DOUT	**GPIO 0 data output value** 0: Output 0 1: Output 1

0xA002_0310 GPIO Data Output Register 1 GPIO_DOUT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPO31**	GPIO31_DOUT	**GPIO 31 data output value** 0: Output 0 1: Output 1
14	**GPO30**	GPIO30_DOUT	**GPIO 30 data output value** 0: Output 0 1: Output 1
13	**GPO29**	GPIO29_DOUT	**GPIO 29 data output value** 0: Output 0 1: Output 1
12	**GPO28**	GPIO28_DOUT	**GPIO 28 data output value** 0: Output 0 1: Output 1
11	**GPO27**	GPIO27_DOUT	**GPIO 27 data output value** 0: Output 0 1: Output 1
10	**GPO26**	GPIO26_DOUT	**GPIO 26 data output value** 0: Output 0 1: Output 1
9	**GPO25**	GPIO25_DOUT	**GPIO 25 data output value** 0: Output 0

Bit(s)	Mnemonic	Name	Description
			1: Output 1
8	**GPO24**	GPIO24_DOUT	**GPIO 24 data output value** 0: Output 0 1: Output 1
7	**GPO23**	GPIO23_DOUT	**GPIO 23 data output value** 0: Output 0 1: Output 1
6	**GPO22**	GPIO22_DOUT	**GPIO 22 data output value** 0: Output 0 1: Output 1
5	**GPO21**	GPIO21_DOUT	**GPIO 21 data output value** 0: Output 0 1: Output 1
4	**GPO20**	GPIO20_DOUT	**GPIO 20 data output value** 0: Output 0 1: Output 1
3	**GPO19**	GPIO19_DOUT	**GPIO 19 data output value** 0: Output 0 1: Output 1
2	**GPO18**	GPIO18_DOUT	**GPIO 18 data output value** 0: Output 0 1: Output 1
1	**GPO17**	GPIO17_DOUT	**GPIO 17 data output value** 0: Output 0 1: Output 1
0	**GPO16**	GPIO16_DOUT	**GPIO 16 data output value** 0: Output 0 1: Output 1

0xA002_0320 GPIO Data Output Register 2 GPIO_DOUT2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPO47**	GPIO47_DOUT	**GPIO 47 data output value** 0: Output 0 1: Output 1
14	**GPO46**	GPIO46_DOUT	**GPIO 46 data output value** 0: Output 0 1: Output 1
13	**GPO45**	GPIO45_DOUT	**GPIO 45 data output value** 0: Output 0 1: Output 1
12	**GPO44**	GPIO44_DOUT	**GPIO 44 data output value** 0: Output 0 1: Output 1
11	**GPO43**	GPIO43_DOUT	**GPIO 43 data output value**

Bit(s)	Mnemonic	Name	Description
			0: Output 0 1: Output 1
10	**GPO42**	GPIO42_DOUT	**GPIO 42 data output value** 0: Output 0 1: Output 1
9	**GPO41**	GPIO41_DOUT	**GPIO 41 data output value** 0: Output 0 1: Output 1
8	**GPO40**	GPIO40_DOUT	**GPIO 40 data output value** 0: Output 0 1: Output 1
7	**GPO39**	GPIO39_DOUT	**GPIO 39 data output value** 0: Output 0 1: Output 1
6	**GPO38**	GPIO38_DOUT	**GPIO 38 data output value** 0: Output 0 1: Output 1
5	**GPO37**	GPIO37_DOUT	**GPIO 37 data output value** 0: Output 0 1: Output 1
4	**GPO36**	GPIO36_DOUT	**GPIO 36 data output value** 0: Output 0 1: Output 1
3	**GPO35**	GPIO35_DOUT	**GPIO 35 data output value** 0: Output 0 1: Output 1
2	**GPO34**	GPIO34_DOUT	**GPIO 34 data output value** 0: Output 0 1: Output 1
1	**GPO33**	GPIO33_DOUT	**GPIO 33 data output value** 0: Output 0 1: Output 1
0	**GPO32**	GPIO32_DOUT	**GPIO 32 data output value** 0: Output 0 1: Output 1

0xA002_0330 GPIO Data Output Register 3 GPIO_DOUT3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPO63**	GPIO63_DOUT	**GPIO 63 data output value** 0: Output 0 1: Output 1
14	**GPO62**	GPIO62_DOUT	**GPIO 62 data output value** 0: Output 0 1: Output 1

Bit(s)	Mnemonic	Name	Description
13	**GPO61**	GPIO61_DOUT	**GPIO 61 data output value** 0: Output 0 1: Output 1
12	**GPO60**	GPIO60_DOUT	**GPIO 60 data output value** 0: Output 0 1: Output 1
11	**GPO59**	GPIO59_DOUT	**GPIO 59 data output value** 0: Output 0 1: Output 1
10	**GPO58**	GPIO58_DOUT	**GPIO 58 data output value** 0: Output 0 1: Output 1
9	**GPO57**	GPIO57_DOUT	**GPIO 57 data output value** 0: Output 0 1: Output 1
8	**GPO56**	GPIO56_DOUT	**GPIO 56 data output value** 0: Output 0 1: Output 1
7	**GPO55**	GPIO55_DOUT	**GPIO 55 data output value** 0: Output 0 1: Output 1
6	**GPO54**	GPIO54_DOUT	**GPIO 54 data output value** 0: Output 0 1: Output 1
5	**GPO53**	GPIO53_DOUT	**GPIO 53 data output value** 0: Output 0 1: Output 1
4	**GPO52**	GPIO52_DOUT	**GPIO 52 data output value** 0: Output 0 1: Output 1
3	**GPO51**	GPIO51_DOUT	**GPIO 51 data output value** 0: Output 0 1: Output 1
2	**GPO50**	GPIO50_DOUT	**GPIO 50 data output value** 0: Output 0 1: Output 1
1	**GPO49**	GPIO49_DOUT	**GPIO 49 data output value** 0: Output 0 1: Output 1
0	**GPO48**	GPIO48_DOUT	**GPIO 48 data output value** 0: Output 0 1: Output 1

0xA002_0340 GPIO Data Output Register 4 GPIO_DOUT4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
8	**GPO72**	GPIO72_DOUT	**GPIO 72 data output value** 0: Output 0 1: Output 1
7	**GPO71**	GPIO71_DOUT	**GPIO 71 data output value** 0: Output 0 1: Output 1
6	**GPO70**	GPIO70_DOUT	**GPIO 70 data output value** 0: Output 0 1: Output 1
5	**GPO69**	GPIO69_DOUT	**GPIO 69 data output value** 0: Output 0 1: Output 1
4	**GPO68**	GPIO68_DOUT	**GPIO 68 data output value** 0: Output 0 1: Output 1
3	**GPO67**	GPIO67_DOUT	**GPIO 67 data output value** 0: Output 0 1: Output 1
2	**GPO66**	GPIO66_DOUT	**GPIO 66 data output value** 0: Output 0 1: Output 1
1	**GPO65**	GPIO65_DOUT	**GPIO 65 data output value** 0: Output 0 1: Output 1
0	**GPO64**	GPIO64_DOUT	**GPIO 64 data output value** 0: Output 0 1: Output 1

0xA002_0400 GPIO Data Input Register 0 GPIO_DIN0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit(s)	Mnemonic	Name	Description
15	**GPIO15**	GPIO15_DIN	GPIO 15 data input value
14	**GPIO14**	GPIO14_DIN	GPIO 14 data input value
13	**GPIO13**	GPIO13_DIN	GPIO 13 data input value
12	**GPIO12**	GPIO12_DIN	GPIO 12 data input value
11	**GPIO11**	GPIO11_DIN	GPIO 11 data input value
10	**GPIO10**	GPIO10_DIN	GPIO 10 data input value
9	**GPIO9**	GPIO9_DIN	GPIO 9 data input value
8	**GPIO8**	GPIO8_DIN	GPIO 8 data input value
7	**GPIO7**	GPIO7_DIN	GPIO 7 data input value
6	**GPIO6**	GPIO6_DIN	GPIO 6 data input value
5	**GPIO5**	GPIO5_DIN	GPIO 5 data input value

Bit(s)	Mnemonic	Name	Description
4	**GPIO4**	GPIO4_DIN	GPIO 4 data input value
3	**GPIO3**	GPIO3_DIN	GPIO 3 data input value
2	**GPIO2**	GPIO2_DIN	GPIO 2 data input value
1	**GPIO1**	GPIO1_DIN	GPIO 1 data input value
0	**GPIO0**	GPIO0_DIN	GPIO 0 data input value

0xA002_0410 GPIO Data Input Register 1 GPIO_DIN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO31	GPIO30	GPIO29	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	GPIO23	GPIO22	GPIO21	PGIO20	GPIO19	GPIO18	GPIO17	GPIO16
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit(s)	Mnemonic	Name	Description
15	**GPIO31**	GPIO31_DIN	GPIO 31 data input value
14	**GPIO30**	GPIO30_DIN	GPIO 30 data input value
13	**GPIO29**	GPIO29_DIN	GPIO 29 data input value
12	**GPIO28**	GPIO28_DIN	GPIO 28 data input value
11	**GPIO27**	GPIO27_DIN	GPIO 27 data input value
10	**GPIO26**	GPIO26_DIN	GPIO 26 data input value
9	**GPIO25**	GPIO25_DIN	GPIO 25 data input value
8	**GPIO24**	GPIO24_DIN	GPIO 24 data input value
7	**GPIO23**	GPIO23_DIN	GPIO 23 data input value
6	**GPIO22**	GPIO22_DIN	GPIO 22 data input value
5	**GPIO21**	GPIO21_DIN	GPIO 21 data input value
4	**GPIO20**	GPIO20_DIN	GPIO 20 data input value
3	**GPIO19**	GPIO19_DIN	GPIO 19 data input value
2	**GPIO18**	GPIO18_DIN	GPIO 18 data input value
1	**GPIO17**	GPIO17_DIN	GPIO 17 data input value
0	**GPIO16**	GPIO16_DIN	GPIO 16 data input value

0xA002_0420 GPIO Data Input Register 2 GPIO_DIN2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit(s)	Mnemonic	Name	Description
15	**GPIO47**	GPIO47_DIN	GPIO 47 data input value
14	**GPIO46**	GPIO46_DIN	GPIO 46 data input value
13	**GPIO45**	GPIO45_DIN	GPIO 45 data input value

Bit(s)	Mnemonic	Name	Description
12	**GPIO44**	GPIO44_DIN	GPIO 44 data input value
11	**GPIO43**	GPIO43_DIN	GPIO 43 data input value
10	**GPIO42**	GPIO42_DIN	GPIO 42 data input value
9	**GPIO41**	GPIO41_DIN	GPIO 41 data input value
8	**GPIO40**	GPIO40_DIN	GPIO 40 data input value
7	**GPIO39**	GPIO39_DIN	GPIO 39 data input value
6	**GPIO38**	GPIO38_DIN	GPIO 38 data input value
5	**GPIO37**	GPIO37_DIN	GPIO 37 data input value
4	**GPIO36**	GPIO36_DIN	GPIO 36 data input value
3	**GPIO35**	GPIO35_DIN	GPIO 35 data input value
2	**GPIO34**	GPIO34_DIN	GPIO 34 data input value
1	**GPIO33**	GPIO33_DIN	GPIO 33 data input value
0	**GPIO32**	GPIO32_DIN	GPIO 32 data input value

0xA002_0430 GPIO Data Input Register 3 GPIO_DIN3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO63	GPIO62	GPIO61	GPIO60	GPIO59	GPIO58	GPIO57	GPIO56	GPIO55	GPIO54	GPIO53	GPIO52	GPIO51	GPIO50	GPIO49	GPIO48
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit(s)	Mnemonic	Name	Description
15	**GPIO63**	GPIO63_DIN	GPIO 63 data input value
14	**GPIO62**	GPIO62_DIN	GPIO 62 data input value
13	**GPIO61**	GPIO61_DIN	GPIO 61 data input value
12	**GPIO60**	GPIO60_DIN	GPIO 60 data input value
11	**GPIO59**	GPIO59_DIN	GPIO 59 data input value
10	**GPIO58**	GPIO58_DIN	GPIO 58 data input value
9	**GPIO57**	GPIO57_DIN	GPIO 57 data input value
8	**GPIO56**	GPIO56_DIN	GPIO 56 data input value
7	**GPIO55**	GPIO55_DIN	GPIO 55 data input value
6	**GPIO54**	GPIO54_DIN	GPIO 54 data input value
5	**GPIO53**	GPIO53_DIN	GPIO 53 data input value
4	**GPIO52**	GPIO52_DIN	GPIO 52 data input value
3	**GPIO51**	GPIO51_DIN	GPIO 51 data input value
2	**GPIO50**	GPIO50_DIN	GPIO 50 data input value
1	**GPIO49**	GPIO49_DIN	GPIO 49 data input value
0	**GPIO48**	GPIO48_DIN	GPIO 48 data input value

0xA002_0440 GPIO Data Input Register 4 GPIO_DIN4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name								GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type								RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Bit(s)	Mnemonic	Name	Description
8	**GPIO72**	GPIO72_DIN	GPIO 72 data input value
7	**GPIO71**	GPIO71_DIN	GPIO 71 data input value
6	**GPIO70**	GPIO70_DIN	GPIO 70 data input value
5	**GPIO69**	GPIO69_DIN	GPIO 69 data input value
4	**GPIO68**	GPIO68_DIN	GPIO 68 data input value
3	**GPIO67**	GPIO67_DIN	GPIO 67 data input value
2	**GPIO66**	GPIO66_DIN	GPIO 66 data input value
1	**GPIO65**	GPIO65_DIN	GPIO 65 data input value
0	**GPIO64**	GPIO64_DIN	GPIO 64 data input value

0xA002_0500 GPIO Pull-up/down Select 0

GPIO_PULLSEL0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO15**	GPIO15_PULLSEL	**GPIO 15 pull-up/down selection** 0: Pull-down 1: Pull-up
14	**GPIO14**	GPIO14_PULLSEL	**GPIO 14 pull-up/down selection** 0: Pull-down 1: Pull-up
13	**GPIO13**	GPIO13_PULLSEL	**GPIO 13 pull-up/down selection** 0: Pull-down 1: Pull-up
12	**GPIO12**	GPIO12_PULLSEL	**GPIO 12 pull-up/down selection** 0: Pull-down 1: Pull-up
11	**GPIO11**	GPIO11_PULLSEL	**GPIO 11 pull-up/down selection** 0: Pull-down 1: Pull-up
10	**GPIO10**	GPIO10_PULLSEL	**GPIO 10 pull-up/down selection** 0: Pull-down 1: Pull-up
9	**GPIO9**	GPIO9_PULLSEL	**GPIO 9 pull-up/down selection** 0: Pull-down 1: Pull-up
8	**GPIO8**	GPIO8_PULLSEL	**GPIO 8 pull-up/down selection** 0: Pull-down

Bit(s)	Mnemonic	Name	Description
			1: Pull-up
7	**GPIO7**	GPIO7_PULLSEL	**GPIO 7 pull-up/down selection** 0: Pull-down 1: Pull-up
6	**GPIO6**	GPIO6_PULLSEL	**GPIO 6 pull-up/down selection** 0: Pull-down 1: Pull-up
5	**GPIO5**	GPIO5_PULLSEL	**GPIO 5 pull-up/down selection** 0: Pull-down 1: Pull-up
4	**GPIO4**	GPIO4_PULLSEL	**GPIO 4 pull-up/down selection** 0: Pull-down 1: Pull-up
3	**GPIO3**	GPIO3_PULLSEL	**GPIO 3 pull-up/down selection** 0: Pull-down 1: Pull-up
2	**GPIO2**	GPIO2_PULLSEL	**GPIO 2 pull-up/down selection** 0: Pull-down 1: Pull-up
1	**GPIO1**	GPIO1_PULLSEL	**GPIO 1 pull-up/down selection** 0: Pull-down 1: Pull-up
0	**GPIO0**	GPIO0_PULLSEL	**GPIO 0 pull-up/down selection** 0: Pull-down 1: Pull-up

0xA002_0510 GPIO Pull-up/down Select 1

GPIO_PULLSEL1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO31**	**GPIO30**	**GPIO29**	**GPIO28**	**GPIO27**	**GPIO26**	**GPIO25**	**GPIO24**	**GPIO23**	**GPIO22**	**GPIO21**	**PGIO20**	**GPIO19**	**GPIO18**	**GPIO17**	**GPIO16**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO31**	GPIO31_PULLSEL	**GPIO 31 pull-up/down selection** 0: Pull-down 1: Pull-up
14	**GPIO30**	GPIO30_PULLSEL	**GPIO 30 pull-up/down selection** 0: Pull-down 1: Pull-up
13	**GPIO29**	GPIO29_PULLSEL	**GPIO 29 pull-up/down selection** 0: Pull-down 1: Pull-up
12	**GPIO28**	GPIO28_PULLSEL	**GPIO 28 pull-up/down selection** 0: Pull-down 1: Pull-up
11	**GPIO27**	GPIO27_PULLSEL	**GPIO 27 pull-up/down selection** 0: Pull-down 1: Pull-up

Bit(s)	Mnemonic	Name	Description
10	**GPIO26**	GPIO26_PULLSEL	**GPIO 26 pull-up/down selection** 0: Pull-down 1: Pull-up
9	**GPIO25**	GPIO25_PULLSEL	**GPIO 25 pull-up/down selection** 0: Pull-down 1: Pull-up
8	**GPIO24**	GPIO24_PULLSEL	**GPIO 24 pull-up/down selection** 0: Pull-down 1: Pull-up
7	**GPIO23**	GPIO23_PULLSEL	**GPIO 23 pull-up/down selection** 0: Pull-down 1: Pull-up
6	**GPIO22**	GPIO22_PULLSEL	**GPIO 22 pull-up/down selection** 0: Pull-down 1: Pull-up
5	**GPIO21**	GPIO21_PULLSEL	**GPIO 21 pull-up/down selection** 0: Pull-down 1: Pull-up
4	**GPIO20**	GPIO20_PULLSEL	**GPIO 20 pull-up/down selection** 0: Pull-down 1: Pull-up
3	**GPIO19**	GPIO19_PULLSEL	**GPIO 19 pull-up/down selection** 0: Pull-down 1: Pull-up
2	**GPIO18**	GPIO18_PULLSEL	**GPIO 18 pull-up/down selection** 0: Pull-down 1: Pull-up
1	**GPIO17**	GPIO17_PULLSEL	**GPIO 17 pull-up/down selection** 0: Pull-down 1: Pull-up
0	**GPIO16**	GPIO16_PULLSEL	**GPIO 16 pull-up/down selection** 0: Pull-down 1: Pull-up

0xA002_0520 GPIO Pull-up/down Select 2

GPIO_PULLSEL2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO47**	**GPIO46**	**GPIO45**	**GPIO44**	**GPIO43**	**GPIO42**	**GPIO41**	**GPIO40**	**GPIO39**	**GPIO38**	**GPIO37**	**GPIO36**	**GPIO35**	**GPIO34**	**GPIO33**	**GPIO32**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO47**	GPIO47_PULLSEL	**GPIO 47 pull-up/down selection** 0: Pull-down 1: Pull-up
14	**GPIO46**	GPIO46_PULLSEL	**GPIO 46 pull-up/down selection** 0: Pull-down 1: Pull-up

Bit(s)	Mnemonic	Name	Description
13	**GPIO45**	GPIO45_PULLSEL	**GPIO 45 pull-up/down selection** 0: Pull-down 1: Pull-up
12	**GPIO44**	GPIO44_PULLSEL	**GPIO 44 pull-up/down selection** 0: Pull-down 1: Pull-up
11	**GPIO43**	GPIO43_PULLSEL	**GPIO 43 pull-up/down selection** 0: Pull-down 1: Pull-up
10	**GPIO42**	GPIO42_PULLSEL	**GPIO 42 pull-up/down selection** 0: Pull-down 1: Pull-up
9	**GPIO41**	GPIO41_PULLSEL	**GPIO 41 pull-up/down selection** 0: Pull-down 1: Pull-up
8	**GPIO40**	GPIO40_PULLSEL	**GPIO 40 pull-up/down selection** 0: Pull-down 1: Pull-up
7	**GPIO39**	GPIO39_PULLSEL	**GPIO 39 pull-up/down selection** 0: Pull-down 1: Pull-up
6	**GPIO38**	GPIO38_PULLSEL	**GPIO 38 pull-up/down selection** 0: Pull-down 1: Pull-up
5	**GPIO37**	GPIO37_PULLSEL	**GPIO 37 pull-up/down selection** 0: Pull-down 1: Pull-up
4	**GPIO36**	GPIO36_PULLSEL	**GPIO 36 pull-up/down selection** 0: Pull-down 1: Pull-up
3	**GPIO35**	GPIO35_PULLSEL	**GPIO 35 pull-up/down selection** 0: Pull-down 1: Pull-up
2	**GPIO34**	GPIO34_PULLSEL	**GPIO 34 pull-up/down selection** 0: Pull-down 1: Pull-up
1	**GPIO33**	GPIO33_PULLSEL	**GPIO 33 pull-up/down selection** 0: Pull-down 1: Pull-up
0	**GPIO32**	GPIO32_PULLSEL	**GPIO 32 pull-up/down selection** 0: Pull-down 1: Pull-up

0xA002_0530 GPIO Pull-up/down Select 3

GPIO_PULLSEL3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO63**	**GPIO62**	**GPIO61**	**GPIO60**	**GPIO59**	**GPIO58**	**GPIO57**	**GPIO56**	**GPIO55**	**GPIO54**	**GPIO53**	**GPIO52**	**GPIO51**	**GPIO50**	**GPIO49**	**GPIO48**
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO63**	GPIO63_PULLSEL	**GPIO 63 pull-up/down selection** 0: Pull-down 1: Pull-up
14	**GPIO62**	GPIO62_PULLSEL	**GPIO 62 pull-up/down selection** 0: Pull-down 1: Pull-up
13	**GPIO61**	GPIO61_PULLSEL	**GPIO 61 pull-up/down selection** 0: Pull-down 1: Pull-up
12	**GPIO60**	GPIO60_PULLSEL	**GPIO 60 pull-up/down selection** 0: Pull-down 1: Pull-up
11	**GPIO59**	GPIO59_PULLSEL	**GPIO 59 pull-up/down selection** 0: Pull-down 1: Pull-up
10	**GPIO58**	GPIO58_PULLSEL	**GPIO 58 pull-up/down selection** 0: Pull-down 1: Pull-up
9	**GPIO57**	GPIO57_PULLSEL	**GPIO 57 pull-up/down selection** 0: Pull-down 1: Pull-up
8	**GPIO56**	GPIO56_PULLSEL	**GPIO 56 pull-up/down selection** 0: Pull-down 1: Pull-up
7	**GPIO55**	GPIO55_PULLSEL	**GPIO 55 pull-up/down selection** 0: Pull-down 1: Pull-up
6	**GPIO54**	GPIO54_PULLSEL	**GPIO 54 pull-up/down selection** 0: Pull-down 1: Pull-up
5	**GPIO53**	GPIO53_PULLSEL	**GPIO 53 pull-up/down selection** 0: Pull-down 1: Pull-up
4	**GPIO52**	GPIO52_PULLSEL	**GPIO 52 pull-up/down selection** 0: Pull-down 1: Pull-up
3	**GPIO51**	GPIO51_PULLSEL	**GPIO 51 pull-up/down selection** 0: Pull-down 1: Pull-up
2	**GPIO50**	GPIO50_PULLSEL	**GPIO 50 pull-up/down selection** 0: Pull-down 1: Pull-up
1	**GPIO49**	GPIO49_PULLSEL	**GPIO 49 pull-up/down selection** 0: Pull-down 1: Pull-up
0	**GPIO48**	GPIO48_PULLSEL	**GPIO 48 pull-up/down selection** 0: Pull-down 1: Pull-up

0xA002_0540 GPIO Pull-up/down Select 4

GPIO_PULLSEL4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								GPIO72	GPIO71	GPIO70	GPIO69	GPIO68	GPIO67	GPIO66	GPIO65	GPIO64
Type								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit(s)	Mnemonic	Name	Description
8	**GPIO72**	GPIO72_PULLSEL	**GPIO 72 pull-up/down selection** 0: Pull-down 1: Pull-up
7	**GPIO71**	GPIO71_PULLSEL	**GPIO 71 pull-up/down selection** 0: Pull-down 1: Pull-up
6	**GPIO70**	GPIO70_PULLSEL	**GPIO 70 pull-up/down selection** 0: Pull-down 1: Pull-up
5	**GPIO69**	GPIO69_PULLSEL	**GPIO 69 pull-up/down selection** 0: Pull-down 1: Pull-up
4	**GPIO68**	GPIO68_PULLSEL	**GPIO 68 pull-up/down selection** 0: Pull-down 1: Pull-up
3	**GPIO67**	GPIO67_PULLSEL	**GPIO 67 pull-up/down selection** 0: Pull-down 1: Pull-up
2	**GPIO66**	GPIO66_PULLSEL	**GPIO 66 pull-up/down selection** 0: Pull-down 1: Pull-up
1	**GPIO65**	GPIO65_PULLSEL	**GPIO 65 pull-up/down selection** 0: Pull-down 1: Pull-up
0	**GPIO64**	GPIO64_PULLSEL	**GPIO 64 pull-up/down selection** 0: Pull-down 1: Pull-up

0xA002_0610 GPIO Schmit Trigger Control 1

GPIO_SMT1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_SMT[15:11]															
Type	R/W	R/W	R/W	R/W	R/W											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**GPIO_SMT**	Reserved	Reserved
14	**GPIO_SMT**	Reserved	Reserved
13	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 26 Schmitt trigger** 0: Disable

Bit(s)	Mnemonic	Name	Description
			1: Enable
12	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 27 Schmitt trigger** 0: Disable 1: Enable
11	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 25 Schmitt trigger** 0: Disable 1: Enable

0xA002_0620 GPIO Schmit Trigger Control 2 GPIO_SMT2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_SMT[47:32]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
47	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 38 Schmitt trigger** 0: Disable 1: Enable
46	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 62/63/64/65 Schmitt trigger** 0: Disable 1: Enable
45	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 22/23 Schmitt trigger** 0: Disable 1: Enable
44	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 20/21 Schmitt trigger** 0: Disable 1: Enable
43	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 48/49/50/51/52/53/54/55/56/57/58 Schmitt trigger** 0: Disable 1: Enable
42	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 28/29/30/31/32/33/34/35/36/37/39/40/42/43/46 Schmitt trigger** 0: Disable 1: Enable
41	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 1/2 Schmitt trigger** 0: Disable 1: Enable
40	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 5/6/7/8/9/11/12/13/14/15 Schmitt trigger** 0: Disable 1: Enable
39	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 61 Schmitt trigger** 0: Disable 1: Enable
38	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 0/16/17/19 Schmitt trigger** 0: Disable 1: Enable
37	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO66/67/68/69/70/71/72 Schmitt trigger** 0: Disable 1: Enable
36	**GPIO_SMT**	Reserved	Reserved

Bit(s)	Mnemonic	Name	Description
35	**GPIO_SMT**	Reserved	Reserved
34	**GPIO_SMT**	Reserved	Reserved
33	**GPIO_SMT**	Reserved	Reserved
32	**GPIO_SMT**	Reserved	Reserved

0xA002_0630 GPIO Schmit Trigger Control 3 GPIO_SMT3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_SMT[63:48]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
63	**GPIO_SMT**	Reserved	Reserved
62	**GPIO_SMT**	Reserved	Reserved
61	**GPIO_SMT**	Reserved	Reserved
60	**GPIO_SMT**	Reserved	Reserved
59	**GPIO_SMT**	Reserved	Reserved
58	**GPIO_SMT**	Reserved	Reserved
57	**GPIO_SMT**	Reserved	Reserved
56	**GPIO_SMT**	Reserved	Reserved
55	**GPIO_SMT**	Reserved	Reserved
54	**GPIO_SMT**	Reserved	Reserved
53	**GPIO_SMT**	Reserved	Reserved
52	**GPIO_SMT**	Reserved	Reserved
51	**GPIO_SMT**	Reserved	Reserved
50	**GPIO_SMT**	Reserved	Reserved
49	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 24 Schmitt trigger** 0: Disable 1: Enable
48	**GPIO_SMT**	GPIO_SMTEN	**Enables GPIO 18 Schmitt trigger** 0: Disable 1: Enable

0xA002_0710 GPIO Slew Rate Control 1 GPIO_SR1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_SR[15:11]															
Type	R/W	R/W	R/W	R/W	R/W											
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15	**GPIO_SR**	Reserved	Reserved

Bit(s)	Mnemonic	Name	Description
14	**GPIO_SR**	Reserved	Reserved
13	**GPIO_SR**	GPIO_SREN	**Enables GPIO 26 slew rate** 0: Disable 1: Enable
12	**GPIO_SR**	GPIO_SREN	**Enables GPIO 27 slew rate** 0: Disable 1: Enable
11	**GPIO_SR**	GPIO_SREN	**Enables GPIO 25 slew rate** 0: Disable 1: Enable

0xA002_0720 GPIO Slew Rate Control 2 GPIO_SR2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_SR[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
47	**GPIO_SR**	GPIO_SREN	**Enables GPIO 38 slew rate** 0: Disable 1: Enable
46	**GPIO_SR**	GPIO_SREN	**Enables GPIO 62/63/64/65 slew rate** 0: Disable 1: Enable
45	**GPIO_SR**	GPIO_SREN	**Enables GPIO 22/23 slew rate** 0: Disable 1: Enable
44	**GPIO_SR**	GPIO_SREN	**Enables GPIO 20/21 slew rate** 0: Disable 1: Enable
43	**GPIO_SR**	GPIO_SREN	**Enables GPIO 48/49/50/51/52/53/54/55/56/57/58/59/60 slew rate** 0: Disable 1: Enable
42	**GPIO_SR**	GPIO_SREN	**Enables GPIO 28/29/30/31/32/33/34/35/36/37/39/40/42/43/46 slew rate** 0: Disable 1: Enable
41	**GPIO_SR**	GPIO_SREN	**Enables GPIO 1/2 slew rate** 0: Disable 1: Enable
40	**GPIO_SR**	GPIO_SREN	**Enables GPIO 5/6/7/8/9/11/12/13/14/15 slew rate** 0: Disable 1: Enable
39	**GPIO_SR**	GPIO_SREN	**Enables GPIO 61 slew rate** 0: Disable 1: Enable
38	**GPIO_SR**	GPIO_SREN	**Enables GPIO 0/16/17/19 slew rate** 0: Disable 1: Enable
37	**GPIO_SR**	GPIO_SREN	**Enables GPIO 66/67/68/69/70/71/72 slew rate**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
36	**GPIO_SR**	Reserved	Reserved
35	**GPIO_SR**	Reserved	Reserved
34	**GPIO_SR**	Reserved	Reserved
33	**GPIO_SR**	Reserved	Reserved
32	**GPIO_SR**	Reserved	Reserved

0xA002_0730 GPIO Slew Rate Control 3 GPIO_SR3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_SR[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
63	**GPIO_SR**	Reserved	Reserved
62	**GPIO_SR**	Reserved	Reserved
61	**GPIO_SR**	Reserved	Reserved
60	**GPIO_SR**	Reserved	Reserved
59	**GPIO_SR**	Reserved	Reserved
58	**GPIO_SR**	Reserved	Reserved
57	**GPIO_SR**	Reserved	Reserved
56	**GPIO_SR**	Reserved	Reserved
55	**GPIO_SR**	Reserved	Reserved
54	**GPIO_SR**	Reserved	Reserved
53	**GPIO_SR**	Reserved	Reserved
52	**GPIO_SR**	Reserved	Reserved
51	**GPIO_SR**	Reserved	Reserved
50	**GPIO_SR**	Reserved	Reserved
49	**GPIO_SR**	GPIO_SREN	**Enables GPIO 24 slew rate** 0: Disable 1: Enable
48	**GPIO_SR**	GPIO_SREN	**Enables GPIO 18 slew rate** 0: Disable 1: Enable

GPIO*n* Output slew rate control. High asserted. SR = 1, slower slew. SR = 0, no slew rate controlled.

0 Disable

1 Enable

0xA002_0800 GPIO Driving Control 0 GPIO_DRV0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_DRV[15:6]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						

Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
14:13	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list: GPIO16/19/28/29/30/31/33/34/42/43/45/46 0: 4mA 1: 8mA 2: 12mA 3: 16mA
12:11	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list: GPIO0/2/16/18/61 0: 4mA 1: 8mA 2: 12mA 3: 16mA
10:8	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list:GPIO35/36/37/38/39/40/41/66/67/68/69/70/71/72 0: 2mA 1: 4mA 2: 6mA 3: 8mA 4: 10mA 5: 12mA 6: 14mA 7: 16mA
7:6	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list: GPIO3/4/10/25/26/27 0: 4mA 1: 8mA 2: 12mA 3: 16mA

0xA002_0810 GPIO Driving Control 1 GPIO_DRV1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_DRV[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO_DRV**	IO_CFG_DRV	Reserved
27:25	**GPIO_DRV**	IO_CFG_DRV	Reserved
24:22	**GPIO_DRV**	IO_CFG_DRV	Reserved
21:20	**GPIO_DRV**	IO_CFG_DRV	Reserved
18:17	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list: GPIO44 0: 2mA 1: 4mA 2: 6mA 3: 8mA
16:15	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration**

Bit(s)	Mnemonic	Name	Description
			For the IO list: GPIO5/6/7/8/9/11/12/13/14/15 0: 4mA 1: 8mA 2: 12mA 3: 16mA

0xA002_0820 GPIO Driving Control 2 GPIO_DRV2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_DRV[47:32]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
48:46	**GPIO_DRV**	Reserved	Reserved
45:43	**GPIO_DRV**	IO_CFG_DRV	Reserved
38:37	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list: GPIO62/63/64/65 0: 4mA 1: 8mA 2: 12mA 3: 16mA
36:35	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list: GPIO20/21/22/23 0: 4mA 1: 8mA 2: 12mA 3: 16mA
34:33	**GPIO_DRV**	IO_CFG_DRV	**IO driving strength configuration** For the IO list: GPIO47/48/49/50/51/52/53/54/55/56/57/58/59/60 0: 4mA 1: 8mA 2: 12mA 3: 16mA
32	**GPIO_DRV**	IO_CFG_DRV	Reserved

0xA002_0910 GPIO Rx Input Buffer Control 1 GPIO_IES1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_IES[15:11]															
Type	R/W	R/W	R/W	R/W	R/W											
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15	**GPIO_IES**	Reserved	Reserved
14	**GPIO_IES**	Reserved	Reserved
13	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 26 Rx input buffer**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
12	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 27 Rx input buffer** 0: Disable 1: Enable
11	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 25 Rx input buffer** 0: Disable 1: Enable

0xA002_0920 GPIO Rx Input Buffer Control 2 GPIO_IES2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_IES[47:32]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
47	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 38 Rx input buffer** 0: Disable 1: Enable
46	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 62/63/64/65 Rx input buffer** 0: Disable 1: Enable
45	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 22/23 Rx input buffer** 0: Disable 1: Enable
44	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 20/21 Rx input buffer** 0: Disable 1: Enable
43	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 48/49/50/51/52/53/54/55/56/57/58/59/60 Rx input buffer** 0: Disable 1: Enable
42	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 28/29/30/31/32/33/34/35/36/37/39/40/42/43/46 Rx input buffer** 0: Disable 1: Enable
41	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 1/2 Rx input buffer** 0: Disable 1: Enable
40	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 5/6/7/8/9/11/12/13/14/15 Rx input buffer** 0: Disable 1: Enable
39	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 61 Rx input buffer** 0: Disable 1: Enable
38	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 0/16/17/19 Rx input buffer** 0: Disable 1: Enable
37	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 66/67/68/69/70/71/72 Rx input buffer** 0: Disable

Bit(s)	Mnemonic	Name	Description
			1: Enable
36	**GPIO_IES**	Reserved	Reserved
35	**GPIO_IES**	Reserved	Reserved
34	**GPIO_IES**	Reserved	Reserved
33	**GPIO_IES**	Reserved	Reserved
32	**GPIO_IES**	Reserved	Reserved

0xA002_0930 GPIO Rx Input Buffer Control 3 GPIO_IES3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_IES[63:48]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
63	**GPIO_IES**	Reserved	Reserved
62	**GPIO_IES**	Reserved	Reserved
61	**GPIO_IES**	Reserved	Reserved
60	**GPIO_IES**	Reserved	Reserved
59	**GPIO_IES**	Reserved	Reserved
58	**GPIO_IES**	Reserved	Reserved
57	**GPIO_IES**	Reserved	Reserved
56	**GPIO_IES**	Reserved	Reserved
55	**GPIO_IES**	Reserved	Reserved
54	**GPIO_IES**	Reserved	Reserved
53	**GPIO_IES**	Reserved	Reserved
52	**GPIO_IES**	Reserved	Reserved
51	**GPIO_IES**	Reserved	Reserved
50	**GPIO_IES**	Reserved	Reserved
49	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 24 Rx input buffer** 0: Disable 1: Enable
48	**GPIO_IES**	GPIO_IESEN	**Enables GPIO 18 Rx input buffer** 0: Disable 1: Enable

GPIO*n* Enables GPIO Rx input buffer. High asserted. Data path: From IO to O. IES = 0, O = 0

0 Disable

1 Enable

0xA002_0a00 GPIO PUPD Enable Control Registers 0 GPIO_PUPDEN0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	GPIO_PUPD[15:6]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						
Reset	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**PUPD**	GPIO6_PUPDEN	**Enables GPIO 6 PUPD** 0: Disable 1: Enable
14	**PUPD**	GPIO7_PUPDEN	**Enables GPIO 7 PUPD** 0: Disable 1: Enable
13	**PUPD**	GPIO15_PUPDEN	**Enables GPIO 15 PUPD** 0: Disable 1: Enable
12	**PUPD**	GPIO5_PUPDEN	**Enables GPIO 5 PUPD** 0: Disable 1: Enable
11	**PUPD**	GPIO11_PUPDEN	**Enables GPIO 11 PUPD** 0: Disable 1: Enable
10	**PUPD**	GPIO12_PUPDEN	**Enables GPIO 12 PUPD** 0: Disable 1: Enable
9	**PUPD**	GPIO9_PUPDEN	**Enables GPIO 9 PUPD** 0: Disable 1: Enable
8	**PUPD**	GPIO26_PUPDEN	**Enables GPIO 26 PUPD** 0: Disable 1: Enable
7	**PUPD**	GPIO27_PUPDEN	**Enables GPIO 27 PUPD** 0: Disable 1: Enable
6	**PUPD**	GPIO25_PUPDEN	**Enables GPIO 25 PUPD** 0: Disable 1: Enable

0xA002_0a10 GPIO PUPD Enable Control Register 1 GPIO_PUPDEN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_PUPD[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0

Bit(s)	Mnemonic	Name	Description
31	**PUPD**	GPIO67_PUPDEN	**Enables GPIO 67 PUPD** 0: Disable 1: Enable
30	**PUPD**	GPIO72_PUPDEN	**Enables GPIO 72 PUPD** 0: Disable 1: Enable

Bit(s)	Mnemonic	Name	Description
29	**PUPD**	GPIO66_PUPDEN	**Enables GPIO 66 PUPD** 0: Disable 1: Enable
28	**PUPD**	GPIO36_PUPDEN	**Enables GPIO 36 PUPD** 0: Disable 1: Enable
27	**PUPD**	GPIO35_PUPDEN	**Enables GPIO 35 PUPD** 0: Disable 1: Enable
26	**PUPD**	GPIO38_PUPDEN	**Enables GPIO 38 PUPD** 0: Disable 1: Enable
25	**PUPD**	GPIO39_PUPDEN	**Enables GPIO 39 PUPD** 0: Disable 1: Enable
24	**PUPD**	GPIO37_PUPDEN	**Enables GPIO 37 PUPD** 0: Disable 1: Enable
23	**PUPD**	GPIO40_PUPDEN	**Enables GPIO 40 PUPD** 0: Disable 1: Enable
22	**PUPD**	GPIO22_PUPDEN	**Enables GPIO 22 PUPD** 0: Disable 1: Enable
21	**PUPD**	GPIO8_PUPDEN	**Enables GPIO 8 PUPD** 0: Disable 1: Enable
20	**PUPD**	GPIO14_PUPDEN	**Enables GPIO 14 PUPD** 0: Disable 1: Enable
19	**PUPD**	GPIO10_PUPDEN	**Enables GPIO 10 PUPD** 0: Disable 1: Enable
18	**PUPD**	GPIO13_PUPDEN	**Enables GPIO 13 PUPD** 0: Disable 1: Enable
17	**PUPD**	GPIO4_PUPDEN	**Enables GPIO 4 PUPD** 0: Disable 1: Enable
16	**PUPD**	GPIO3_PUPDEN	**Enables GPIO 3 PUPD** 0: Disable 1: Enable

0xA002_0a20 GPIO PUPD Enable Control Register 2 GPIO_PUPDEN2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_PUPD[47:32]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit(s)	Mnemonic	Name	Description
47	**PUPD**	Reserved	Reserved
45	**PUPD**	Reserved	Reserved
45	**PUPD**	Reserved	Reserved
44	**PUPD**	Reserved	Reserved
43	**PUPD**	Reserved	Reserved
42	**PUPD**	Reserved	Reserved
41	**PUPD**	GPIO55_PUPDEN	**Enables GPIO 55 PUPD** 0: Disable 1: Enable
40	**PUPD**	GPIO54_PUPDEN	**Enables GPIO 54 PUPD** 0: Disable 1: Enable
39	**PUPD**	GPIO53_PUPDEN	**Enables GPIO 53 PUPD** 0: Disable 1: Enable
38	**PUPD**	GPIO51_PUPDEN	**Enables GPIO 51 PUPD** 0: Disable 1: Enable
37	**PUPD**	GPIO52_PUPDEN	**Enables GPIO 52 PUPD** 0: Disable 1: Enable
36	**PUPD**	GPIO50_PUPDEN	**Enables GPIO 50 PUPD** 0: Disable 1: Enable
35	**PUPD**	GPIO68_PUPDEN	**Enables GPIO 68 PUPD** 0: Disable 1: Enable
34	**PUPD**	GPIO70_PUPDEN	**Enables GPIO 70 PUPD** 0: Disable 1: Enable
33	**PUPD**	GPIO69_PUPDEN	**Enables GPIO 69 PUPD** 0: Disable 1: Enable
32	**PUPD**	GPIO71_PUPDEN	**Enables GPIO 71 PUPD** 0: Disable 1: Enable

0xA002_0b00 GPIO Resistance Enable Control Register 0 GPIO_RESEN0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_R0[15:6]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						
Reset	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**R0**	GPIO6_R0EN	**Enables GPIO 6 R0** 0: Disable 1: Enable
14	**R0**	GPIO7_R0EN	**Enables GPIO 7 R0** 0: Disable

Bit(s)	Mnemonic	Name	Description
			1: Enable
13	**R0**	GPIO15_R0EN	**Enables GPIO 15 R0** 0: Disable 1: Enable
12	**R0**	GPIO5_R0EN	**Enables GPIO 5 R0** 0: Disable 1: Enable
11	**R0**	GPIO11_R0EN	**Enables GPIO 11 R0** 0: Disable 1: Enable
10	**R0**	GPIO12_R0EN	**Enables GPIO 12 R0** 0: Disable 1: Enable
9	**R0**	GPIO9_R0EN	**Enables GPIO 9 R0** 0: Disable 1: Enable
8	**R0**	GPIO26_R0EN	**Enables GPIO 26 R0** 0: Disable 1: Enable
7	**R0**	GPIO27_R0EN	**Enables GPIO 27 R0** 0: Disable 1: Enable
6	**R0**	GPIO25_R0EN	**Enables GPIO 25 R0** 0: Disable 1: Enable

0xA002_0b10 GPIO Resistance Enable Control Register 1 GPIO_RESEN1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_R0[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0

Bit(s)	Mnemonic	Name	Description
31	**R0**	GPIO67_R0EN	**Enables GPIO 67 R0** 0: Disable 1: Enable
30	**R0**	GPIO72_R0EN	**Enables GPIO 72 R0** 0: Disable 1: Enable
29	**R0**	GPIO66_R0EN	**Enables GPIO 66 R0** 0: Disable 1: Enable
28	**R0**	GPIO36_R0EN	**Enables GPIO 36 R0** 0: Disable 1: Enable
27	**R0**	GPIO35_R0EN	**Enables GPIO 35 R0** 0: Disable 1: Enable
26	**R0**	GPIO38_R0EN	**Enables GPIO 38 R0**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
25	**R0**	GPIO39_R0EN	**Enables GPIO 39 R0** 0: Disable 1: Enable
24	**R0**	GPIO37_R0EN	**Enables GPIO 37 R0** 0: Disable 1: Enable
23	**R0**	GPIO40_R0EN	**Enables GPIO 40 R0** 0: Disable 1: Enable
22	**R0**	GPIO22_R0EN	**Enables GPIO 22 R0** 0: Disable 1: Enable
21	**R0**	GPIO8_R0EN	**Enables GPIO 8 R0** 0: Disable 1: Enable
20	**R0**	GPIO14_R0EN	**Enables GPIO 14 R0** 0: Disable 1: Enable
19	**R0**	GPIO10_R0EN	**Enables GPIO 10 R0** 0: Disable 1: Enable
18	**R0**	GPIO13_R0EN	**Enables GPIO 13 R0** 0: Disable 1: Enable
17	**R0**	GPIO4_R0EN	**Enables GPIO 4 R0** 0: Disable 1: Enable
16	**R0**	GPIO3_R0EN	**Enables GPIO 3 R0** 0: Disable 1: Enable

0xA002_0b20 GPIO Resistance Enable Control Register 2 GPIO_RESEN2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_R0[47:32]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit(s)	Mnemonic	Name	Description
47	**R1**	Reserved	Reserved
46	**R1**	Reserved	Reserved
45	**R0**	Reserved	Reserved
44	**R0**	Reserved	Reserved
43	**R0**	Reserved	Reserved
42	**R0**	Reserved	Reserved
41	**R0**	GPIO55_R0EN	**Enables GPIO 55 R0** 0: Disable 1: Enable

Bit(s)	Mnemonic	Name	Description
40	**R0**	GPIO54_R0EN	**Enables GPIO 54 R0** 0: Disable 1: Enable
39	**R0**	GPIO53_R0EN	**Enables GPIO 53 R0** 0: Disable 1: Enable
38	**R0**	GPIO51_R0EN	**Enables GPIO 51 R0** 0: Disable 1: Enable
37	**R0**	GPIO52_R0EN	**Enables GPIO 52 R0** 0: Disable 1: Enable
36	**R0**	GPIO50_R0EN	**Enables GPIO 50 R0** 0: Disable 1: Enable
35	**R0**	GPIO68_R0EN	**Enables GPIO 68 R0** 0: Disable 1: Enable
34	**R0**	GPIO70_R0EN	**Enables GPIO 70 R0** 0: Disable 1: Enable
33	**R0**	GPIO69_R0EN	**Enables GPIO 69 R0** 0: Disable 1: Enable
32	**R0**	GPIO71_R0EN	**Enables GPIO 71 R0** 0: Disable 1: Enable

0xA002_0b30 GPIO Resistance Enable Control Register 3 GPIO_RESEN3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_R1[15:6]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**R1**	GPIO6_R1EN	**Enables GPIO 6 R1** 0: Disable 1: Enable
14	**R1**	GPIO7_R1EN	**Enables GPIO 7 R1** 0: Disable 1: Enable
13	**R1**	GPIO15_R1EN	**Enables GPIO 15 R1** 0: Disable 1: Enable
12	**R1**	GPIO5_R1EN	**Enables GPIO 5 R1** 0: Disable 1: Enable
11	**R1**	GPIO11_R1EN	**Enables GPIO 11 R1** 0: Disable

Bit(s)	Mnemonic	Name	Description
			1: Enable
10	**R1**	GPIO12_R1EN	**Enables GPIO 12 R1** 0: Disable 1: Enable
9	**R1**	GPIO9_R1EN	**Enables GPIO 9 R1** 0: Disable 1: Enable
8	**R1**	GPIO26_R1EN	**Enables GPIO 26 R1** 0: Disable 1: Enable
7	**R1**	GPIO27_R1EN	**Enables GPIO 27 R1** 0: Disable 1: Enable
6	**R1**	GPIO25_R1EN	**Enables GPIO 25 R1** 0: Disable 1: Enable

0xA002_0b40 GPIO Resistance Enable Control Register 4 GPIO_RESEN4

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_R1[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31	**R1**	GPIO67_R1EN	**Enables GPIO 67 R1** 0: Disable 1: Enable
30	**R1**	GPIO72_R1EN	**Enables GPIO 72 R1** 0: Disable 1: Enable
29	**R1**	GPIO66_R1EN	**Enables GPIO 66 R1** 0: Disable 1: Enable
28	**R1**	GPIO36_R1EN	**Enables GPIO 36 R1** 0: Disable 1: Enable
27	**R1**	GPIO35_R1EN	**Enables GPIO 35 R1** 0: Disable 1: Enable
26	**R1**	GPIO38_R1EN	**Enables GPIO 38 R1** 0: Disable 1: Enable
25	**R1**	GPIO39_R1EN	**Enables GPIO 39 R1** 0: Disable 1: Enable
24	**R1**	GPIO37_R1EN	**Enables GPIO 37 R1** 0: Disable 1: Enable
23	**R1**	GPIO40_R1EN	**Enables GPIO 40 R1**

Bit(s)	Mnemonic	Name	Description
			0: Disable 1: Enable
22	**R1**	GPIO22_R1EN	**Enables GPIO 22 R1** 0: Disable 1: Enable
21	**R1**	GPIO8_R1EN	**Enables GPIO 8 R1** 0: Disable 1: Enable
20	**R1**	GPIO14_R1EN	**Enables GPIO 14 R1** 0: Disable 1: Enable
19	**R1**	GPIO10_R1EN	**Enables GPIO 10 R1** 0: Disable 1: Enable
18	**R1**	GPIO13_R1EN	**Enables GPIO 13 R1** 0: Disable 1: Enable
17	**R1**	GPIO4_R1EN	**Enables GPIO 4 R1** 0: Disable 1: Enable
16	**R1**	GPIO3_R1EN	**Enables GPIO 3 R1** 0: Disable 1: Enable

0xA002_0b50 GPIO Resistance Enable Control Register 5 GPIO_RESEN5

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_R1[47:32]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
47	**R1**	Reserved	Reserved
46	**R1**	Reserved	Reserved
45	**R1**	Reserved	Reserved
44	**R1**	Reserved	Reserved
43	**R1**	Reserved	Reserved
42	**R1**	Reserved	Reserved
41	**R1**	GPIO55_R1EN	**Enables GPIO 55 R1** 0: Disable 1: Enable
40	**R1**	GPIO54_R1EN	**Enables GPIO 54 R1** 0: Disable 1: Enable
39	**R1**	GPIO53_R1EN	**Enables GPIO 53 R1** 0: Disable 1: Enable
38	**R1**	GPIO51_R1EN	**Enables GPIO 51 R1** 0: Disable 1: Enable

Bit(s)	Mnemonic	Name	Description
37	**R1**	GPIO52_R1EN	**Enables GPIO 52 R1** 0: Disable 1: Enable
36	**R1**	GPIO50_R1EN	**Enables GPIO 50 R1** 0: Disable 1: Enable
35	**R1**	GPIO68_R1EN	**Enables GPIO 68 R1** 0: Disable 1: Enable
34	**R1**	GPIO70_R1EN	**Enables GPIO 70 R1** 0: Disable 1: Enable
33	**R1**	GPIO69_R1EN	**Enables GPIO 69 R1** 0: Disable 1: Enable
32	**R1**	GPIO71_R1EN	**Enables GPIO 71 R1** 0: Disable 1: Enable

GPIO*n* PUPD weak pull-up/pull-down control register. See the tables below for details. E means GPIO DRV control.

KeyPad

KCOL0-4

E	PUPD	R1	R0	weak Pull up/down state
0	0	0	0	Hi-Z
0	0	0	1	PU-75K ohms
0	0	1	0	PU-200K ohms
0	0	1	1	PU-75K//200K ohms
0	1	0	0	Hi-Z
0	1	0	1	PD-75K ohms
0	1	1	0	PD-200K ohms
0	1	1	1	PD-75K//200K ohms
1	X	X	X	Hi-Z

KROW0-4

E	PUPD	R1	R0	weak Pull up/down state
0	0	0	0	Hi-Z
0	0	0	1	PU-75K ohms
0	0	1	0	PU-2K ohms
0	0	1	1	PU-75K//2K ohms
0	1	0	0	Hi-Z
0	1	0	1	PD-75K ohms
0	1	1	0	PD-2K ohms
0	1	1	1	PD-75K//2K ohms
1	X	X	X	Hi-Z

MSDC/KCOL6/KCOL7/KROW7

MSDC PAD Control

* No keeper function is required

pu_pd	ctrl_r0	ctrl_r1	Description
0	0	0	both resistors disabled
0	0	1	enable pull up 47K
0	1	0	enable pull up 47K
0	1	1	enable pull up 23.5K
1	0	0	both resistors disabled
1	0	1	enable pull down 47K
1	1	0	enable pull down 47K
1	1	1	enable pull down 23.5K

Others :Using GPIO_PULLEN0 ~ 6 &GPIO_PULLSEL0 ~ 6 control

GPIO_PULLSEL*n* GPIO pull-up/down selection. Only valid if the corresponding GPIO_PULLEN is high.

0 GPIO_PULLSELs pull-down is selected.

1 GPIO_PULLSELs pull-up is selected.

0xA002_0c00 GPIO Mode Control Register 0 GPIO_MODE0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO7_M				GPIO6_M				GPIO5_M				GPIO4_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO3_M				GPIO2_M				GPIO1_M				GPIO0_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO7_M**	GPIO7_MODE	**GPIO 7 mode selection** 0: GPIO7 (IO) 1: KCOL2 (IO) 2: Reserved 3: Reserved 4: FMJTCK(I) 5: JTCK(I) 6: BTJTCK (I) 7: DEBUG0MON5(O)
27:24	**GPIO6_M**	GPIO6_MODE	**GPIO 6 mode selection** 0: GPIO6 (IO) 1: KCOL3 (IO) 2: EDI2CK (O) 3: Reserved 4: FMJTMS(I) 5: JTMS(I) 6: BTJTMS(I) 7: DEBUG0MON4(O)
23:20	**GPIO5_M**	GPIO5_MODE	**GPIO 5 mode selection**

Bit(s)	Mnemonic	Name	Description
			0: GPIO5 (IO) 1: KCOL4 (IO) 2: Reserved 3: EINT1 (I) 4: FMJTDI(I) 5: JTDI (I) 6: BTJDI(I) 7: DEBUG0MON3(O)
19:16	**GPIO4_M**	GPIO4_MODE	**GPIO 4 mode selection** 0: GPIO4 (IO) 1: MCDA2 (IO) 2: WIFITOBT (I) 3: Reserved 4:FMJTMS (I) 5: JTMS (I) 6: Reserved 7: Reserved
15:12	**GPIO3_M**	GPIO3_MODE	**GPIO 3 mode selection** 0: GPIO3 (IO) 1: MCDA1 (IO) 2: Reserved 3: Reserved 4: FMJTDI (I) 5: JTDI (I) 6: Reserved 7: Reserved
11:8	**GPIO2_M**	GPIO2_MODE	**GPIO 2 mode selection** 0: GPIO2 (IO) 1: SDA (IO) 2: Reserved 3: Reserved 4: SPICS0 (O) 5: D1_IMS (I) 6: Reserved 7: DEBUG0MON2(O)
7:4	**GPIO1_M**	GPIO1_MODE	**GPIO 1 mode selection** 0: GPIO1 (IO) 1: SCL (IO) 2: Reserved 3: Reserved 4: SPISCK0 (O) 5: D1_ICK (I) 6: Reserved 7: DEBUG0MON1(O)
3:0	**GPIO0_M**	GPIO0_MODE	**GPIO 0 mode selection** 0: GPIO0 (IO) 1: EDICK (O) 2: PWM (O) 3: EINT0 (O) 4: Reserved 5: Reserved 6: Reserved 7: DEBUG0MON0(O)

0xA002_0c10 GPIO Mode Control Register 1 GPIO_MODE1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name		GPIO15_M				GPIO14_M				GPIO13_M				GPIO12_M			
Type		R/W				R/W				R/W				R/W			
Reset		0				0				0				0			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name		GPIO11_M				GPIO10_M				GPIO9_M				GPIO8_M			
Type		R/W				R/W				R/W				R/W			
Reset		0				0				0				0			

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO15_M**	GPIO15_MODE	**GPIO 15 Mode Selection** 0: GPIO15 (IO) 1: KROW0 (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: DEBUG0MON12(O)
27:24	**GPIO14_M**	GPIO14_MODE	**GPIO 14 mode selection** 0: GPIO14 (IO) 1: KROW1 (IO) 2: Reserved 3: Reserved 4: Reserved 5: D1_IDA(IO) 6: BTDBGIN (O) 7: DEBUG0MON11 (O)
23:20	**GPIO13_M**	GPIO13_MODE	**GPIO 13 mode selection** 0: GPIO13 (IO) 1: KROW2 (IO) 2: Reserved 3: Reserved 4: Reserved 5: JRTCK(O) 6: BTDBGACKN(O) 7: DEBUG0MON10 (O)
19:16	**GPIO12_M**	GPIO12_MODE	**GPIO 12 mode selection** 0: GPIO12 (IO) 1: KROW3 (IO) 2: EDI2DAT (IO) 3: Reserved 4: FMJTDO(O) 5: JTDO(O) 6: BTJDO(O) 7: DEBUG0MON9(O)
15:12	**GPIO11_M**	GPIO11_MODE	**GPIO 11 mode selection** 0: GPIO11 (IO) 1: KROW4 (IO) 2: EDI2WS (O) 3: EINT3(I) 4: FMJTRSTB (I) 5: JTRSTB(I) 6: BTJTRSTB(I) 7: DEBUG0MON8(O)
11:8	**GPIO10_M**	GPIO10_MODE	**GPIO 10 mode selection** 0: GPIO10 (IO) 1: MCDA3 (IO)

Bit(s)	Mnemonic	Name	Description
			2: Reserved 3: EINT2(I) 4: FMJTCK(I) 5: JTCK(I) 6: Reserved 7: Reserved
7:4	**GPIO9_M**	GPIO9_MODE	**GPIO 9 mode selection** 0: GPIO9 (IO) 1: KCOL0 (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: DEBUG0MON7(O)
3:0	**GPIO8_M**	GPIO8_MODE	**GPIO 8 mode selection** 0: GPIO8 (IO) 1: KCOL1 (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: DEBUG0MON4(O)

0xA002_0c20 GPIO Mode Control Register 2 GPIO_MODE2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO23_M				GPIO22_M				GPIO21_M				GPIO20_M		
Type		R/W				R/W				R/W				R/W		
Reset		1				1				0				0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO19_M				GPIO18_M				GPIO17_M				GPIO16_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO23_M**	GPIO23_MODE	**GPIO 23 mode selection** 0: GPIO23 (IO) 1: U1TXD (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
27:24	**GPIO22_M**	GPIO22_MODE	**GPIO 22 mode selection** 0: GPIO22 (IO) 1: U1RXD (I) 2: Reserved 3: EINT4(I) 4: Reserved 5: Reserved 6: Reserved

Bit(s)	Mnemonic	Name	Description
			7: Reserved
23:20	**GPIO21_M**	GPIO21_MODE	**GPIO 21 mode selection** 0: GPIO21 (IO) 1: U2TXD (O) 2: U1CTS(I) 3: CLKO0 (O) 4: SPIMISO0(O) 5: Reserved 6: Reserved 7: Reserved
19:16	**GPIO20_M**	GPIO20_MODE	**GPIO 20 mode selection** 0: GPIO20 (IO) 1: U2RXD (I) 2: U1RTS (O) 3: BTPRI(IO) 4: SPIMOSI0(O) 5: Reserved 6: Reserved 7: Reserved
15:12	**GPIO19_M**	GPIO19_MODE	**GPIO 19 mode selection** 0: GPIO19 (IO) 1: BPI_BUS4 (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: DEBUG0MON15(O)
11:8	**GPIO18_M**	GPIO18_MODE	**GPIO 18 mode selection** 0: GPIO18 (IO) 1: EDIWS (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: DEBUG0MON14(O)
7:4	**GPIO17_M**	GPIO17_MODE	**GPIO 17 mode selection** 0: GPIO17 (IO) 1: BPI_BUS5(O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: DEBUG0MON13(O)
3:0	**GPIO16_M**	GPIO16_MODE	**GPIO 16 mode selection** 0: GPIO16 (IO) 1: GPSFSYNC (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved

0xA002_0c30 GPIO Mode Control Register 3 GPIO_MODE3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO31_M				GPIO30_M				GPIO29_M				GPIO28_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO27_M				GPIO26_M				GPIO25_M				GPIO24_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO31_M**	GPIO31_MODE	**GPIO 31 mode selection** 0: GPIO31 (IO) 1: NLD5 (IO) 2: Reserved 3: Reserved 4: Reserved 5: CMCSD0(I) 6: DEBUG1MIN6(I) 7: DEBUG1MON6(O)
27:24	**GPIO30_M**	GPIO30_MODE	**GPIO 30 mode selection** 0: GPIO30 (IO) 1: NLD6 (IO) 2: Reserved 3: Reserved 4: Reserved 5: CMRST(O) 6: DEBUG1MIN5(I) 7: DEBUG1MON5(O)
23:20	**GPIO29_M**	GPIO29_MODE	**GPIO 29 mode selection** 0: GPIO29 (IO) 1: NLD7(IO) 2: Reserved 3: Reserved 4: Reserved 5: CMCSK(I) 6: DEBUG1MIN4(I) 7: DEBUG1MON4(O)
19:16	**GPIO28_M**	GPIO28_MODE	**GPIO 28 mode selection** 0: GPIO28 (IO) 1: NLD8 (IO) 2: Reserved 3: Reserved 4: Reserved 5: CMMCLK(O) 6: DEBUG1MIN3(I) 7: DEBUG1MON3(O)
15:12	**GPIO27_M**	GPIO27_MODE	**GPIO 27 mode selection** 0: GPIO27 (IO) 1: MCCM0 (IO) 2: Reserved 3: Reserved 4: FMJTDO(O) 5: JTDO(O) 6: DEBUG1MIN2(I) 7: DEBUG1MON2(O)

Bit(s)	Mnemonic	Name	Description
11:8	**GPIO26_M**	GPIO26_MODE	**GPIO 26 mode selection** 0: GPIO26 (IO) 1: MCDA0 (I) 2: Reserved 3: Reserved 4: FMJTRSTB(I) 5: JTRSTB(I) 6: DEBUG1MIN1(I) 7: DEBUG1MON1(O)
7:4	**GPIO25_M**	GPIO25_MODE	**GPIO 25 mode selection** 0: GPIO25 (IO) 1: MCCK (IO) 2: Reserved 3: Reserved 4: Reserved 5: JTRCK(O) 6: DEBUG1MIN0(I) 7: DEBUG1MON0(O)
3:0	**GPIO24_M**	GPIO24_MODE	**GPIO 24 mode selection** 0: GPIO24 (IO) 1: EINT5(I) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved

0xA002_0c40 GPIO Mode Control Register 4 GPIO_MODE4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO39_M				GPIO38_M				GPIO37_M				GPIO36_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO35_M				GPIO34_M				GPIO33_M				GPIO32_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO39_M**	GPIO39_MODE	**GPIO 39 mode selection** 0: GPIO39 (IO) 1: LPA0 (O) 2: LSDI0 (I) 3: SFOUT(IO) 4: Reserved 5: Reserved 6: BT_RGPIO4 (IO) 7: DEBUG1MON14(O)
27:24	**GPIO38_M**	GPIO38_MODE	**GPIO 38 mode selection** 0: GPIO38 (IO) 1: LRDB (O) 2: LSDA0 (IO) 3: SFIN(IO)

Bit(s)	Mnemonic	Name	Description
			4: Reserved 5: Reserved 6: BT_RGPIO3 (IO) 7: DEBUG1MON13(O)
23:20	**GPIO37_M**	GPIO37_MODE	**GPIO 37 mode selection** 0: GPIO37 (IO) 1: LWRB (O) 2: LSCK0(O) 3: SFCK(IO) 4: Reserved 5: Reserved 6: BT_RGPIO2 (IO) 7: DEBUG1MON12(O)
19:16	**GPIO36_M**	GPIO36_MODE	**GPIO 36 mode selection** 0: GPIO36 (IO) 1: NLD0 (IO) 2: LSA0DA0(O) 3: SFCS0(O) 4: Reserved 5: Reserved 6: BT_RGPIO1(IO) 7: DEBUG1MON11(O)
15:12	**GPIO35_M**	GPIO35_MODE	**GPIO 35 mode selection** 0: GPIO35 (IO) 1: NLD1 (IO) 2: Reserved 3: SFWP(IO) 4: Reserved 5: Reserved 6: BT_RGPIO0(IO) 7: DEBUG1MON10(O)
11:8	**GPIO34_M**	GPIO34_MODE	**GPIO 34 mode selection** 0: GPIO34 (IO) 1: NLD2 (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: DEBUG1MON9(O)
7:4	**GPIO33_M**	GPIO33_MODE	**GPIO 33 mode selection** 0: GPIO33 (IO) 1: NLD3 (IO) 2: Reserved 3: Reserved 4: Reserved 5: CMCSD1 (I) 6: Reserved 7: DEBUG1MON8(O)
3:0	**GPIO32_M**	GPIO32_MODE	**GPIO 32 mode selection** 0: GPIO32 (IO) 1: NLD4 (IO) 2: Reserved 3: Reserved 4: Reserved 5: CMPDN (O) 6: Reserved 7: DEBUG1MON7(O)

0xA002_0c50 GPIO Mode Control Register 5 GPIO_MODE5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO47_M				GPIO46_M				GPIO45_M				GPIO44_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				1		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO43_M				GPIO42_M				GPIO41_M				GPIO40_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				1				1				1		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO47_M**	GPIO47_MODE	**GPIO 47 mode selection** 0: GPIO47 (IO) 1: CMDAT0 (O) 2: CMCSD0 (I) 3: U2RTS (I) 4: FMJTDI (I) 5: JTDI (I) 6: BTJTDI (I) 7: Reserved
27:24	**GPIO46_M**	GPIO46_MODE	**GPIO 46 mode selection** 0: GPIO46 (IO) 1: LPRSTB (O) 2: LSRSTB (O) 3: Reserved 4: LPTE1 (I) 5: LPCE3B (O) 6: Reserved 7: Reserved
23:20	**GPIO45_M**	GPIO45_MODE	**GPIO 45 mode selection** 0: GPIO45 (IO) 1: LPTE0 (O) 2: Reserved 3: CLKO1(O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
19:16	**GPIO44_M**	GPIO44_MODE	**GPIO 44 mode selection** 0: GPIO44 (IO) 1: WATCHDOG (O) 2: LPCE2B(O) 3: EINT6(I) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
15:12	**GPIO43_M**	GPIO43_MODE	**GPIO 43 mode selection** 0: GPIO43 (IO) 1: LSRSTB (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved

Bit(s)	Mnemonic	Name	Description
			7: EGND37 (I)
11:8	**GPIO42_M**	GPIO42_MODE	**GPIO 42 mode selection** 0: GPIO42 (IO) 1: LPCE1B (O) 2: LPTE1(I) 3: SDA(IO) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
7:4	**GPIO41_M**	GPIO41_MODE	**GPIO 41 mode selection** 0: GPIO41 (IO) 1: LSCE1B (O) 2: LPCE2B (O) 3: SCL(IO) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
3:0	**GPIO40_M**	GPIO40_MODE	**GPIO 40 mode selection** 0: GPIO40 (IO) 1: LPCE0B (O) 2: LSCE0B0 (O) 3: SFHOLD(IO) 4: Reserved 5: BT_RGPIO5 (IO) 6: DEBUG1MON15(O) 7:DEBUG1MON3 (O)

0xA002_0c60 GPIO Mode Control Register 6 GPIO_MODE6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO55_M				GPIO54_M				GPIO53_M				GPIO52_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO51_M				GPIO50_M				GPIO49_M				GPIO48_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO55_M**	GPIO55_MODE	**GPIO 55 mode selection** 0: GPIO55 (IO) 1: CMHREF (I) 2: LSCE0B1 (O) 3: MC2CK (IO) 4: DAISYNC(O) 5: Reserved 6: DEBUG2MIN1 (I) 7: DEBUG2MON1(O)
27:24	**GPIO54_M**	GPIO54_MODE	**GPIO 54 mode selection** 0: GPIO54 (IO) 1: CMDAT7 (I)

Bit(s)	Mnemonic	Name	Description
			2: LSDI1 (I) 3: MC2DA3(IO) 4: DAIPCMOUT(O) 5: Reserved 6: DEBUG2MIN0 (I) 7: DEBUG2MON0(O)
23:20	**GPIO53_M**	GPIO53_MODE	**GPIO 53 mode selection** 0: GPIO53 (IO) 1: CMDAT6 (I) 2: LSDA1 (I) 3: MC2DA2(IO) 4: DAIPCMIN(I) 5: Reserved 6: BTDBGIN(I) 7: Reserved
19:16	**GPIO52_M**	GPIO52_MODE	**GPIO 52 mode selection** 0: GPIO52 (IO) 1: CMDAT5 (I) 2: LSCK1 (I) 3: MC2DA1(IO) 4: FMJTDO(O) 5: JTDO(O) 6: BTJTDO(O) 7: Reserved
15:12	**GPIO51_M**	GPIO51_MODE	**GPIO 51 mode selection** 0: GPIO51 (IO) 1: CMDAT4 (I) 2: LSA0DA1 (O) 3: MC2DA0 (IO) 4: FMJTRSTB (I) 5: JTRST_B (I) 6: BTJTRSTB(I) 7: Reserved
11:8	**GPIO50_M**	GPIO50_MODE	**GPIO 50 mode selection** 0: GPIO50 (IO) 1: CMDAT3 (I) 2: LSRSTB (O) 3: MC2CM(IO) 4: DAICLK(O) 5: JRTCK(O) 6: BTDBGACKN (O) 7: Reserved
7:4	**GPIO49_M**	GPIO49_MODE	**GPIO 49 mode selection** 0: GPIO49 (IO) 1: CMDAT2 (I) 2: SCL (IO) 3: Reserved 4: FMJTCK (I) 5: JTCK (I) 6: BTJTCK(I) 7: Reserved
3:0	**GPIO48_M**	GPIO48_MODE	**GPIO 48 mode selection** 0: GPIO48 (IO) 1: CMDAT1 (I) 2: CMCSD1 (I) 3: U2CTS (I) 4: FMJTMS (I) 5: JTMS (I) 6: BTJTMS (I)

Bit(s)	Mnemonic	Name	Description
			7: Reserved

0xA002_0c70 GPIO Mode Control Register 7 GPIO_MODE7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO63_M				GPIO62_M				GPIO61_M				GPIO60_M		
Type		R/W				R/W				R/W				R/W		
Reset		1				1				0				0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO59_M				GPIO58_M				GPIO57_M				GPIO56_M		
Type		R/W				R/W				R/W				R/W		
Reset		0				0				0				0		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO63_M**	GPIO63_MODE	**GPIO 63 mode selection** 0: GPIO63 (IO) 1: BPI_BUS2 (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
27:24	**GPIO62_M**	GPIO62_MODE	**GPIO 62 mode selection** 0: GPIO62 (IO) 1: BPI_BU3 (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
23:20	**GPIO61_M**	GPIO61_MODE	**GPIO 61 mode selection** 0: GPIO61 (IO) 1: EDIDAT (IO) 2: PWM (O) 3: EINT8(I) 4: Reserved 5: Reserved 6: Reserved 7: Reserved)
19:16	**GPIO60_M**	GPIO60_MODE	**GPIO 60 mode selection** 0: GPIO60 (IO) 1: CMRST (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: DEBUG2MIN6 (I) 7: DEBUG2MON6(O)
15:12	**GPIO59_M**	GPIO59_MODE	**GPIO 59 mode selection** 0: GPIO59 (IO) 1: CMPCLK (I)

Bit(s)	Mnemonic	Name	Description
			2: CMCSK (I) 3: Reserved 4: Reserved 5: Reserved 6: DEBUG2MIN5 (I) 7: DEBUG2MON5(O)
11:8	**GPIO58_M**	GPIO58_MODE	**GPIO 58 mode selection** 0: GPIO58 (IO) 1: CMMCLK (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: DEBUG2MIN4 (I) 7: DEBUG2MON4(O)
7:4	**GPIO57_M**	GPIO57_MODE	**GPIO 57 mode selection** 0: GPIO57 (IO) 1: CMPDN (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: DEBUG2MIN3 (I) 7: DEBUG2MON3(O)
3:0	**GPIO56_M**	GPIO56_MODE	**GPIO 56 mode selection** 0: GPIO56 (IO) 1: CMVREF (I) 2: SDA (IO) 3: EINT7 (I) 4: DAIRST (I) 5: LPTE0(I) 6: DEBUG2MIN2 (I) 7: DEBUG2MON2(O)

0xA002_0c80 GPIO Mode Control Register 8 GPIO_MODE8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO71_M				GPIO70_M				GPIO69_M				GPIO68_M		
Type		R/W				R/W				R/W				R/W		
Reset		1				1				1				1		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO67_M				GPIO66_M				GPIO65_M				GPIO64_M		
Type		R/W				R/W				R/W				R/W		
Reset		1				0				1				1		

Bit(s)	Mnemonic	Name	Description
31:28	**GPIO71_M**	GPIO71_MODE	**GPIO 71 mode selection** 0: GPIO71 (IO) 1: SFOUT (IO) 2: MC2CM(IO) 3: Reserved 4: Reserved 5: Reserved 6: Reserved

Bit(s)	Mnemonic	Name	Description
			7: Reserved
27:24	**GPIO70_M**	GPIO70_MODE	**GPIO 70 mode selection** 0: GPIO70 (IO) 1: SFIN (IO) 2: MC2DA2(IO) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
23:20	**GPIO69_M**	GPIO69_MODE	**GPIO 69 mode selection** 0: GPIO69 (IO) 1: SFCK (IO) 2: MC2DA0(IO) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
19:16	**GPIO68_M**	GPIO68_MODE	**GPIO 68 mode selection** 0: GPIO68 (IO) 1: SFCS0 (O) 2: MC2DA1(IO) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
15:12	**GPIO67_M**	GPIO67_MODE	**GPIO 67 mode selection** 0: GPIO67 (IO) 1: SFWP (IO) 2: MC2CK(IO) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
11:8	**GPIO66_M**	GPIO66_MODE	**GPIO 66 mode selection** 0: GPIO66 (IO) 1: SFCS1(O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
7:4	**GPIO65_M**	GPIO65_MODE	**GPIO 65 mode selection** 0: GPIO65 (IO) 1: BPI_BUS0 (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
3:0	**GPIO64_M**	GPIO64_MODE	**GPIO 64 mode selection** 0: GPIO64 (IO) 1: BPI_BUS1 (O)

Bit(s)	Mnemonic	Name	Description
			2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved

0xA002_0c90 GPIO Mode Control Register 9 GPIO_MODE9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset		1				1				1				1		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														GPIO72_M		
Type														R/W		
Reset		1				1				1				1		

Bit(s)	Mnemonic	Name	Description
3:0	**GPIO72_M**	GPIO72_MODE	**GPIO 72 mode selection** 0: GPIO72 (IO) 1: SFSHOLD (IO) 2: MC2DA3(IO) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: EGND68 (I)

0xA002_0d00 GPIO Rx Duty Select Control Register GPIO_RDSEL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_RDSEL[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:13	**GPIO_RDSEL**	Reserved	Reserved
12:11	**GPIO_RDSEL**	Reserved	Reserved
10:8	**GPIO_RDSEL**	Reserved	Reserved
7:6	**GPIO_RDSEL**	Reserved	Reserved
5:4	**GPIO_RDSEL**	Reserved	Reserved
3:2	**GPIO_RDSEL**	GPIO_RDSEL	**Rx duty configuration** For the IO list: GPIO66/67/68/69/70/71/72 0: Input buffer duty high when asserted 1: Input buffer duty low when asserted
1:0	**GPIO_RDSEL**	Reserved	Reserved

RDSEL[1:0] Selects Rx duty

Bit 0 Input buffer duty high when asserted (high pulse width adjustment)

Bit 1 Input buffer duty low when asserted(low pulse width adjustment)

0xA002_0d10 GPIO Tx Duty Select Control Register GPIO_TDSEL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO_TDSEL[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:13	GPIO_TDSEL	Reserved	Reserved
12:11	GPIO_TDSEL	Reserved	Reserved
10:8	GPIO_TDSEL	Reserved	Reserved
7:6	GPIO_TDSEL	Reserved	Reserved
5:4	GPIO_TDSEL	Reserved	Reserved
3:2	GPIO_TDSEL	D2D_RDSEL	**Tx duty configuration** For the IO list: GPIO66/67/68/69/70/71/72 0: Output buffer duty high when asserted 1: Output buffer duty low when asserted
1:0	GPIO_RDSEL	GPIO_RDSEL	Reserved

TDSEL[1:0] Selects Tx duty

Bit 0 Output level shifter duty high when asserted (high pulse width adjustment)

Bit 1 Output level shifter duty low when asserted(low pulse width adjustment)

0xA002_0e00 CLK_OUT0 Setting CLKO_MODE0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_MODE0			
Type													R/W			
Reset													0			

CLK_MODE0 Selects the CLKO0 output source

0 sfc_ck_div 2 (65MHz, 52MHz, 39MHz, etc.)

1 f26m_ck (26MHz)

2 emi1x_ck_div2 (65MHz, 52MHz, etc.)

3 command d2d_ck_div2 (37.5MHz, 26MHz, etc.)

4 f32k_ck (32kHz)

5 command d2d_ck (65MHz, 52MHz, 26MHz)

6 BT ahb_bus_clk_div2 (65MHz, 52MHz, etc.)

7 MMsys_emi1x_ck_div2 (65MHz, 52MHz)

8 MCUSYS ahb bus clk (65MHz, 52MHz)

9 MSDC clk_div2 (~49MHz)

A ARM clk_div4 (65MHz, 52MHz, etc.)

B FM 128Khz clk (128kHz)
C USB 48Mhz clk (48MHz)
D x4g md2g clk (52MHz)
E BSI_clk_div2 (65MHz)
F DSP_clk_div2 (78MHz)

0xA002_0e10 CLK_OUT1 Setting CLKO_MODE1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_MODE1			
Type													R/W			
Reset													0			

CLK_MODE1 Selects the CLKO1 output source

0 sfc_ck_div 2 (65MHz, 52MHz, 39MHz, etc.)
1 f26m_ck (26MHz)
2 emi1x_ck_div2 (65MHz, 52MHz, etc.)
3 command d2d_ck_div2 (37.5MHz, 26MHz, etc.)
4 f32k_ck (32kHz)
5 command d2d_ck (65MHz, 52MHz, 26MHz)
6 BT ahb_bus_clk_div2 (65MHz, 52MHz, etc.)
7 MMsys_emi1x_ck_div2 (65MHz, 52MHz)
8 MCUSYS ahb bus clk (65MHz, 52MHz)
9 MSDC clk_div2 (~49MHz)
A ARM clk_div4 (65MHz, 52MHz, etc.)
B FM 128Khz clk (128kHz)
C USB 48Mhz clk (48MHz)
D x4g md2g clk (52MHz)
E BSI_clk_div2 (65MHz)
F DSP_clk_div2 (78MHz)

Example :
If you would like to select clock out to pin, set up the GPIO mode and choose the GPIO pin first. Then set up the CLK_MODE0/CLK_MODE1 register for chose clock frequency. The related GPIO PIN_MUX of clock out will output the correct frequency.

A0730104 ACFG_PIN_PULLEN1 Pin Control 1 0078

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										IO_CFG_PULLEN1						
Type										RW						
Reset										1	1	1	1	0	0	0

Bit(s)	Mnemonic	Name	Description
6:0		IO_CFG_PULLEN1	**Digital IO pull-up/down control** Bit[0]: N/A Bit[1]: GPIO74 Bit[2]: GPIO73 Bit[3]: AGPI78 Bit[4]: AGPI77 Bit[5]: AGPI76 Bit[6]: AGPI75

Bit(s)	Mnemonic	Name	Description
			0: Disable pull-up/down 1: Enable pull-up/down

A073010C ACFG_PIN_PULLSEL1 Pin Control 3 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										IO_CFG_PULLSEL1						
Type										RW						
Reset										0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
6:0		IO_CFG_PULLSEL1	**Selects digital IO pull-up/down** Bit[0]: N/A Bit[1]: GPIO74 Bit[2]: GPIO73 Bit[3]: AGPI78 Bit[4]: AGPI77 Bit[5]: AGPI76 Bit[6]: AGPI75 0: Pulled-up 1: Pulled-down

A0730118 ACFG_PIN_DRV2 Pin Control 6 003F

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											IO_CFG_DRV2					
Type											RW					
Reset											1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
5:0		IO_CFG_DRV2	**Digital IO output driving status** Bit[1:0]: N/A Bit[3:2]: GPIO74 Bit[5:4]: GPIO73 2'b00: 4mA 2'b01: 8mA 2'b10: 12mA 1'b11: 16mA

A0730120 ACFG_PIN_SMT1 Pin Control 8 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														IO_CFG_SMT1		
Type														RW		
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2:0		IO_CFG_SMT1	**Digital IO schmitt trigger status** Bit[0]: N/A Bit[1]: GPIO74

Bit(s)	Mnemonic	Name	Description
			Bit[2]: GPIO73 0: Schmitt trigger off 1: Schmitt trigger on

A0730128 ACFG_PIN_IES1 Pin Control 10 0007

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														IO_CFG_IES1		
Type														RW		
Reset														1	1	1

Bit(s)	Mnemonic	Name	Description
2:0		IO_CFG_IES1	**Digital IO IES status** Bit[0]: N/A Bit[1]: GPIO74 Bit[2]: GPIO73 0: Enable input isolation 1: Normal

A073012C ACFG_PIN_G Pin Control 11 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																IO_CFG_G
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0		IO_CFG_G	**Analog AGPI gating control** To support R-touch panel, this bit has to be set to 0 to enable XP/XM/YP/YM as analog IO. To enable XP/XM/YP/YM as GPI or EINT, this bit has to be set to 1. 0: Analog IO 1: Digital IO

A0730130 ACFG_PIN_DINV Pin Control 12 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		IO_CFG_DINV														
Type		RW														
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
14:0		IO_CFG_DINV	**GPIO data-in polarity inversion** GPIO_DIN[5:0]: N/A GPIO_DIN[6]: GPIO74 GPIO_DIN[7]: GPIO73 GPIO_DIN[8]: AGPI75 GPIO_DIN[9]: AGPI76 GPIO_DIN[10]: AGPI77 GPIO_DIN[11]: AGPI78 GPIO_DIN[12]: AGPI80

Bit(s)	Mnemonic	Name	Description
			GPIO_DIN[13]: AGPI81 GPIO_DIN[14]: AGPI82 0: Keep polarity 1: Invert polarity

A0730134 ACFG_GPIO_OE GPIO OE Control 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															GPIO_OE	
Type															RW	
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1:0		GPIO_OE	**GPIO direction control** GPIO_OE[0]: GPIO74 GPIO_OE[1]: GPIO73 0: Input 1: Output

A0730138 ACFG_GPIO_DOUT GPIO DOUT Control 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														GPIO_DOUT		
Type														RW		
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2:0		GPIO_DOUT	**GPIO data-out value** GPIO_DOUT[0]: GPIO74 GPIO_DOUT[1]: GPIO73 GPIO_DOUT[2]: AGPO79 0: Data-out is 0. 1: Data-out is 1.

A073013C ACFG_GPIO_DIN GPIO DIN Status 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO_DIN														
Type		RO														
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
14:0		GPIO_DIN	**GPIO data-in read-out** GPIO_DIN[5:0]: N/A GPIO_DIN[6]: GPIO74 GPIO_DIN[7]: GPIO73 GPIO_DIN[8]: AGPI75 GPIO_DIN[9]: AGPI76 GPIO_DIN[10]: AGPI77 GPIO_DIN[11]: AGPI78 GPIO_DIN[12]: AGPI80

Bit(s)	Mnemonic	Name	Description
			GPIO_DIN[13]: AGPI81 GPIO_DIN[14]: AGPI82 0: Data-in is 0. 1: Data-in is 1.

A0730144 ACFG_GPIO_MODE1 GPIO Mode Control 1 0111

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			GPIO7_M				GPIO6_M									
Type			RW				RW									
Reset			0	0			0	1								

Bit(s)	Mnemonic	Name	Description
13:12		GPIO7_M	**Selects GPIO73 mode** 0: GPIO73 1: SRCLKENAI
9:8		GPIO6_M	**Selects GPIO74 mode** 0: GPIO74 1: RESETB

A0730148 ACFG_GPIO_MODE2 GPIO Mode Control 2 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			GPIO11_M				GPIO10_M				GPIO9_M				GPIO8_M	
Type			RW				RW				RW				RW	
Reset			0	0			0	0			0	0			0	0

Bit(s)	Mnemonic	Name	Description
13:12		GPIO11_M	**Selects AGPI78 mode** Effective only when IO_CFG_G = 1 0: AGPI78 1: N/A 2: EINT33 3: N/A
9:8		GPIO10_M	**Selects AGPI77 mode** Effective only when IO_CFG_G = 1 0: AGPI77 1: N/A 2: EINT32 3: N/A
5:4		GPIO9_M	**Selects AGPI76 mode** Effective only when IO_CFG_G = 1 0: AGPI76 1: N/A 2: EINT31 3: N/A
1:0		GPIO8_M	**Selects AGPI75 mode** Effective only when IO_CFG_G = 1 0: AGPI75 1: N/A 2: EINT30 3: N/A

A073014C ACFG_GPIO_MODE3 GPIO Mode Control 3 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								GPIO14_M				GPIO13_M				GPIO12_M
Type								RW				RW				RW
Reset								0				0				0

Bit(s)	Mnemonic	Name	Description
8		GPIO14_M	**Selects AGPI82 mode** 0: AGPI82 1: EINT36
4		GPIO13_M	**Selects AGPI81 mode** 0: AGPI81 1: EINT35
0		GPIO12_M	**Selects AGPI80 mode** 0: AGPI80 1: EINT34

item	idx	Power (SiP)	Pin Name	GPIO	#	Mode	default Func.0	Aux Func.0	Aux Func.1	Aux Func.2	Aux Func.3	Aux Func.4	Aux Func.5	Aux Func.6	Aux Func.7
3	#N/A	DVDD33	EDICK	1	0	0	B:GPIO0	B:GPIO0	O:EDICK	O:PWM	I1:EINT0				O:DEBUG0MON0
4	1	DVDD33	SCL28	1	1	0	B:GPIO1	B:GPIO1	B1:SCL			O:SPISCK0	I0:D1_ICK		O:DEBUG0MON1
5	0	DVDD33	SDA28	1	2	0	B:GPIO2	B:GPIO2	B1:SDA			O:SPICS0	I0:D1_IMS		O:DEBUG0MON2
6	#N/A	DVDD_MSDC	MCDA1	1	3	0	B:GPIO3	B:GPIO3	B1:MCDA1			I1:FMJTDI	I1:JTDI		
7	#N/A	DVDD_MSDC	MCDA2	1	4	0	B:GPIO4	B:GPIO4	B1:MCDA2	I0:WIFITOBT		I1:FMJTMS	I1:JTMS		
9	1	DVDD33	KCOL4	1	5	0	B:GPIO5	B:GPIO5	B1:KCOL4		I1:EINT1	I1:FMJTDI	I1:JTDI	I1:BTJTDI	O:DEBUG0MON3
18	1	DVDD33	KCOL3	1	6	0	B:GPIO6	B:GPIO6	B1:KCOL3	O:EDI2CK		I1:FMJTMS	I1:JTMS	I1:BTJTMS	O:DEBUG0MON4
19	1	DVDD33	KCOL2	1	7	0	B:GPIO7	B:GPIO7	B1:KCOL2			I1:FMJTCK	I1:JTCK	I1:BTJTCK	O:DEBUG0MON5
20	1	DVDD33	KCOL1	1	8	0	B:GPIO8	B:GPIO8	B1:KCOL1						O:DEBUG0MON6
21	0	DVDD33	KCOL0	1	9	0	B:GPIO9	B:GPIO9	B1:KCOL0						O:DEBUG0MON7
22	#N/A	DVDD_MSDC	MCDA3	1	10	0	B:GPIO10	B:GPIO10	B1:MCDA3		I1:EINT2	I1:FMJTCK	I1:JTCK		
25	#N/A	DVDD33	KROW4	1	11	0	B:GPIO11	B:GPIO11	B1:KROW4	O:EDI2WS	I1:EINT3	I1:FMJTRSTB	I0:JTRST_B	I0:BTJTRSTB	O:DEBUG0MON8
26	1	DVDD33	KROW3	1	12	0	B:GPIO12	B:GPIO12	B1:KROW3	B0:EDI2DAT		O:FMJTDO	O:JTDO	O:BTJTDO	O:DEBUG0MON9
27	1	DVDD33	KROW2	1	13	0	B:GPIO13	B:GPIO13	B1:KROW2				O:JTRCK	O:BTDBGACKN	O:DEBUG0MON10
28	1	DVDD33	KROW1	1	14	0	B:GPIO14	B:GPIO14	B1:KROW1				B:D1_IDA	I1:BTDBGIN	O:DEBUG0MON11
29	#N/A	DVDD33	KROW0	1	15	0	B:GPIO15	B:GPIO15	B1:KROW0						O:DEBUG0MON12
30	#N/A	DVDD33	GPIO16	1	16	0	B:GPIO16	B:GPIO16	O:GPSFSYNC						
31	#N/A	DVDD33	GPIO17	1	17	0	B:GPIO17	B:GPIO17	O:BPIBUS5						O:DEBUG0MON13
32	#N/A	DVDD33	EDIWS	1	18	0	B:GPIO18	B:GPIO18	O:EDIWS						O:DEBUG0MON14
33	#N/A	DVDD33	GPIO19	1	19	0	B:GPIO19	B:GPIO19	O:BPIBUS4						O:DEBUG0MON15
34	0	DVDD33	URXD2	1	20	0	B:GPIO20	B:GPIO20	I1:U2RXD	O:U1RTS	B:BTPRI	O:SPIMOSI0			
35	0	DVDD33	UTXD2	1	21	0	B:GPIO21	B:GPIO21	O:U2TXD	I1:U1CTS	O:CLKO0	I0:SPIMISO0			
36	0	DVDD33	URXD1	1	22	1	I1:URXD1	B:GPIO22	I1:U1RXD		I1:EINT4				
37	0	DVDD33	UTXD1	1	23	1	O:UTXD1	B:GPIO23	O:U1TXD						
63	0	DVDD_EMI	MCINS	1	24	0	B:GPIO24	B:GPIO24	I1:EINT5						
67	1	DVDD_MSDC	MCCK	1	25	0	B:GPIO25	B:GPIO25	B1:MCCK				O:JTRCK	I0:DEBUG1MIN0	O:DEBUG1MON0
69	1	DVDD_MSDC	MCDA0	1	26	0	B:GPIO26	B:GPIO26	B1:MCDA0			I1:FMJTRSTB	I0:JTRST_B	I0:DEBUG1MIN1	O:DEBUG1MON1
76	1	DVDD_MSDC	MCCM0	1	27	0	B:GPIO27	B:GPIO27	B1:MCCM0			O:FMJTDO	O:JTDO	I0:DEBUG1MIN2	O:DEBUG1MON2
72	0	DVDD_EMI	NLD8	1	28	0	B:GPIO28	B:GPIO28	B:NLD8				O:CMMCLK	I0:DEBUG1MIN3	O:DEBUG1MON3
58	0	DVDD_EMI	NLD7	1	29	0	B:GPIO29	B:GPIO29	B:NLD7				I0:CMCSK	I0:DEBUG1MIN4	O:DEBUG1MON4
56	1	DVDD_EMI	NLD6	1	30	0	B:GPIO30	B:GPIO30	B:NLD6				O:CMRST	I0:DEBUG1MIN5	O:DEBUG1MON5
71	0	DVDD_EMI	NLD5	1	31	0	B:GPIO31	B:GPIO31	B:NLD5				I0:CMCSD0	I0:DEBUG1MIN6	O:DEBUG1MON6

item	idx	Power (SiP)	Pin Name	GPIO	#	Mode	default Func.0	Aux Func.0	Aux Func.1	Aux Func.2	Aux Func.3	Aux Func.4	Aux Func.5	Aux Func.6	Aux Func.7
62	0	DVDD_EMI	NLD4	1	32	0	B:GPIO32	B:GPIO32	B:NLD4				O:CMPDN		O:DEBUG1MON7
57	0	DVDD_EMI	NLD3	1	33	0	B:GPIO33	B:GPIO33	B:NLD3				I0:CMCSD1		O:DEBUG1MON8
59	0	DVDD_EMI	NLD2	1	34	0	B:GPIO34	B:GPIO34	B:NLD2						O:DEBUG1MON9
65	0	DVDD_EMI	NLD1	1	35	0	B:GPIO35	B:GPIO35	B:NLD1		B:SFWP			B0:BT_RGPIO0	O:DEBUG1MON10
61	0	DVDD_EMI	NLD0	1	36	0	B:GPIO36	B:GPIO36	B:NLD0	O:LSA0DA0	O:SFCS0			B0:BT_RGPIO1	O:DEBUG1MON11
64	1	DVDD_EMI	LWR_B	1	37	0	B:GPIO37	B:GPIO37	O:LWRB	O:LSCK0	B:SFCK			B0:BT_RGPIO2	O:DEBUG1MON12
60	0	DVDD_EMI	LRD_B	1	38	0	B:GPIO38	B:GPIO38	O:LRDB	B1:LSDA0	B:SFIN			B0:BT_RGPIO3	O:DEBUG1MON13
82	1	DVDD_EMI	LPA0	1	39	0	B:GPIO39	B:GPIO39	O:LPA0	I0:LSDI0	B:SFOUT			B0:BT_RGPIO4	O:DEBUG1MON14
87	1	DVDD_EMI	LPCE0_B	1	40	1	O:LPCE0B	B:GPIO40	O:LPCE0B	O:LSCE0B0	B:SFHOLD			B0:BT_RGPIO5	O:DEBUG1MON15
79	#N/A	DVDD_EMI	LSCE1_B	1	41	1	O:LSCE1B	B:GPIO41	O:LSCE1B	O:LPCE2B	B1:SCL				
73	0	DVDD_EMI	LPCE1_B	1	42	1	O:LPCE1B	B:GPIO42	O:LPCE1B	I0:LPTE1	B1:SDA				
68	0	DVDD_EMI	LSRSTB	1	43	0	B:GPIO43	B:GPIO43	O:LSRSTB						
75	1	DVDD_EMI	WATCHDOG	1	44	1	O:WATCHDOG	B:GPIO44	O:WATCHDOG	O:LPCE2B	I1:EINT6				
84	1	DVDD_EMI	LPTE	1	45	0	B:GPIO45	B:GPIO45	I0:LPTE0		O:CLKO1				
80	0	DVDD_EMI	LPRSTB	1	46	0	B:GPIO46	B:GPIO46	O:LPRSTB	O:LSRSTB		I0:LPTE1	O:LPCE3B		
86	0	DVDD33	CMDAT0	1	47	0	B:GPIO47	B:GPIO47	I0:CMDAT0	I0:CMCSD0	O:U2RTS	I1:FMJTDI	I1:JTDI	I1:BTJTDI	
66	1	DVDD33	CMDAT1	1	48	0	B:GPIO48	B:GPIO48	I0:CMDAT1	I0:CMCSD1	I1:U2CTS	I1:FMJTMS	I1:JTMS	I1:BTJTMS	
83	1	DVDD33	CMDAT2	1	49	0	B:GPIO49	B:GPIO49	I0:CMDAT2	B1:SCL		I1:FMJTCK	I1:JTCK	I1:BTJTCK	
70	1	DVDD33	CMDAT3	1	50	0	B:GPIO50	B:GPIO50	I0:CMDAT3	O:LSRSTB	B1:MC3CM0	O:DAICLK	O:JTRCK	O:BTDBGACKN	
77	#N/A	DVDD33	CMDAT4	1	51	0	B:GPIO51	B:GPIO51	I0:CMDAT4	O:LSA0DA1	B1:MC3DA0	I1:FMJTRSTB	I0:JTRST_B	I0:BTJTRSTB	
54	1	DVDD33	CMDAT5	1	52	0	B:GPIO52	B:GPIO52	I0:CMDAT5	O:LSCK1	B1:MC3DA1	O:FMJTDO	O:JTDO	O:BTJTDO	
47	#N/A	DVDD33	CMDAT6	1	53	0	B:GPIO53	B:GPIO53	I0:CMDAT6	B1:LSDA1	B1:MC3DA2	I0:DAIPCMIN		I1:BTDBGIN	
46	1	DVDD33	CMDAT7	1	54	0	B:GPIO54	B:GPIO54	I0:CMDAT7	I0:LSDI1	B1:MC3DA3	O:DAIPCMOUT		I0:DEBUG2MIN0	O:DEBUG2MON0
48	1	DVDD33	CMHREF	1	55	0	B:GPIO55	B:GPIO55	I0:CMHREF	O:LSCE0B1	B1:MC3CK	O:DAISYNC		I0:DEBUG2MIN1	O:DEBUG2MON1
50	1	DVDD33	CMVREF	1	56	0	B:GPIO56	B:GPIO56	I0:CMVREF	B1:SDA	I1:EINT7	I1:DAIRST	I0:LPTE0	I0:DEBUG2MIN2	O:DEBUG2MON2
52	1	DVDD33	CMPDN	1	57	0	B:GPIO57	B:GPIO57	O:CMPDN					I0:DEBUG2MIN3	O:DEBUG2MON3
49	0	DVDD33	CMMCLK	1	58	0	B:GPIO58	B:GPIO58	O:CMMCLK					I0:DEBUG2MIN4	O:DEBUG2MON4
51	1	DVDD33	CMPCLK	1	59	0	B:GPIO59	B:GPIO59	I0:CMPCLK	I0:CMCSK				I0:DEBUG2MIN5	O:DEBUG2MON5
74	0	DVDD33	CMRST	1	60	0	B:GPIO60	B:GPIO60	O:CMRST					I0:DEBUG2MIN6	O:DEBUG2MON6
78	#N/A	DVDD33	EDIDAT	1	61	0	B:GPIO61	B:GPIO61	B0:EDIDAT	O:PWM	I1:EINT8				
53	0	DVDD33	BPI_BUS3	1	62	1	O:BPIBUS3	B:GPIO62	O:BPIBUS3						
88	0	DVDD33	BPI_BUS2	1	63	1	O:BPIBUS2	B:GPIO63	O:BPIBUS2						

item	idx	Power (SiP)	Pin Name	GPIO	#	Mode	default Func.0	Aux Func.0	Aux Func.1	Aux Func.2	Aux Func.3	Aux Func.4	Aux Func.5	Aux Func.6	Aux Func.7
85	0	DVDD33	BPI_BUS1	1	64	1	O:BPIBUS1	B:GPIO64	O:BPIBUS1						
55	0	DVDD33	BPI_BUS0	1	65	1	O:BPIBUS0	B:GPIO65	O:BPIBUS0						
81	#N/A	DVDD28_SFF	SFSCS1	1	66	0	B:GPIO66	B:GPIO66	O:SFCS1						
0	#N/A	DVDD28_SFF	SFSWP	1	67	1	B:SFWP	B:GPIO67	B:SFWP	B1:MC2CK					
1	#N/A	DVDD28_SFF	SFSCS0	1	68	1	O:SFCS0	B:GPIO68	O:SFCS0	B1:MC2DA1					
2	#N/A	DVDD28_SFF	SFSCK	1	69	1	B:SFCK	B:GPIO69	B:SFCK	B1:MC2DA0					
3	#N/A	DVDD28_SFF	SFSIN	1	70	1	B:SFIN	B:GPIO70	B:SFIN	B1:MC2DA2					
8	#N/A	DVDD28_SFF	SFSOUT	1	71	1	B:SFOUT	B:GPIO71	B:SFOUT	B1:MC2CM0					
9	0	DVDD28_SFF	SFSHOLD	1	72	1	B:SFHOLD	B:GPIO72	B:SFHOLD	B1:MC2DA3					
9	0	VDD33_VSIM	SIM1_SIO	1	73	1	B:SIMSIO	B:GPIO73	B:SIMSIO						
9	0	VDD33_VSIM	SIM1_SRST	1	74	1	B:SIMSRST	B:GPIO74	B:SIMSRST						
13	0	VDD33_VSIM	SIM2_SIO	1	75	1	B:SIM2SIO	B:GPIO75	B:SIM2SIO						
13	0	VDD33_VSIM	SIM1_SCLK	1	76	1	B:SIMSCLK	B:GPIO76	B:SIMSCLK						
14	0	VDD33_VSIM	SIM2_SCLK	1	77	1	B:SIM2SCLK	B:GPIO77	B:SIM2SCLK						
15	0	VDD33_VSIM	SIM2_SRST	1	78	1	B:SIM2SRST	B:GPIO78	B:SIM2SRST						
		DVDD33	PRTC_B	0	78	1	I:PRTC_B		I:PRTC_B						
		DVDD33	BONDOPT0	0	78	1	I:BONDOPT0		I:BONDOPT0						
		DVDD33	BONDOPT1	0	78	1	I:BONDOPT1		I:BONDOPT1						
			D2D_BSI_EN	1	79	1	O:D2D_BSI_EN	B:GPIO79	O:D2D_BSI_EN				O:D2D_DEBUG2I	I0:D2D_DEBUG2MON3	
			D2D_BSI_DATAI1	1	80	1	B0:D2D_BSI_DAT	B:GPIO80	B0:D2D_BSI_DATAI1				O:D2D_DEBUG2I	I0:D2D_DEBUG2MON1	
			D2D_BSI_DATAI0	1	81	1	B0:D2D_BSI_DAT	B:GPIO81	B0:D2D_BSI_DATAI0				O:D2D_DEBUG2I	I0:D2D_DEBUG2MON2	
			D2D_BSI_CLK	1	82	1	O:D2D_BSI_CLK	B:GPIO82	O:D2D_BSI_CLK				O:D2D_DEBUG2I	I0:D2D_DEBUG2MON3	
			D2D_ANA_CON_SDA1	0	82	1	B:D2D_ANA_CON_SDA1		B:D2D_ANA_CON_SDA1						
			D2D_ANA_CON_SDA0	0	82	1	B:D2D_ANA_CON_SDA0		B:D2D_ANA_CON_SDA0						
			D2D_ANA_CON_SCK	0	82	1	O:D2D_ANA_CON_SCK		O:D2D_ANA_CON_SCK						
			D2D_ABB_DPH_SCK	0	82	1	O:D2D_ABB_DPH_SCK		O:D2D_ABB_DPH_SCK						
			D2D_ABB_DPH_SDA2	1	83	1	B0:D2D_ABB_DP	B:GPIO83	B0:D2D_ABB_DPH_SDA2						
			D2D_ABB_DPH_SDA1	1	84	1	B0:D2D_ABB_DP	B:GPIO84	B0:D2D_ABB_DPH_SDA1						
			D2D_ABB_DPH_SDA0	1	85	1	B0:D2D_ABB_DP	B:GPIO85	B0:D2D_ABB_DPH_SDA0						
			D2D_XOSC_32K_CK	0	85	1	I:D2D_XOSC_32K_CK		I:D2D_XOSC_32K_CK						
			D2D_SYS_RESETB	0	85	1	I:D2D_SYS_RESETB		I:D2D_SYS_RESETB						
			D2D_RGU_RESETB	0	85	1	O:D2D_RGU_RESETB		O:D2D_RGU_RESETB						
			D2D_TESTMODE	0	85	1	I:D2D_TESTMODE		I:D2D_TESTMODE						
			D2D_BB_SRCLKENA	0	85	1	O:D2D_BB_SRCLKENA		O:D2D_BB_SRCLKENA						
			D2D_ANA_INT	1	86	1	I1:D2D_ANA_INT	B:GPIO86	I1:D2D_ANA_INT				O:D2D_DEBUG2I	I0:D2D_DEBUG2MON4	
			D2D_ANA_WAKEUP	1	87	1	I1:D2D_ANA_WA	B:GPIO87	I1:D2D_ANA_WAKEUP				O:D2D_DEBUG2I	I0:D2D_DEBUG2MON5	

3.5 General-purpose Timer

3.5.1 General Descriptions

Four general-purpose timers are provided. Three timers are 16 bits long and one timer is 32 bits long. Each runs independently. GPT1 ~ 3 use 32k clock source to count, whereas GPT4 uses 26M clock source. The 26M clock source can be gated when the system enters the sleep mode, and this will cause GPT4 to stop counting, whereas 32k clock source is always toggling. GPT1 and GPT2 can operate in one of the two modes: one-shot mode and auto-repeat mode. GPT3 and GPT4 are free running timer. In the one-shot mode, when the timer counts down and reaches 0, it will be halted. In the auto-repeat mode, when the timer reaches 0, it will simply be reset to counting down the initial value and repeating the count-down to 0. This loop keeps repeating until the disabling signal is set to 1. Regardless of the timer's mode, if the countdown initial value (i.e. GPTIMER1_DAT for GPT1 or GPTIMER_DAT2 for GPT2) is written when the timer is running, the new initial value will not take effect until the next time the timer is restarted. In the auto-repeat mode, the new countdown start value is used on the next countdown iteration. Therefore, before enabling the timer, the desired values for GPTIMER_DAT and the GPTIMER_PRESCALER registers must first be set.

3.5.2 Register Definition

Module name: GPTimer base address: (+A00C0000h)

Address	Name	Width	Register function
A00C0000	**GPTIMER1_CON**	32	GPT1 control register
A00C0004	**GPTIMER1_DAT**	32	GPT1 time-out interval register
A00C0008	**GPTIMER2_CON**	32	GPT2 control register
A00C000C	**GPTIMER2_DAT**	32	GPT2 time-out Interval register
A00C0010	**GPTIMER_STA**	32	GPT status register
A00C0014	**GPTIMER1_PRESCALER**	32	GPT1 prescaler register
A00C0018	**GPTIMER2_PRESCALER**	32	GPT2 prescaler register
A00C001C	**GPTIMER3_CON**	32	GPT3 control register
A00C0020	**GPTIMER3_DAT**	32	GPT3 time-out interval register
A00C0024	**GPTIMER3_PRESCALER**	32	GPT3 prescaler register
A00C0028	**GPTIMER4_CON**	32	GPT4 control register
A00C002C	**GPTIMER4_DAT**	32	GPT4 data register

A00C0000 GPTIMER1_CON GPT1 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN	MODE														
Type	RW	RW														
Reset	0	0														

Bit(s)	Mnemonic	Name	Description
15	**EN**	EN	**Controls GPT1 to start counting or to stop** 0: Disable GPT1 1: Enable GPT1
14	**MODE**	MODE	**Controls GPT1 to count repeatedly (in a loop) or just one-shot** 0: One-shot mode is selected. 1: Auto-repeat mode is selected.

A00C0004 GPTIMER1_DAT GPT1 Time-out Interval Register 0000FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0	**CNT**	CNT	**Initial counting value** GPT1 counts down from GPTIMER1_DAT. When GPT1 counts down to 0, a GPT1 interrupt will be generated.

A00C0008 GPTIMER2_CON GPT2 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN	MODE														
Type	RW	RW														
Reset	0	0														

Bit(s)	Mnemonic	Name	Description
15	**EN**	EN	**Controls GPT2 to start counting or to stop** 0: Disable GPT2 1: Enable GPT2
14	**MODE**	MODE	**Controls GPT2 to count repeatedly (in a loop) or just one-shot** 0: One-shot mode is selected. 1: Auto-repeat mode is selected.

A00C000C GPTIMER2_DAT GPT2 Time-out Interval Register 0000FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0	**CNT**	CNT	**Initial counting value** GPT2 counts down from GPTIMER2_DAT. When GPT2 counts down to 0, a GPT2 interrupt will be generated.

A00C0010 GPTIMER_STA GPT Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															GPT2	GPT1
Type															RC	RC
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**GPT2**	GPT2	**GP timer time-out status** Each flag is set when the corresponding timer count-down is completed. Can be cleared when the CPU reads the status register.
0	**GPT1**	GPT1	**GP timer time-out status** Each flag is set when the corresponding timer count-down is completed. Can be cleared when the CPU reads the status register.

A00C0014 GPTIMER1_PRESCALER GPT1 Prescaler Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRESCALER		
Type														RW		
Reset														1	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**PRESCALER**	PRESCALER	**Controls the counting clock for GP timer 1** 0: 16,384Hz 1: 8,192Hz 2: 4,096Hz 3: 2,048Hz 4: 1,024Hz 5: 512Hz 6: 256Hz 7: 128Hz

A00C0018 GPTIMER2_PRESCALER GPT2 Prescaler Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRESCALER		
Type														RW		
Reset														1	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**PRESCALER**	PRESCALER	**Controls the counting clock for GP timer 2** 0: 16,384Hz 1: 8,192Hz 2: 4,096Hz 3: 2,048Hz 4: 1,024Hz 5: 512Hz 6: 256Hz 7: 128Hz

A00C001C GPTIMER3_CON GPT3 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																EN
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**EN**	EN	**Controls GPT3 to start counting or to stop** 0: Disable GPT3 1: Enable GPT3

A00C0020 GPTIMER3_DAT GPT3 Time-out Interval Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**CNT**	CNT	If EN = 1, GPT3 will be a free running timer. This register records the GPT3 value. If EN = 0, this register will be cleared to 0.

A00C0024 GPTIMER3_PRESCALER GPT3 Prescaler Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRESCALER		
Type														RW		
Reset														1	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**PRESCALER**	PRESCALER	**Controls the counting clock for GP timer 3** 0: 16,384Hz 1: 8,192Hz 2: 4,096Hz 3: 2,048Hz 4: 1,024Hz 5: 512Hz 6: 256Hz 7: 128Hz

A00C0028 GPTIMER4_CON GPT4 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															LOCK	EN
Type															RW	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**LOCK**	LOCK	**Controls GPT4 EN bit can be modified or not** If LOCK = 0, EN can be modified. If LOCK = 1, the EN value will be fixed, and the LOCK bit will always be 1 and cannot be modified until hardware reset. 0: Unlock 1: Lock
0	**EN**	EN	**Controls GPT4 to start counting or to stop** 0: Disable GPT4 1: Enable GPT4

A00C002C GPTIMER4_DAT GPT4 Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CNT[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description

Bit(s)	Mnemonic	Name	Description
31:0	**CNT**	CNT	If EN = 1, GPT4 will be a free running timer. This register records the GPT4 value. If EN = 0, this register will be cleared to 0. This register does not allow continuous read. It requires at least 1 26M clock cycle between 2 APB reads.

3.5.3 Application Note

When the GPT is in running status, GPTIMER_DAT cannot be configured. To start GPT1 or GPT2, SW should make sure the timer has finished counting for at least 3 cycles of the 32kHz clock.

3.6 MCU OSTIMER

3.6.1 Overview

Figure 33. *OS timer system view*

The OS timer is a hardware timer which specifies the OS time frame duration and generates time-out interrupts by programming the frame counter number. The OS timer has the pause mode. The user can specify pause duration period before the pause mode, and the timer will resume from the pause mode by the external wakeup sources or when the pause duration is timed out.

3.6.2 Terminology

Table 3. Abbreviations

Abbreviation	
R/W	Read/Write
RO	Read only
WO	Write only

Abbreviation	
W1C	Write 1 to clear
R/W1C	Read/Write 1 to clear
FRC	Free running counter in the system
OST	OS timer

3.6.3 Introduction to Wakeup Source

The OS timer only accepts level trigger wakeup source. The wakeup sources are all treated as asynchronous input (will be changed) and will be synchronized by OST clock in OST wakeup source controller.

The possible wakeup sources are listed in the table below.

Table 2. Wakeup sources

No	Wakeup source
0	GPT
1	EINT
2	Timer trigger
3	KP
4	MSDC
5	ANALOG
6	DSP
7	GPI
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	DSP_ASYNC
13	GPI_ASYNC
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved

Table 3. Characteristics of wakeup sources

No	Wakeup source	Edge/Level	SW clear	Clock domain
0	GPT	Edge		F32k_CK

No	Wakeup source	Edge/Level	SW clear	Clock domain
1	EINT	Level		F32k_CK
2	Timer trigger	Level		F32k_CK
3	KP	Level		
4	MSDC	Level		
5	ANALOG	Level		
6	DSP	Level		DSP_CK
7	GPI	Level		
8	Reserved			
9	Reserved			
10	Reserved			
11	Reserved			
12	DSP_ASYNC	Level		DSP_CK
13	GPI_ASYNC	Level		MCLK_CK
14	Reserved			
15	Reserved			
16	Reserved			
17	Reserved			
18	Reserved			

Figure 4. Wakeup event and Irq_b integration diagram

wakeup_irq_b may be asserted before *wakeup_event_b* is asserted from the wakeup source peripheral to ensure the software will enter wakeup ISR after the pause command is set.

Recommended software programming sequence:

1. I-BIT is set when a pause command is executed.
2. Set up IRQ mask registers to select IRQ wakeup sources
3. Pause criteria are met.
4. Set up pause command.
5. Confirm the pause command is executed at OST, and check if the pause command is pending or not.
6. Set up processor in request for interrupt state.

7. The processor clock will be off if there is no interrupt.
8. Check if the pause request command is completed (after the processor clock is active or resumes).
9. Clear I-BIT.

3.6.4 Register Definition

Address	Name	Width	Register function
0XA01F0000	**OST_CON**	32	**OS timer control register** Only updated when OST_CMD.OST_WR is set and OST_STA.WR_RDY is high.
0XA01F0004	**OST_CMD**	32	**OS timer command** Only valid when the write data BIT31 to BIT16 is 0x1153.
0XA01F0008	**OST_STA**	32	**OS timer status**
0XA01F000C	**OST_FRM**	32	**OS timer frame duration** Specifies OS timer frame duration by micron second resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before OS timer is enabled. The hardware will set up OST_ISR[0] periodically at frame duration period when the OS timer is enabled.
0XA01F0010	**OST_FRM_F32K**	32	**OS timer frame duration by 32K clock** Specifies OS timer frame duration with 30.5176 micron second (32kHz) resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before the OS timer is enabled. OST_FRAM_F32K*30.5176us should be less than OST_FRM_NUM*OST_FRM*1us - 30.5176us. Set up OST_FRM_NUM if the frame duration is shorter than system settling time.
0XA01F0014	**OST_UFN**	32	**OS timer un-alignment frame number** Specifies OS timer un-alignment event frame number count. This register value is updated to OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time Un-alignment frame number. The hardware will count down the OST_UFN counter at frame time-out and stop the count-down when the current value is 0. When the OS timer is in the pause mode, this counter is still active and wakes up the OS timer when OST_UFN becomes 0 at frame time-out boundary. The software should enable the OS timer pause mode when OST_UFN is larger than 1, or the pause command will be ignored by the hardware.
0XA01F0018	**OST_AFN**	32	**OS timer alignment frame number** Specifies OS timer alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time

Address	Name	Width	Register function
			alignment frame number. The hardware will count down the OST_AFN counter at frame time-out and stop the count-down when the current value is 0. The hardware will set OST_ISR[1] when OST_AFN is 1 or 0 at frame time-out, and the OS timer is in normal mode. OST_AFN is current hardware AFN down counter value, and the software cannot read this register directly. The software can only read the current frame number through the OST_AFN_R register, and there is synchronization latency from OST_AFN to OST_AFN_R.
0XA01F001C	**OST_AFN_DLY**	32	**OS timer alignment frame delay number count** Specifies the OS timer alignment event frame delay count due to OS timer pause mode. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.
0XA01F0020	**OST_UFN_R**	32	**Current OS timer un-alignment frame number** Specifies the OS timer current un-alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.
0XA01F0024	**OST_AFN_R**	32	**Current OS timer alignment frame number** Specifies the OS timer current alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.
0XA01F0030	**OST_INT_MASK**	32	**OS timer interrupt mask** Specifies the OS timer interrupt mask control.
0XA01F0040	**OST_ISR**	32	**OS timer interrupt status** Specifies the OS timer interrupt status. The software has to write 1 at the corresponding bit to clear the interrupt status bit. OSR_ISR[2-0] are also cleared when the OSR_WR command is executed and AFN, UFN are updated.
0XA01F0050	**OST_EVENT_MASK**	32	**OS timer event mask** Specifies which wakeup event will be masked to wake up the OS timer in the OS timer pause mode.
0XA01F0054	**OST_WAKEUP_STA**	32	**OS timer event wakeup status**
0XA01F0060	**OST_DBG_WAKEUP**	32	**OS timer debug wakeup**

0XA01F0000 **OS Timer Control Register** **OST_CON**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name														OST_DBG	UFN_DOWN	EN
Type														RW	RW	RW
Reset														0	1	0

Overview This registers is only updated when OST_CMD.OST_WR is set and OST_STA.WR_RDY is high.

Bit(s)	Name	Description
2	OST_DBG	**Enables OST wakeup debugging function** 0: Disable 1: Enable
1	UFN_DOWN	**Enables OST_UFN count-down** 0: Disable OST_UFN count-down 1: Enable OST_UFN count-down
0	EN	**Enables OS timer** 0: OS timer is disabled. Then all internal timers are stopped. 1: OS Timer is enabled. The software has to ensure OST_AFN, OST_UFN, OST_FRAM are configured before enabling the OS timer.

0XA01F0004 **OS Timer Command** **OST_CMD**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_KEY															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_CON_WR	OST_AFN_WR	OST_UFN_WR											OST_WR	OST_RD	PAUSE_STR
Type	RW	RW	RW											WO	WO	WO
Reset	0	0	0											0	0	0

Overview: The command is only valid when the write data BIT31 to BIT16 is 0x1153.

Bit(s)	Name	Description
31:16	OST_KEY	
15	OST_CON_WR	Updates OST_CON when the OST_WR command is active.
14	OST_AFN_WR	Updates OST_AFN when the OST_WR command is active.
13	OST_UFN_WR	Updates OST_UFN when the OST_WR command is active.
2	OST_WR	Write 1 to this bit to update the bus clock domain OS timer configuration into the OST SYSCLK domain
1	OST_RD	Write 1 to this bit to update the current OS timer status to the bus clock domain
0	PAUSE_STR	Write 1 to this bit to enable the OS timer pause function. The command will be ignored if the current OST UFN is less than 2. The software has to ensure OST_CMD.OST_WR is completed before the next software pause sequence.

0XA01F0008 **OS Timer Status** **OST_STA**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**CPU_SLEEP**									**AFN_DLY_OVER**		**PAUSE_REQ**			**CMD_CPL**	**READY**
Type	**RO**									**RO**		**RO**			**RO**	**RO**
Reset	**0**									**0**		**0**	**0**		**1**	**0**

Figure 34. Pause command complete and pause request state

Bit(s)	Name	Description
15	CPU_SLEEP	**The processor is in the sleep mode. (for debugging)** 0: Active 1: Sleep
6	AFN_DLY_OVER	**AFN_DLY counter overflows. This bit is cleared when AFN is updated.** 0: Does not overflow 1: Overflow
4:3	PAUSE_REQ	**An OS timer pause request is pending. A pause command will be completed when the pause command is set.** 1. Processor is in the sleep mode (processor clock is off), UFN ≥ 2 and no wakeup sources 2. Any wakeup sources are sensed after the pause command is set. 3. UFN < 2 00: The last pause command request is not completed yet (CP15 is not enable and no wakeup source). 01: The last pause command request is completed with OST pause mode being active. 10: The last pause command request is completed with wakeup sources. 11: The last pause command request is completed with UFN < 2.
1	CMD_CPL	**OST command is completed. It takes several clocks from OST_CMD being updated to command being active.** 0: OST command is not completed. 1: OST command is completed.
0	READY	**OS timer status** 0: OST is in pause mode. 1: OST is in normal mode.

0XA01F000C **OS Timer Frame Duration** **OST_FRM**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				OST_FRM												
Type				RW												
Reset				1	0	0	0	0	0	1	0	0	1	0	0	1

Overview: This register specifies the OS timer frame duration by micron second resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before the OS timer is enabled. The hardware will set up OST_ISR[0] periodically at frame duration period when the OS timer is enabled.

Bit(s)	Name	Description
12:0	OST_FRM	

0XA01F0010 OS Timer Frame Duration by 32K Clock OST_FRM_F32K

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN								OST_FRM_F32K							
Type	RW								RW							
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer frame duration with 30.5176 micron second (32kHz) resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set up before the OS timer is enabled. OST_FRAM_F32K*30.5176us should be less than OST_FRM_NUM*OST_FRM*1us - 30.5176us. Set up OST_FRM_NUM if the frame duration is shorter than the system settling time.

Bit(s)	Name	Description
15:12	OST_UFN	
11:0	OST_FRM_F32K	

0XA01F0014 OS Timer Un-alignment Frame Number OST_UFN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_UFN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN[15:0]															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer un-alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time un-alignment frame number. The hardware will count down the OST_UFN counter at frame time-out and stop the count-down when the current value is 0. When the OS timer is in the pause mode, this counter is still active and wakes up the OS timer when OST_UFN becomes 0 at frame time-out boundary. The software should enable the OS timer pause mode when OST_UFN is larger than 1, or the pause command will be ignored by hardware.

Bit(s)	Name	Description
31:0	OST_UFN	

0XA01F0018 OS Timer Alignment Frame Number OST_AFN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time alignment frame number. The hardware will count down the OST_AFN counter at frame time-out and stop the count-down when the current value is 0. The hardware will set up OST_ISR[1] when OST_AFN is 1 or 0 at frame time-out and the OS timer is in the normal mode. OST_AFN is the current hardware AFN down counter value, and the software cannot read this register directly. The software can only read the current frame number through the OST_AFN_R register, and there is synchronization latency from OST_AFN to OST_AFN_R.

Bit(s)	Name	Description
31:0	OST_AFN	

0XA01F001C OS Timer Alignment Frame Delay Number Count OST_AFN_DLY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN_DLY[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN_DLY[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS Timer Alignment event frame delay count due to OS timer pause mode. Software has to use OST_CMD.OST_RD command to update this register value, or the value maybe out of date.

Bit(s)	Name	Description
31:0	OST_AFN_DLY	

0XA01F0020 Current OS Timer Un-alignment Frame Number OST_UFN_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_UFN_R[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN_R[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer current un-alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.

Bit(s)	Name	Description
31:0	OST_UFN_R	

0XA01F0024 Current OS Timer Alignment Frame Number OST_AFN_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN_R[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN_R[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer current alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.

Bit(s)	Name	Description
31:0	OST_AFN_R	

0XA01F0030 OS Timer Interrupt Mask OST_INT_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												OST_INT_MASK				
Type												RW				
Reset												1	1	1	1	1

Overview: This register specifies the OS timer interrupt mask control.

Bit(s)	Name	Description
4:0	OST_INT_MASK	0: Mask OS timer frame time-out interrupt 1: Mask OS timer alignment frame time-out interrupt 2: Mask OS timer un-alignment frame time-out interrupt 3: Mask OS timer pause abort interrupt 4: Mask OS timer pause interrupt

0XA01F0040 OS Timer Interrupt Status OST_ISR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												OST_ISR				
Type												W1C				
Reset												0	0	0	0	0

Overview: This register specifies the OS timer interrupt status. The software has to write 1 to the corresponding bit to clear the interrupt status bit. OSR_ISR[2-0] are also cleared when the OSR_WR command is executed and AFN, UFN are updated.

Bit(s)	Name	Description
4:0	OST_ISR	0: OS timer frame time-out interrupt status 1: OS timer alignment frame time-out interrupt status 2: OS timer un-alignment frame time-out interrupt status 3: OS timer pause abort interrupt status 4: OS timer pause interrupt status

0XA01F0050 OS Timer Event Mask OST_EVENT_MASK

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														OST_EVENT_MASK [18:16]		
Type														RW		
Reset														0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	OST_EVENT_MASK[15:0]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview: This register specifies which wakeup event will be masked to wake up the OS timer in the OS timer pause mode.

Bit(s)	Name	Description
18:0	OST_EVENT_MASK	

0XA01F0054 **OS Timer Event Wakeup Status** **OST_WAKEUP_STA**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														OST_WAKEUP_STA[18:16]		
Type														RO		
Reset														0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_WAKEUP_STA[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
18:0	OST_WAKEUP_STA	

0XA01F0060 **OS Timer Debug Wakeup** **OST_DBG_WAKEOSUP**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CIRQ_MASK_EN													OST_DBG_WAKEUP[18:16]		
Type	RW															
Reset	0													0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_DBG_WAKEUP[15:0]															
Type																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	CIRQ_MASK_EN	0: Disable cirq mask function 1: Enable cirq mask function
18:0	OST_DBG_WAKEUP	Controls wakeup status in debug timing event1

Figure 35. Debug wakeup events

3.7 UART

3.7.1 General Description

The baseband chipset houses two UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external dev ices.
The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.
In common with the M16550A, the UART supports word lengths **from five to eight bits**, **an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided. The UART also includes two DMA handshake lines, used to indicate when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.
Note: The UART has been designed so that all internal operations are synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and the industry standard 16550A device, which means that the core is not clock for clock identical to the original device.
After a hardware reset, the UART is in M16C450 mode. Its FIFOs can be enabled and the UART can then enter M16550A mode. The UART adds further functionality beyond M16550A mode. Each of the extended functions can be selected individually under software control.
The UART provides more powerful enhancements than the industry-standard 16550:

- Hardware flow control. This feature is very useful when the ISR latency is hard to predict and control in the embedded applications. The MCU is relieved of having to fetch the received data within a fixed amount of time.

Note: In order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If

EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The Enhanced Mode bit ensures that the UART is backward compatible with software that has been written for 16C450 and 16550A devices.

Figure 36 shows the block diagram of the UART device.

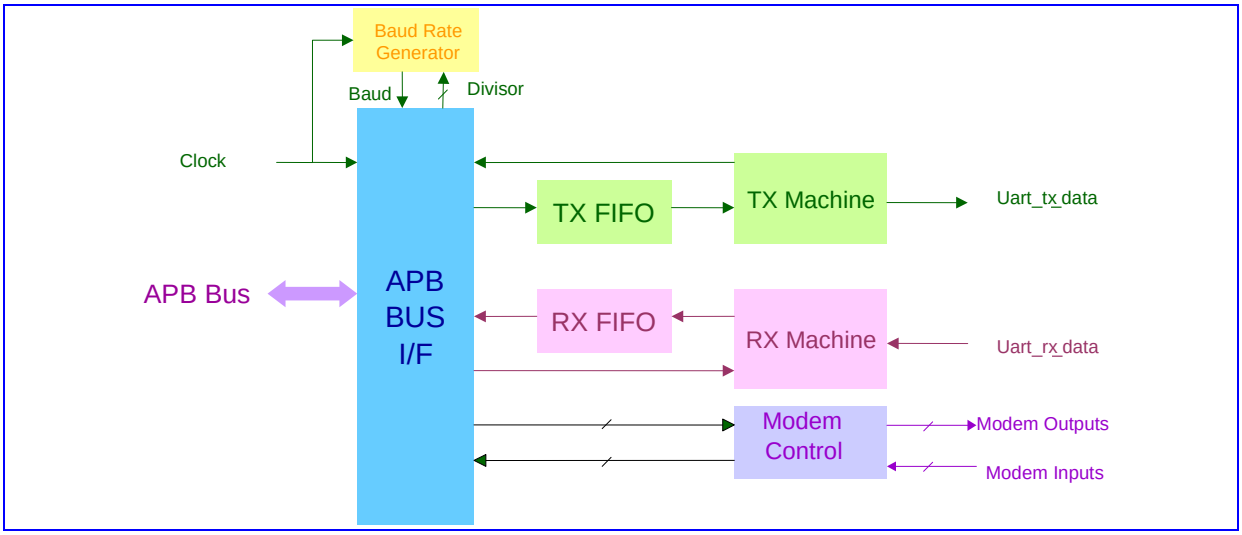


Figure 36 ***Block Diagram of UART***

3.7.2 Register Definitions

Module name: UART1 Base address: (+A0080000h)

Address	Name	Width	Register function
A0080000	**RBR**	8	RX buffer register
A0080000	**THR**	8	TX holding register
A0080000	**DLL**	8	Divisor latch (LS) register
A0080004	**IER**	8	**Interrupt enable register** By storing “1” to a specific bit position, the interrupt associated with that bit will be enabled. Otherwise, the interrupt will be disabled. IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.
A0080004	**DLM**	8	Divisor latch (MS) register
A0080008	**IIR**	8	**Interrupt identification register** Identifies if there are pending interrupts. ID4 and ID3 are presented only when EFR[4] = 1.
A0080008	**FCR**	8	**FIFO control register** FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh
A0080008	**EFR**	8	**Enhanced feature register** *Note: Only when LCR='hBF*
A008000C	**LCR**	8	**Line control register** Determines characteristics of serial communication

Address	Name	Width	Register function
			signals.
A0080010	**MCR**	8	**Modem control register** Controls interface signals of the UART. MCR[4:0] are modified when LCR[7] = 0. MCR[7:6] are modified when LCR[7] = 0 & EFR[4] = 1.
A0080010	**XON1**	8	**XON1 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A0080014	**LSR**	8	**Line status register** Modified when LCR[7] = 0.
A0080014	**XON2**	8	**XON2 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A0080018	**MSR**	8	**Modem status register** *Note: After a reset, bit4-bit5 will be inputs.* A modem status interrupt can be cleared by writing 0 or set by writing 1 to this register. bit0-bit1 can be written. Modified when LCR[7] = 0.
A0080018	**XOFF1**	8	**XOFF1 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A008001C	**SCR**	8	**Scratch register** A general purpose read/write register. After reset, its value will be un-defined. Modified when LCR[7] = 0.
A008001C	**XOFF2**	8	**XOFF2 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A0080020	**AUTOBAUD_EN**	8	Auto baud detect enable register
A0080024	**HIGHSPEED**	8	High speed mode register
A0080028	**SAMPLE_COUNT**	8	**Sample counter register** When HIGHSPEED = 3, sample_count will be the threshold value for UART sample counter (sample_num). Counts from 0 to sample_count.
A008002C	**SAMPLE_POINT**	8	**Sample point register** When HIGHSPEED = 3, UART will obtain the input data when sample_count = sample_num, e.g. system clock = 13MHz, baud rate = 921600. Therefore, sample_count = 13MHz/921600 = 14, and sample point = 13/2 = 6 (sampling the central point to decrease the inaccuracy) SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.
A0080030	**AUTOBAUD_REG**	8	Auto baud monitor register
A0080034	**RATEFIX_AD**	8	**Clock rate fix register** UART operation clock rate is fixed for low power

Address	Name	Width	Register function
A0080038	**AUTOBAUDSAMPLE**	8	**Auto baud sample register** Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13. *Note: When AUTO_SEL = 1, autobaudsample should be 15.*
A008003C	**GUARD**	8	Guard time added register
A0080040	**ESCAPE_DAT**	8	Escape char register
A0080044	**ESCAPE_EN**	8	Escape enable register
A0080048	**SLEEP_EN**	8	Sleep enable register
A008004C	**DMA_EN**	8	DMA enable register
A0080050	**RXTRI_AD**	8	Rx trigger register
A0080054	**FRACDIV_L**	8	Fractional divider LSB register
A0080058	**FRACDIV_M**	8	Fractional divider MSB register
A008005C	**FCR_RD**	8	FIFO control read out register

A0080000 RBR RX Buffer Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									RBR							
Type									RO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**RBR**	RBR	**RX buffer register** A read-only register. The received data can be read by accessing this register. Modified when LCR[7] = 0.

A0080000 THR TX Holding Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									THR							
Type									WO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**THR**	THR	**TX holding register** A write-only register. The data to be transmitted are written to this register and sent to the PC via serial communication. Modified when LCR[7] = 0.

A0080000 DLL Divisor Latch (LS) Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									DLL							

Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
7:0	**DLL**	DLL	Refer to the descriptions of DLM.

A0080004 IER Interrupt Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									**CTSI**	**RTSI**	**XOFFI**	**VFF_FC_EN**	**EDSSI**	**ELSI**	**ETBEI**	**ERBFI**
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	0	0	0	0	0	0	0

Overview: By storing “1” to a specific bit position, the interrupt associated with that bit will be enabled. Otherwise, the interrupt will be disabled. IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.

Bit(s)	Mnemonic	Name	Description
7	**CTSI**	CTSI	**Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated when a rising edge is detected on the CTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the CTS modem control line.
6	**RTSI**	RTSI	**Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated when a rising edge is detected on the RTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the RTS modem control line.
5	**XOFFI**	XOFFI	**Masks an interrupt that is generated when an XOFF character is received.** *Note: This interrupt is only enabled when software flow control is enabled.* 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.
4	**VFF_FC_EN**	VFF_FC_EN	**Enables flow control triggered by RX FIFO full when VFF_FC_EN is set.**
3	**EDSSI**	EDSSI	**When set to 1, an interrupt will be generated if DDSR or DCTS (MSR[1:0]) becomes set.** 0: No interrupt is generated if DDSR or DCTS (MSR[1:0]) becomes set. 1: An interrupt is generated if DDSR or DCTS (MSR[1:0]) becomes set.
2	**ELSI**	ELSI	**When set to 1, an interrupt will be generated if BI, FE, PE or OE (LSR[4:1]) becomes set.** 0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.

Bit(s)	Mnemonic	Name	Description
			1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
1	**ETBEI**	ETBEI	**When set to 1, an interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.** 0: No interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level. 1: An interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.
0	**ERBFI**	ERBFI	**When set to 1, an interrupt will be generated if the RX buffer contains data.** 0: No interrupt will be generated if the RX buffer contains data. 1: An interrupt will be generated if the RX buffer contains data.

A0080004 DLM Divisor Latch (MS) Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									DLM							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**DLM**	DLM	DLL & DLM can only be updated when DLAB (LCR[7]) is set to 1. Division by 1 will generate a BAUD signal that is constantly high. DLL & DLM setting formula is {DLH,DLL} = (system clock frequency/baud_pulse/baud_rate). ▪ When RATE_FIX(RATEFIX_AD[0]) = 0, system clock frequency = 52MHz. ▪ When RATE_FIX(RATEFIX_AD[0]) = 1 and RATE_FIX(RATEFIX_AD[2]) = 0, system clock frequency = 26MHz. ▪ When RATE_FIX(RATEFIX_AD[0]) = 1 and RATE_FIX(RATEFIX_AD[2]) = 1, system clock frequency = 13MHz. For baud_pulse value, refer to the HIGH_SPEED (offset = 24H) register. For example: When at 52MHz, default speed mode and 115200 baud rate, {DLH,DLL} = 52MHz/16/115,200 = 28.

A0080008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									FIFOE		ID					
Type									RO		RO					
Reset									0	0	0	0	0	0	0	1

Overview: Identifies if there are pending interrupts. ID4 and ID3 are presented only when EFR[4] = 1.

Bit(s)	Mnemonic	Name	Description
7:6	**FIFOE**	FIFOE	**FIFO enable read out**
5:0	**ID**	ID	**IIR[5:0] Priority level interrupt source** 000001 - No interrupt pending 000110 1 Line status interrupt:

Bit(s)	Mnemonic	Name	Description
			BI, FE, PE or OE set in LSR. (Under IER[2]=1) 000100 2 RX data received: RX data received or RX trigger level reached. (Under IER[0]=1) 001100 2 RX data time-out: Time-out on character in RX FIFO. (Under IER[0]=1) 000010 3 TX holding register empty: TX holding register empty or TX FIFO trigger level reached. (Under IER[1]=1) 000000 4 modem status change: DDSR or DCTS set in MSR. (Under IER[3]=1) 010000 5 Software flow control: XOFF character received. (Under IER[5]=1) 100000 6 Hardware flow control: CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1) **Line status interrupt:** A RX line status interrupt (IIR[5:0] = 000110b) will be generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the line status register. **RX data received interrupt:** A RX received interrupt (IER[5:0] = 000100b) will be generated if EFRBI (IER[0]) is set and either RX data are placed in the RX buffer register or the RX trigger level is reached. The interrupt is cleared by reading the RX buffer register or the RX FIFO (if enabled). **RX data time-out interrupt:** When the virtual FIFO DMA mode is disabled, RX data time-out interrupt will be generated if all of the following conditions are applied: 1. FIFO contains at least one character. 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits). 3. The most recent CPU read of the FIFO is longer than four character periods ago. The time-out timer/interrupt is restarted/cleared upon receipt of a new byte from the RX shift register or upon CPU read access to RBR. The RX data time-out interrupt is enabled by setting EFRBI (IER[0]) to 1. When the virtual FIFO DMA mode is enabled, RX data time-out Interrupt will be generated if all of the following conditions are applied: 1. FIFO is empty. 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits). 3. The most recent CPU read of the FIFO is longer than four character periods ago. The time-out timer/interrupt is restarted/cleared upon receipt of a new byte from the RX shift register or upon CPU read access to DMA_EN. *Note: No matter the virtual FIFO DMA mode is enable or not, the time-out timer/interrupt will be auto restarted/cleared by setting TO_CNT_AUTORST(DMA_EN[2]) to 1.* **TX holding register empty interrupt:** A TX holding register empty interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX holding register or if FIFOs are enabled, and the TX FIFO becomes empty. The interrupt is cleared by writing to the TX holding register or TX FIFO if FIFO is enabled. **Modem status change interrupt:** A modem status change Interrupt (IIR[5:0] = 000000b) will be generated if EDSSI (IER[3]) is set and either DDSR or DCTS (MSR[1:0]) becomes set. The interrupt is cleared by reading the modem status register.

Bit(s)	Mnemonic	Name	Description
			Software flow control interrupt: A software flow control interrupt (IIR[5:0] = 010000b) will be generated if the software flow control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the interrupt identification register. **Hardware flow control interrupt:** A hardware flow control interrupt (IER[5:0] = 100000b) will be generated if the hardware flow control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS modem control line. The interrupt is cleared by reading the interrupt identification register.

A0080008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RFTL1_RFTL0**		**TFTL1_TFTL0**			**CLRT**	**CLRR**	**FIFOE**
Type									WO		WO			WO	WO	WO
Reset									0	0	0	0		0	0	0

Overview: FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh

Bit(s)	Mnemonic	Name	Description
7:6	**RFTL1_RFTL0**	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	**TFTL1_TFTL0**	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 16 bytes. 0: 1 1: 4 2: 8 3: 14 (FIFOSIZE - 2)
2	**CLRT**	CLRT	**Clears Transmit FIFO** This bit is self-clearing. 0: Leave TX FIFO intact. 1: Clear all the bytes in the TX FIFO.
1	**CLRR**	CLRR	**Clears Receive FIFO** This bit is self-clearing. 0: Leave RX FIFO intact. 1: Clear all the bytes in the RX FIFO.
0	**FIFOE**	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect. 0: Disable both the RX and TX FIFOs. 1: Enable both the RX and TX FIFOs.

A0080008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**AUTO**	**AUTO**	**LOWE**	**ENAB**	**SW_FLOW_CONT**			

									_CTS	_RTS	R_RATEFIX	LE_E				
Type									RW	RW	RW	RW	RW			
Reset									0	0	0	0	0	0	0	0

Overview: Only when LCR='hBF

Bit(s)	Mnemonic	Name	Description
7	**AUTO_CTS**	AUTO_CTS	**Enables hardware transmission flow control** 0: Disabled 1: Enabled
6	**AUTO_RTS**	AUTO_RTS	**Enables hardware reception flow control** 0: Disabled 1: Enabled
5	**LOWER_RATEFIX**	LOWER_RATEFIX	**Enable ratefix function for lower clock** 0: Disabled 1: Enabled
4	**ENABLE_E**	ENABLE_E	**Enable enhancement features** 0: Disabled 1: Enabled
3:0	**SW_FLOW_CONT**	SW_FLOW_CONT	**Software flow control bits** 00xx: No TX flow control 10xx: Transmit XON1/XOFF1 as flow control bytes 01xx: Transmit XON2/XOFF2 as flow control bytes 11xx: Transmit XON1 & XON2 and XOFF1 & XOFF2 as flow control words xx00: No RX flow control xx10: Receive XON1/XOFF1 as flow control bytes xx01: Receive XON2/XOFF2 as flow control bytes xx11: Receive XON1 & XON2 and XOFF1 & XOFF2 as flow control words. *Note: When 11xx or xx11, ESC character is not supported and ESC_EN must be 0. When xx00 or xx11, XON1, XON2, XOFF1 and XOFF2 will be received as valid data. When xx10, XON1&XOFF1 will be omitted; When xx01, XON2&XOFF2 will be omitted.*

A008000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1_WLS0	
Type									RW	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	0

Overview: Determines characteristics of serial communication signals.

Bit(s)	Mnemonic	Name	Description
7	**DLAB**	DLAB	**Divisor latch access bit** 0: RX and TX registers are read/written at Address 0, and the IER register is read/written at Address 4. 1: Divisor Latch LS is read/written at Address 0, and the Divisor Latch MS is read/written at Address 4.
6	**SB**	SB	**Set break** 0: No effect 1: SOUT signal is forced to the 0 state.
5	**SP**	SP	**Stick parity**

			0: No effect 1: The parity bit is forced to a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the parity bit will be set and checked if = 0. If EPS=0 & PEN=1, the parity bit will be set and checked if = 1.
4	**EPS**	EPS	**Selects even parity** 0: When EPS = 0, an odd number of ones is sent and checked. 1: When EPS = 1, an even number of ones is sent and checked.
3	**PEN**	PEN	**Enables parity** 0: The parity is neither transmitted nor checked. 1: The parity is transmitted and checked.
2	**STB**	STB	**Number of STOP bits** 0: One STOP bit is always added. 1: Two STOP bits are added after each character is sent, unless the character length is 5 when 1 STOP bit is added.
1:0	**WLS1_WLS0**	WLS1_WLS0	**Selects word length** 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits

A0080010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF_STATUS			LOOP			RTS	DTR
Type									RO			RW			RW	RW
Reset									0			0			0	0

Overview: Controls interface signals of the UART. MCR[4:0] are modified when LCR[7] = 0, MCR[7:6] are modified when LCR[7] = 0 & EFR[4] = 1.

Bit(s)	Mnemonic	Name	Description
7	**XOFF_STATUS**	XOFF_STATUS	**Read-only bit** 0: When an XON character is received. 1: When an XOFF character is received.
4	**LOOP**	LOOP	**Loop-back control bit** 0: No loop-back is enabled. 1: Loop-back mode is enabled.
1	**RTS**	RTS	**Controls the state of the output NRTS, even in loop mode** 0: NRTS = 1 1: NRTS = 0
0	**DTR**	DTR	**Controls the state of the output NDTR, even in loop mode** 0: NDTR = 1 1: NDTR = 0

A0080010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XON1**	XON1	XON1 character for software flow control

A0080014 LSR Line Status Register 60

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	1	1	0	0	0	0	0

Overview: Modified when LCR[7] = 0.

Bit(s)	Mnemonic	Name	Description
7	**FIFOERR**	FIFOERR	**RX FIFO error indicator** 0: No PE, FE, BI set in the RX FIFO. 1: Set to 1 when there is at least one PE, FE or BI in the RX FIFO.
6	**TEMT**	TEMT	**TX holding register (or TX FIFO) and the TX shift register are empty.** 0: Empty conditions below are not met. 1: If FIFOs are enabled, the bit will be set whenever the TX FIFO and the TX shift register are empty. If FIFOs are disabled, the bit will be set whenever TX holding register and TX shift register are empty.
5	**THRE**	THRE	**Indicates if there is room for TX holding register or TX FIFO is reduced to its trigger level.** 0: Reset whenever the contents of the TX FIFO are more than its trigger level (FIFOs are enabled), or whenever TX holding register is not empty (FIFOs are disabled). 1: Set whenever the contents of the TX FIFO are reduced to its trigger level (FIFOs are enabled), or whenever TX holding register is empty and ready to accept new data (FIFOs are disabled).
4	**BI**	BI	**Break interrupt** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). If the FIFOs are enabled, this error will be associated with a corresponding character in the FIFO and flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: The next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.
3	**FE**	FE	**Framing error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid STOP bit. If the FIFOs are enabled, the state of this bit will be revealed when the byte it refers to is the next to be read.
2	**PE**	PE	**Parity error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid parity bit. If the FIFOs are enabled, the state of this bit will be revealed when the referred byte is the next to be read.
1	**OE**	OE	**Overrun error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the RX buffer is not read by the CPU before the new data from the RX shift register

Bit(s)	Mnemonic	Name	Description
			overwrites the previous contents. If the FIFOs are enabled, an overrun error will occur when the RX FIFO is full and the RX shift register becomes full. OE will be set as soon as this happens. The character in the shift register is then overwritten, but not transferred to the FIFO.
0	**DR**	DR	**Data ready** 0: Cleared by the CPU reading the RX buffer or by reading all the FIFO bytes. 1: Set by the RX buffer becoming full or by a byte being transferred into the FIFO.

A0080014 XON2 XON2 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON2							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XON2**	XON2	XON2 character for software flow control

A0080018 MSR Modem Status Register NA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											DSR	CTS	RESERVED	RESERVED	DDSR	DCTS
Type											RW	RW	RW	RW	RW	RW
Reset											N/A	N/A	0	0	0	0

Overview: After a reset, bit4-bit5 will be inputs. A modem status interrupt can be cleared by writing 0 or set by writing 1 to this register. bit0-bit1 can be written. Modified when LCR[7] = 0.

Bit(s)	Mnemonic	Name	Description
5	**DSR**	DSR	**Data set ready** When Loop = 0, this value is the complement of the NDSR input signal. When Loop = 1, this value is equal to the DTR bit in the modem control register.
4	**CTS**	CTS	**Clear to send** When Loop = 0, this value is the complement of the NCTS input signal. When Loop = "1", this value is equal to the RTS bit in the modem control register.
1	**DDSR**	DDSR	**Delta data set ready** 0: Cleared if the state of DSR has not changed since this register is last read. 1: Set if the state of DSR has changed since this register is last read.
0	**DCTS**	DCTS	**Delta clear to send** 0: Cleared if the state of CTS has not changed since this register is last read. 1: Set if the state of CTS has changed since this register is last read.

A0080018 XOFF1 XOFF1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XOFF1**	XOFF1	XOFF1 character for software flow control

A008001C SCR Scratch Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SCR							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: A general-purpose read/write register. After reset, its value will be un-defined. Modified when LCR[7] = 0.

Bit(s)	Mnemonic	Name	Description
7:0	**SCR**	SCR	

A008001C XOFF2 XOFF2 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF2							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XOFF2**	XOFF2	XOFF2 character for software flow control

A0080020 AUTOBAUD_EN Auto Baud Detect Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															AUTO_SEL	AUTO_EN
Type															RW	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**AUTO_SEL**	AUTOBAUD_SEL	**Selects auto-baud feature** 0: Auto-baud function detects baud and transfers it to standard baud. 1: Auto-baud function detects baud (only supports baud from 300 to

Bit(s)	Mnemonic	Name	Description
			115,200; 52MHZ to auto fix is recommended).
0	**AUTO_EN**	AUTOBAUD_EN	**Auto-baud enabling signal** *Note: When AUTOBAUD_EN is active, there should not A*/a* char before the auto baud char AT/at. If A*/a* is inevitable, the autobaud will fail, and you should disable AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.* 0: Disable auto-baud function 1: Enable auto-baud function (UARTn+0024h SPEED should be set to 0.)

A0080024 HIGHSPEED High Speed Mode Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SPEED	
Type															RW	
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1:0	**SPEED**	SPEED	**UART sample counter base** 0: Based on 16*baud_pulse,baud_rate = system clock frequency/16/{DLH, DLL} 1: Based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL} 2: Based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL} 3: Based on sampe_count*baud_pulse, baud_rate = system clock frequency/(sampe_count+1)/{DLM, DLL}

A0080028 SAMPLE_COUNT Sample Counter Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLECOUNT							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: When HIGHSPEED = 3, sample_count will be the threshold value for UART sample counter (sample_num). Counts from 0 to sample_count.

Bit(s)	Mnemonic	Name	Description
7:0	**SAMPLECOUNT**	SAMPLECOUNT	Only useful when HIGHSPEED mode = 3.

A008002C SAMPLE_POINT Sample Point Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLEPOINT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Overview: When HIGHSPEED = 3, UART will obtain the input data when sample_count = sample_num, e.g. system clock = 13MHz, baud rate = 921,600. Therefore, sample_count = 13MHz/921,600 = 14, and sample point

= 13/2 = 6 (sampling the central point to decrease the inaccuracy) The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.

Bit(s)	Mnemonic	Name	Description
7:0	**SAMPLEPOINT**	SAMPLEPOINT	

A0080030 AUTOBAUD_REG Auto Baud Monitor Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BAUD_STAT				BAUD_RATE			
Type									RO				RO			
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:4	**BAUD_STAT**	BAUD_STAT	**Autobaud format** 0: Autobaud is detecting. 1: AT_7N1 2: AT_7O1 3: AT_7E1 4: AT_8N1 5: AT_8O1 6: AT_8E1 7: at_7N1 8: at_7E1 9: at_7O1 10: at_8N1 11: at_8E1 12: at_8O1 13: Autobaud detection fails.
3:0	**BAUD_RATE**	BAUD_RATE	**Autobaud baud rate** 0: 115,200 1: 57,600 2: 38,400 3: 19,200 4: 9,600 5: 4,800 6: 2,400 7: 1,200 8: 300 9: 110

A0080034 RATEFIX_AD Clock Rate Fix Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														FREQ_SEL	AUTOBAUD_RATE_FIX	RATE_FIX
Type														RW	RW	RW
Reset														0	0	0

Overview: UART operation clock rate is fixed for low power.

Bit(s)	Mnemonic	Name	Description
2	**FREQ_SEL**	FREQ_SEL	0: Select 26MHz as system clock

Bit(s)	Mnemonic	Name	Description
			1: Select 13MHz as system clock
1	**AUTOBAUD_RATE_FIX**	AUTOBAUD_RATE_FIX	0: Use 52MHz as system clock for UART auto baud detection 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART auto baud detection
0	**RATE_FIX**	RATE_FIX	0: Use 52MHz as system clock for UART TX/RX 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART TX/RX

A0080038 AUTOBAUDSAMPLE Auto Baud Sample Register 0D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											AUTOBAUDSAMPLE					
Type											RW					
Reset											0	0	1	1	0	1

Overview: Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6. When system clock = 26MHz, autobaudsample = 13. Note that when AUTO_SEL = 1, autobaudsample should be 15.

Bit(s)	Mnemonic	Name	Description
5:0	**AUTOBAUDSAMPLE**	AUTOBAUDSAMPLE	**For standard baud rate detection:** For system clk 52m, set to 'd 27; for system clk 26m, set to 'd 13; for system clk 13m, set to 'd 6. **For non-standard baud rate detection, set to 'd15.**

A008003C GUARD Guard Time Added Register 0F

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT			
Type												RW	RW			
Reset												0	1	1	1	1

Bit(s)	Mnemonic	Name	Description
4	**GUARD_EN**	GUARD_EN	**Guard interval added enabling signal** 0: No guard interval added. 1: Add guard interval after stop bit.
3:0	**GUARD_CNT**	GUARD_CNT	**Guard interval count value** Guard interval = [1/(system clock/div_step/div)]*GUARD_CNT.

A0080040 ESCAPE_DAT Escape Char Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ESCAPE_DAT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
7:0	**ESCAPE_DAT**	ESCAPE_DAT	**Escape character added before software flow control data and escape character** i.e. if TX data are xon (31h), with esc_en =1, UART will transmit data as esc + CEh (~xon).

A0080044 ESCAPE_EN Escape Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**ESC_EN**	ESC_EN	**Adds escape character in transmitter and removes escape character in receiver by UART** 0: Does not deal with the escape character 1: Add escape character in transmitter and remove escape character in receiver

A0080048 SLEEP_EN Sleep Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**SLEEP_EN**	SLEEP_EN	**For sleep mode issue** 0: Does not deal with sleep mode indicate signal 1: Activate hardware flow control or software control according to software initial setting when the chip enters sleep mode. Release hardware flow when the chip wakes up. However, for software control, UART sends xon when awaken and when FIFO does not reach threshold level.

A008004C DMA_EN DMA Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													LSR_SEL	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Mnemonic	Name	Description
3	**LSR_SEL**	LSR_SEL	**LSR selection just for FIFO enable mode** Accordingly for FIFO disable mode, LSR will be updated when RX received new data. 0: LSR stores the current and past errors until CPU reads LSR clear in FIFO mode. 1: LSR is updated when RX receives new data in FIFO mode.
2	**TO_CNT_AUTORST**	TO_CNT_AUTORST	**Time-out counter auto reset register** 0: After RX time-out happens, SW will reset the interrupt by reading UART 0x4C. 1: The time-out counter will be auto reset. Set this register when new DMA is used.

Bit(s)	Mnemonic	Name	Description
1	**TX_DMA_EN**	TX_DMA_EN	**TX_DMA mechanism enabling signal** 0: Does not use DMA in TX. 1: Use DMA in TX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt for DMA.
0	**RX_DMA_EN**	RX_DMA_EN	**RX_DMA mechanism enabling signal** 0: Does not use DMA in RX 1: Use DMA in RX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt

A0080050 RXTRI_AD Rx Trigger Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												RXTRIG				
Type												RW				
Reset												0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
4:0	**RXTRIG**	RXTRIG	**When {rtm,rtl}=2'b11, the Rx FIFO threshold will be Rxtrig.** The value is suggested to be smaller than half the RX FIFO size, which is 24 bytes.

A0080054 FRACDIV_L Fractional Divider LSB Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**FRACDIV_L**	FRACDIV_L	Adds sampling count (+1) from state data 7 to data 0 in order to contribute fractional divisor

A0080058 FRACDIV_M Fractional Divider MSB Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															FRACDIV_M	
Type															RW	
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1:0	**FRACDIV_M**	FRACDIV_M	Adds sampling count when in state stop to parity in order to contribute fractional divisor

A008005C FCR_RD FIFO Control Read Out Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									RO		RO			RO	RO	RO
Reset									0	0	0	0		0	0	0

Bit(s)	Mnemonic	Name	Description
7:6	**RFTL1_RFTL0**	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 24 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	**TFTL1_TFTL0**	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 16 bytes. 0: 1 1: 4 2: 8 3: 14 (FIFOSIZE - 2)
2	**CLRT**	CLRT	**Clears Transmit FIFO.** This bit is self-clearing. 0: Leave TX FIFO intact. 1: Clear all the bytes in the TX FIFO.
1	**CLRR**	CLRR	**Clears Receive FIFO** This bit is self-clearing. 0: Leave RX FIFO intact. 1: Clear all the bytes in the RX FIFO.
0	**FIFOE**	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect. 0: Disable both the RX and TX FIFOs. 1: Enable both the RX and TX FIFOs.

Module name: UART2 Base address: (+A0090000h)

Address	Name	Width	Register function
A0090000	**RBR**	8	RX buffer register
A0090000	**THR**	8	TX holding register
A0090000	**DLL**	8	Divisor latch (LS) register
A0090004	**IER**	8	**Interrupt enable register** By storing “1” to a specific bit position, the interrupt associated with that bit will be enabled. Otherwise, the interrupt will be disabled. IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.
A0090004	**DLM**	8	Divisor latch (MS) register
A0090008	**IIR**	8	**Interrupt identification register** Identifies if there are pending interrupts. ID4 and ID3 are presented only when EFR[4] = 1.
A0090008	**FCR**	8	**FIFO control register** FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh
A0090008	**EFR**	8	**Enhanced feature register** *Note: Only when LCR='hBF*
A009000C	**LCR**	8	**Line control register** Determines characteristics of serial communication signals.
A0090010	**MCR**	8	**Modem control register** Controls interface signals of the UART. MCR[4:0] are modified when LCR[7] = 0. MCR[7:6] are modified when LCR[7] = 0 & EFR[4] = 1.
A0090010	**XON1**	8	**XON1 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A0090014	**LSR**	8	**Line status register** Modified when LCR[7] = 0.
A0090014	**XON2**	8	**XON2 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A0090018	**MSR**	8	**Modem status register** *Note: After a reset, bit4-bit5 will be inputs.* A modem status interrupt can be cleared by writing 0 or set by writing 1 to this register. bit0-bit1 can be written. Modified when LCR[7] = 0.
A0090018	**XOFF1**	8	**XOFF1 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A009001C	**SCR**	8	**Scratch register** A general purpose read/write register. After reset, its

Address	Name	Width	Register function
			value will be un-defined. Modified when LCR[7] = 0.
A009001C	**XOFF2**	8	**XOFF2 char register** *Note: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR=BFh.*
A0090020	**AUTOBAUD_EN**	8	Auto baud detect enable register
A0090024	**HIGHSPEED**	8	High speed mode register
A0090028	**SAMPLE_COUNT**	8	**Sample counter register** When HIGHSPEED = 3, sample_count will be the threshold value for UART sample counter (sample_num). Counts from 0 to sample_count.
A009002C	**SAMPLE_POINT**	8	**Sample point register** When HIGHSPEED = 3, UART will obtain the input data when sample_count = sample_num, e.g. system clock = 13MHz, baud rate = 921600. Therefore, sample_count = 13MHz/921600 = 14, and sample point = 13/2 = 6 (sampling the central point to decrease the inaccuracy) SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.
A0090030	**AUTOBAUD_REG**	8	Auto baud monitor register
A0090034	**RATEFIX_AD**	8	**Clock rate fix register** UART operation clock rate is fixed for low power
A0090038	**AUTOBAUDSAMPLE**	8	**Auto baud sample register** Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13. *Note: When AUTO_SEL = 1, autobaudsample should be 15.*
A009003C	**GUARD**	8	Guard time added register
A0090040	**ESCAPE_DAT**	8	Escape char register
A0090044	**ESCAPE_EN**	8	Escape enable register
A0090048	**SLEEP_EN**	8	Sleep enable register
A009004C	**DMA_EN**	8	DMA enable register
A0090050	**RXTRI_AD**	8	Rx trigger register
A0090054	**FRACDIV_L**	8	Fractional divider LSB register
A0090058	**FRACDIV_M**	8	Fractional divider MSB register
A009005C	**FCR_RD**	8	FIFO control read out register

A0090000 RBR RX Buffer Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									RBR							
Type									RO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**RBR**	RBR	**RX buffer register** A read-only register. The received data can be read by accessing this register. Modified when LCR[7] = 0.

A0090000 THR TX Holding Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									THR							
Type									WO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**THR**	THR	**TX holding register** A write-only register. The data to be transmitted are written to this register and sent to the PC via serial communication. Modified when LCR[7] = 0.

A0090000 DLL Divisor Latch (LS) Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									DLL							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
7:0	**DLL**	DLL	Refer to the descriptions of DLM.

A0090004 IER Interrupt Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									CTSI	RTSI	XOFFI	VFF_FC_EN	EDSSI	ELSI	ETBEI	ERBFI
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	0	0	0	0	0	0	0

Overview: By storing “1” to a specific bit position, the interrupt associated with that bit will be enabled. Otherwise, the interrupt will be disabled. IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.

Bit(s)	Mnemonic	Name	Description
7	**CTSI**	CTSI	**Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated when a rising edge is detected on the CTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the CTS modem control line.

Bit(s)	Mnemonic	Name	Description
6	**RTSI**	RTSI	**Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated when a rising edge is detected on the RTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the RTS modem control line.
5	**XOFFI**	XOFFI	**Masks an interrupt that is generated when an XOFF character is received.** *Note: This interrupt is only enabled when software flow control is enabled.* 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.
4	**VFF_FC_EN**	VFF_FC_EN	**Enables flow control triggered by RX FIFO full when VFF_FC_EN is set.**
3	**EDSSI**	EDSSI	**When set to 1, an interrupt will be generated if DDSR or DCTS (MSR[1:0]) becomes set.** 0: No interrupt is generated if DDSR or DCTS (MSR[1:0]) becomes set. 1: An interrupt is generated if DDSR or DCTS (MSR[1:0]) becomes set.
2	**ELSI**	ELSI	**When set to 1, an interrupt will be generated if BI, FE, PE or OE (LSR[4:1]) becomes set.** 0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
1	**ETBEI**	ETBEI	**When set to 1, an interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.** 0: No interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level. 1: An interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.
0	**ERBFI**	ERBFI	**When set to 1, an interrupt will be generated if the RX buffer contains data.** 0: No interrupt will be generated if the RX buffer contains data. 1: An interrupt will be generated if the RX buffer contains data.

A0090004 DLM Divisor Latch (MS) Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									DLM							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**DLM**	DLM	DLL & DLM can only be updated when DLAB (LCR[7]) is set to 1. Division by 1 will generate a BAUD signal that is constantly high. DLL & DLM setting formula is {DLH,DLL} = (system clock frequency/baud_pulse/baud_rate). ▪ When RATE_FIX(RATEFIX_AD[0]) = 0, system clock frequency =

Bit(s)	Mnemonic	Name	Description
			52MHz. ▪ When RATE_FIX(RATEFIX_AD[0]) = 1 and RATE_FIX(RATEFIX_AD[2]) = 0, system clock frequency = 26MHz. ▪ When RATE_FIX(RATEFIX_AD[0]) = 1 and RATE_FIX(RATEFIX_AD[2]) = 1, system clock frequency = 13MHz. For baud_pulse value, refer to the HIGH_SPEED (offset = 24H) register. For example: When at 52MHz, default speed mode and 115200 baud rate, {DLH,DLL} = 52MHz/16/115,200 = 28.

A0090008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									FIFOE		ID					
Type									RO		RO					
Reset									0	0	0	0	0	0	0	1

Overview: Identifies if there are pending interrupts. ID4 and ID3 are presented only when EFR[4] = 1.

Bit(s)	Mnemonic	Name	Description
7:6	**FIFOE**	FIFOE	**FIFO enable read out**
5:0	**ID**	ID	**IIR[5:0] Priority level interrupt source** 000001 - No interrupt pending 000110 1 Line status interrupt: BI, FE, PE or OE set in LSR. (Under IER[2]=1) 000100 2 RX data received: RX data received or RX trigger level reached. (Under IER[0]=1) 001100 2 RX data time-out: Time-out on character in RX FIFO. (Under IER[0]=1) 000010 3 TX holding register empty: TX holding register empty or TX FIFO trigger level reached. (Under IER[1]=1) 000000 4 modem status change: DDSR or DCTS set in MSR. (Under IER[3]=1) 010000 5 Software flow control: XOFF character received. (Under IER[5]=1) 100000 6 Hardware flow control: CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1) **Line status interrupt:** A RX line status interrupt (IIR[5:0] = 000110b) will be generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the line status register. **RX data received interrupt:** A RX received interrupt (IER[5:0] = 000100b) will be generated if EFRBI (IER[0]) is set and either RX data are placed in the RX buffer register or the RX trigger level is reached. The interrupt is cleared by reading the RX buffer register or the RX FIFO (if enabled). **RX data time-out interrupt:** When the virtual FIFO DMA mode is disabled, RX data time-out interrupt will be generated if all of the following conditions are applied: 4. FIFO contains at least one character. 5. The most recent character is received longer than four character periods ago (including all start, parity and stop bits). 6. The most recent CPU read of the FIFO is longer than four

Bit(s)	Mnemonic	Name	Description
			character periods ago. The time-out timer/interrupt is restarted/cleared upon receipt of a new byte from the RX shift register or upon CPU read access to RBR. The RX data time-out interrupt is enabled by setting EFRBI (IER[0]) to 1. When the virtual FIFO DMA mode is enabled, RX data time-out Interrupt will be generated if all of the following conditions are applied: 4. FIFO is empty. 5. The most recent character is received longer than four character periods ago (including all start, parity and stop bits). 6. The most recent CPU read of the FIFO is longer than four character periods ago. The time-out timer/interrupt is restarted/cleared upon receipt of a new byte from the RX shift register or upon CPU read access to DMA_EN. *Note: No matter the virtual FIFO DMA mode is enable or not, the time-out timer/interrupt will be auto restarted/cleared by setting TO_CNT_AUTORST(DMA_EN[2]) to 1.* **TX holding register empty interrupt:** A TX holding register empty interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX holding register or if FIFOs are enabled, and the TX FIFO becomes empty. The interrupt is cleared by writing to the TX holding register or TX FIFO if FIFO is enabled. **Modem status change interrupt:** A modem status change Interrupt (IIR[5:0] = 000000b) will be generated if EDSSI (IER[3]) is set and either DDSR or DCTS (MSR[1:0]) becomes set. The interrupt is cleared by reading the modem status register. **Software flow control interrupt:** A software flow control interrupt (IIR[5:0] = 010000b) will be generated if the software flow control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the interrupt identification register. **Hardware flow control interrupt:** A hardware flow control interrupt (IER[5:0] = 100000b) will be generated if the hardware flow control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS modem control line. The interrupt is cleared by reading the interrupt identification register.

A0090008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RFTL1_RFTL0**		**TFTL1_TFTL0**			**CLRT**	**CLRR**	**FIFOE**
Type									WO		WO			WO	WO	WO
Reset									0	0	0	0		0	0	0

Overview: FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh

Bit(s)	Mnemonic	Name	Description
7:6	**RFTL1_RFTL0**	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 32 bytes. 0: 1

Bit(s)	Mnemonic	Name	Description
			1: 6 2: 12 3: RXTRIG
5:4	**TFTL1_TFTL0**	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 16 bytes. 0: 1 1: 4 2: 8 3: 14 (FIFOSIZE - 2)
2	**CLRT**	CLRT	**Clears Transmit FIFO** This bit is self-clearing. 0: Leave TX FIFO intact. 1: Clear all the bytes in the TX FIFO.
1	**CLRR**	CLRR	**Clears Receive FIFO** This bit is self-clearing. 0: Leave RX FIFO intact. 1: Clear all the bytes in the RX FIFO.
0	**FIFOE**	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect. 0: Disable both the RX and TX FIFOs. 1: Enable both the RX and TX FIFOs.

A0090008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTO_CTS	AUTO_RTS	LOWER_RATEFIX	ENABLE_E	SW_FLOW_CONT			
Type									RW	RW	RW	RW	RW			
Reset									0	0	0	0	0	0	0	0

Overview: Only when LCR='hBF

Bit(s)	Mnemonic	Name	Description
7	**AUTO_CTS**	AUTO_CTS	**Enables hardware transmission flow control** 0: Disabled 1: Enabled
6	**AUTO_RTS**	AUTO_RTS	**Enables hardware reception flow control** 0: Disabled 1: Enabled
5	**LOWER_RATEFIX**	LOWER_RATEFIX	**Enable ratefix function for lower clock** 0: Disabled 1: Enabled
4	**ENABLE_E**	ENABLE_E	**Enable enhancement features** 0: Disabled 1: Enabled
3:0	**SW_FLOW_CONT**	SW_FLOW_CONT	**Software flow control bits** 00xx: No TX flow control 10xx: Transmit XON1/XOFF1 as flow control bytes 01xx: Transmit XON2/XOFF2 as flow control bytes 11xx: Transmit XON1 & XON2 and XOFF1 & XOFF2 as flow control words xx00: No RX flow control

Bit(s)	Mnemonic	Name	Description
			xx10: Receive XON1/XOFF1 as flow control bytes xx01: Receive XON2/XOFF2 as flow control bytes xx11: Receive XON1 & XON2 and XOFF1 & XOFF2 as flow control words. *Note: When 11xx or xx11, ESC character is not supported and ESC_EN must be 0. When xx00 or xx11, XON1, XON2, XOFF1 and XOFF2 will be received as valid data. When xx10, XON1&XOFF1 will be omitted; When xx01, XON2&XOFF2 will be omitted.*

A009000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1_WLS0	
Type									RW	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	0

Overview: Determines characteristics of serial communication signals.

Bit(s)	Mnemonic	Name	Description
7	**DLAB**	DLAB	**Divisor latch access bit** 0: RX and TX registers are read/written at Address 0, and the IER register is read/written at Address 4. 1: Divisor Latch LS is read/written at Address 0, and the Divisor Latch MS is read/written at Address 4.
6	**SB**	SB	**Set break** 0: No effect 1: SOUT signal is forced to the 0 state.
5	**SP**	SP	**Stick parity** 0: No effect 1: The parity bit is forced to a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the parity bit will be set and checked if = 0. If EPS=0 & PEN=1, the parity bit will be set and checked if = 1.
4	**EPS**	EPS	**Selects even parity** 0: When EPS = 0, an odd number of ones is sent and checked. 1: When EPS = 1, an even number of ones is sent and checked.
3	**PEN**	PEN	**Enables parity** 0: The parity is neither transmitted nor checked. 1: The parity is transmitted and checked.
2	**STB**	STB	**Number of STOP bits** 0: One STOP bit is always added. 1: Two STOP bits are added after each character is sent, unless the character length is 5 when 1 STOP bit is added.
1:0	**WLS1_WLS0**	WLS1_WLS0	**Selects word length** 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits

A0090010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF_STAT			LOOP			RTS	DTR

									US							
Type									RO			RW			RW	RW
Reset									0			0			0	0

Overview: Controls interface signals of the UART. MCR[4:0] are modified when LCR[7] = 0, MCR[7:6] are modified when LCR[7] = 0 & EFR[4] = 1.

Bit(s)	Mnemonic	Name	Description
7	**XOFF_STATUS**	XOFF_STATUS	**Read-only bit** 0: When an XON character is received. 1: When an XOFF character is received.
4	**LOOP**	LOOP	**Loop-back control bit** 0: No loop-back is enabled. 1: Loop-back mode is enabled.
1	**RTS**	RTS	**Controls the state of the output NRTS, even in loop mode** 0: NRTS = 1 1: NRTS = 0
0	**DTR**	DTR	**Controls the state of the output NDTR, even in loop mode** 0: NDTR = 1 1: NDTR = 0

A0090010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XON1**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XON1**	XON1	XON1 character for software flow control

A0090014 LSR Line Status Register 60

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFOERR**	**TEMT**	**THRE**	**BI**	**FE**	**PE**	**OE**	**DR**
Type									RW	RW	RW	RW	RW	RW	RW	RW
Reset									0	1	1	0	0	0	0	0

Overview: Modified when LCR[7] = 0.

Bit(s)	Mnemonic	Name	Description
7	**FIFOERR**	FIFOERR	**RX FIFO error indicator** 0: No PE, FE, BI set in the RX FIFO. 1: Set to 1 when there is at least one PE, FE or BI in the RX FIFO.
6	**TEMT**	TEMT	**TX holding register (or TX FIFO) and the TX shift register are empty.** 0: Empty conditions below are not met. 1: If FIFOs are enabled, the bit will be set whenever the TX FIFO and the TX shift register are empty. If FIFOs are disabled, the bit will be set whenever TX holding register and TX shift register are empty.

Bit(s)	Mnemonic	Name	Description
5	**THRE**	THRE	**Indicates if there is room for TX holding register or TX FIFO is reduced to its trigger level.** 0: Reset whenever the contents of the TX FIFO are more than its trigger level (FIFOs are enabled), or whenever TX holding register is not empty (FIFOs are disabled). 1: Set whenever the contents of the TX FIFO are reduced to its trigger level (FIFOs are enabled), or whenever TX holding register is empty and ready to accept new data (FIFOs are disabled).
4	**BI**	BI	**Break interrupt** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). If the FIFOs are enabled, this error will be associated with a corresponding character in the FIFO and flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: The next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.
3	**FE**	FE	**Framing error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid STOP bit. If the FIFOs are enabled, the state of this bit will be revealed when the byte it refers to is the next to be read.
2	**PE**	PE	**Parity error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid parity bit. If the FIFOs are enabled, the state of this bit will be revealed when the referred byte is the next to be read.
1	**OE**	OE	**Overrun error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the RX buffer is not read by the CPU before the new data from the RX shift register overwrites the previous contents. If the FIFOs are enabled, an overrun error will occur when the RX FIFO is full and the RX shift register becomes full. OE will be set as soon as this happens. The character in the shift register is then overwritten, but not transferred to the FIFO.
0	**DR**	DR	**Data ready** 0: Cleared by the CPU reading the RX buffer or by reading all the FIFO bytes. 1: Set by the RX buffer becoming full or by a byte being transferred into the FIFO.

A0090014 XON2 XON2 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON2							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XON2**	XON2	XON2 character for software flow control

A0090018 MSR Modem Status Register NA

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											DSR	CTS	RESERVED	RESERVED	DDSR	DCTS
Type											RW	RW	RW	RW	RW	RW
Reset											N/A	N/A	0	0	0	0

Overview: After a reset, bit4-bit5 will be inputs. A modem status interrupt can be cleared by writing 0 or set by writing 1 to this register. bit0-bit1 can be written. Modified when LCR[7] = 0.

Bit(s)	Mnemonic	Name	Description
5	**DSR**	DSR	**Data set ready** When Loop = 0, this value is the complement of the NDSR input signal. When Loop = 1, this value is equal to the DTR bit in the modem control register.
4	**CTS**	CTS	**Clear to send** When Loop = 0, this value is the complement of the NCTS input signal. When Loop = "1", this value is equal to the RTS bit in the modem control register.
1	**DDSR**	DDSR	**Delta data set ready** 0: Cleared if the state of DSR has not changed since this register is last read. 1: Set if the state of DSR has changed since this register is last read.
0	**DCTS**	DCTS	**Delta clear to send** 0: Cleared if the state of CTS has not changed since this register is last read. 1: Set if the state of CTS has changed since this register is last read.

A0090018 XOFF1 XOFF1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XOFF1**	XOFF1	XOFF1 character for software flow control

A009001C SCR Scratch Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SCR							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: A general-purpose read/write register. After reset, its value will be un-defined. Modified when LCR[7] = 0.

Bit(s)	Mnemonic	Name	Description
7:0	**SCR**	SCR	

A009001C XOFF2 XOFF2 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF2							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: XON1, XON2, XOFF1 and XOFF2 are valid only when LCR = BFh.

Bit(s)	Mnemonic	Name	Description
7:0	**XOFF2**	XOFF2	XOFF2 character for software flow control

A0090020 AUTOBAUD_EN Auto Baud Detect Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															AUTO_SEL	AUTO_EN
Type															RW	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**AUTO_SEL**	AUTOBAUD_SEL	**Selects auto-baud feature** 0: Auto-baud function detects baud and transfers it to standard baud. 1: Auto-baud function detects baud (only supports baud from 300 to 115,200; 52MHZ to auto fix is recommended).
0	**AUTO_EN**	AUTOBAUD_EN	**Auto-baud enabling signal** *Note: When AUTOBAUD_EN is active, there should not A*/a* char before the auto baud char AT/at. If A*/a* is inevitable, the autobaud will fail, and you should disable AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.* 0: Disable auto-baud function 1: Enable auto-baud function (UARTn+0024h SPEED should be set to 0.)

A0090024 HIGHSPEED High Speed Mode Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SPEED	
Type															RW	
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1:0	**SPEED**	SPEED	**UART sample counter base** 0: Based on 16*baud_pulse,baud_rate = system clock frequency/16/{DLH, DLL} 1: Based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL} 2: Based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL} 3: Based on sampe_count*baud_pulse, baud_rate = system clock

Bit(s)	Mnemonic	Name	Description
			frequency/(sampe_count+1)/{DLM, DLL}

A0090028 SAMPLE_COUNT Sample Counter Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLECOUNT							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Overview: When HIGHSPEED = 3, sample_count will be the threshold value for UART sample counter (sample_num). Counts from 0 to sample_count.

Bit(s)	Mnemonic	Name	Description
7:0	**SAMPLECOUNT**	SAMPLECOUNT	Only useful when HIGHSPEED mode = 3.

A009002C SAMPLE_POINT Sample Point Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLEPOINT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Overview: When HIGHSPEED = 3, UART will obtain the input data when sample_count = sample_num, e.g. system clock = 13MHz, baud rate = 921,600. Therefore, sample_count = 13MHz/921,600 = 14, and sample point = 13/2 = 6 (sampling the central point to decrease the inaccuracy) The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.

Bit(s)	Mnemonic	Name	Description
7:0	**SAMPLEPOINT**	SAMPLEPOINT	

A0090030 AUTOBAUD_REG Auto Baud Monitor Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BAUD_STAT				BAUD_RATE			
Type									RO				RO			
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:4	**BAUD_STAT**	BAUD_STAT	**Autobaud format** 0: Autobaud is detecting. 1: AT_7N1 2: AT_7O1 3: AT_7E1 4: AT_8N1 5: AT_8O1 6: AT_8E1 7: at_7N1 8: at_7E1 9: at_7O1

Bit(s)	Mnemonic	Name	Description
			10: at_8N1 11: at_8E1 12: at_8O1 13: Autobaud detection fails.
3:0	**BAUD_RATE**	BAUD_RATE	**Autobaud baud rate** 0: 115,200 1: 57,600 2: 38,400 3: 19,200 4: 9,600 5: 4,800 6: 2,400 7: 1,200 8: 300 9: 110

A0090034 RATEFIX_AD Clock Rate Fix Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**FREQ_SEL**	**AUTOBAUD_RATE_FIX**	**RATE_FIX**
Type														RW	RW	RW
Reset														0	0	0

Overview: UART operation clock rate is fixed for low power.

Bit(s)	Mnemonic	Name	Description
2	**FREQ_SEL**	FREQ_SEL	0: Select 26MHz as system clock 1: Select 13MHz as system clock
1	**AUTOBAUD_RATE_FIX**	AUTOBAUD_RATE_FIX	0: Use 52MHz as system clock for UART auto baud detection 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART auto baud detection
0	**RATE_FIX**	RATE_FIX	0: Use 52MHz as system clock for UART TX/RX 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART TX/RX

A0090038 AUTOBAUDSAMPLE Auto Baud Sample Register 0D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**AUTOBAUDSAMPLE**					
Type											RW					
Reset											0	0	1	1	0	1

Overview: Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6. When system clock = 26MHz, autobaudsample = 13. Note that when AUTO_SEL = 1, autobaudsample should be 15.

Bit(s)	Mnemonic	Name	Description
5:0	**AUTOBAUDSAMPLE**	AUTOBAUDSAMPLE	**For standard baud rate detection:** For system clk 52m, set to 'd 27; for system clk 26m, set to 'd 13; for system clk 13m, set to 'd 6. **For non-standard baud rate detection, set to 'd15.**

A009003C GUARD Guard Time Added Register 0F

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT			
Type												RW	RW			
Reset												0	1	1	1	1

Bit(s)	Mnemonic	Name	Description
4	**GUARD_EN**	GUARD_EN	**Guard interval added enabling signal** 0: No guard interval added. 1: Add guard interval after stop bit.
3:0	**GUARD_CNT**	GUARD_CNT	**Guard interval count value** Guard interval = [1/(system clock/div_step/div)]*GUARD_CNT.

A0090040 ESCAPE_DAT Escape Char Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ESCAPE_DAT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
7:0	**ESCAPE_DAT**	ESCAPE_DAT	**Escape character added before software flow control data and escape character** i.e. if TX data are xon (31h), with esc_en =1, UART will transmit data as esc + CEh (~xon).

A0090044 ESCAPE_EN Escape Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**ESC_EN**	ESC_EN	**Adds escape character in transmitter and removes escape character in receiver by UART** 0: Does not deal with the escape character 1: Add escape character in transmitter and remove escape character in receiver

A0090048 SLEEP_EN Sleep Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description

Bit(s)	Mnemonic	Name	Description
0	**SLEEP_EN**	SLEEP_EN	**For sleep mode issue** 0: Does not deal with sleep mode indicate signal 1: Activate hardware flow control or software control according to software initial setting when the chip enters sleep mode. Release hardware flow when the chip wakes up. However, for software control, UART sends xon when awaken and when FIFO does not reach threshold level.

A009004C DMA_EN DMA Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													LSR_SEL	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Mnemonic	Name	Description
3	**LSR_SEL**	LSR_SEL	**LSR selection just for FIFO enable mode** Accordingly for FIFO disable mode, LSR will be updated when RX received new data. 0: LSR stores the current and past errors until CPU reads LSR clear in FIFO mode. 1: LSR is updated when RX receives new data in FIFO mode.
2	**TO_CNT_AUTORST**	TO_CNT_AUTORST	**Time-out counter auto reset register** 0: After RX time-out happens, SW will reset the interrupt by reading UART 0x4C. 1: The time-out counter will be auto reset. Set this register when new DMA is used.
1	**TX_DMA_EN**	TX_DMA_EN	**TX_DMA mechanism enabling signal** 0: Does not use DMA in TX. 1: Use DMA in TX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt for DMA.
0	**RX_DMA_EN**	RX_DMA_EN	**RX_DMA mechanism enabling signal** 0: Does not use DMA in RX 1: Use DMA in RX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt

A0090050 RXTRI_AD Rx Trigger Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												RXTRIG				
Type												RW				
Reset												0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
4:0	**RXTRIG**	RXTRIG	**When {rtm,rtl}=2'b11, the Rx FIFO threshold will be Rxtrig.** The value is suggested to be smaller than half the RX FIFO size, which is 24 bytes.

A0090054 FRACDIV_L Fractional Divider LSB Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**FRACDIV_L**	FRACDIV_L	Adds sampling count (+1) from state data 7 to data 0 in order to contribute fractional divisor

A0090058 FRACDIV_M Fractional Divider MSB Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															FRACDIV_M	
Type															RW	
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1:0	**FRACDIV_M**	FRACDIV_M	Adds sampling count when in state stop to parity in order to contribute fractional divisor

A009005C FCR_RD FIFO Control Read Out Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									RO		RO			RO	RO	RO
Reset									0	0	0	0		0	0	0

Bit(s)	Mnemonic	Name	Description
7:6	**RFTL1_RFTL0**	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 24 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	**TFTL1_TFTL0**	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 16 bytes. 0: 1 1: 4 2: 8 3: 14 (FIFOSIZE - 2)
2	**CLRT**	CLRT	**Clears Transmit FIFO.** This bit is self-clearing. 0: Leave TX FIFO intact. 1: Clear all the bytes in the TX FIFO.
1	**CLRR**	CLRR	**Clears Receive FIFO** This bit is self-clearing. 0: Leave RX FIFO intact. 1: Clear all the bytes in the RX FIFO.
0	**FIFOE**	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect.

Bit(s)	Mnemonic	Name	Description
			0: Disable both the RX and TX FIFOs. 1: Enable both the RX and TX FIFOs.

3.8 I2C/SCCB Controller

3.8.1 General Description

I2C (Inter-IC)/SCCB (Serial Camera Control Bus) is a two-wire serial interface. The two signals are SCL and SDA. SCL is a clock signal that is driven by the master. SDA is a bi-directional data signal that can be driven by either the master or the slave. This generic controller supports the master role and conforms to the I2C specification.

3.8.1.1 Feature

I2C compliant master mode operation
Adjustable clock speed for LS/FS mode operation
Supports 7-bit/10-bit addressing
Supports high-speed mode
Supports slave clock extension
START/STOP/REPEATED START condition
Manual transfer mode
Multi-write per transfer (up to 8 data bytes for non-DMA mode)
Multi-read per transfer (up to 8 data bytes for non-DMA mode)
Multi-transfer per transaction
Combined format transfer with length change capability
Active drive/wired-and I/O configuration

3.8.1.2 Manual Transfer Mode

The controller offers one transfer mode, the manual mode.

When the manual mode is selected, in addition to the slave address register, the controller has a built-in 8byte deep FIFO which allows MCU to prepare up to 8 bytes of data for a write transfer, or read up to 8 bytes of data for a read transfer.

3.8.1.3 Transfer Format Support

This controller has been designed to be as generic as possible in order to support a wide range of devices that may utilize different combinations of transfer formats. Here are the transfer format types that can be supported through different software configuration:

Wording convention note

Transfer = Anything encapsulated within a Start and Stop or Repeated Start.
Transfer length = Number of bytes within the transfer
Transaction = This is the top unit. Everything combined equals 1 transaction.
Transaction length = Number of transfers to be conducted.

Single-byte access

Multi-byte access

Multi-byte transfer + multi-transfer (same direction)

Multi-byte transfer + multi-transfer w RS (same direction)

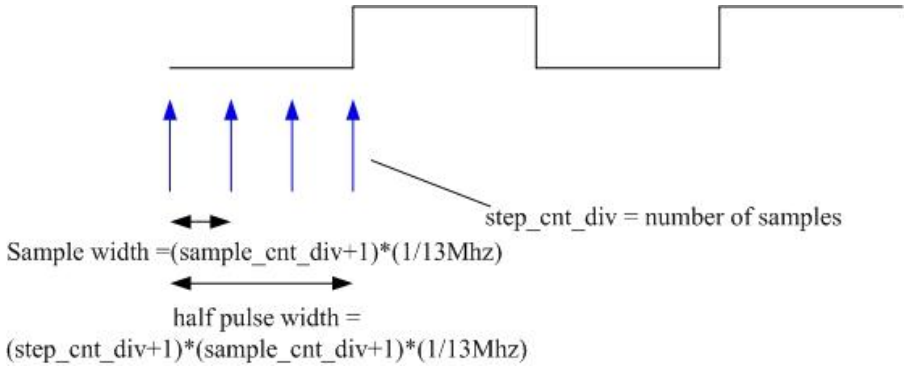
Combined write/read with Repeated Start (direction change)

Note: Only supports write and then read sequence. Read and then write is not supported.

3.8.2 Programming Examples

Common transfer programmable parameters

Output waveform timing programmable parameters



3.8.3 Register Definition

Module name: I2C_SCCB_Controller base address: (+A0120000h)

Address	Name	Width	Register function
A0120000	DATA_PORT	16	Data port register
A0120004	SLAVE_ADDR	16	Slave address register
A0120008	INTR_MASK	16	**Interrupt mask register** This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = Allow interrupt 0 = Disable interrupt *Note: While disabled, the corresponding interrupt will not be asserted, however intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.*
A012000C	INTR_STAT	16	**Interrupt status register** When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.
A0120010	CONTROL	16	Control register
A0120014	TRANSFER_LEN	16	Transfer length register (number of bytes per transfer)
A0120018	TRANSAC_LEN	16	Transaction length register (number of transfers per transaction)
A012001C	DELAY_LEN	16	Inter delay length register
A0120020	TIMING	16	**Timing control register** LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz
A0120024	START	16	Start register
A0120030	FIFO_STAT	16	FIFO status register
A0120038	FIFO_ADDR_CLR	16	FIFO address clear register
A0120040	IO_CONFIG	16	**IO config register** This register is used to configure the I/O for the SDA

Address	Name	Width	Register function
			and SCL lines to select between normal I/O mode, or open-drain mode to support wired-and bus.
A0120048	HS	16	**High speed mode register** This register contains options for supporting high speed operation features. Each HS half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz
A0120050	SOFTRESET	16	Soft reset register
A0120060	SPARE	16	SPARE
A0120064	DEBUGSTAT	16	Debug status register
A0120068	DEBUGCTRL	16	Debug control register
A012006C	TRANSFER_LEN_AUX	16	Transfer length register (number of bytes per transfer)
A0120074	TIMEOUT	16	Timeout timing register

A0120000 DATA_PORT Data Port Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA_PORT							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**DATA_PORT**	DATA_PORT	**FIFO access port** During master write sequences (slave_addr[0] = 0), this port can be written by APB, and during master read sequences (slave_addr[0] = 1), this port can be read by APB. *Note: Slave_addr must be set correctly before accessing FIFO.* For debugging only: If the fifo_apb_debug bit is set, FIFO can be read and written by the APB.

A0120004 SLAVE_ADDR Slave Address Register 00BF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SLAVE_ADDR							
Type									RW							
Reset									1	0	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
7:0	**SLAVE_ADDR**	SLAVE_ADDR	**Specifies the slave address of the device to be accessed** Bit 0 is defined by the I2C protocol as a bit that indicates the direction of transfer. 0: Master write 1: Master read

A0120008 INTR_MASK Interrupt Mask Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												MASK_TIMEOUT		MASK_HS_NACKERR	MASK_ACKERR	MASK_TRANSAC_COMP
Type												RW		RW	RW	RW
Reset												0		0	0	0

Overview: This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = allow interrupt; 0 = disable interrupt Note that while disabled, the corresponding interrupt will not be asserted. However, intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.

Bit(s)	Mnemonic	Name	Description
4	**MASK_TIMEOUT**	MASK_TIMEOUT	Setting this value to 0 will mask TIMEOUT interrupt signal.
2	**MASK_HS_NACKERR**	MASK_HS_NACKERR	Setting this value to 0 will mask HS_NACKERR interrupt signal.
1	**MASK_ACKERR**	MASK_ACKERR	Setting this value to 0 will mask ACK_ERR interrupt signal.
0	**MASK_TRANSAC_COMP**	MASK_TRANSAC_COMP	Setting this value to 0 will mask TRANSAC_COMP interrupt signal.

A012000C INTR_STAT **Interrupt Status Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												TIMEOUT	ARB_LOST	HS_NACKERR	ACKERR	TRANSAC_COMP
Type												W1C	W1C	W1C	W1C	W1C
Reset												0	0	0	0	0

Overview: When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.

Bit(s)	Mnemonic	Name	Description
4	**TIMEOUT**	TIMEOUT_IRQ	This status is asserted if time-out is enabled and the timer expires. The internal master state machine will stop, and MCU will need to manually clear the state machine by either issuing software reset by disabling the transact_en bit. Time-out can be used to detect PCB or I2C malfunction.
3	**ARB_LOST**	SPARE	**Reserved**
2	**HS_NACKERR**	HS_NACKERR	This status is asserted if HS master code NACK error detection is enabled. If enabled, HS master code NACK err will cause transaction to end, and stop will be issued.
1	**ACKERR**	ACKERR	This status is asserted if ACK error detection is enabled. If enabled, ACKERR will cause transaction to end, and stop will be issued.
0	**TRANSAC_COMP**	TRANSAC_COMP	This status is asserted when a transaction is completed successfully.

A0120010 CONTROL **Control Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								TIMEOUT_EN	RESET_BUS_BUSY_EN	TRANSFER_LEN_CHANGE	ACKERR_DET_EN	DIR_CHANGE	CLK_EXT_EN		RS_STOP	
Type								RW	RW	RW	RW	RW	RW		RW	
Reset								0	0	0	0	0	0		0	

Bit(s)	Mnemonic	Name	Description
8	**TIMEOUT_EN**	TIMEOUT_EN	**Enables time-out mechanism** When enabled, if SCL stays at 0 for too long period of time due to I2C slave holds SCL at 0 for too long or SCL stucks at 0 due to PCB issue, the master shall terminate the transaction, stop the internal state machine and assert time-out interrupt. MCU shall handle this case appropriately and reset the master and FIFO address before reissuing the transaction again. If this option is disabled, the HW timer will not count and nor expire forever. 0: Disable 1: Enable
7	**RESET_BUS_BUSY_EN**	SPARE	**Reserved**
6	**TRANSFER_LEN_CHANGE**	TRANSFER_LEN_CHANGE	**Specifies whether or not to change the transfer length after the fist transfer is completed** If enabled, the transfers after the first transfer will use the transfer_len_aux parameter. 0: Disable 1: Enable
5	**ACKERR_DET_EN**	ACKERR_DET_EN	**Enables slave ACK error detection** When enabled, if slave ACK error is detected, the master shall terminate the transaction by issuing a STOP condition and then assert the ACKERR interrupt. MCU handles this case appropriately then resets the FIFO address before reissuing transaction. If this option is disabled, the controller will ignore slave ACK error and keep on scheduled transaction. 0: Disable 1: Enable
4	**DIR_CHANGE**	DIR_CHANGE	**Combined transfer format, where the direction of transfer is to be changed from write to read after the FIRST RS condition** *Note: When set to 1, the transfers after the direction change will be based on the transfer_len_aux parameter.* 0: Disable 1: Enable
3	**CLK_EXT_EN**	CLK_EXT_EN	**I2C spec allows slaves to hold the SCL line low if it is not yet ready for further processing** Therefore, if this bit is set to 1, the master controller will enter a high wait state until the slave releases the SCL line.
1	**RS_STOP**	RS_STOP	**In LS/FS mode, this bit affects multi-transfer transaction only.** It controls whether or not the REPEATED-START condition is used between transfers. The last ending transfer always ends with a STOP. In HS mode, this bit must be set to 1. 0: Use STOP 1: Use REPEATED-START

A0120014 TRANSFER_LEN **Transfer Length Register (Number of Bytes per Transfer)** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name													TRANSFER_LEN_AUX			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	**TRANSFER_LEN_AUX**	TRANSFER_LEN	**Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte)** *Note: The value must be set to be bigger than 1; otherwise no transfer will take place.*

A0120018 TRANSAC_LEN Transaction Length Register (Number of Transfers per Transaction) 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									TRANSAC_LEN							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
7:0	**TRANSAC_LEN**	TRANSAC_LEN	**Indicates the number of transfers to be transferred in 1 transaction** *Note: The value must be set to be bigger than 1; otherwise no transfer will take place.*

A012001C DELAY_LEN Inter Delay Length Register 0002

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DELAY_LEN							
Type									RW							
Reset									0	0	0	0	0	0	1	0

Bit(s)	Mnemonic	Name	Description
7:0	**DELAY_LEN**	DELAY_LEN	**Sets up wait delay between consecutive transfers when RS_STOP bit is set to 0** Unit: Half the pulse width

A0120020 TIMING Timing Control Register 1303

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA_READ_ADJ	DATA_READ_TIME				SAMPLE_CNT_DIV					STEP_CNT_DIV					
Type	RW	RW				RW					RW					
Reset	0	0	0	1		0	1	1			0	0	0	0	1	1

Overview: LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz

Bit(s)	Mnemonic	Name	Description
15	**DATA_READ_ADJ**	DATA_READ_ADJ	When set to 1, data latch in sampling time during master reads are adjusted according to the DATA_READ_TIME value. Otherwise, by default, the data are latched in at half of the high pulse width point.

Bit(s)	Mnemonic	Name	Description
			This value must be set to be smaller than or equal to half the high pulse width.
14:12	**DATA_READ_TIME**	DATA_READ_TIME	This value is valid only when DATA_READ_ADJ is set to 1. This can be used to adjust so that the data are latched in at earlier sampling points (assuming data are settled by then).
10:8	**SAMPLE_CNT_DIV**	SAMPLE_CNT_DIV	Used for LS/FS only. This adjusts the width of each sample. Sample width = (sample_cnt_div +1)/13MHz
5:0	**STEP_CNT_DIV**	STEP_CNT_DIV	Specifies the number of samples per half pulse width, i.e. each high or low pulse

A0120024 START **Start Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**START**
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**START**	START	**Starts the transaction on the bus** It is auto de-asserted at the end of the transaction.

A0120030 FIFO_STAT **FIFO Status Register** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RD_ADDR**				**WR_ADDR**				**FIFO_OFFSET**						**WR_FULL**	**RD_EMPTY**
Type	RO				RU				RU						RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0			0	1

Bit(s)	Mnemonic	Name	Description
15:12	**RD_ADDR**	RD_ADDR	**Current RD address pointer** Only bit [2:0] have physical meanings.
11:8	**WR_ADDR**	WR_ADDR	**Current WR address pointer** Only bit [2:0] have physical meanings.
7:4	**FIFO_OFFSET**	FIFO_OFFSET	**wr_addr[3:0] - rd_addr[3:0]**
1	**WR_FULL**	WR_FULL	**Indicates FIFO is full**
0	**RD_EMPTY**	RD_EMPTY	**Indicates FIFO is empty**

A0120038 FIFO_ADDR_CLR **FIFO Address Clear Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**FIFO_ADDR_CLR**
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**FIFO_ADDR**	FIFO_ADDR_CLR	When written with 1'b1, a 1 pulse fifo_addr_clr is generated to clear

Bit(s)	Mnemonic	Name	Description
	_CLR		the FIFO address back to 0.

A0120040 IO_CONFIG **IO Config Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**IDLE_OE_EN**		**SDA_IO_CONFIG**	**SCL_IO_CONFIG**
Type													RW		RW	RW
Reset													0		0	0

Overview: This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode or open-drain mode to support wired-and bus.

Bit(s)	Mnemonic	Name	Description
3	**IDLE_OE_EN**	IDLE_OE_EN	0: Does not drive bus in idle state 1: Drive bus in idle state
1	**SDA_IO_CONFIG**	SDA_IO_CONFIG	0: Normal tristate I/O mode 1: Open-drain mode
0	**SCL_IO_CONFIG**	SCL_IO_CONFIG	0: Normal tristate I/O mode 1: Open-drain mode

A0120048 HS **High Speed Mode Register** **0102**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**HS_SAMPLE_CNT_DIV**				**HS_STEP_CNT_DIV**				**MASTER_CODE**					**HS_NACKERR_DET_EN**	**HS_EN**
Type		RW				RW				RW					RW	RW
Reset		0	0	0		0	0	1		0	0	0			1	0

Overview: This register contains options for supporting high speed operation features Each HS half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz.

Bit(s)	Mnemonic	Name	Description
14:12	**HS_SAMPLE_CNT_DIV**	HS_SAMPLE_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the sample width will become dependent on this parameter.
10:8	**HS_STEP_CNT_DIV**	HS_STEP_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the number of samples per half pulse width will become dependent on this value.
6:4	**MASTER_CODE**	MASTER_CODE	This is the 3 bit programmable value for the master code to be transmitted.
1	**HS_NACKERR_DET_EN**	HS_NACKERR_DET_EN	**Enables NACKERR detection during the master code transmission** When enabled, if NACK is not received after the master code is transmitted, the transaction will be terminated with a STOP condition.
0	**HS_EN**	HS_EN	**Enables high-speed transaction** *Note: rs_stop must be set to 1.*

A0120050 SOFTRESET Soft Reset Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SOFT_RESET
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	SOFT_RESET	SOFT_RESET	When written with 1'b1, a 1 pulse soft reset is used as synchronous reset to reset the I2C internal hardware circuits.

A0120060 SPARE SPARE 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													SPARE			
Type													RW			
Reset													0	0	0	0

Bit(s)	Mnemonic	Name	Description
3:0	SPARE	SPARE	Reserved for future use

A0120064 DEBUGSTAT Debug Status Register 0020

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BUS_BUSY	MASTER_WRITE	MASTER_READ	MASTER_STATE				
Type									RO	RO	RO	RO				
Reset									0	0	1	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7	BUS_BUSY	SPARE	**Reserved**
6	MASTER_WRITE	MASTER_WRITE	**For debugging only** 1: Current transfer is in the master write dir.
5	MASTER_READ	MASTER_READ	**For debugging only** 1: Current transfer is in the master read dir.
4:0	MASTER_STATE	MASTER_STATE	**(For debugging only) Reads back the current master_state.** 0: Idle state 1: I2c master is preparing to send out the start bit, SCL=1, SDA=1. 2: I2C master is sending out the start bit, SCL=1, SDA=0. 3: I2C master/slave is preparing to transmit data bit, SCL=0, SDA=data bit. (Data bit can be changed when SCL=0.) 4: I2C master/slave is transmitting data bit, SCL=1, SDA=data bit. (Data bit is stable when SCL=1.) 5: I2C master/slave is preparing to transmit the ACK bit, SCL=0, SDA=ack. (The ACK bit can be changed when SCL=0.) 6: I2C master/slave is transmitting the ACK bit, SCL=1, SDA=0. (The ACK bit is stable when SCL=1.) 7: I2C master is preparing to send out stop bit or repeated-start bit, SCL=0, SDA=0/1. (0: Stop bit; 1: Repeated-start bit) 8: I2C master is sending out stop bit or repeated-start bit, SCL=1, SDA=1/0. (0: Repeated-start bit; 1: Stop bit)

Bit(s)	Mnemonic	Name	Description
			9: I2C master is in delay start between two transfers, SCL=1, SDA=1. 10: I2C master is in FIFO wait state; For writing transaction, it means FIFO is empty and I2C master is waiting for DMA controller to write data into FIFo. For reading transaction, it means FIFO is full and I2C master is waiting for DMA controller to read data from FIFOo, SCL=0, SDA=don't care. 12: I2C master is preparing to send out data bit of master code. This state is used only in high-speed transaction, SCL=0, SDA=data bit of master code. (Data bit of master code can be changed when SCL=0.) 13: I2C master is sending out data bit of master code. This state is used only in high-speed transaction, SCL=1, SDA=data bit of master code. (Data bit of master code is stable when SCL=1.) 14: I2C master/slave is preparing to transmit the NACK bit, SCL=0, SDA=nack bit. (The NACK bit can be changed when SCL=0.) This state is used only in high-speed transaction. 15: I2C master/slave is transmitting the NACK bit, SCL=1, SDA=1. This state is used only in high-speed transaction.

A0120068 DEBUGCTRL **Debug Control Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															APB_DEBUG_RD	FIFO_APB_DEBUG
Type															WO	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**APB_DEBUG_RD**	APB_DEBUG_RD	**Only valid when fifo_apb_debug is set to 1** Writing to this register will generate a 1 pulsed FIFO APB RD signal for reading the FIFO data.
0	**FIFO_APB_DEBUG**	FIFO_APB_DEBUG	**Used for trace 32 debugging** When using trace 32, and the memory map is shown, turning this bit on will block the normal APB read access. The APB read access to the FIFO will then be enabled by writing to apb_debug_rd. 0: Disable 1: Enable

A012006C TRANSFER_LEN_AUX **Transfer Length Register (Number of Bytes per Transfer)** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TRANSFER_LEN			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	**TRANSFER_LEN**	TRANSFER_LEN_AUX	**Only valid when dir_change or transfer_len_change is set to 1. Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte) for the transfers following the direction change or transfer_len_change** If dir_change =1, the first write transfer length will depend on transfer_len, while the second read transfer length will depend on

Bit(s)	Mnemonic	Name	Description
			transfer_len_aux. Dir change is always after the first transfer. Similarly, transfer length change is always after the first transfer. *Note: The value must be set to be bigger than 1; otherwise no transfer will take place.*

A0120074 TIMEOUT **Timeout Timing Register** **FFFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIMEOUT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0		TIMEOUT	**Indicates the number of steps to count before time-out** The time-out counter counts only when the time-out mechanism is enabled and has started a transaction.

3.9 Real Time Clock

3.9.1 General Descriptions

The Real-Time Clock (RTC) module provides time and data information, as well as 32.768 kHz clock. By configure pin XOSC32_ENB, the usage of the 32k crystal can be determined. That is, to use a 32k crystal, or not to use a 32k crystal. The RTC block has an independent power supply. When the mobile handset is powered off, a dedicated regulator supplies the RTC block. If the main battery is not present, a backup supply such as a small mercury cell battery or a large capacitor is used. In addition to providing timing data, an alarm interrupt is generated and can be used to power up the baseband core. Regulator interrupts corresponding to seconds, minutes, hours and days can be generated whenever the time counter value reaches a maximum value (e.g., 59 for seconds and minutes, 23 for hours, etc.). The year span is supported up to 2127. The maximum day-of-month values, which depend on the leap year condition, are stored in the RTC block.

3.9.2 Register Definition

Module name: RTC Base address: (+A0710000h)

Address	Name	Width	Register function
A0710000	**RTC_BBPU**	16	**Baseband power up**
A071 0004	**RTC_IRQ_STA**	16	**RTC IRQ status** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A071 0008	**RTC_IRQ_EN**	16	**RTC IRQ enable** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A071 000C	**RTC_CII_EN**	16	**Counter increment IRQ enable** This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.

Address	Name	Width	Register function
A071 0010	**RTC_AL_MASK**	16	**RTC alarm mask** The alarm condition for alarm IRQ generation depends on whether or not the corresponding bit in this register is masked. Warning: If you set all bits to 1 in RTC_AL_MASK (i.e. RTC_AL_MASK=0x7f) and PWREN=1 in RTC_BBPU, it means the alarm will come every second, not disabled.
A071 0014	**RTC_TC_SEC**	16	**RTC seconds time counter register**
A071 0018	**RTC_TC_MIN**	16	**RTC minutes time counter register**
A071001C	**RTC_TC_HOU**	16	**RTC hours time counter register**
A0710020	**RTC_TC_DOM**	16	**RTC day-of-month time counter register**
A0710024	**RTC_TC_DOW**	16	**RTC day-of-week time counter register**
A0710028	**RTC_TC_MTH**	16	**RTC month time counter register**
A071002C	**RTC_TC_YEA**	16	**RTC year time counter register**
A0710030	**RTC_AL_SEC**	16	**RTC second alarm setting register**
A0710034	**RTC_AL_MIN**	16	**RTC minute alarm setting register**
A0710038	**RTC_AL_HOU**	16	**RTC hour alarm setting register**
A071003C	**RTC_AL_DOM**	16	**RTC day-of-month alarm setting register**
A0710040	**RTC_AL_DOW**	16	**RTC day-of-week alarm setting register**
A0710044	**RTC_AL_MTH**	16	**RTC month alarm setting register**
A0710048	**RTC_AL_YEA**	16	**RTC year alarm setting register**
A071004C	**RTC_OSC32CON**	16	**OSC32 control** The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.
A0710050	**RTC_POWERKEY1**	16	**RTC_POWERKEY1 register**
A0710054	**RTC_POWERKEY2**	16	**RTC_POWERKEY2 register**
A0710058	**RTC_PDN1**	16	**PDN1**
A071005C	**RTC_PDN2**	16	**PDN2**
A0710060	**RTC_SPAR0**	16	**Spare register for specific purpose**
A0710064	**RTC_SPAR1**	16	**Spare register for specific purpose**
A0710068	**RTC_PROT**	16	**Lock/unlock scheme to prevent RTC miswriting**
A071006C	**RTC_DIFF**	16	**One-time calibration offset** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A0710070	**RTC_CALI**	16	**Repeat calibration offset** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A0710074	**RTC_WRTGR**	16	**Enable the transfers from core to RTC in the queue**
A0710078	**RTC_CON**	16	**Other RTC control register** Note:LPRST,LPEN are tie0 internally when RTC_POWERKEY1 & RTC_POWERKEY2 do not match the correct values. After changing RTC_CON, write WRTGR = 1 to take effect.

*Table 1. **RTC register map***

A0710000 RTC_BBPU Baseband power up 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	KEY_BBPU									CBUSY	RELOAD	CLRPKY	AUTO	BBPU		PWREN
Type	WO									RO	WO	WO	RW	RW		RW
Reset	0	0	0	0	0	0	0	0		0	0	0	0	0		0

Bit (s)	Mnemonic	Name	Description
15:8	**KEY_BBPU**	KEY_BBPU	A bus write is acceptable only when KEY_BBPU=0x43.
6	**CBUSY**	CBUSY	The read/write channels between RTC/Core is busy. This bit indicates high after the software program sequence to anyone of RTC data registers and enables the transfer by RTC_WRTGR = 1. In addition, it is high after the reset from low to high due to RTC reload process.
5	**RELOAD**	RELOAD	**Reloads the values from RTC domain to core domain** Generally the RTC will reload and synchronize the data from RTC to core when being reset from 0 to 1. This bit can be treated as a debugging bit.
4	**CLRPKY**	CLRPKY	**Clears powerkey1 and powerkey2 at the same time** In some cases, the software may clear powerkey1 & powerkey2. BBWAKEUP depends on the matching specific patterns of powerkey1 and powerkey2. If any one of powerkey1 or powerkey2 or BBPU is cleared, BBWAKEUP will go low immediately. The software cannot program the other control bits without power. By programming RTC_BBPU with CLRPKY = 1 and BBPU = 0 condition, RTC can clear powerkey1, powerkey2 and BBPU at the same time.
3	**AUTO**	AUTO	**Controls if BBWAKEUP is automatically in the low state when SYSRST transitions from high to low.** 0: BBWAKEUP is not automatically in the low state when SYSRST# transitions are from high to low. 1: BBWAKEUP is automatically in the low state when SYSRST# transitions are from high to low. The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.
2	**BBPU**	BBPU	**Controls the power of PMU** If powerkey1 = A357h and powerkey2 = 67D2h, PMU takes on the value programmed by software; otherwise PMU is low. 0: Power down 1: Power on
0	**PWREN**	PWREN	0: RTC alarm has no action on power switch. 1: When an RTC alarm occurs, BBPU is set to 1 and the system is powered on by RTC alarm wakeup.

A0710004 RTC_IRQ_STA RTC IRQ status 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													LPST		TCST	ALST

													A		A	A
Type													RC		RC	RC
Reset													0		0	0

Overview This register is read-cleared, and is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
3	**LPSTA**	LPSTA	**Indicates the IRQ status and whether or not the LPD is asserted** (LPD function is either provided by XOSC32 or EOSC32[6], depending on XOSC32_ENB) 0: No IRQ occurred; the 32K clock is good. 1: IRQ occurred; the 32K clock stopped or stops. This can be masked by LP_EN or cleared by initialize LPD.
1	**TCSTA**	TCSTA	**Indicates the IRQ status and whether or not the tick condition has been met.** 0: No IRQ occurred; the tick condition has not been met. 1: IRQ occurred; the tick condition has been met.
0	**ALSTA**	ALSTA	**Indicates the IRQ status and whether or not the alarm condition has been met.** 0: No IRQ occurred; the alarm condition has not been met. 1: IRQ occurred; the alarm condition has been met.

A0710008 **RTC_IRQ_EN** **RTC IRQ enable** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													**LP_EN**	**ONESHOT**	**TC_EN**	**AL_EN**
Type													RW	RW	RW	RW
Reset													0	0	0	0

Overview This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
3	**LP_EN**	LP_EN	**Enables the control bit for IRQ generation if low power detected (32k clock off).** 0: Disable IRQ generations. 1: Enable LPD.
2	**ONESHOT**	ONESHOT	**Controls automatic reset of AL_EN and TC_EN**
1	**TC_EN**	TC_EN	Enables the control bit for IRQ generation if the tick condition has been met. 0: Disable IRQ generations 1: Enable the tick time match interrupt. Clear the interrupt

Bit (s)	Mnemonic	Name	Description
			when ONESHOT is high upon generation of the corresponding IRQ.
0	**AL_EN**	AL_EN	**Enables the control bit for IRQ generation if the alarm condition has been met.** 0: Disable IRQ generations 1: Enable the alarm time match interrupt. Clear the interrupt when ONESHOT is high upon generation of the corresponding IRQ.

A071000C **RTC_CII_EN** **Counter increment IRQ enable** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne							1/8SECCII	1/4SECCII	1/2SECCII	YEACII	MTHCII	DOWCII	DOMCII	HOUCII	MINCII	SECCII
Type							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset							0	0	0	0	0	0	0	0	0	0

Overview This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.

Bit (s)	Mnemonic	Name	Description
9	**1/8SECCII**	SECCII_1_8	Sets the bit to 1 to activate the IRQ at each 1/8 of a second update
8	**1/4SECCII**	SECCII_1_4	Sets the bit to 1 to activate the IRQ at each 1/4 of a second update
7	**1/2SECCII**	SECCII_1_2	Sets the bit to 1 to activate the IRQ at each 1/2 of a second update
6	**YEACII**	YEACII	Sets the bit to 1 to activate the IRQ at each year update
5	**MTHCII**	MTHCII	Sets the bit to 1 to activate the IRQ at each month update
4	**DOWCII**	DOWCII	Sets the bit to 1 to activate the IRQ at each day-of-week update
3	**DOMCII**	DOMCII	Sets the bit to 1 to activate the IRQ at each day-of-month update
2	**HOUCII**	HOUCII	Sets the bit to 1 to activate the IRQ at each hour update
1	**MINCII**	MINCII	Sets the bit to 1 to activate the IRQ at each minute update
0	**SECCII**	SECCII	Sets this bit to 1 to activate the IRQ at each second update

A0710010 **RTC_AL_MASK** **RTC alarm mask** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne										YEA_MSK	MTH_MSK	DOW_MSK	DOM_MSK	HOU_MSK	MIN_MSK	SEC_MSK
Type										RW	RW	RW	RW	RW	RW	RW
Reset										0	0	0	0	0	0	0

Overview The alarm condition for alarm IRQ generation depends on whether or not the

corresponding bit in this register is masked. Note that if all bits 1 in RTC_AL_MASK are set (i.e. RTC_AL_MASK = 0x7f) and PWREN = 1 in RTC_BBPU, it means the alarm will come every second, not disabled.

Bit (s)	Mnemonic	Name	Description
6	**YEA_MSK**	YEA_MSK	0: Condition (RTC_TC_YEA = RTC_AL_YEA) is checked to generate the alarm signal. 1: Condition (RTC_TC_YEA = RTC_AL_YEA) is masked, i.e. the value of RTC_TC_YEA does not affect the alarm IRQ generation.
5	**MTH_MSK**	MTH_MSK	0: Condition (RTC_TC_MTH = RTC_AL_MTH) is checked to generate the alarm signal. 1: Condition (RTC_TC_MTH = RTC_AL_MTH) is masked, i.e. the value of RTC_TC_MTH does not affect the alarm IRQ generation.
4	**DOW_MSK**	DOW_MSK	0: Condition (RTC_TC_DOW = RTC_AL_DOW) is checked to generate the alarm signal. 1: Condition (RTC_TC_DOW = RTC_AL_DOW) is masked, i.e. the value of RTC_TC_DOW does not affect the alarm IRQ generation.
3	**DOM_MSK**	DOM_MSK	0: Condition (RTC_TC_DOM = RTC_AL_DOM) is checked to generate the alarm signal. 1: Condition (RTC_TC_DOM = RTC_AL_DOM) is masked, i.e. the value of RTC_TC_DOM does not affect the alarm IRQ generation.
2	**HOU_MSK**	HOU_MSK	0: Condition (RTC_TC_HOU = RTC_AL_HOU) is checked to generate the alarm signal. 1: Condition (RTC_TC_HOU = RTC_AL_HOU) is masked, i.e. the value of RTC_TC_HOU does not affect the alarm IRQ generation.
1	**MIN_MSK**	MIN_MSK	0: Condition (RTC_TC_MIN = RTC_AL_MIN) is checked to generate the alarm signal. 1: Condition (RTC_TC_MIN = RTC_AL_MIN) is masked, i.e. the value of RTC_TC_MIN does not affect the alarm IRQ generation.
0	**SEC_MSK**	SEC_MSK	0: Condition (RTC_TC_SEC = RTC_AL_SEC) is checked to generate the alarm signal. 1: Condition (RTC_TC_SEC = RTC_AL_SEC) is masked, i.e. the value of RTC_TC_SEC does not affect the alarm IRQ generation.

A0710014 RTC_TC_SEC RTC seconds time counter register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											TC_SECOND					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**TC_SECOND**	TC_SECOND	Second initial value for the time counter. Range: 0 ~ 59

A0710018 RTC_TC_MIN RTC minutes time counter register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											TC_MINUTE					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**TC_MINUTE**	TC_MINUTE	Minute initial value for the time counter. Range: 0 ~ 59

A071001C **RTC_TC_HOU** **RTC hours time counter register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												TC_HOUR				
Type												RW				
Reset												0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	**TC_HOUR**	TC_HOUR	Hour initial value for the time counter. Range: 0 ~ 23

A0710020 **RTC_TC_DOM** **RTC day-of-month time counter register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												TC_DOM				
Type												RW				
Reset												0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	**TC_DOM**	TC_DOM	Day-of-month initial value for the time counter. The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

A0710024 **RTC_TC_DOW** **RTC day-of-week time counter register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														TC_DOW		
Type														RW		
Reset														0	0	0

Bit (s)	Mnemonic	Name	Description
2:0	**TC_DOW**	TC_DOW	Day-of-week initial value for the time counter. Range: 1 ~ 7

A0710028 **RTC_TC_MTH** **RTC month time counter register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													TC_MONTH			
Type													RW			
Reset													0	0	0	0

Bit (s)	Mnemonic	Name	Description

Bit (s)	Mnemonic	Name	Description
3:0	**TC_MONTH**	TC_MONTH	Month initial value for the time counter. Range: 1 ~ 12

A071002C **RTC_TC_YEA** **RTC year time counter register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne										TC_YEAR						
Type										RW						
Reset										0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
6:0	**TC_YEAR**	TC_YEAR	**Year initial value for the time counter. Range: 0 ~ 127 (2000-2127).** The software can bias the year as multiples of 4 for the internal leap-year formula. Here are 3 examples: 2000 ~ 2127, 1972 ~ 2099 and 1904 ~ 2031. To simplify the process, the RTC hardware treats all 4-multiple as leap years. If the range you define includes non-leap 4-multiple year (e.g. 2100), please adjust it to the correct date by yourselves. (e.g. change Feb. 29th, 2100 to Mar. 1st, 2100). It is suggested to bias the range to be bigger than 1900 and smaller than 2100 to evade the manual adjustment, i.e. the bias values are suggested to be in the range of [-28,-96], that are (1972~ 2099) ~ (1904~ 2031). The formal leap formula: If year modulo 400 is 0 then leap Else if year modulo 100 is 0 then no_leap Else if year modulo 4 is 0 then leap Else no_leap

A0710030 **RTC_AL_SEC** **RTC second alarm setting register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											AL_SECOND					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**AL_SECOND**	AL_SECOND	Second value of the alarm counter setting. Range: 0 ~ 59

A0710034 **RTC_AL_MIN** **RTC minute alarm setting register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											AL_MINUTE					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**AL_MINUTE**	AL_MINUTE	Minute value of the alarm counter setting. Range: 0 ~ 59

A0710038 **RTC_AL_HOU** **RTC hour alarm setting register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												AL_HOUR				
Type												RW				
Reset												0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	**AL_HOUR**	AL_HOUR	Hour value of the alarm counter setting. Range: 0 ~ 23

A071003C **RTC_AL_DOM** **RTC day-of-month alarm setting register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												AL_DOM				
Type												RW				
Reset												0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	**AL_DOM**	AL_DOM	Day-of-month value of the alarm counter setting. The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

A0710040 **RTC_AL_DOW** **RTC day-of-week alarm setting register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														AL_DOW		
Type														RW		
Reset														0	0	0

Bit (s)	Mnemonic	Name	Description
2:0	**AL_DOW**	AL_DOW	Day-of-week value of the alarm counter setting. Range: 1 ~ 7

A0710044 **RTC_AL_MTH** **RTC month alarm setting register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													AL_MONTH			
Type													RW			
Reset													0	0	0	0

Bit (s)	Mnemonic	Name	Description
3:0	**AL_MONTH**	AL_MONTH	Month value of the alarm counter setting. Range: 1 ~ 12

A0710048 **RTC_AL_YEA** **RTC year alarm setting register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne										AL_YEAR						
Type										RW						
Reset										0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
6:0	**AL_YEAR**	AL_YEAR	Year value of the alarm counter setting. Range: 0 ~ 127 (2000-2127)

A071004C **RTC_OSC32CON** **OSC32 control** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	EMBCK_SEL_OPTION	RESERVED	RESERVED	RESERVED	RESERVED	EOSC_CHOP_EN	EOSC_OPT	RESERVED	EMBCK_SEL	EMBCK_SEL_MODE	XOSC32_ENB	XOSCCALI				
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RO	RW				
Reset	-					1	0		0	0	-	0	1	1	1	1

Overview The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

Bit (s)	Mnemonic	Name	Description
15	**EMBCK_SEL_OPTION**	EMBCK_SEL_OPTION	**Reserved bit** Please keep 0x0.
10	**EOSC_CHOP_EN**	EOSC_CHOP_EN	**EOSC32 chopper enable.** 0: Disable 1: Enable Please keep 0x1.
9	**EOSC_OPT**	EOSC_OPT	**EOSC32 optional bits** Please keep 0x0.
7:6	**EMB_MODE**	EMBCK_SEL	**Mode setting for crystal removal case.** Please keep 0x0.
5	**XOSC32_ENB**	XOSC32_ENB	**XOSC32_ENB** **Can read pin XOSC32_ENB configuration to know the 32k crystal usage.** 0: Use 32k crystal. 1: Do not use 32k crystal.
4:0	**XOSCCALI**	XOSCCALI	**Controls XOSC32/EOSC32 cali.** If XOSC32_ENB(RTC_OSC32CON[5]) = 0, the XOSCCALI controls the bias current for 32k xtal in XOSC32. When powerkeys do not match, the default value is 0xf. If XOSC32_ENB(RTC_OSC32CON[5]) = 1, the XOSCCALI is the trimming value for EOSC32. SW needs to find the best trimming value for EOSC32 when 1st power-on by frequency meter. When powerkeys do not match, the default value is 0xf.

OSC32CON is not protected by RTC_PROT because RTC_PROT needs 32.768 kHz clock to unlock. Similarly, to modify RTC_OSC32CON, writing RTC_WRTGR is not needed, neither. To protect the OSC32 control bits, follow the *update sequence* to update RTC_OSC32CON. After the updating sequence is completed, reload to acquire the internal RTC_OSC32CON. This register needs to be initialized **before** writing powerkeys match.

Update sequence:

Step 1: Write RTC_OSC32CON = 0x1a57 and wait until CBUSY=0

Step 2: Write RTC_OSC32CON = 0x2b68 and wait until CBUSY=0

Step 3: Write the real value you would like to write RTC_OSC32CON and wait until CBUSY=0

Step 4: Return to step 1 if you need to modify RTC_OSC32CON again.

Notices:

The RTC_OSC32CON should be set *before* write POWERKEY1 & POWERKEY2 to the correct value.

A0710050 RTC_POWERKEY1 RTC_POWERKEY1 register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_POWERKEY1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_POWERKEY1**	RTC_POWERKEY1	

A0710054 RTC_POWERKEY2 RTC_POWERKEY2 register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_POWERKEY2															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_POWERKEY2**	RTC_POWERKEY2	

These register sets are used to determine if the real time clock has been programmed by the software, i.e. the time value in real time clock is correct. When the real-time clock is first powered on, the register contents are all undefined, and therefore the time values shown are incorrect. The software needs to know if the real-time clock has been programmed. Hence, the two registers are defined to solve this power-on issue. After the software programs the correct value, the two register sets will not need to be updated. In addition to programming the correct time value, when the contents of the register sets are wrong, the interrupt will not be generated. Therefore, the-real time clock will not generate the interrupts before the software programs the registers, and unwanted interrupt due to wrong time value will not occur. The correct values of these two register sets are:

RTC_POWERKEY1 A357h
RTC_POWERKEY2 67D2h

A0710058 RTC_PDN1 PDN1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Mne	RTC_PDN1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_PDN1**	RTC_PDN1	Spare registers for software to keep the power-on and power-off state information

A071005C **RTC_PDN2** **PDN2** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_PDN2															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_PDN2**	RTC_PDN2	Spare registers for software to keep power-on and power-off state information

A0710060 **RTC_SPAR0** **Spare register for specific purpose** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_SPAR0															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_SPAR0**	RTC_SPAR0	Rreserved for specific purposes

A0710064 **RTC_SPAR1** **Spare register for specific purpose** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_SPAR1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_SPAR1**	RTC_SPAR1	Reserved for specific purposes

A0710068 **RTC_PROT** **Lock/unlock scheme to prevent RTC miswriting** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_PROT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_PROT**	RTC_PROT	**Protects RTC write interface by RTC_PROT** Whether the RTC writing interface is enabled or not is

Bit (s)	Mnemonic	Name	Description

			decided by RTC_PROT contents. When RTC_POWERKEY1 & RTC_POWERKEY2 are not equal to the correct values, the RTC writing interface will always be enabled. However, when they match, the user has to perform the unlock flow to enable the writing interface.
			
			Unlock flow:
			Step1: *RTC_PROT=0x586a;
			Step2: *RTC_WRTGR=1;
			Step3: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec
			Step4: *RTC_PROT=0x9136;
			Step5: *RTC_WRTGR=1;
			Step6: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec
			
			Note: Always keep RTC in the unlock state in the power-on mode. Once the normal RTC content writing is completed, DO NOT modify the RTC_PROT content to lock RTC. The RTC_PROT contents will be cleared automatically when being powered off immediately.

A071006C **RTC_DIFF** **One-time calibration offset** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne					RTC_DIFF											
Type					RW											
Reset					0	0	0	0	0	0	0	0	0	0	0	0

Overview This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
11:0	**RTC_DIFF**	RTC_DIFF	**Adjusts the internal counter of RTC. It takes effect once and returns to zero when done.** In some cases, RTC is faster or slower than the standard. To change RTC_TC_SEC being coarse may cause alarm problem. RTC_DIFF provides a finer time unit. An internal 15-bit counter accumulates in each 32,768-Hz clock. Entering a non-zero value to RTC_DIFF causes the internal RTC counter to increase or decrease RTC_DIFF when RTC_DIFF changes to zero again. RTC_DIFF represents 2's completement form. For example, if you fill in 0xfff into RTC_DIFF, the internal counter will decrease by 1 when RTC_DIFF returns to 0. In other words, you can only use RTC_DIFF continuously if RTC_DIFF is equal to 0 now. Note: RTC_DIFF ranges from 0x800 (-2048) to 0x7fd (2045). 0x7ff & 0x7fe are forbidden.

A0710070 **RTC_CALI** **Repeat calibration offset** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne										RTC_CALI						
Type										RW						
Reset										0	0	0	0	0	0	0

Overview This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
6:0	**RTC_CALI**	RTC_CALI	**Provides a repeated calibration scheme** RTC_CALI provides 7-bit calibration capability in 8-second duration, i.e. 5-bit calibration capability in each second. RTC_CALI represents 2's complement form for the user to adjust RTC increase or decrease Due to RTC_CALI is revealed in 8 seconds, the resolution is less than a 1/32768 clock. Avg. resolution: 1/32768/8 = 3.81us Avg. adjust range: -0.244~0.240ms/sec in 2's complement: -0x40 ~ 0x3f (-64 ~ 63)

A0710074 **RTC_WRTGR** **Enable the transfers from core to RTC in the queue** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																WRTGR
Type																WO
Reset																0

Bit (s)	Mnemonic	Name	Description
0	**WRTGR**	WRTGR	**Enables the transfers from core to RTC** After you modify all the RTC registers and would like to change, write 1 to RTC_WRTGR to trigger the transfer. The prior writing operation is queued in the core power domain. The pending data will not be transferred to the RTC domain until WRTGR = 1. After WRTGR=1, the pending data will be transferred to the RTC domain sequentially in order of register addresses, from low to high. For example: RTC_BBPU -> RTC_IRQ_EN -> RTC_CII_EN -> RTC_AL_MASK -> RTC_TC_SEC -> etc. CBUSY in RTC_BBPU is equal to 1 in the writing process. Observe CBUSY to determine when the transmission is completed.

A0710078 **RTC_CON** **Other RTC control register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	LPSTA_RAW		RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	LPRST	LPEN	RESERVED	VBAT_LPSTA_R

																AW
Type	W1C		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RO	RW
Reset																

Bit (s)	Mnemonic	Name	Description
15	**LPSTA_RAW**	LPSTA_RAW	**Raw status of LP_STA** Re-initialize LPD to clear this bit. Note that this bit is always high before LPD initialization sequence after the first power-on.
3	**LPRST**	LPRST	**Resets LPDETB** This only takes effect when LPEN = 1.
2	**LPEN**	LPEN	**Enables LPDETB** LP initialization sequence: 1. Write LPEN = 1, LPRST = 0. Write RTC_WRTGR = 1. wait cbusy down. 2. Write LPEN = 1, LPRST = 1. Write RTC_WRTGR = 1. wait cbusy down. 3. Write LPEN = 1, LPRST = 0. Write RTC_WRTGR = 1. wait cbusy down.
0	**VBAT_LPSTA_RAW**	VBAT_LPSTA_RAW	**Indicates the battery has been in LP state** Software needs to clear this bit for the next time use 0: VBAT has not been in LP state. 1: VBAT has been in LP state. Note: VBAT LP state = VBAT < 2.5V

When Vcore always exists, the software can trust the registers and wait for the LP interrupt from RTC if the 32.768 kHz clock stopped or has been stopped.

However, nothing can be trusted after the battery is off (Vrtc may drop). In every booting time, the software has to check if LPSTA_RAW = 1. (LP_STA = LPSTA_RAW & LP_IRQ_EN) If true, then the RTC contents will no longer be trusted just like powerkeys do not match. You have to initialize the RTC contents in this case.

3.10 Auxiliary ADC Unit

The auxiliary ADC unit is used to monitor the status of the battery and charger, to identify the plugged peripheral, and to perform temperature measurement. There are 7 input channels allowing diverse applications in this unit.

Figure 37: ***Auxadc Architecture***

Each channel operates in one of the two modes: immediate mode or timer-triggered mode. The mode of each channel can be individually selected through register AUXADC_CON0. For example, if the flag SYN0 in register AUXADC_CON0 is set, channel 0 will be set in the timer-triggered mode. Otherwise, the channel will operate in the immediate mode.

In the immediate mode, the A/D converter samples the value once only when the flag in the AUXADC_CON1 register is set. For example, if the flag IMM0 in AUXADC_CON1 is set, the A/D converter will sample the data for channel 0. The IMM flags must be cleared and set again to initialize another sampling.

The value sampled for channel 0 is stored in register AUXADC_DAT0, and the value for channel 1 is stored in register AUXADC_DAT1, and so on.

If the AUTOSET flag in register AUXADC_CON3 is set, the auto-sample function will be enabled. The A/D converter samples the data for the channel in which the corresponding data register is read. For example, in the case where the SYN1 flag is not set, the AUTOSET flag is set. When the data register AUXADC_DAT0 is read, the A/D converter will sample the next value for channel 1 immediately.

If multiple channels are selected at the same time, the task will be performed sequentially on every selected channel. For example, if AUXADC_CON1 is set to 0x3f, i.e. 6 channels are selected, the state machine in the unit will start sampling from channel 6 to channel 0 and save the values of each input channel in respective registers. The same process also applies to the timer-triggered mode.

In the timer-triggered mode, the A/D converter samples the value for the channels in which the corresponding SYN flags are set when the TDMA timer counts to the value specified in register TDMA_AUXEV1 placed in the TDMA timer. For example, if AUXADC_CON0 is set to 0x3f, the 6 channels will be selected to be in the timer-triggered mode. The state machine will sample the 6 channels sequentially and save the values in registers from AUXADC_DAT0 to AUXADC_DAT5, as it does in the immediate mode.

AUTOCLR*n* in register AUXADC_CON3 is set when it is intended to sample only once after setting up the timer-triggered mode. If the AUTOCLR1 flag is set, after the data for the channels in the timer-triggered mode are stored, the SYN*n* flags in register AUXADC_CON0 will be cleared.

The uses of the immediate mode and timer-triggered mode are mutually exclusive in terms of individual channels.

There are only two external pins (channel 4 ~ 5) for voltage detection. Other channels (0 ~ 3) are for battery voltage, battery current, charger and battery temperature respectively. Channel 9 is used for audio.

Touch panel

Figure 38: ***Touch Panel Circuit Structure***

Besides the normal sampling of external input voltage, the AUXADC includes the sampling of the touch panel function. For specified axis, the software should program AUX_TS_CMD first then trigger the sample of touch panel in register AUX_TS_CON. The touch panel sampling waveform is shown as the following. After the software polls the status bit in register AUXADC_CON3 to know that the touch panel sample is finished, the software can read back the specified axis value from register AUX_TS_DAT0.

Figure 39: ***Touch Panel Sampling Waveform***

S: Start bit

A2 ~ A0: Addressing bits
Mode: 10-bit or 8-bit
SE/DF: Single end or differential mode
PD1 ~ 0: Power down command

These values are defined in register AUX_TS_CMD. The table below shows the relationship between AUX_TS_CMD and touch panel control signals.

A2	A1	A0	SE/DFB	*CHN_SEL*	ADC In	X switches	Y switches	+REF	-REF
0	0	1	0	C	X+	OFF	ON	Y+	Y-
0	1	0	0	F	X-	OFF	ON	Y+	Y-
0	1	1	0	C	X+	X+ OFF X- ON	Y+ ON Y- OFF	Y+	X-
1	0	0	0	E	Y-	X+ OFF X- ON	Y+ ON Y- OFF	Y+	X-
1	0	1	0	D	Y+	ON	OFF	X+	X-
1	1	0	0	E	Y-	ON	OFF	X+	X-

Table 47: Relationship between commands and touch panel control signals

AuxADC Channel ID	Description
Channel 0	VBAT
Channel 1	ISENSE
Channel 2	CHRIN
Channel 3	BATON (BATtemp)
Channel 4	AUXIN4 (external)
Channel 5	ACCDET (external)
Channel 9	ClassAB
Channel 12	XP / EINT
Channel 13	YP / EINT
Channel 14	YM / EINT
Channel 15	XM / EINT

Table 2: Relationship between commands and touch panel control signals

3.10.1 Register Definition

Module name: AUXADC Base address: (+A0790000h)

Address	Name	Width	Register Function

A0790000	AUXADC_CON0	16	**AuxiliaryADC Control Register 0** These bits define whether the corresponding channel is sampled or not in the timer-triggered mode. It is associated with the timing offset register TDMA_AUXEV1. It supports multiple flags. The flags can be automatically cleared after those channels are sampled if AUTOCLR1 in register AUXADC_CON3 is set.
A0790004	AUXADC_CON1	16	**AuxiliaryADC Control Register 1** These bits are set individually to sample the data for the corresponding channel. It supports multiple flags.
A079000C	AUXADC_CON3	16	**AuxiliaryADC Control Register 3**
A0790010	AUXADC_DAT0	16	**AuxiliaryADC Channel 0 Register (VBAT)**
A0790014	AUXADC_DAT1	16	**AuxiliaryADC Channel 1 Register (ISENSE)**
A0790018	AUXADC_DAT2	16	**AuxiliaryADC Channel 2 Register (CHRIN)**
A079001C	AUXADC_DAT3	16	**AuxiliaryADC Channel 3 Register (VBATTMP)**
A0790020	AUXADC_DAT4	16	**AuxiliaryADC Channel 4 Register (External)**
A0790024	AUXADC_DAT5	16	**AuxiliaryADC Channel 5 Register (External/ACCDET)**
A0790034	AUXADC_DAT9	16	**AuxiliaryADC Channel 9 Register (ClassAB)**
A0790054	AUX_TS_CMD0	16	**Touch Screen Sample Command 0**
A0790058	AUX_TS_CON	16	**Touch Screen Control**
A079005C	AUX_TS_DAT0	16	**Touch Screen Sample DATA 0**
A07900D0	AUXADC_CON4	16	**AuxiliaryADC Control Register 4**
A07900D4	AUX_TS_CMD1	16	**Touch Screen Sample Command 1**
A07900D8	AUX_TS_DAT1	16	**Touch Screen Sample DATA 1**

Table 3: Auxadc Registers

A0790000 AUXADC_CON0 AuxiliaryADC Control Register 0 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							SYN9				SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
Type							R/W				R/W	R/W	R/W	R/W	R/W	R/W
Reset							0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9	SYN9	SYN9	**Channel 9 sync mode** 0: The channel is not selected. 1: The channel is selected.
5	SYN5	SYN5	**Channel 5 sync mode** 0: The channel is not selected. 1: The channel is selected.
4	SYN4	SYN4	**Channel 4 sync mode** 0: The channel is not selected. 1: The channel is selected.
3	SYN3	SYN3	**Channel 3 sync mode** 0: The channel is not selected. 1: The channel is selected.
2	SYN2	SYN2	**Channel 2 sync mode** 0: The channel is not selected. 1: The channel is selected.
1	SYN1	SYN1	**Channel 1 sync mode** 0: The channel is not selected. 1: The channel is selected.
0	SYN0	SYN0	**Channel 0 sync mode** 0: The channel is not selected. 1: The channel is selected.

A0790004 AUXADC_CON1 AuxiliaryADC Control Register 1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							IMM9				IMM5	IMM4	IMM3	IMM2	IMM1	IMM0
Type							R/W				R/W	R/W	R/W	R/W	R/W	R/W
Reset							0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9	IMM9	IMM9	**Channel 9 immediate mode** 0: The channel is not selected. 1: The channel is selected.
5	IMM5	IMM5	**Channel 5 immediate mode** 0: The channel is not selected. 1: The channel is selected.
4	IMM4	IMM4	**Channel 4 immediate mode** 0: The channel is not selected. 1: The channel is selected.
3	IMM3	IMM3	**Channel 3 immediate mode** 0: The channel is not selected. 1: The channel is selected.
2	IMM2	IMM2	**Channel 2 immediate mode** 0: The channel is not selected. 1: The channel is selected.
1	IMM1	IMM1	**Channel 1 immediate mode** 0: The channel is not selected. 1: The channel is selected.
0	IMM0	IMM0	**Channel 0 immediate mode** 0: The channel is not selected. 1: The channel is selected.

A079000C AUXADC_CON3 AuxiliaryADC Control Register 3 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AUTOSET				RSV		AUTOCLR1		SOFT_RST							AUXADC_STA
Type	R/W				R/W		R/W		R/W							RO
Reset	0				0		0		0							0

Bit(s)	Mnemonic	Name	Description
15	AUTOSET	AUTOSET	**Defines the auto-sample mode of the module. In auto-sample mode, each channel with its sample register being read can start sampling immediately without configuring the control register AUXADC_CON1 again.**
11	RSV	RSV	**Please keep 1'b0**
9	AUTOCLR1	AUTOCLR1	**Defines the auto-clear mode of the module for event 1. In the auto-clear mode, each timer-triggered channel acquires samples of specified channels once the SYNn bit in register AUXADC_CON0 is set. The SYNn bits are automatically cleared and the channel is not enabled again by the timer event except when the SYNn flags are set again.** 0: The automatic clear mode is not enabled. 1: The automatic clear mode is enabled.
7	SOFT_RST	SOFT_RST	**Software reset AUXADC state machine** 0: Normal function 1: Reset AUXADC state machine
0	AUXADC_STA	AUXADC_STA	**Defines the state of the module** 0: This module is idle. 1: This module is busy.

A0790010 AUXADC_DAT0 AuxiliaryADC Channel 0 Register (VBAT) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT	DAT	Sampled data for channel0

A0790014 AUXADC_DAT1 AuxiliaryADC Channel 1 Register (ISENSE) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT	DAT	Sampled data for channel1

A0790018 AUXADC_DAT2 AuxiliaryADC Channel 2 Register (CHRIN) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT	DAT	Sampled data for channel2

A079001C AUXADC_DAT3 AuxiliaryADC Channel 3 Register (VBATTMP) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT	DAT	Sampled data for channel3

A0790020 AUXADC_DAT4 AuxiliaryADC Channel 4 Register (External) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT	DAT	Sampled data for channel4

A0790024 AUXADC_DAT5 AuxiliaryADC Channel 5 Register (External/ACCDET) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT									

Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT	DAT	**Sampled data for channel5**

A0790034 AUXADC_DAT9 AuxiliaryADC Channel 9 Register (ClassAB) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	DAT	DAT	**Sampled data for channel9**

A0790054 AUX_TS_CMD0 Touch Screen Sample Command 0 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL	TS_MAGIC_KEY								ADDRESS			MODE	SEDF	PD	
Type	R/W	WO								R/W			R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	TS_SPL	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to "AB0600D6260_PMU" document "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.
14:7	TS_MAGIC_KEY	TS_MAGIC_KEY	**The TS commands in AUX_TS_CMD0 can only take effect when the TS_MAGIC_KEY matches the correct value.** TS_MAGIC_KEY=0xaa .
6:4	ADDRESS	ADDRESS	**Defines which x or y or z data will be sampled** 001: Y position 011: Z1 position 100: Z2 position 101: X position Others: Reserved
3	MODE	MODE	**Selects sample resolution** 0: 10-bit 1: 8-bit
2	SEDF	SEDF	**Selects mode** 0: Differential mode 1: Single-end mode
1:0	PD	PD	**Power-down control for analog IRQ signal and touch screen sample control signal** 00: Turn on Y-_drive signal and PDN_sh_ref 01: Turn on PDN_IRQ and PDN_sh_ref 10: Reserved 11: Turn on PDN_IRQ

A0790058 AUX_TS_CON Touch Screen Control 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																TS_SPL

Type																R/W
Reset																0

Bit(s)	Mnemonic	Name	Description
0	TS_SPL	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to "AB0600D6260_PMU" document "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.

A079005C AUX_TS_DAT0 Touch Screen Sample DATA 0 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL						TS_DAT									
Type	RO						RO									
Reset	0						0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	TS_SPL	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to "AB0600D6260_PMU" document "AUX_CON6" register bit 14 "ts_status"
9:0	TS_DAT	TS_DAT	**Touch screen sample result data**

A07900D0 AUXADC_CON4 AuxiliaryADC Control Register 4 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL	AUXADC_STA					SYN9				SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
Type	RO	RO					RO				RO	RO	RO	RO	RO	RO
Reset	0	0					0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	TS_SPL	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to "AB0600D6260_PMU" document "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.
14	AUXADC_STA	AUXADC_STA	**Defines the state of the module** 0: This module is idle. 1: This module is busy.
9	SYN9	SYN9	**Channel 9 sync mode** 0: The channel is not selected. 1: The channel is selected.
5	SYN5	SYN5	**Channel 5 sync mode** 0: The channel is not selected. 1: The channel is selected.
4	SYN4	SYN4	**Channel 4 sync mode** 0: The channel is not selected. 1: The channel is selected.
3	SYN3	SYN3	**Channel 3 sync mode** 0: The channel is not selected. 1: The channel is selected.
2	SYN2	SYN2	**Channel 2 sync mode**

			0: The channel is not selected. 1: The channel is selected.
1	SYN1	SYN1	**Channel 1 sync mode** 0: The channel is not selected. 1: The channel is selected.
0	SYN0	SYN0	**Channel 0 sync mode** 0: The channel is not selected. 1: The channel is selected.

A07900D4 AUX_TS_CMD1 Touch Screen Sample Command 1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										ADDRESS			MODE	SEDF	PD	
Type										R/W			R/W	R/W	R/W	
Reset										0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
6:4	ADDRESS	ADDRESS	**Defines which x or y or z data will be sampled** 001: Y position 011: Z1 position 100: Z2 position 101: X position Others: Reserved
3	MODE	MODE	**Selects sample resolution** 0: 10-bit 1: 8-bit
2	SEDF	SEDF	**Selects mode** 0: Differential mode 1: Single-end mode
1:0	PD	PD	**Power-down control for analog IRQ signal and touch screen sample control signal** 00: Turn on Y-_drive signal and PDN_sh_ref 01: Turn on PDN_IRQ and PDN_sh_ref 10: Reserved 11: Turn on PDN_IRQ

A07900D8 AUX_TS_DAT1 Touch Screen Sample DATA 1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							TS_DAT									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	TS_DAT	TS_DAT	**Touch screen sample result data**

※ Please refer to ***"AM0000B6260 Analog Baseband.doc"*** for other AuxADC related settings:

- Sampling cycle control: Please see "AUX_CON4" and "AUX_CON5"

3.10.2 General Programming Guide

All register writes will occur in sequence. However, due to synchronization time, any reads after writes need to be delayed by three dummy reads.

3.10.3 Power on/off Programming Guide

The Die-to-Die(D2D) interface is required when using AUXADC. Please turn on/off the D2D when turning on/off AUXADC. Please follow the following sequence when turning on/off D2D:

AUXADC			
Turn on			
step	**Programming line**	**Register setting**	**W/R**
1	D die D2D 26MHz clock enable	(D2D_D_APC_AUX_CON1,16'h0001)	W
2	A die D2D **aux_en** setting	(D2D_A_APC_AUX_CON1,16'h8000)	W
3	Dummy read for order confirm	(D2D_A_APC_AUX_CON1,16'h8000)	R
4	D die D2D **aux_en** setting	(D2D_D_APC_AUX_CON1,16'h8001)	W
Turn off			
step	Programming line	Register setting	**W/R**
1	D die D2D **aux_en** disable setting	(D2D_D_APC_AUX_CON1,16'h0001)	W
2	A die D2D **aux_en** disable setting	(D2D_A_APC_AUX_CON1,16'h0000)	W
3	Wait for 50T(26MHz)		
4	D die D2D **f26m_aux_en** disable	(D2D_D_APC_AUX_CON1,16'h0000)	W

Table 4: Auxadc Registers

3.10.4 Usage Programming Guide

There are two modes to program the AUXADC to sample: immediate mode and synchronous mode. The following are notes on programming each mode:

Immediate mode sampling is accomplished by programming AUXADC_CON1 with the channels to be sampled. After programming, it is necessary to perform three dummy reads on AUXADC_CON3. After the dummy reads, the next read of AUXADC_CON3 will be valid. After sampling is done, it is necessary to program AUXADC_CON1 back to zero before sampling again.

Synchronous mode sampling is accomplished by programming AUXADC_CON0 with the channels to be sampled. Then it is necessary to program TDMA_EVTENA7 to 0x2. Please refer to ***"AK0000B6260 Timing Generator.doc"*** document for detailed register definitions. After the sample is done, TDMA_EVTENA7 must be programmed to 0x0 with waiting of two frames before sampling again.

3.10.5 Performance Programming Guide

For details on adjusting the performance of ADC sampling, please refer to registers AUX_CON4, AUX_CON5 and AUX_CON6 in the **"*AM0000B6260 Analog Baseband.doc*"** document.

3.10.6 AUXADC PDN

AUXADC is located in A-die. Due to limitation, one of UART1,2,3 clocks must be turned on in order to use AUXADC. Please clear the corresponding PDN bits to enable the AUXADC clock (bit 2 or 3 of ACFG_CLK_CG). Please refer to "MT6101_cfgsys.doc" for more information.

3.10.7 Notice

The 4 TP pins – PAD_XP, PAD_XM, PAD_YP, PAD_YM, are used as EINT pins when no R-touch is used in the system. Therefore, please make sure the pin settings are correct. Please refer to **"*MT6101_cfgsys.doc*"** for more information.

3.11 USB Device Controller

3.11.1 General Description

This chip provides a USB function interface which complies with Universal Serial Bus Specification Rev 1.1. The USB device controller supports only full-speed (12Mbps) operation. The chip can make use of this widely available USB interface to transmit/receive data with USB hosts, typically PC/laptop.

There are 6 endpoints in the USB device controller besides the mandatory control endpoint, 4 of which are for IN transactions and 2 are for OUT transactions. Word, half-word, and byte access are all allowed for loading and unloading the FIFO. The controller features 3 DMA channels to accelerate the data transfer for ACL and SCO data streams. The features of the endpoints are:

- Endpoint 0: The control endpoint feature 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. DMA transfer is not supported.
- IN endpoint 1: It features 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. GDMA Channel 4 Write Transfer is supported.
- IN endpoint 2: It features 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. GDMA Channel 6 Write Transfer is supported.
- IN endpoint 3: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.
- IN endpoint 4: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.
- OUT endpoint 1: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Channel 5 Read Transfer is supported.
- OUT endpoint 2: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Transfer is not supported.

For each endpoint except for the control endpoint, when the packet size is smaller than half the size of the FIFO, at most 2 packets can be buffered.

This unit is highly software configurable. All endpoints except for the control endpoint can be configured to be a bulk, interrupt or isochronous endpoints. Composite devices are also supported. IN endpoint 1 and OUT endpoint 1 share the same endpoint number, but they can be used separately. So is the situation for the endpoint 2.

The USB device uses the cable-powered feature for the transceiver but only drains little current. An internal pull-up resistor is integrated across Vbus and D+ signal. The switch on/off of the pull-up resistor can be configured through the internal register. Two additional external serial resistors might be needed to place on the output of D+ and D- signals to make the output impedance equivalent to 28 ~ 44Ohm.

***Figure 40**. USB11 controller system diagram*

3.11.2 Register Definition

A0900000 USB_FADDR **USB Function Address Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									UPD	FADDR						
Type									RW	RW						
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	UPD	**This is an 8-bit register that should be written with the functions 7-bit address (received through a SET_ADDRESS description).It is then used for decoding the function address in subsequent token packets.** When set by the MCU, the core will wait for an SOF token from the time INPKTRDY is set before sending the packet.
6:0	FADDR	**Function address of device**

A0900001 USB_POWER **USB Power Control Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name									ISOUPDATE			SWRSTENAB	RESET	RESUME	SUSPMODE	SUSPENAB
Type									RW			RU	RU	RW	RU	RW
Reset									0			0	0	0	0	0

Bit(s)	Name	Description
7	ISOUPDATE	When set by the MCU, the core will wait for an SOF token from the time INPKTRDY is set before sending the packet
4	SWRSTENAB	Set by the MCU to enable the mode in which the device can only be reset by the software after reset signals are detected on the bus. In case the software is delayed by other high priority processes and cannot make it to read the command from the buffer before the hardware resets the device after the reset signal is detected on the bus, the command will be lost. That is why the software reset mode is effective. When the flag is enabled, the hardware state machine cannot reset itself but by the software. In that sense, the software and hardware can keep synchronous detecting the reset signal.
3	RESET	The read-only bit is set when Reset signaling is present on the bus
2	RESUME	Set by the MCU to generate Resume signaling when the function is in the suspend mode. The MCU should clear this bit after 10ms (maximum 15ms) to end Resume signaling
1	SUSPMODE	Set by the USB core when the Suspend mode is entered. Cleared when the Resume bit of this register is set.
0	SUSPENAB	Set by the MCU to enable device into the Suspend mode when Suspend signaling is received on the bus.

A0900002 USB_INTRIN **USB IN Endpoints Interrupt Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												EP4_IN	EP3_IN	EP2_IN	EP1_IN	EP0
Type												RC	RC	RC	RC	RC
Reset												0	0	0	0	0

Bit(s)	Name	Description
4	EP4_IN	IN Endpoint 4 interrupt event
3	EP3_IN	IN Endpoint 3 interrupt event
2	EP2_IN	IN Endpoint 2 interrupt event
1	EP1_IN	IN Endpoint 1 interrupt event
0	EP0	Endpoint 0 interrupt event

A0900004 USB_INTROUT **USB OUT Endpoints Interrupt Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														EP2_OUT	EP1_OUT	
Type														RC	RC	
Reset														0	0	

Bit(s)	Name	Description
2	EP2_OUT	OUT Endpoint 2 interrupt event
1	EP1_OUT	OUT Endpoint 1 interrupt event

A0900006 USB_INTRUSB **USB General Interrupt Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												POWERDWN	SOF	RESET	RESUME	SUSPEND
Type												RC	RC	RC	RC	RC
Reset												0	0	0	0	0

Bit(s)	Name	Description
4	POWERDWN	**Set at SUSPMODE and LineState is JState** The programmer should have de bounce scheme in SW code when using this interrupt.
3	SOF	**Set at the start of each frame**
2	RESET	**Set when Reset signaling is detected on the bus**
1	RESUME	**Set when Resume signaling is detected on the bus while the USB core is in suspend mode**
0	SUSPEND	**Set when Suspend signaling is detected on the bus**

A0900007 USB_INTRINE **USB IN Endpoints Interrupt Enable Register** **00FF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												EP4_INE	EP3_INE	EP2_INE	EP1_INE	EP0_E
Type												RW	RW	RW	RW	RW
Reset												1	1	1	1	1

Bit(s)	Name	Description
4	EP4_INE	1b0: Disable IN Endpoint 4 interrupt event 1b1: Enable IN Endpoint 4 interrupt event
3	EP3_INE	1b0: Disable IN Endpoint 3 interrupt event 1b1: Enable IN Endpoint 3 interrupt event
2	EP2_INE	1b0: Disable IN Endpoint 2 interrupt event 1b1: Enable IN Endpoint 2 interrupt event
1	EP1_INE	1b0: Disable IN Endpoint 1 interrupt event 1b1: Enable IN Endpoint 1 interrupt event
0	EP0_E	1b0: Disable IN Endpoint 0 interrupt event 1b1: Enable IN Endpoint 0 interrupt event

A0900009 USB_INTROUTE **USB OUT Endpoints Interrupt Enable Register** **00FE**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														EP2_OUTE	EP1_OUTE	
Type														RW	RW	
Reset														1	1	

Bit(s)	Name	Description
2	EP2_OUTE	1b0: Disable OUT Endpoint 2 interrupt event 1b1: Enable OUT Endpoint 2 interrupt event
1	EP1_OUTE	1b0: Disable OUT Endpoint 1 interrupt event 1b1: Enable OUT Endpoint 1 interrupt event

A090000B INTRUSBE **USB General Interrupt Enable Register** **06**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												POWERDWN_E	SOF_E	RESET_E	RESEUM_E	SUSPEND_E
Type												RW	RW	RW	RW	RW
Reset												0	0	1	1	0

Bit(s)	Name	Description
4	POWERDWN_E	Enables power-down interrupt
3	SOF_E	Enables SOF interrupt
2	RESET_E	Enables Reset/Babble interrupt
1	RESEUM_E	Enables Resume interrupt
0	SUSPEND_E	Enables Suspend interrupt

A090000C USB_FRAME1 **USB Frame Count #1 Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									NUML							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	NUML	The lower 8 bits of the frame number

A090000D USB_FRAME2 **USB Frame Count #2 Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														NUMH		
Type														RU		
Reset														0	0	0

Bit(s)	Name	Description
2:0	NUMH	The upper 3 bits of the frame number

A090000E USB_INDEX **USB Endpoint Register Index** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													INDEX			
Type													RW			
Reset													0	0	0	0

The register determines which endpoint control/status registers are to be accessed at addresses USB+10h to USB+17h. Each IN endpoint and OUT endpoint has it own set of control/status registers. Only one set of IN control/status and one set of OUT control/status registers appear in the memory map at a time. Before accessing the control/status registers of an endpoints, the endpoint number should be written to the USB_INDEX register to ensure that the correct control/status registers appear in the memory map.

Bit(s)	Name	Description
3:0	INDEX	Index of the endpoint

A090000F USB_RSTCTRL **USB Reset Control** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SWRST				RSTCNTR			
Type									RW				RW			
Reset									0				0	0	0	0

Bit(s)	Name	Description
7	SWRST	If the flag SWRSTENAB in register USB_POWER is set to 1, the software enable mode will be enabled, and the device can be reset by writing 1 to this flag.
3:0	RSTCNTR	**Signifies the duration of the reset operation after reset signal is detected on the bus** It is only enabled when software reset is not enabled. If the value is 0, the duration will be 2.5us. Otherwise, the duration will be this value multiplied by 341 then added by 2.5 (unit: us). The range consequently starts from 2.5us to 5122.5us

A0900010 USB_EP_INMAXP **USB Maximum Packet Size Register for IN Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MAXP							
Type									RW							
Reset									0	0	0	0	0	0	0	0

The register holds the maximum packet size for transactions through the currently selected IN endpoint - in units of one byte. When setting up the value, the programmer should note the constraints placed by the USB Specification on the packet size for bulk interrupt, and isochronous transactions in full speed operation. There is an INMAXP register for each IN endpoint except for endpoint 0. The registers are active when the USB_INDEX register is set to 1, 2, 3, and 4 respectively.

The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch may cause unexpected results. If a value is bigger than the configured IN FIFO size for the endpoint written to the register, the value will be automatically changed to the IN FIFO size. If the value written to the register is smaller than or equal to half the IN FIFO size, two IN packets can be buffered. The configured IN FIFO size for the endpoint 1, 2, and 3 are 16 bytes, 64 bytes and 64 bytes, respectively.

The register is reset to 0. If the register is changed after the packets are sent from the endpoint, the endpoint IN FIFO should be completely flushed after writing the new value to the register.

Bit(s)	Name	Description
7:0	MAXP	Maximum packet size (unit: one byte)

A0900011 USB_EP0_CSR **USB Control/Status Register for Endpoint 0** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SSETUPEN	SOUTPKTR	SENDSTAL	SETUPEND	DATAEND	SENTSTALL	INPKTRDY	OUTPKTRD

									D	DY	L					Y
Type									A0	A0	A0	RU	A0	A1	A0	RU
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	SSETUPEND	The MCU writes 1 to this bit to clear the SETUPEND bit. It is cleared automatically. Only active when a transaction is started.
6	SOUTPKTRDY	The MCU writes 1 to this bit to clear the OUTPKTRDY bit. It is cleared automatically. Only active when an OUT transaction is started.
5	SENDSTALL	The MCU writes 1 to this bit to terminate the current transaction. The STALL handshake will be transmitted, and this bit will be cleared automatically.
4	SETUPEND	This bit will be set when a control transaction ends before the DATAEND bit is set. An interrupt will be generated and FIFO flushed at this time. The bit is cleared by the MCU writing 1 to the SSETUPEND bit.
3	DATAEND	The MCU sets up this bit: 1. When setting INPKTRDY for the last data packet 2. When clearing OUTPKTRDY after unloading the last data packet 3. When setting INPKTRDY for a zero length data packet It is cleared automatically.
2	SENTSTALL	This bit is set when a STALL handshake is transmitted. The MCU should clear this bit by writing 0 to it.
1	INPKTRDY	The MCU sets up this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet is transmitted. An interrupt is generated when this bit is set.
0	OUTPKTRDY	This bit is set when a data packet is received. An interrupt is generated when this bit is set. The MCU clears this bit by setting up the SOUTPKTRDY bit.

A0900011 USB_EP_INCSR1 **USB Control/Status Register #1 for IN Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ABORTPKT_EN	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONOTEMPTY	INPKTRDY
Type									RW	A0	A1	RW	A0	A1	RU	A0
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	ABORTPKT_EN	When MCU writes 1 to ABORTPKT_EN, FLUSHFIFO switches to abort the packet function. This bit should be enabled before FLUSHFIFO is set. If FLUSHFIFO is set and ABORTPKT_EN is enabled the data loaded into FIFO will be discarded. After the packet is aborted, EP will issue an interrupt. The programmer should wait for this interrupt to make sure the packet is aborted
6	CLRDATATOG	The MCU writes 1 to this bit to reset the endpoint IN data toggle to 0
5	SENTSTALL	The bit is set when a STALL handshake is transmitted. The FIFO is flushed and the INPKTRDY bit is cleared. The MCU should clear this bit by writing 0 to this bit
4	SENDSTALL	The MCU writes 1 to this bit to issue a STALL handshake to an IN token. The MCU clears this bit to terminate the stall condition.
3	FLUSHFIFO	The MCU writes 1 to this bit to flush the next packet to be transmitted from the endpoint IN FIFO. The FIFO pointer is reset and the INPKTRDY bit is cleared. If the FIFO contains two packets, FLUSHFIFO will need to be set twice to completely clear the FIFO. FLUSHFIFO should only be used when INPKTRDY is set. At other times, it may cause data corruption. If ABORTPKT_EN is enabled and this bit is set , the function of this bit will become ABORTPKT to abort the next packet to be transmitted from the endpoint IN FIFO and does not need to set INPKTRDY. It is the same with the

Bit(s)	Name	Description
		FLUSHFIFO function. This bit is only active when the endpoint is idle.
2	UNDERRUN	In isochronous mode, this bit is set when a zero length data packet is sent after receiving an IN token with the INPKTRDY bit not set. In Bulk/Interrupt mode, this bit is set when a NAK is returned in response to an IN token. The MCU should clear this bit by writing a 0 to this bit.
1	FIFONOTEMPTY	This bit is set when there is at least 1 packet in the IN FIFO.
0	INPKTRDY	The MCU sets up this bit after loading a data packet into the FIFO. Only active when an IN transaction is started. It is cleared automatically when a data packet is transmitted. An interrupt will be generated (if enabled) when the bit is cleared

A0900012 USB_EP_INCSR2 **USB Control/Status Register #2 for IN Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTOSET	ISO	MODE	DMAENAB	FRCDATATOG			
Type									RW	RW	RW	RW	RW			
Reset									0	0	0	0	0			

Bit(s)	Name	Description
7	AUTOSET	If the MCU sets up the bit, INPKTRDY will be automatically set when the data of the maximum packet size (value in INMAXP) is loaded into the IN FIFO. If a packet of smaller than the maximum packet size is loaded, INPKTRDY will have to be set manually. When 2 packets are in the IN FIFO, INPKTRDY will also be automatically set when the first packet is sent if the second packet is the maximum packet size.
6	ISO	The MCU sets up this bit to enable the IN endpoint for isochronous transfer, and clears it to enable the IN endpoint for bulk/interrupt transfers
5	MODE	The MCU sets this bit to enable the endpoint direction as IN and clears it to enable the endpoint direction as OUT. Its valid only where the same endpoint FIFO is used for both IN and OUT transaction
4	DMAENAB	The MCU sets up this bit to enable the DMA request for the IN endpoint.
3	FRCDATATOG	The MCU sets up this bit to force the endpoints IN data toggle to switch after each data packet is sent regardless of whether an ACK has been received. This can be used by interrupt IN endpoints which are used to communicate rate feedback for isochronous endpoints.

A0900013 USB_EP_OUTMAXP **USB Maximum Packet Size Register for OUT Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MAXP							
Type									RW							
Reset									0	0	0	0	0	0	0	0

This register holds the maximum packet size for transactions through the currently selected OUT endpoint (unit: one byte). When setting up this value, the programmer should note the constraints placed by the USB specification on packet sizes for bulk, interrupt, and isochronous transactions in full speed operations. There is an OUTMAXP register for each OUT endpoint except for endpoint 0. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch may cause unexpected results. The total amount of data represented by the

value written to this register must not exceed the FIFO size for the OUT endpoint, and should not exceed half the FIFO size if double buffering is required. If a value is bigger than the configured OUT FIFO size for the endpoint is written to the register, the value will be automatically changed to the OUT FIFO size. If the value written to the register is smaller than or equal to half the OUT FIFO size, two OUT packets can be buffered. The configured IN FIFO size for the endpoint 1, 2, and 3 are both 16, 64, and 64 bytes respectively.

Bit(s)	Name	Description
7:0	MAXP	**M**aximum packet size (unit: one byte)

A0900014 USB_EP_OUTCSR1 USB Control/Status Register #1 for OUT Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CLRDTATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERROR	OVERRUN	FIFOFULL	RXPKTRDY
Type									A0	A1	RW	A0	RU	A1	RU	A1
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	CLRDTATOG	The MCU writes 1 to this bit to reset the endpoint data toggle to 0.
6	SENTSTALL	The bit is set when a STALL handshake is transmitted. The MCU should clear this bit by writing 0 to it.
5	SENDSTALL	The MCU writes 1 to this bit to issue a STALL handshake. The MCU clears this bit to terminate the stall condition. This bit will have no effect if the OUT endpoint is in the isochronous mode.
4	FLUSHFIFO	The MCU writes 1 to this bit to flush the next packet to be read from the endpoint OUT FIFO. If the FIFO contains two packets, FLUSHFIFO will need to be set twice to completely clear the FIFO. FLUSHFIFO should only be used when OUTPKTRDY is set. At other times, it may cause data corruption.
3	DATAERROR	The bit is set when OUTPKTRDY is set if the data packet has a CRC or bit stuff error. It is cleared when OUTPKTRDY is cleared. This bit is only valid in the isochronous mode.
2	OVERRUN	The bit will be set if an OUT packet cannot be loaded into the OUT FIFO. The MCU should clear the bit by writing 0 to it. This bit is only valid in the isochronous mode.
1	FIFOFULL	This bit is set when no more packets can be loaded into the OUT FIFO
0	RXPKTRDY	The bit is set when a data packet has been received. The MCU should clear (write 0 to it) the bit when the packet is unloaded from the OUT FIFO. An interrupt will be generated when the bit is set. When the null packet is received, OUTPKTRDY will be set after USB_INTROUT1 is high.

A0900015 USB_EP_OUTCSR2 USB Control/Status Register #2 for OUT Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTOCLEAR	ISO	DMAENAB	DMAMODE				
Type									RW	RW	RW	RW				
Reset									0	0	0	0				

Bit(s)	Name	Description
7	AUTOCLEAR	If the MCU sets up this bit, the OUTPKTRDY bit will be automatically cleared when a packet of OUTMAXP bytes is unloaded from the OUT FIFO. When packets of smaller than the maximum packet size are unloaded, OUTPKTRDY

Bit(s)	Name	Description
		will have to be cleared manually.
6	ISO	The MCU sets up this bit to enable the OUT endpoint for isochronous transfers and clears it to enable the OUT endpoint for bulk/interrupt transfers
5	DMAENAB	The MCU sets up this bit to enable the DMA request for the OUT endpoint.
4	DMAMODE	Two modes of DMA operation are supported: DMA mode 0 in which a DMA request is generated for all received packets, together with an interrupt (if enabled). DMA mode 1 in which a DMA request (but no interrupt) is generated for OUT packets of size OUTMAXP bytes and an interrupt (but no DMA request) is generated for OUT packets of any other size. The MCU sets up the bit to select DMA mode 1 and clears this bit to select DMA mode 0

A0900016 USB_EP0_COUNT **EP0 Received Bytes Count Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										EP0_COUNT						
Type										RU						
Reset										0	0	0	0	0	0	0

Bit(s)	Name	Description
6:0	EP0_COUNT	**Number of received data bytes in endpoint 0** The value returned is valid while the OUTPKTRDY bit of the USB_EP0_CSR register is set. The register is active when the USB_INDEX register is set to 0.

A0900016 USB_EP_COUNT1 **USB OUT Endpoint Byte Counter Register LSB Part for Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									NUML							
Type									RU							
Reset									0	0	0	0	0	0	0	0

The register holds the lower 8 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY in register USB_OUTCSR1 is set. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

Bit(s)	Name	Description
7:0	NUML	The lower 8 bits of the number of received data bytes for the OUT endpoint

A0900017 USB_EP_COUNT2 **USB OUT Endpoint Byte Counter Register MSB Part for Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														NUMH		
Type														RU		
Reset														0	0	0

The register holds the upper 3 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY in register USB_EP_OUTCSR1 is set. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

Bit(s)	Name	Description
2:0	NUMH	The upper 8 bits of the number of received data bytes for the OUT endpoint

A0900020 USB_EP0_FIFO_DB0 **USB Endpoint 0 FIFO Register DB0** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 0. Writing to this register will load data to the FIFO for the endpoint 0. Reading this register will unload data from the FIFO for the endpoint 0.

The register provides word, half word and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO0_DB0	The first byte to be loaded into or unloaded from the FIFO

A0900021 USB_EP0_FIFO_DB1 **USB Endpoint 0 FIFO Register DB1** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO0_DB1	The second byte to be loaded into or unloaded from the FIFO

A0900022 USB_EP0_FIFO_DB2 **USB Endpoint 0 FIFO Register DB2** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO0_DB2	The third byte to be loaded into or unloaded from the FIFO

A0900023 USB_EP0_FIFO_DB3 **USB Endpoint 0 FIFO Register DB3** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO0_DB3	The fourth byte to be loaded into or unloaded from the FIFO

A0900024 USB_EP1_FIFO_DB0 USB Endpoint 1 FIFO Register DB0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO1_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 1. Writing to this register will load data to the FIFO for the endpoint 1. Reading this register will unload data from the FIFO for the endpoint 1.

The register provides word, half word and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO1_DB0	The first byte to be loaded into or unloaded from the FIFO

A0900025 USB_EP1_FIFO_DB1 USB Endpoint 1 FIFO Register DB1 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO1_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB1	The second byte to be loaded into or unloaded from the FIFO

A0900026 USB_EP1_FIFO_DB2 USB Endpoint 1 FIFO Register DB2 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO1_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB2	The third byte to be loaded into or unloaded from the FIFO

A0900027 USB_EP1_FIFO_DB3 USB Endpoint 1 FIFO Register DB3 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO1_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB3	The fourth byte to be loaded into or unloaded from the FIFO

A0900028 USB_EP2_FIFO_DB0 USB Endpoint 2 FIFO Register DB0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO2_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 2. Writing to this register will load data to the FIFO for the endpoint 2. Reading this register will unloads data from the FIFO for the endpoint 2.

The register provides word, half word and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO2_DB0	The first byte to be loaded into or unloaded from the FIFO

A0900029 USB_EP2_FIFO_DB1 USB Endpoint 2 FIFO Register DB1 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO2_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB1	The second byte to be loaded into or unloaded from the FIFO

A090002A USB_EP2_FIFO_DB2 USB Endpoint 2 FIFO Register DB2 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO2_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB2	The third byte to be loaded into or unloaded from the FIFO

A090002B USB_EP2_FIFO_DB3 USB Endpoint 2 FIFO Register DB3 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO2_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB3	The fourth byte to be loaded into or unloaded from the FIFO

A090002C USB_EP3_FIFO_DB0 USB Endpoint 3 FIFO Register DB0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO3_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 3. Writing to this register will load data to the FIFO for the endpoint 3. Reading this register will unload data from the FIFO for the endpoint 3.

The register provides word, half word and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO3_DB0	The first byte to be loaded into or unloaded from the FIFO

A090002D USB_EP3_FIFO_DB1 **USB Endpoint 3 FIFO Register DB1** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO3_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB1	The second byte to be loaded into or unloaded from the FIFO

A090002E USB_EP3_FIFO_DB2 **USB Endpoint 3 FIFO Register DB2** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO3_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB2	The third byte to be loaded into or unloaded from the FIFO

A090002F USB_EP3_FIFO_DB3 **USB Endpoint 3 FIFO Register DB3** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO3_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB3	The fourth byte to be loaded into or unloaded from the FIFO

A0900030 USB_EP4_FIFO_DB0 **USB Endpoint 4 FIFO Register DB0** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 4. Writing to this register will load data into the FIFO for the endpoint 4. Reading this register will unload data from the FIFO for the endpoint 4.

The register provides word, half word and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO4_DB0	The first byte to be loaded into or unloaded from the FIFO

A0900031 USB_EP4_FIFO_DB1 USB Endpoint 4 FIFO Register DB1 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO4_DB1	The second byte to be loaded into or unloaded from the FIFO

A0900032 USB_EP4_FIFO_DB2 USB Endpoint 4 FIFO Register DB2 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO4_DB2	The third byte to be loaded into or unloaded from the FIFO

A0900033 USB_EP4_FIFO_DB3 USB Endpoint 4 FIFO Register DB3 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO4_DB3	The fourth byte to be loaded into or unloaded from the FIFO

A0900240 USB_CON USB PHY Control 20

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											NULLPKT_FIX				DMPULLUP	DPPULLUP
Type											RW				RW	RW
Reset											1				0	0

Bit(s)	Name	Description
5	NULLPKT_FIX	If NULLPKT_FIX is set to 1, the USB controller will not issue a DMAreq when a null packet is received.
1	DMPULLUP	**Pull-up enable pin** Enables the 1.5KOhm pull-up on D- pin as a full-speed device by setting it to

Bit(s)	Name	Description
		high.
0	DPPULLUP	**Pull-up enable pin** Enables the 1.5KOhm pull-up on D+ pin as a full-speed device by setting it to high.

3.12 Accessory Detector

1.1.1 General Description

The hardware accessory detector (ACCDET) detects plug-in/out of multiple types of external components. Based on the suggested circuit (see Figure 41), this design supports 3 types of external components, which are microphone, hook-switch and TV-OUT line. This design uses the internal 2-bit comparator to separate external components. The de-bounce scheme is also supported to resist uncertain input noises. When the plug-in/out state is stable, the PWM unit of ACCDET will enable the comparator, MBIAS and threshold voltage of the comparator periodically for the plugging detection. With suitable PWM settings, very low-power consumption can be achieved when the detection feature is enabled. In order to compensate the delay between the detection login and comparator, the delay enabling scheme is adopted. Given the suitable delay number compared to the rising edge of PWM high pulse, the stable plugging state can be prorogated to digital detection logic. Then the correct plugging state can be detected and reported.

Figure 42 shows the state machine without TV-OUT mode. The state machine is executed by the software to control the ACCDET design. The ACCDET design will send one interrupt to acknowledge the software after the ACCDET input state is changed and the duration of the state is longer than de-bounce time. The software needs to read out the memorized ACCDET input state and follow the recommend state machine to program the register in it.

Figure 41. Suggested Accessory Detection Circuit.

Figure 42. The Sate machine between Microphone and Hook-Switch plug-in/out change.

1.1.2 Pulse Width Modulation

The ACCDET design also provides one Pulse-Width-Modulation (PWM) to enable the comparator, microphone's bias current and the threshold voltage of the comparator periodically. With suitable PWM and settings for delayed enabling, the ACCDET can achieve very low power consumption and accurate plug-in/out detection. Figure 43 shows a timing diagram example of such PWM design. The output from PWM keeps being at "0" until the value of the counter is smaller than the programmed threshold.

Figure 43. PWM waveform.

1.1.3 Register Definition

Module name: ACCDET base address: (+A0750000h)

Address	Name	Width	Register function
A0750000	ACCDET_RSTB	32	ACCDET software reset register
A0750004	ACCDET_CTRL	32	ACCDET control register
A0750008	ACCDET_STATE_SWCTRL	32	ACCDET state switch control register
A075000C	ACCDET_PWM_WIDTH	32	ACCDET PWM width register
A0750010	ACCDET_PWM_THRESH	32	ACCDET PWM threshold register
A0750024	ACCDET_EN_DELAY_NUM	32	ACCDET enable delay number register
A0750028	ACCDET_PWM_IDLE_VALUE	32	ACCDET PWM IDLE value register
A075002C	ACCDET_DEBOUNCE0	32	ACCDET debounce0 register
A0750030	ACCDET_DEBOUNCE1	32	ACCDET debounce1 register
A0750038	ACCDET_DEBOUNCE3	32	ACCDET debounce3 register
A075003C	ACCDET_IRQ_STS	32	ACCDET interrupt status register
A0750040	ACCDET_CURR_IN	32	ACCDET current input status register
A0750044	ACCDET_SAMPLE_IN	32	ACCDET sampled input status register
A0750048	ACCDET_MEMOIZED_IN	32	ACCDET memorized input status register
A075004C	ACCDET_LAST_MEMOIZED_IN	32	ACCDET last memorized input status register
A0750050	ACCDET_FSM_STATE	32	ACCDET FSM status register
A0750054	ACCDET_CURR_DEBOUNCE	32	ACCDET current de-bounce status register
A0750058	ACCDET_VERSION	32	ACCDET version code
A075005C	ACCDET_IN_DEFAULT	32	default value of accdet_in

A0750000 ACCDET_RSTB ACCDET Software Reset Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																RSTB
Type																RW
Reset																1

Overview: After applying the setting to register, software reset is necessary for state initialization. Without this process, ACCDET may detect incorrect plug state.

Bit(s)	Mnemonic	Name	Description
0	**RSTB**	RSTB	**Set to 0 to reset the ACCDET unit and set to 1 after the reset process is finished.** This software reset will clear ACCDET's enable signal but keep all ACCDET's settings. After the reset process, ACCDET will return to the IDLE state.

A0750004 ACCDET_CTRL ACCDET Control Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																ACCDET_EN
Type																RW
Reset																1

Bit(s)	Mnemonic	Name	Description
0	**ACCDET_EN**	EN	Set to 1 to enable the ACCDET unit.

A0750008 ACCDET_STATE_SWCTRL ACCDET State Switch Control Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												MBIAS_PWM_EN	VTH_PWM_EN	CMP_PWM_EN		
Type												RW	RW	RW		
Reset												0	0	0		

Bit(s)	Mnemonic	Name	Description
4	**MBIAS_PW**	MBIAS_PWM_EN	Enables PWM of ACCDET MBIAS unit

Bit(s)	Mnemonic	Name	Description
	M_EN		
3	**VTH_PWM_EN**	VTH_PWM_EN	Enables PWM of ACCDET voltage threshold unit
2	**CMP_PWM_EN**	CMP_PWM_EN	Enables PWM of ACCDET comparator

A075000C **ACCDET_PWM_WIDTH** **ACCDET PWM Width Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	PWM_WIDTH															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**PWM_WIDTH**	PWM_WIDTH	**ACCDET PWM width** It is PWM max. counter value. It will be the initial value for the internal counter. The PWM internal counter always counts down to zero to finish one compete period, and the value of the internal counter will return to the value of PWM_WIDTH. PWM output frequency = (32k/PWM_WIDTH) Hz.

A0750010 **ACCDET_PWM_THRESH** **ACCDET PWM Threshold Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	PWM_THRESH															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**PWM_THRESH**	PWM_THRESH	**ACCDET PWM threshold** When the internal counter value is bigger than or equal to PWM_THRESH, the PWM output signal will be "0". When the internal counter is smaller than PWM_THRESH, the PWM output signal will be "1". PWM output duty cycle = (PWM_THRESH)x(1/32) ms.

Figure 1 shows the PWM waveform with register value present.

Figure 1. PWM waveform with register value present

A0750024 ACCDET_EN_DELAY_NUM ACCDET Enable Delay Number Register 00000101

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**FALL_DELAY_NUM**	**RISE_DELAY_NUM**														
Type	RW	RW														
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
15	**FALL_DELAY_NUM**	FALL_DELAY_NUM	**Falling delay cycle compared to CMP PWM waveform** In order to make sure the plug state is stable after disabling ACCDET, the suitable delay cycle is necessary. This number indicates the clock cycle number between the point when the digital part of ACCDET stops receiving accdet_in and the point when the analog part of ACCDET stops working.
14:0	**RISE_DELAY_NUM**	RISE_DELAY_NUM	**Rising delay cycle compared to PWM waveform** In order to make sure the plug state is stable before activating ACCDET, the suitable delay cycle is necessary. This number indicates the clock cycle number between the point when the analog part of ACCDET starts working and the point when the digital part of ACCDET starts receiving stable accdet_in. This number should be fine tuned depending on different project requirements.

A0750028 ACCDET_PWM_IDLE_VALUE ACCDET PWM IDLE Value Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														**MBIAS**	**VTH**	**CMP**
Type														RW	RW	RW
Reset														0	0	1

Bit(s)	Mnemonic	Name	Description
2	**MBIAS**	MBIAS	IDLE value of MBIAS PWM

Bit(s)	Mnemonic	Name	Description
1	**VTH**	VTH	IDLE value of VTH PWM
0	**CMP**	CMP	IDLE value of CMP PWM

A075002C **ACCDET_DEBOUNCE0** **ACCDET Debounce0 Register** **00000010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE0															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	**DEBOUNCE0**	DEBOUNCE0	**De-bounce time control of the next state = 2'b00** De-bounce time = DEBOUNCE/32 ms

A0750030 **ACCDET_DEBOUNCE1** **ACCDET Debounce1 Register** **00000010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	**DEBOUNCE1**	DEBOUNCE1	**De-bounce time control of the next state = 2'b01** De-bounce time = DEBOUNCE/32 ms

A0750034 **ACCDET_DEBOUNCE2** **ACCDET Debounce2 Register** **00000010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE2															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	**DEBOUNCE2**	DEBOUNCE2	**De-bounce time control of the next state = 2'b10** De-bounce time = DEBOUNCE/32 ms

A0750038 ACCDET_DEBOUNCE3 ACCDET Debounce3 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE3															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	**DEBOUNCE3**	DEBOUNCE3	**De-bounce time control of the next state = 2'b11** De-bounce time = DEBOUNCE/32 ms

A075003C ACCDET_IRQ_STS ACCDET Interrupt Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne								**IRQ_CLR**								**IRQ**
Type								RW								RW
Reset								0								0

Overview: When the interrupt of ACCDET is asserted, IRQ_CLR must be set to 1 to clear the interrupt status. This bit will pause all activities in the ACCDET design until both interrupt status and IRQ_CLR are cleared. The software should write 1 to IRQ_CLR first to clear the interrupt (IRQ). After that, if pclk gating is enabled, the software should read ACCDET_IRQ_STS again to make IRQ_CLR self-reset to 0.

Bit(s)	Mnemonic	Name	Description

Bit(s)	Mnemonic	Name	Description
8	**IRQ_CLR**	IRQ_CLR	**Clears interrupt status of ACCDET unit**
0	**IRQ**	IRQ	**Interrupt status of ACCDET unit** Because this register will be cleared by hardware, the interrupt edge-sensitive scheme should be adopted for this design.

A0750040 ACCDET_CURR_IN ACCDET Current Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															CURR_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**CURR_IN**	CURR_IN	Current input status of ACCDET unit

A0750044 ACCDET_SAMPLE_IN ACCDET Sampled Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															SAMPLE_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**SAMPLE_IN**	SAMPLE_IN	**Samples input status of ACCDET unit** When the plug-in/out state is changed, the ACCDET unit will do sampling.

A0750048 ACCDET_MEMOIZED_IN ACCDET Memorized Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															MEMORIZED_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description

Bit(s)	Mnemonic	Name	Description
1:0	**MEMORIZED_IN**	MEMORIZED_IN	**Memorized input status of ACCDET unit** When the plug-in/out states is changed and held longer than the de-bounce time, the ACCDET unit will save the sampled input state to the memorized state. The interrupt will also be asserted to acknowledge the software.

A075004C ACCDET_LAST_MEMOIZED_IN ACCDET Last Memorized Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															LAST_MEMORIZED_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**LAST_MEMORIZED_IN**	LAST_MEMORIZED_IN	Last memorized input status of ACCDET unit

A0750050 ACCDET_FSM_STATE ACCDET FSM Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														FSM_STATE		
Type														RO		
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**FSM_STATE**	FSM_SATE	**State of ACCDET unit finite-state-machine** 0: ACCDET_IDLE 1: ACCDET_SAMPLE 2: ACCDET_DEBOUNCE 3: ACCDET_CHECK 4: ACCDET_MEMORIZED 5: ACCDET_IRQ

A0750054 ACCDET_CURR_DEBOUNCE ACCDET Current De-bounce Status Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Mne	CURR_DEBOUNCE															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**CURR_DEBOUNCE**	CURR_DEBOUNCE	Currently used de-bounce time setting

A0750058 **ACCDET_VERSION** **ACCDET Version Code** **000000 03**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**ACCDET_VERSION**	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**ACCDET_VERSION**	ACCDET_VERSION	Version code for ACCDET

A075005C **ACCDET_IN_DEFAULT** **Default Value of accdet_in** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**ACCDET_IN_DEFAULT_REFRESH_EN**			**ACCDET_IN_DEFAULT**	
Type												RW			RW	
Reset												0			0	0

Overview: The default value of sample_accdet_in and memorised_accdet_in can be set by software instead of using the default value set by hardware(i.e. 3). ACCDET_DEFAULT_REFRESH_EN is the enable bit controlling whether to use this additional function. The value of sample_accdet_in and memorized_accdet_in will change when accdet_en rises from low to high. *Note that if software reset is applied when accdet_en is high, the default value of sample_accdet_in and memorized_accdet_in will also be loaded when the software reset is de-asserted.*

Bit(s)	Mnemonic	Name	Description
4	**ACCDET_IN_DEFAULT_REFR**	ACCDET_IN_DEFAULT_REFRESH_EN	**Enable signal for whether to load accdet_in_default** 0: accdet_in_default will not be loaded. 1: accdet_in_default will be loaded.

Bit(s)	Mnemonic	Name	Description
	ESH_EN		
1:0	**ACCDET_IN_DEFAULT**	ACCDET_IN_DEFAULT	**Default value of accdet_in set by software**

3.13 SD Memory Card Controller

3.13.1 Introduction

The controller fully supports the SD memory card bus protocol as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 2.0.

Furthermore, the controller also partially supports the SDIO card specification version 2.0. However, the controller can only be configured as the host of the SD memory card. Hereafter, the controller is also abbreviated as the SD controller. The following are the main features of the controller.

- Interface with MCU by APB bus
- 16/32-bit access on APB bus
- 16/32-bit access for control registers
- 32-bit access for FIFO
- Built-in 32 bytes FIFO buffers for transmit and receive, FIFO is shared for transmit and receive
- Built-in CRC circuit
- CRC generation can be disabled
- DMA supported
- Interrupt capabilities
- Data rate up to 48 Mbps in serial mode, 48x4 Mbps in parallel model, the module is targeted at 48 MHz operating clock
- Serial clock rate on SD bus is programmable
- Card detection capabilities during sleep mode
- Controllability of power for memory card
- Does not support SPI mode for SD memory card
- Does not support multiple SD memory cards

3.13.2 Overview

3.13.2.1 Pin Assignment

The following lists pins required for the SD memory card. Table 48 shows how the pins are shared. Note that all I/O pads have embedded both pull-up and pull-down resistors because they are shared by the SD memory card. The pull-down resistors for these pins can be used for power saving. If optimal pull-up or pull-down resistors are required on the system board, all embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers. The VDDPD pin is used for power saving. Power for the SD memory card can be shut down by programming the corresponding control register. The WP (Write Protection) pin is used to detect the status of the Write Protection Switch on the SD memory card.

Table 48: Sharing of pins for SD memory card controller

3.13.2.2 Card Detection

For SD memory card, detection of card insertion/removal by hardware is supported. Because a pull-down resistor with about 470 KΩ resistance which is impractical to embed in an I/O pad is needed on the signal CD/DAT3, and it has to be capable of being connected or disconnected dynamically onto the signal CD during initialization period, an additional I/O pad is needed to switch on/off the pull-down resistor on the system board. The card detection for SD memory card is shown in Figure 45. Before SD memory card is inserted or powered on, SW1 and SW2 must be opened for card detection on the host side. Meanwhile, the pull-down resistor R_{CD} on the system board must attach onto the signal CD/DAT3 by the output signal RCDEN. In addition, the default position of SW3 on the card is closed.

Upon insertion of the SD memory card, the signal CD/DAT3 will transit from low to high. If the SD memory card is removed, the CD/DAT3 signal will return to logic low state. After the card identification process, the pull-down resistor R_{CD} on the system board shall be disconnected with the CD/DAT3 signal, and SW3 on the card must be opened for normal operation.

Since the scheme above needs a mechanical switch such as a relay on the system board, it is not an ideal scheme. Thus, a dedicated pin “INS” is used to perform card insertion and removal for SD. The pin “INS” will be connected to the pin “VSS2” of a SD connector (see Figure 44).

Note that this SD memory card controller just support scheme 2 which use dedicated pin “INS” to perform card insertion and removal for SD.

Figure 44: ***Card detection for SD memory card (Scheme 1)***

Figure 45: ***Card detection for SD memory card (Scheme 2)***

3.13.3 Register Definition

Module name: MSDC0 base address: (+A0130000h)

Address	Name	Width	Register function
A0130000	**MSDC_CFG**	32	**SD memory card controller configuration register** For general configuration of the SD controller. *Note: MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0130004	**MSDC_STA**	32	**SD memory card controller status register** Contains the status of FIFO, interrupts and data requests.
A0130008	**MSDC_INT**	32	**SD memory card controller interrupt register** Contains the status of interrupts. Note that the register still shows the status of interrupt even though the interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting up register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.
A013000C	**MSDC_PS**	32	**SD memory card pin status register** Used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull down when the system is powered on. The scheme of card detection for MS is the same as that for SD. For detecting card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.
A0130010	**MSDC_DAT**	32	**SD memory card controller data register** Reads/Writes data from/to FIFO inside SD controller. Data access unit: 32 bits
A0130014	**MSDC_IOCON**	32	**SD memory card controller IO control register** Specifies output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current will depend on the PAD type selected for the chip.
A0130018	**MSDC_IOCON1**	32	**SD memory card controller IO control register 1**
A0130020	**SDC_CFG**	32	**SD memory card controller configuration register** Configures the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is

Address	Name	Width	Register function
			configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. *Note: SDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0130024	**SDC_CMD**	32	**SD memory card controller command register** Defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.
A0130028	**SDC_ARG**	32	**SD memory card controller argument register** Contains argument of the SD memory card command.
A013002C	**SDC_STA**	32	**SD memory card controller status register** Contains various statuses of SD controller as the controller is configured as the host of SD memory card.
A0130030	**SDC_RESP0**	32	**SD memory card controller response register 0** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0130034	**SDC_RESP1**	32	**SD memory card controller response register 1** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0130038	**SDC_RESP2**	32	**SD memory card controller response register 2** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A013003C	**SDC_RESP3**	32	**SD memory card controller response register 3** Contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For responses of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.
A0130040	**SDC_CMDSTA**	32	**SD memory card controller command status register** Contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register will also be used as the interrupt source. The register is cleared when

Address	Name	Width	Register function
			being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0130044	**SDC_DATSTA**	32	**SD memory card controller data status register** Contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register is also used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0130048	**SDC_CSTA**	32	**SD memory card status register** After commands with R1 and R1b response this register containing the status of the SD card, it will be used as the response interrupt source. In all register fields, logic high indicates error, and logic low indicates there is no error. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A013004C	**SDC_IRQMASK0**	32	**SD memory card IRQ mask register 0** Contains parts of SD memory card interrupt mask register. See the descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. ‘1’ in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CMDSTA and SDC_DATSTA.
A0130050	**SDC_IRQMASK1**	32	**SD memory card IRQ mask register 1** Contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bit does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.
A0130054	**SDIO_CFG**	32	**SDIO configuration register** Configures functions for SDIO.
A0130058	**SDIO_STA**	32	**SDIO status register** Identifies if thers is SDIO interrupt during the interrupt period on data line.

Address	Name	Width	Register function
A0130080	**CLK_RED**	32	**CLK latch configuration register** Configures the MSDC sample data/response clock. *Note: When MSDC_IOCON[19] = 1, the host will latch response; otherwise MSDC FSM will handle the response from PAD directly.*
A0130098	**DAT_CHECKSUM**	32	**MSDC Rx data checksum register** Competes the checksum value of Rx read data

A0130000 MSDC_CFG SD Memory Card Controller Configuration Register 04000020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FIFOTHD				CLKSRC_PAT		VDDPD	RCDEN	DIRQEN	PINEN	DMAEN	INTEN
Type					RW				RW		RW	RW	RW	RW	RW	RW
Reset					0	1	0	0	0		0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCLKF								SCLKON	CRED	STDBY	CLKSRC		NOCRC	RST	MSDC
Type	RW								RW	RW	RW	RW		RW	W1C	RW
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Overview: For general configuration of the SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
27:24	**FIFOTHD**	FIFOTHD	**FIFO threshold** The register field determines when to issue a DMA request. For write transactions, DMA requests will be asserted if the number of free entries in FIFO are bigger than or equal to the value in the register field. For read transactions, DMA requests will be asserted if the number of valid entries in FIFO are bigger than or equal to the value in the register field. The register field must be set according to the setting of data transfer count in DMA burst mode. If single mode for DMA transfer is used, the register field shall be set to 0b0001. 0000: Invalid 0001: Threshold value is 1. 0010: Threshold value is 2. 0011~01111: ... 1000: Threshold value is 8. Others: Invalid
23	**CLKSRC_PA** **T**	CLKSRC_PAT	**CLKSRC patch when {CLKSRC_PAT,CLKSRC} equals to** 4: EPLL/5 (EPLL = 494M) as MSDC source clock 5: EPLL/8 (EPLL = 780M) as MSDC source clock 6: EPLL/6.5 (EPLL = 650 or 624M) as MSDC source clock 7: EPLL/6 (EPLL = 598M) as MSDC source clock Therefore, CLKSRC_PAT means: 0: MPLL as the MSDC source clock 1: EPLL as the MSDC source clock
21	**VDDPD**	VDDPD	**Controls output pin VDDPD used for power saving** Output pin VDDPD controls the power for memory card. 0: Output pin VDDPD outputs logic low. The power for memory card will be turned off. 1: Output pin VDDPD outputs logic high. The power for memory card will be turned on.

Bit(s)	Mnemonic	Name	Description
20	**RCDEN**	RCDEN	**Controls output pin RCDEN used for card identification process when the controller is for SD memory card** Its output controls the pull-down resistor on the system board to connect to or disconnect from signal CD/DAT3. 0: The output pin RCDEN outputs logic low. 1: The output pin RCDEN outputs logic high.
19	**DIRQEN**	DIRQEN	**Enables data request interrupt** The register bit is used to control if data request is used as an interrupt source. 0: Data request is not used as an interrupt source. 1: Data request is used as an interrupt source.
18	**PINEN**	PINEN	**Enables pin interrupt** The register bit is used to control if the pin for card detection is used as an interrupt source. 0: The pin for card detection is not used as an interrupt source. 1: The pin for card detection is used as an interrupt source.
17	**DMAEN**	DMAEN	**Enables DMA** *Note: If DMA capability is disabled, the application software must poll the status of register MSDC_STA to check on any data transfer request. If DMA is desired, the register bit must be set up before command register is written.* 0: DMA request induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: DMA request induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
16	**INTEN**	INTEN	**Enables interrupt** *Note: If interrupt capability is disabled, the application software must poll the status of register MSDC_STA to check on any interrupt request.* 0: Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: Interrupt induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
15:8	**SCLKF**	SCLKF	**Controls clock frequency of serial clock on SD bus and denotes clock frequency of SD bus serial clock as fslave and clock frequency of the SD controller as fhost which is 98.3 or 96.2 MHz** *Note: The allowed maximum frequency of fslave is 49.15MHz.* While changing the clock rate, "1T clock period before change + 1T clock period after change" is required for HW signal to re-synchronize. 00000000b: fslave =(1/2)*fhost 00000001b: fslave = [1/(4*1)]*fhost 00000010b: fslave = [1/(4*2)]*fhost 00000011b: fslave = [1/(4*3)]*fhost 00000100b~111111110b: ... 11111111b: fslave = [1/(4*255)]*fhost
7	**SCLKON**	SCLKON	**Serial clock always on** For debugging. 0: Serial clock not always on 1: Serial clock always on
6	**CRED**	CRED	**Rising edge data** The register bit is used to determine the serial data input is latched at the falling edge or rising edge of the serial clock. The default setting is at the rising edge. If the serial data have bad timing, set the register

Bit(s)	Mnemonic	Name	Description
			bit to 1. When the memory card has bad timing on returned read data, set the register bit to 1. 0: Serial data input is latched at the rising edge of serial clock. 1: Serial data input is latched at the falling edge of serial clock.
5	**STDBY**	STDBY	**Standby mode** If the module is powered down, operating clock to the module will be stopped. At the same time, the clock to card detection circuitry will also be stopped. If detection on memory card insertion and removal is desired, write 1 to the register bit. If interrupt for detection on memory card insertion and removal is enabled, the interrupt will take place whenever the memory is inserted or removed. 0: Standby mode is disabled. 1: Standby mode is enabled.
4:3	**CLKSRC**	CLKSRC	**Specifies which clock is used as source clock of memory card** Use MPLL (598MHz) or EPLL (598MHz/650MHz/780MHz/494MHz) as the source clock of memory card when clock hopping is not enabled. If clock hopping is enabled, MPLL clock's hopping rate will be 0 ~ -8%. EPLL is 598MHz (0 ~ -5%), 650MHz (3 ~ -5% or 0 ~ -5%), 780MHz (2.5 ~ -5%) and 494MHz (0.8 ~ -5%). *Note: EPLL's frequency depends on the EMI device type.* 00: Use MPLL clock divide by 7.5 as source clock of memory card. The source clock is 97MHz. 01: Use MPLL clock divide by 8 as source clock of memory card. The source clock is 91MHz. 10: Use MPLL clock divide by 10 as source clock of memory card. The source clock is 73MHz. 11: Use MPLL clock divide by 12 as source clock of memory card. The source clock is 60.6MHz.
2	**NOCRC**	NOCRC	**Disable CRC** '1' indicates data transfer without CRC is desired. For write data block, the data are transmitted without CRC. For read data block, CRC will not be checked. For testing purpose. 0: Data transfer with CRC is desired. 1: Data transfer without CRC is desired.
1	**RST**	RST	**Software reset** Writing 1 to the register bit will cause internal synchronous reset of SD controller but will not reset register settings. 0: Read 0 stands for the reset process is finished. 1: Write 1 to reset SD controller.
0	**MSDC**	MSDC	**Configures the controller as SD memory card mode** CLK/CMD/DAT line is pulled low when SD memory card mode is disable. 0: Disable SD memory card 1: Enable SD memory card

A0130004 MSDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSY	FIFOCLR							FIFOCNT				INT	DRQ	BE	BF
Type	R	W1C							RO				RO	RO	RO	RO
Reset	0	0							0	0	0	0	0	0	0	0

Overview: The register contains the status of FIFO, interrupts and data requests.

Bit(s)	Mnemonic	Name	Description
15	**BUSY**	BUSY	**Status of the controller** If the controller is in busy state, the register bit will be 1; otherwise 0. 0: The controller is in busy state. 1: The controller is in idle state.
14	**FIFOCLR**	FIFOCLR	**Clears FIFO** **Writing 1 to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.** 0: Read 0 stands for the FIFO clear process is finished. 1: Write 1 to clear the content of FIFO clear and reset the status of FIFO controller.
7:4	**FIFOCNT**	FIFOCNT	**FIFO count** The register field shows how many valid entries are there in FIFO. 0000: 0 valid entry in FIFO 0001: 1 valid entry in FIFO 0010: 2 valid entries in FIFO 0011~0111: ... 1000: 8 valid entries in FIFO
3	**INT**	INT	**Indicates if there is any interrupt existing** When there is interrupt existing, the register bit will still be active even if register bit INTEN in register MSDC_CFG is disabled. The SD controller can interrupt MCU by issuing interrupt request to the interrupt controller, or the software/application will poll the register endlessly to check if there is any interrupt request existing in the SD controller. When register bit INTEN in register MSDC_CFG is disabled, the second method is used. For read commands, it is possible that time-out error takes place. The software can read the status register to check if the time-out error takes place without OS time tick support or data request asserted. *Note: The register bit will be cleared when register MSDC_INT is read.* 0: No interrupt request existing. 1: Interrupt request exists.
2	**DRQ**	DRQ	**Indicates if any data transfer is required** When a data transfer is required, the register bit will still be active even if register bit DIRQEN in register MSDC_CFG is disabled. Data transfer can be achieved by DMA channel alleviating MCU loading, or by polling the register bit to check if any data transfer is requested. When register bit DIRQEN in register MSDC_CFG is disabled, the second method is used. 0: No DMA request existing. 1: DMA request exists.
1	**BE**	BE	**Indicates if FIFO in SD controller is empty** 0: FIFO in SD controller is not empty. 1: FIFO in SD controller is empty.
0	**BF**	BF	**Indicates if FIFO in SD controller is full** 0: FIFO in SD controller is not full. 1: FIFO in SD controller is full.

A0130008 MSDC_INT SD Memory Card Controller Interrupt Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SDIOIRQ	SDR1BIRQ		SDMCIRQ	SDDATIRQ	SDCMDIRQ	PINIRQ	DIRQ
Type									RC	RC		RC	RC	RC	RC	RC
Reset									0	0		0	0	0	0	0

Overview: The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.

Bit(s)	Mnemonic	Name	Description
7	**SDIOIRQ**	SDIOIRQ	**SDIO interrupt** The register bit indicates if there is any interrupt for SDIO existing. Whenever an interrupt for SDIO exists, the register bit will be set to 1 if the interrupt is enabled. It will be reset when the register is read. 0: No SDIO interrupt 1: Interrupt for SDIO exists.
6	**SDR1BIRQ**	SDR1BIRQ	**SD R1b response interrupt** The register bit will be active when a SD command with R1b response is finished and the DAT0 line is transited from busy to idle state. Single block write commands with R1b response will cause interrupt when the command is completed either successfully or with CRC error. However, multi-block write commands with R1b response do not cause interrupt because multi-block write commands are always stopped by STOP_TRANS commands. STOP_TRANS commands (with R1b response) behind multi-block write commands will cause interrupt. Single block read command with R1b response will cause interrupt when the command is completed, but multi-block read commands do not. *Note: STOP_TRANS commands (with R1b response) behind multi-block read commands will cause interrupt.* 0: No interrupt for SD R1b response. 1: Interrupt for SD R1b response exists.
4	**SDMCIRQ**	SDMCIRQ	**SD memory card interrupt** The register bit indicates if there is any interrupt for SD memory card existing. Whenever an interrupt for SD memory card exists, i.e. any bit in register SDC_CSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. *Note: This bit will not trigger MSDC hardware interrupt.* 0: No SD memory card interrupt 1: SD memory card interrupt exists.
3	**SDDATIRQ**	SDDATIRQ	**SD bus DAT interrupt** The register bit indicates if there is any interrupt for SD DAT line existing. Whenever interrupt for SD DAT line exists, i.e. any bit in register SDC_ DATSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. 0: No SD DAT line interrupt 1: SD DAT line interrupt exists.
2	**SDCMDIRQ**	SDCMDIRQ	**SD bus CMD interrupt** The register bit indicates if there is any interrupt for SD CMD line existing. Whenever interrupt for SD CMD line exists, i.e. any bit in the register SDC_CMDSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. 0: No SD CMD line interrupt 1: SD CMD line interrupt exists.
1	**PINIRQ**	PINIRQ	**Pin change interrupt**

Bit(s)	Mnemonic	Name	Description
			The register bit indicates if there is any interrupt for memory card insertion/removal existing. Whenever the memory card is inserted or removed and card detection interrupt is enabled, i.e. register bit PINEN in register MSDC_CFG is set to 1, the register bit will be set to 1. It will be reset when the register is read. 0: Otherwise 1: Card is inserted or removed.
0	**DIRQ**	DIRQ	**Data request interrupt** The register bit indicates if there is any interrupt for data request existing. Whenever data request exists and data request as an interrupt source is enabled, i.e. register bit DIRQEN in register MSDC_CFG is set to 1, the register bit will be active. It will be reset when being read. For software, data requests can be recognized by polling register bit DRQ or by data request interrupt. Data request interrupts will be generated every FIFOTHD data transfers. 0: No data request interrupt 1: Data request interrupt occurs.

A013000C MSDC_PS SD Memory Card Pin Status Register 00000008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								CMD	DAT							
Type								RO	RO							
Reset								0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CDDEBOUNCE											PINCHG	PIN0	POEN0	PIEN0	CDEN
Type	RW											RC	RO	RW	RW	RW
Reset	0	0	0	0								0	1	0	0	0

Overview: The register is used for card detection. When the memory card controller and system are powered on, , the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull-down when the system is powered on. The scheme of card detection for MS is the same as that for SD. To detect card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.

Bit(s)	Mnemonic	Name	Description
24	**CMD**	CMD	**Memory card/SDIO card/MMC card command lines**
23:16	**DAT**	DAT	**Memory card/SDIO card/MMC card data lines**
15:12	**CDDEBOUNCE**	CDDEBOUNCE	**Specifies the time interval for card detection de-bounce** Default value: 0. It means the de-bounce interval is 32-cycle time at 32kHz. The interval can extend one cycle time at 32kHz by increasing the counter by 1.
4	**PINCHG**	PINCHG	**Pin change** The register bit indicates the status of card insertion/removal. If the memory card is inserted or removed, the register bit will be set to 1 no matter pin change interrupt is enabled or not. It will be cleared when the register is read. 0: Otherwise 1: Card is inserted or removed.
3	**PIN0**	PIN0	**Shows the value of input pin for card detection** 0: The value of input pin for card detection is logic low. 1: The value of input pin for card detection is logic high.
2	**POEN0**	POEN0	**Controls output of input pin for card detection**

Bit(s)	Mnemonic	Name	Description
			0: Output of input pin for card detection is disabled. 1: Output of input pin for card detection is enabled.
1	**PIEN0**	PIEN0	**Controls input pin for card detection** 0: Input pin for card detection is disabled. 1: Input pin for card detection is enabled.
0	**CDEN**	CDEN	**Enables card detection** The register bit is used to enable or disable card detection. 0: Card detection is disabled. 1: Card detection is enabled.

A0130010 MSDC_DAT SD Memory Card Controller Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to read/write data from/to FIFO inside the SD controller. Data access unit: 32 bits.

Bit(s)	Mnemonic	Name	Description
31:0	**DATA**	DATA	**Reads/Writes data from/to FIFO inside SD controller** Data access unit: 32 bits

A0130014 MSDC_IOCON SD Memory Card Controller IO Control Register 010000C3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							SAMPLEDLY		FIXDLY		SAMPON	CRCDIS	CMDSEL	INTLH		DSW
Type							RW		RW		RW	RW	RW	RW		RW
Reset							0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDRE					HIGH_SPEED	DMABURST		SRCFG1	SRCFG0	ODCCFG1			ODCCFG0		
Type	RW					RW	RW		RW	RW	RW			RW		
Reset	0					0	0	0	1	1	0	0	0	0	1	1

Overview: The register specifies the output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current depends on the PAD type selected for the chip.

Bit(s)	Mnemonic	Name	Description
25:24	**SAMPLEDLY**	SAMPLEDLY	**Used for SW to select the turn-around delay cycle between write data end bit and CRC status for SD card** 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay

Bit(s)	Mnemonic	Name	Description
23:22	**FIXDLY**	FIXDLY	**Used for SW to select the delay cycle after clock fix high for the host controller to SD card** 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
21	**SAMPON**	SAMPON	**Data sample enabling always on** The bit is suggested to be set to 1 when the feedback clock is used and to 0 when the multiple phase clock is used. 0: Data sample enabling not always on 1: Data sample enabling always on
20	**CRCDIS**	CRCDIS	**Switches off data CRC check for SD read data** 0: CRC check is on. 1: CRC check is off.
19	**CMDSEL**	CMDSEL	**Determines whether the host should delay 1-T to latch response from card** 0: Host latches response without 1-T delay 1: Host latches response with 1-T delay.
18:17	**INTLH**	INTLH	**Selects latch timing for SDIO multi-block read interrupt** 00: Host latches INT at the second backend clock after the end bit of the current data block from card is received. (Default) 01: Host latches INT at the first backend clock after the end bit of the current data block from card is received. 10: Host latches INT at the second backend clock after the end bit of the current data block from card is received. 11: Host latches INT at the third backend clock after the end bit of the current data block from card is received.
16	**DSW**	DSW	**Determines whether the host should latch data with 1-T delay or not** For SD card, this bit is suggested to be 0. For MSPRO cards, it is suggested to be 1. 0: Host latches the data with 1-T delay. 1: Host latches the data without 1-T delay.
15	**CMDRE**	CMDRE	**Determines whether the host should latch response token (sent from card on CMD line) at rising edge or falling edge of serial clock** (T.B.D this bit is un-useful) 0: Host latches response at rising edge of serial clock. 1: Host latches response at falling edge of serial clock.
10	**HIGH_SPEED**	HIGH_SPEED	**For high-speed mode when internal sample clock is used** High-speed mode means the SD/MMC serial bus clock rate is bigger than 25MHz. The default speed mode means that the SD/MMC serial bus clock rate is bigger than 25MHz. 0: Default speed 1: High speed
9:8	**DMABURST**	DMABURST	**Used for SW to select burst type when data are transferred by DMA** *Note: Only single mode can support non-4N bytes data transfer in read operation.* 00: Single mode 01: 4-beat incrementing burst 10: 8-beat incrementing burst 11: Reserved
7	**SRCFG1**	SRCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 0: Fast slew rate 1: Slow slew rate

Bit(s)	Mnemonic	Name	Description
6	**SRCFG0**	SRCFG0	**Output driving capability for pins CMD/BS and SCLK** 0: Fast slew rate 1: Slow slew rate
5:3	**ODCCFG1**	ODCCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 000: 4mA 001: 8mA 010: 12mA 011: 16mA
2:0	**ODCCFG0**	ODCCFG0	**Output driving capability for pins CMD/BS and SCLK** 000: 4mA 001: 8mA 010: 12mA 011: 16mA

A0130018 MSDC_IOCON1 SD Memory Card Controller IO Control Register 1 00022022

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														PRCFG_RST/WP	PRVAL_RST/WP	
Type														RW	RW	
Reset														0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		PRCFG_CK	PRVAL_CK			PRCFG_CM	PRVAL_CM			PRCFG_DA	PRVAL_DA			PRCFG_INS	PRVAL_INS	
Type		RW	RW			RW	RW			RW	RW			RW	RW	
Reset		0	1	0		0	0	0		0	1	0		0	1	0

Bit(s)	Mnemonic	Name	Description
18	**PRCFG_RST/WP**	PRCFG_RST_WP	**Pull-up/down register configuration for pin RST/WP** Default value: 0 0: Pull-up resistor in the I/O pad of pin WP is enabled. 1: Pull-down resistor in the I/O pad of pin WP is enabled.
17:16	**PRVAL_RST/WP**	PRVAL_RST_WP	**Pull-up/down register value for pin RST/WP** Default value: 10 00: Pull–up/down resistor in the I/O pad of pin WP are all disabled. 01: Pull-up/down resistor in the I/O pad of pin WP value is 47k. 10: Pull-up/down resistor in the I/O pad of pin WP value is 47k. 11: Pull-up/down resistor in the I/O pad of pin WP value is 23.5k.
14	**PRCFG_CK**	PRCFG_CK	**Pull-up/down register configuration for pin CK** Default value: 0 0: Pull-up resistor in the I/O pad of pin CK is enabled. 1: Pull-down resistor in the I/O pad of pin CK is enabled.
13:12	**PRVAL_CK**	PRVAL_CK	**Pull-up/down register value for pin CLK** Default value: 10 00: Pull-up/down resistor in the I/O pad of pin CLK are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CLK value is 23.5k.
10	**PRCFG_CM**	PRCFG_CM	**Pull-up/down register configuration for pin CM** Default value is 0. 0: Pull-up resistor in the I/O pad of pin CM is enabled. 1: Pull-down resistor in the I/O pad of pin CM is enabled.
9:8	**PRVAL_CM**	PRVAL_CM	**Pull-up/down register value for pin CMD/BS**

Bit(s)	Mnemonic	Name	Description
			Default value: 00 00: Pull-up/down resistor in the I/O pad of pin CMD/BS are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 23.5k.
6	**PRCFG_DA**	PRCFG_DA	**Pull-up/down register configuration for pin DAT0, DAT1, DAT2 and DAT3** Default value: 0 0: Pull-up resistor in the I/O pad of pin DAT is enabled. 1: Pull-down resistor in the I/O pad of pin DAT is enabled.
5:4	**PRVAL_DA**	PRVAL_DA	**Pull-up/down register value for pin DAT0, DAT1, DAT2 and DAT3** Default value: 10 00: Pull-up/ down resistor in the I/O pad of pin DAT are all disabled. 01: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 10: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 11: Pull-up/down resistor in the I/O pad of pin DAT value is 23.5k.
2	**PRCFG_INS**	PRCFG_INS	**Pull-up/down register configuration for pin INS** Default value: 0 0: Pull-up resistor in the I/O pad of pin WP is enabled. 1: Pull-down resistor in the I/O pad of pin WP is enabled.
1:0	**PRVAL_INS**	PRVAL_INS	**Pull-up/down register value for pin INS** Default value: 10 00: Pull-up/down resistor in the I/O pad of pin INS are all disabled. 01: Pull-up/down resistor in the I/O pad of pin INS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin INS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin INS value is 23.5k.

A0130020 SDC_CFG SD Memory Card Controller Configuration Register 00008000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DTOC								WDOD				SDIO		MDLEN	SIEN
Type	RW								RW				RW		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BSYDLY				BLKLEN											
Type	RW				RW											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to configure the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
31:24	**DTOC**	DTOC	**Data time-out counter** The period from finish of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit of 65,536 serial clock. See the register field descriptions of register bit RDINT for reference. 00000000: Extend 65,536 more serial clock cycles 00000001: Extend 65,536x2 more serial clock cycles 00000010: Extend 65,536x3 more serial clock cycles

Bit(s)	Mnemonic	Name	Description
			00000011~11111110: ... 11111111: Extend 65,536x 256 more serial clock cycles
23:20	**WDOD**	WDOD	**Write data output delay** The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
19	**SDIO**	SDIO	**Enables SDIO** 0: Disable SDIO mode 1: Enable SDIO mode
17	**MDLEN**	MDLEN	**Enables multiple data line** The register can be enabled only when SD memory card is applied and detected by software application. It is the responsibility of the application to program the bit correctly when an multi-media card is applied. If an multi-media card is applied and 4-bit data line is enabled, the 4 bits will be output every serial clock. Therefore, data integrity will fail. 0: Disable 4-bit data line 1: Enable 4-bit data line
16	**SIEN**	SIEN	**Enables serial interface** It should be enabled as soon as possible before any command. 0: Disable serial interface for SD 1: Enable serial interface for SD
15:12	**BSYDLY**	BSYDLY	**Only valid for the commands with R1b response** If the command has a response of R1b type, the SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if the operation in SD memory card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, the controller will abandon the detection. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
11:0	**BLKLEN**	BLKLEN	**Block length** The register field is used to define the length of one block in unit of byte in a data transaction. The maximum value of block length is 2,048 bytes. 000000000000: Reserved 000000000001: Block length is 1 byte. 000000000010: Block length is 2 bytes. 000000000011~011111111110: ... 011111111111: Block length is 2,047 bytes. 100000000000: Block length is 2,048 bytes.

A0130024 SDC_CMD SD Memory Card Controller Command Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name																CMDFAIL
Type																RW
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTC	STOP	RW	DTYPE		IDRT	RSPTYP			BREAK	CMD					
Type	RW	RW	RW	RW		RW	RW			RW	RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto the SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.

Bit(s)	Mnemonic	Name	Description
16	**CMDFAIL**	CMDFAIL	**If 4-bit SDIO mode is enabled and when CMD/DAT error occurrs, set this bit to select whether to "wait stop command" or "wait data state machine idle".** 0: Wait stop command 1: Wait data state machine idle
15	**INTC**	INTC	**Indicates if the command is GO_IRQ_STATE** If the command is GO_IRQ_STATE, the period between command token and response token will not be limited. 0: The command is not GO_IRQ_STATE. 1: The command is GO_IRQ_STATE.
14	**STOP**	STOP	**Indicates if the command is a stop transmission command** 0: The command is not a stop transmission command. 1: The command is a stop transmission command.
13	**RW**	RW	**Defines the command is a read command or write command** The register bit is valid only when the command causes a transaction with data token. 0: The command is a read command. 1: The command is a write command.
12:11	**DTYPE**	DTYPE	**Defines data token type for the command** 00: No data token for the command 01: Single block transaction 10: Multiple block transaction, i.e. the command is a multiple block read or write command. 11: Stream operation. It can only be used when an multi-media card is applied.
10	**IDRT**	IDRT	**Identification response time** The register bit indicates if the command has a response with NID (i.e. 5 serial clock cycles as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 1.0) response time. The register bit is valid only when the command has a response token. Thus the register bit must be set to 1 for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD). 0: Otherwise 1: The command has a response with NID response time.
9:7	**RSPTYP**	RSPTYP	**Defines response type for the command** For commands with R1 and R1b response, register SDC_CSTA (not SDC_STA) updates after response token is received. This register SDC_CSTA contains the status of the SD, and it can be used as a response interrupt source. *Note: If CMD7 is used with all 0's RCA, then RSPTYP must be "000". Command "GO_TO_IDLE" also has RSPTYP='000'.*

Bit(s)	Mnemonic	Name	Description
			000: There is no response for the command, e.g. broadcast command without response and GO_INACTIVE_STATE command. 001: The command has R1 response. R1 response token is 48-bit. 010: The command has R2 response. R2 response token is 136-bit. 011: The command has R3 response. Even though R3 is 48-bit response, it does not contain CRC checksum. 100: The command has R4 response. R4 response token is 48-bit. (only for MMC) 101: The command has R5 response. R5 response token is 48-bit. (only for MMC) 110: The command has R6 response. R6 response token is 48-bit. 111: The command has R1b response. If the command has a response of R1b type, SD controller must monitor the data line 0 for card busy status from the bit time that is 2 or 4 serial clock cycles after the command end bit to check if the operation in SD memory card has finished. There are two cases for detection of card busy status. The first case is that the host stops the data transmission during an active write data transfer. The card will assert busy signal after the stop transmission command end bit followed by 4 serial clock cycles. The second case is that the card is in idle state or receiving a stop transmission command between data blocks when multiple block write command is in progress. The register bit will be valid only when the command has a response token.
6	**BREAK**	BREAK	**Aborts pending MMC GO_IRQ_MODE command** It is only valid for a pending GO_IRQ_MODE command waiting for MMC interrupt response. 0: Other fields are valid. 1: Break a pending MMC GO_IRQ_MODE command in the controller. Other fields are invalid.
5:0	**CMD**	CMD	**SD memory card command** Total 6 bits.

A0130028 SDC_ARG SD Memory Card Controller Argument Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ARG[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ARG[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains the argument of the SD memory card command.

Bit(s)	Mnemonic	Name	Description
31:0	**ARG**	ARG	Contains argument of the SD memory card command.

A013002C SDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WP											FEDATBUS	FECMDBUS	BEDATBUS	BECMDBUS	BESDCBUS

												Y	Y	Y	Y	Y
Type	RO											RO	RO	RO	RO	RO
Reset	0											0	0	0	0	0

Overview: The register contains various statuses of SD controller as the controller is configured as the host of SD memory card.

Bit(s)	Mnemonic	Name	Description
15	**WP**	WP	**Detects the status of write protection switch on SD memory card** The register bit shows the status of write protection switch on SD memory card. There is no default reset value. Pin WP (Write Protection) is only useful when the controller is configured for SD memory card. 1: Write protection switch on, i.e. memory card is desired to be write-protected. 0: Write protection switch off, i.e. memory card is writable.
4	**FEDATBUSY**	FEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** This bit indicates directly the CMD line at card clock domain. For those commands without data but still involving DAT line, the register bit is useless. For example, if an erase command is issued, checking if the register bit is 0 before issuing the next command with data will not guarantee that the controller is idle. In this case, use register bit BESDCBUSY. 0: No transmission is going on DAT line on SD bus. 1: There is transmission going on DAT line on SD bus.
3	**FECMDBUSY**	FECMDBUSY	**Indicates if there is any transmission going on CMD line on SD bus** This bit indicates directly the CMD line at card clock domain. 0: No transmission is going on CMD line on SD bus. 1: There is transmission going on CMD line on SD bus.
2	**BEDATBUSY**	BEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** 0: Backend SDC controller gets the info that no transmission is going on DAT line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on DAT line on SD bus.
1	**BECMDBUSY**	BECMDBUSY	**Indicates if there is any transmission going on CMD line on SD bus** This bit shows backend controller's CMD busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SDC controller gets the info that no transmission is going on CMD line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on CMD line on SD bus.
0	**BESDCBUSY**	BESDCBUSY	**Indicates if SD controller is busy, i.e. is there any transmission going on CMD or DAT line on SD bus** This bit shows backend controller's SDC busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SD controller is idle. 1: Backend SD controller is busy.

A0130030 SDC_RESP0 SD Memory Card Controller Response Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESP[31:0][31:16]**															
Type	RO															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[31:0][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[31:0]**	RESP_31_0	

A0130034 SDC_RESP1 SD Memory Card Controller Response Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[63:32][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[63:32][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[63:32]**	RESP_63_32	

A0130038 SDC_RESP2 SD Memory Card Controller Response Register 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[95:64][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[95:64][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[95:64]**	RESP_95_64	

A013003C SDC_RESP3 SD Memory Card Controller Response Register 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[127:96][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	RESP[127:96][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, i.e. response of commands ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[127:96]**	RESP_127_96	

A0130040 SDC_CMDSTA SD Memory Card Controller Command Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**RSPCRCERR**	**CMDTO**	**CMDRDY**
Type														RC	RC	RC
Reset														0	0	0

Overview: The register contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register can also be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
2	**RSPCRCERR**	RSPCRCERR	**CRC error on CMD detected** '1' indicates the SD controller detects a CRC error after reading a response from the CMD line. 0: Otherwise 1: SD controller detects a CRC error after reading a response from the CMD line.
1	**CMDTO**	CMDTO	**Time-out on CMD detected** '1' indicates the SD controller detects a time-out condition while waiting for a response on the CMD line. 0: Otherwise 1: SD controller detects a timeout condition while waiting for a response on the CMD line.
0	**CMDRDY**	CMDRDY	**For command without response, the register bit will be 1 once the command is completed on SD bus** For command with response, the register bit will be 1 whenever the command is issued onto the SD bus and its corresponding response is received without CRC error. 0: Otherwise 1: Command with/without response is finished successfully without CRC error.

A0130044 SDC_DATSTA SD Memory Card Controller Data Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DATCRCERR								DATTO	BLKDONE
Type							RC								RC	RC
Reset							0	0	0	0	0	0	0	0	0	0

Overview: The register contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register can be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
9:2	**DATCRCERR**	DATCRCERR	**CRC error on DAT detected** '1' indicates that the SD controller detects a CRC error for bit n after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. 0: Otherwise 1: SD controller detects a CRC error after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. *Note: n is 7 ~ 1 for 8-bits mode. Each bit is read and cleared individually.*
1	**DATTO**	DATTO	**Time-out on DAT detected** A '1' indicates that the SD controller detects a time-out condition while waiting for data token on the DAT line. 0: Otherwise 1: SD controller detects a time-out condition while waiting for data token on the DAT line.
0	**BLKDONE**	BLKDONE	**Indicates the status of data block transfer** 0: Otherwise 1: A data block is successfully transferred.

A0130048 SDC_CSTA SD Memory Card Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSTA [31:0][31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSTA [31:0][15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: After commands with R1 and R1b respond this register containing the status of the SD card, it will be used as a response interrupt source. In all register fields, logic high indicates error, and logic low indicates no error. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description

31:0	**CSTA [31:0]** CSTA_31_0	CSTA31: OUT_OF_RANGE. The command's argument is out of the allowed range for this card. CSTA30: ADDRESS_ERROR. A misaligned address that does not match the block length is used in the command. CSTA29: BLOCK_LEN_ERROR. The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length. CSTA28: ERASE_SEQ_ERROR. An error in the sequence of erase commands occurs. CSTA27: ERASE_PARAM. An invalid selection of write-blocks for erase occurs. CSTA26: WP_VIOLATION. Attempt to program a write-protected block. CSTA25: Reserved. Return to 0. CSTA24: LOCK_UNLOCK_FAILED. Set when a sequence or password error is detected in lock/unlock card command or if there is an attempt to access a locked card. CSTA23: COM_CRC_ERROR. The CRC check of the previous command fails. CSTA22: ILLEGAL_COMMAND. Command not legal for the card state. CSTA21: CARD_ECC_FAILED. Card internal ECC is applied but fails to correct the data. CSTA20: CC_ERROR. Internal card controller error. CSTA19: ERROR. A general or unknown error occurs during the operation. CSTA18: UNDERRUN. The card cannot sustain data transfer in stream read mode. CSTA17: OVERRUN. The card cannot sustain data programming in stream write mode. CSTA16: CID/CSD_OVERWRITE. It can be either one of the following errors: 1) The CID register has been already written and cannot be overwritten; 2) The read-only section of the CSD does not match the card; 3) An attempt to reverse the copy (set as the original) or permanent WP (unprotected) bits is made. CSTA[15: 4]: Reserved. Return to 0. CSTA3: AKE_SEQ_ERROR. Error in the sequence of authentication process CSTA[2: 0]: Reserved. Return to 0.

A013004C SDC_IRQMASK0 SD Memory Card IRQ Mask Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [31:0][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [31:0][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. See the register descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from

register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from registers SDC_CMDSTA and SDC_DATSTA.

Bit(s)	Mnemonic	Name	Description
31:0	**IRQMASK [31:0]**	IRQMASK_31_0	

A0130050 SDC_IRQMASK1 SD Memory Card IRQ Mask Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [63:32][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [63:32][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then the interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.

Bit(s)	Mnemonic	Name	Description
31:0	**IRQMASK [63:32]**	IRQMASK_63_32	

A0130054 SDIO_CFG SDIO Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**DISSEL**		**INTCSEL**	**DSBSEL**		**INTEN**
Type											RW		RW	RW		RW
Reset											0		0	0		0

Overview: The register is used to configure functions for SDIO.

Bit(s)	Mnemonic	Name	Description
5	**DISSEL**	DISSEL	**Selects data block interrupt source** 0: The host detects SDIO interrupt during interrupt period between two data blocks of multiple block data access. 1: The host ignores SDIO interrupt during interrupt period between two data blocks of multiple block data access.
3	**INTCSEL**	INTCSEL	**Selects interrupt control** 0: The host detects DAT1 low as SDIO interrupt. 1: The host detects DAT3/DAT2/DAT1/DAT0 4'b1101 as SDIO interrupt.
2	**DSBSEL**	DSBSEL	**Selects data block start bit**

Bit(s)	Mnemonic	Name	Description
			0: Use data line 0 as start bit of data block. Other data lines are ignored. 1: Start bit of a data block is received only when all data line 0-3 become low.
0	**INTEN**	INTEN	**Enables interrupt for SDIO** 0: Disable 1: Enable

A0130058 SDIO_STA SDIO Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**IRQ**
Type																RO
Reset																0

Overview: This register is used to identify if there is SDIO interrupt during the interrupt period on data line.

Bit(s)	Mnemonic	Name	Description
0	**IRQ**	IRQ	**SDIO interrupt exists on the data line.** For example, when in the interrupt period or the 1 -bit data line mode and DAT1/5 goes low from high, this bit will become 1 from 0. If DAT1/5 goes high from low, this bit will become 0 from 1. 0: There is no SDIO interrupt existing on the data line. 1: There is SDIO interrupt existing on the data line.

A0130080 CLK_RED CLK Latch Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**CMD_RED**													
Type			RW													
Reset			0													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**DAT_RED**						**CLKPAD_RED**	**CLK_LATCH**						
Type			RW						RW	RW						
Reset			0						0	0						

Overview: The register is used to configure the MSDC sample data/response clock. Note that only when MSDC_IOCON[19] = 1 will the host latch response; otherwise MSDC FSM will handle the response from PAD directly.

Bit(s)	Mnemonic	Name	Description
29	**CMD_RED**	CMD_RED	**Determines the command response from card output is latched at falling edge or rising edge of internal clock** Only effective when CLK_LATCH = 1 0: Internal clock rising edge to latch response 1: Internal clock falling edge to latch response
13	**DAT_RED**	DAT_RED	**Determines the input data from card output is latched at falling**

Bit(s)	Mnemonic	Name	Description
			edge or rising edge of internal sample clock Only effective when CLK_LATCH = 1 0: Internal clock rising edge to latch data 1: Internal clock falling edge to latch data
7	**CLKPAD_RED**	CLKPAD_RED	**Determines the input data from card is latched at falling edge or rising edge of the feedback clock from pad** The suggested setting is 0 when SD serial clock is lower than 25MHz. The suggestion setting is 1 when SD serial clock is higher than 25MHz. The suggestion setting is 0 for MMC card no matter the serial clock rate is high speed or default speed. Only effective when CLK_LATCH = 0. 0: Internal feedback clock rising edge to latch data/response 1: Internal feedback clock falling edge to latch data/response
6	**CLK_LATCH**	CLK_LATCH	**Determines which clock to latch data from card** The suggested setting is 1 if SCLKF in register field MSDC_CFG is 0x0. Otherwise, the suggested setting is 0. 0: Internal feedback clock is used to latch data/response from card. 1: Internal clock is used to latch data/response from card.

A0130098 DAT_CHECKSUM MSDC Rx Data Checksum Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAT_CHECKSUM[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT_CHECKSUM[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to compete the checksum value of Rx read data

Bit(s)	Mnemonic	Name	Description
31:0	**DAT_CHECKSUM**	DAT_CHECKSUM	The checksum algorithm is 32-bit XOR.

Module name: MSDC1 base address: (+A0270000h)

Address	Name	Width	Register function
A0270000	**MSDC_CFG**	32	**SD memory card controller configuration register** For general configuration of the SD controller. *Note: MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0270004	**MSDC_STA**	32	**SD memory card controller status register** Contains the status of FIFO, interrupts and data requests.
A0270008	**MSDC_INT**	32	**SD memory card controller interrupt register** Contains the status of interrupts. Note that the register still shows the status of interrupt even though the interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting up register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.
A027000C	**MSDC_PS**	32	**SD memory card pin status register** Used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull down when the system is powered on. The scheme of card detection for MS is the same as that for SD. For detecting card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.
A0270010	**MSDC_DAT**	32	**SD memory card controller data register** Reads/Writes data from/to FIFO inside SD controller. Data access unit: 32 bits
A0270014	**MSDC_IOCON**	32	**SD memory card controller IO control register** Specifies output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current will depend on the PAD type selected for the chip.
A0270018	**MSDC_IOCON1**	32	**SD memory card controller IO control register 1**
A0270020	**SDC_CFG**	32	**SD memory card controller configuration register** Configures the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the

Address	Name	Width	Register function
			operation of the controller. *Note: SDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0270024	**SDC_CMD**	32	**SD memory card controller command register** Defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.
A0270028	**SDC_ARG**	32	**SD memory card controller argument register** Contains argument of the SD memory card command.
A027002C	**SDC_STA**	32	**SD memory card controller status register** Contains various statuses of SD controller as the controller is configured as the host of SD memory card.
A0270030	**SDC_RESP0**	32	**SD memory card controller response register 0** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0270034	**SDC_RESP1**	32	**SD memory card controller response register 1** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0270038	**SDC_RESP2**	32	**SD memory card controller response register 2** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A027003C	**SDC_RESP3**	32	**SD memory card controller response register 3** Contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For responses of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.
A0270040	**SDC_CMDSTA**	32	**SD memory card controller command status register** Contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register will also be used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Address	Name	Width	Register function
A0270044	**SDC_DATSTA**	32	**SD memory card controller data status register** Contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register is also used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0270048	**SDC_CSTA**	32	**SD memory card status register** After commands with R1 and R1b response this register containing the status of the SD card, it will be used as the response interrupt source. In all register fields, logic high indicates error, and logic low indicates there is no error. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A027004C	**SDC_IRQMASK0**	32	**SD memory card IRQ mask register 0** Contains parts of SD memory card interrupt mask register. See the descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. ‘1’ in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CMDSTA and SDC_DATSTA.
A0270050	**SDC_IRQMASK1**	32	**SD memory card IRQ mask register 1** Contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bit does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.
A0270054	**SDIO_CFG**	32	**SDIO configuration register** Configures functions for SDIO.
A0270058	**SDIO_STA**	32	**SDIO status register** Identifies if thers is SDIO interrupt during the interrupt period on data line.
A0270080	**CLK_RED**	32	**CLK latch configuration register** Configures the MSDC sample data/response clock. *Note: When MSDC_IOCON[19] = 1, the host will latch*

Address	Name	Width	Register function
			response; otherwise MSDC FSM will handle the response from PAD directly.
A0270098	**DAT_CHECKSUM**	32	**MSDC Rx data checksum register** Competes the checksum value of Rx read data

A0270000 MSDC_CFG SD Memory Card Controller Configuration Register 04000020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FIFOTHD				CLKSRC_PAT		VDDPD	RCDEN	DIRQEN	PINEN	DMAEN	INTEN
Type					RW				RW		RW	RW	RW	RW	RW	RW
Reset					0	1	0	0	0		0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCLKF								SCLKON	CRED	STDBY	CLKSRC		NOCRC	RST	MSDC
Type	RW								RW	RW	RW	RW		RW	W1C	RW
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Overview: For general configuration of the SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
27:24	**FIFOTHD**	FIFOTHD	**FIFO threshold** The register field determines when to issue a DMA request. For write transactions, DMA requests will be asserted if the number of free entries in FIFO are bigger than or equal to the value in the register field. For read transactions, DMA requests will be asserted if the number of valid entries in FIFO are bigger than or equal to the value in the register field. The register field must be set according to the setting of data transfer count in DMA burst mode. If single mode for DMA transfer is used, the register field shall be set to 0b0001. 0000: Invalid 0001: Threshold value is 1. 0010: Threshold value is 2. 0011~01111: ... 1000: Threshold value is 8. Others: Invalid
23	**CLKSRC_PAT**	CLKSRC_PAT	**CLKSRC patch when {CLKSRC_PAT,CLKSRC} equals to** 4: EPLL/5 (EPLL = 494M) as MSDC source clock 5: EPLL/8 (EPLL = 780M) as MSDC source clock 6: EPLL/6.5 (EPLL = 650 or 624M) as MSDC source clock 7: EPLL/6 (EPLL = 598M) as MSDC source clock Therefore, CLKSRC_PAT means: 0: MPLL as the MSDC source clock 1: EPLL as the MSDC source clock
21	**VDDPD**	VDDPD	**Controls output pin VDDPD used for power saving** Output pin VDDPD controls the power for memory card. 0: Output pin VDDPD outputs logic low. The power for memory card will be turned off. 1: Output pin VDDPD outputs logic high. The power for memory card will be turned on.
20	**RCDEN**	RCDEN	**Controls output pin RCDEN used for card identification process when the controller is for SD memory card** Its output controls the pull-down resistor on the system board to

Bit(s)	Mnemonic	Name	Description
			connect to or disconnect from signal CD/DAT3. 0: The output pin RCDEN outputs logic low. 1: The output pin RCDEN outputs logic high.
19	**DIRQEN**	DIRQEN	**Enables data request interrupt** The register bit is used to control if data request is used as an interrupt source. 0: Data request is not used as an interrupt source. 1: Data request is used as an interrupt source.
18	**PINEN**	PINEN	**Enables pin interrupt** The register bit is used to control if the pin for card detection is used as an interrupt source. 0: The pin for card detection is not used as an interrupt source. 1: The pin for card detection is used as an interrupt source.
17	**DMAEN**	DMAEN	**Enables DMA** *Note: If DMA capability is disabled, the application software must poll the status of register MSDC_STA to check on any data transfer request. If DMA is desired, the register bit must be set up before command register is written.* 0: DMA request induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: DMA request induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
16	**INTEN**	INTEN	**Enables interrupt** *Note: If interrupt capability is disabled, the application software must poll the status of register MSDC_STA to check on any interrupt request.* 0: Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: Interrupt induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
15:8	**SCLKF**	SCLKF	**Controls clock frequency of serial clock on SD bus and denotes clock frequency of SD bus serial clock as fslave and clock frequency of the SD controller as fhost which is 98.3 or 96.2 MHz** *Note: The allowed maximum frequency of fslave is 49.15MHz.* While changing the clock rate, "1T clock period before change + 1T clock period after change" is required for HW signal to re-synchronize. 00000000b: fslave =(1/2)*fhost 00000001b: fslave = [1/(4*1)]*fhost 00000010b: fslave = [1/(4*2)]*fhost 00000011b: fslave = [1/(4*3)]*fhost 00000100b~111111110b: ... 11111111b: fslave = [1/(4*255)]*fhost
7	**SCLKON**	SCLKON	**Serial clock always on** For debugging. 0: Serial clock not always on 1: Serial clock always on
6	**CRED**	CRED	**Rising edge data** The register bit is used to determine the serial data input is latched at the falling edge or rising edge of the serial clock. The default setting is at the rising edge. If the serial data have bad timing, set the register bit to 1. When the memory card has bad timing on returned read data, set the register bit to 1. 0: Serial data input is latched at the rising edge of serial clock.

Bit(s)	Mnemonic	Name	Description
			1: Serial data input is latched at the falling edge of serial clock.
5	**STDBY**	STDBY	**Standby mode** If the module is powered down, operating clock to the module will be stopped. At the same time, the clock to card detection circuitry will also be stopped. If detection on memory card insertion and removal is desired, write 1 to the register bit. If interrupt for detection on memory card insertion and removal is enabled, the interrupt will take place whenever the memory is inserted or removed. 0: Standby mode is disabled. 1: Standby mode is enabled.
4:3	**CLKSRC**	CLKSRC	**Specifies which clock is used as source clock of memory card** Use MPLL (598MHz) or EPLL (598MHz/650MHz/780MHz/494MHz) as the source clock of memory card when clock hopping is not enabled. If clock hopping is enabled, MPLL clock's hopping rate will be 0 ~ -8%. EPLL is 598MHz (0 ~ -5%), 650MHz (3 ~ -5% or 0 ~ -5%), 780MHz (2.5 ~ -5%) and 494MHz (0.8 ~ -5%). *Note: EPLL's frequency depends on the EMI device type.* 00: Use MPLL clock divide by 7.5 as source clock of memory card. The source clock is 97MHz. 01: Use MPLL clock divide by 8 as source clock of memory card. The source clock is 91MHz. 10: Use MPLL clock divide by 10 as source clock of memory card. The source clock is 73MHz. 11: Use MPLL clock divide by 12 as source clock of memory card. The source clock is 60.6MHz.
2	**NOCRC**	NOCRC	**Disable CRC** '1' indicates data transfer without CRC is desired. For write data block, the data are transmitted without CRC. For read data block, CRC will not be checked. For testing purpose. 0: Data transfer with CRC is desired. 1: Data transfer without CRC is desired.
1	**RST**	RST	**Software reset** Writing 1 to the register bit will cause internal synchronous reset of SD controller but will not reset register settings. 0: Read 0 stands for the reset process is finished. 1: Write 1 to reset SD controller.
0	**MSDC**	MSDC	**Configures the controller as SD memory card mode** CLK/CMD/DAT line is pulled low when SD memory card mode is disable. 0: Disable SD memory card 1: Enable SD memory card

A0270004 MSDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSY	FIFOCLR							FIFOCNT				INT	DRQ	BE	BF
Type	R	W1C							RO				RO	RO	RO	RO
Reset	0	0							0	0	0	0	0	0	0	0

Overview: The register contains the status of FIFO, interrupts and data requests.

Bit(s)	Mnemonic	Name	Description
15	**BUSY**	BUSY	**Status of the controller** If the controller is in busy state, the register bit will be 1; otherwise 0. 0: The controller is in busy state. 1: The controller is in idle state.
14	**FIFOCLR**	FIFOCLR	**Clears FIFO** **Writing 1 to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.** 0: Read 0 stands for the FIFO clear process is finished. 1: Write 1 to clear the content of FIFO clear and reset the status of FIFO controller.
7:4	**FIFOCNT**	FIFOCNT	**FIFO count** The register field shows how many valid entries are there in FIFO. 0000: 0 valid entry in FIFO 0001: 1 valid entry in FIFO 0010: 2 valid entries in FIFO 0011~0111: ... 1000: 8 valid entries in FIFO
3	**INT**	INT	**Indicates if there is any interrupt existing** When there is interrupt existing, the register bit will still be active even if register bit INTEN in register MSDC_CFG is disabled. The SD controller can interrupt MCU by issuing interrupt request to the interrupt controller, or the software/application will poll the register endlessly to check if there is any interrupt request existing in the SD controller. When register bit INTEN in register MSDC_CFG is disabled, the second method is used. For read commands, it is possible that time-out error takes place. The software can read the status register to check if the time-out error takes place without OS time tick support or data request asserted. *Note: The register bit will be cleared when register MSDC_INT is read.* 0: No interrupt request existing. 1: Interrupt request exists.
2	**DRQ**	DRQ	**Indicates if any data transfer is required** When a data transfer is required, the register bit will still be active even if register bit DIRQEN in register MSDC_CFG is disabled. Data transfer can be achieved by DMA channel alleviating MCU loading, or by polling the register bit to check if any data transfer is requested. When register bit DIRQEN in register MSDC_CFG is disabled, the second method is used. 0: No DMA request existing. 1: DMA request exists.
1	**BE**	BE	**Indicates if FIFO in SD controller is empty** 0: FIFO in SD controller is not empty. 1: FIFO in SD controller is empty.
0	**BF**	BF	**Indicates if FIFO in SD controller is full** 0: FIFO in SD controller is not full. 1: FIFO in SD controller is full.

A0270008 MSDC_INT SD Memory Card Controller Interrupt Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SDIOIRQ	SDR1BIRQ		SDMCIRQ	SDDATIRQ	SDCMDIRQ	PINIRQ	DIRQ

Type									RC	RC		RC	RC	RC	RC	RC
Reset									0	0		0	0	0	0	0

Overview: The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.

Bit(s)	Mnemonic	Name	Description
7	**SDIOIRQ**	SDIOIRQ	**SDIO interrupt** The register bit indicates if there is any interrupt for SDIO existing. Whenever an interrupt for SDIO exists, the register bit will be set to 1 if the interrupt is enabled. It will be reset when the register is read. 0: No SDIO interrupt 1: Interrupt for SDIO exists.
6	**SDR1BIRQ**	SDR1BIRQ	**SD R1b response interrupt** The register bit will be active when a SD command with R1b response is finished and the DAT0 line is transited from busy to idle state. Single block write commands with R1b response will cause interrupt when the command is completed either successfully or with CRC error. However, multi-block write commands with R1b response do not cause interrupt because multi-block write commands are always stopped by STOP_TRANS commands. STOP_TRANS commands (with R1b response) behind multi-block write commands will cause interrupt. Single block read command with R1b response will cause interrupt when the command is completed, but multi-block read commands do not. *Note: STOP_TRANS commands (with R1b response) behind multi-block read commands will cause interrupt.* 0: No interrupt for SD R1b response. 1: Interrupt for SD R1b response exists.
4	**SDMCIRQ**	SDMCIRQ	**SD memory card interrupt** The register bit indicates if there is any interrupt for SD memory card existing. Whenever an interrupt for SD memory card exists, i.e. any bit in register SDC_CSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. *Note: This bit will not trigger MSDC hardware interrupt.* 0: No SD memory card interrupt 1: SD memory card interrupt exists.
3	**SDDATIRQ**	SDDATIRQ	**SD bus DAT interrupt** The register bit indicates if there is any interrupt for SD DAT line existing. Whenever interrupt for SD DAT line exists, i.e. any bit in register SDC_ DATSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. 0: No SD DAT line interrupt 1: SD DAT line interrupt exists.
2	**SDCMDIRQ**	SDCMDIRQ	**SD bus CMD interrupt** The register bit indicates if there is any interrupt for SD CMD line existing. Whenever interrupt for SD CMD line exists, i.e. any bit in the register SDC_CMDSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. 0: No SD CMD line interrupt 1: SD CMD line interrupt exists.
1	**PINIRQ**	PINIRQ	**Pin change interrupt** The register bit indicates if there is any interrupt for memory card insertion/removal existing. Whenever the memory card is inserted or removed and card detection interrupt is enabled, i.e. register bit

Bit(s)	Mnemonic	Name	Description
			PINEN in register MSDC_CFG is set to 1, the register bit will be set to 1. It will be reset when the register is read. 0: Otherwise 1: Card is inserted or removed.
0	**DIRQ**	DIRQ	**Data request interrupt** The register bit indicates if there is any interrupt for data request existing. Whenever data request exists and data request as an interrupt source is enabled, i.e. register bit DIRQEN in register MSDC_CFG is set to 1, the register bit will be active. It will be reset when being read. For software, data requests can be recognized by polling register bit DRQ or by data request interrupt. Data request interrupts will be generated every FIFOTHD data transfers. 0: No data request interrupt 1: Data request interrupt occurs.

A027000C MSDC_PS SD Memory Card Pin Status Register 00000008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								CMD	DAT							
Type								RO	RO							
Reset								0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CDDEBOUNCE											PINCHG	PIN0	POEN0	PIEN0	CDEN
Type	RW											RC	RO	RW	RW	RW
Reset	0	0	0	0								0	1	0	0	0

Overview: The register is used for card detection. When the memory card controller and system are powered on, , the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull-down when the system is powered on. The scheme of card detection for MS is the same as that for SD. To detect card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.

Bit(s)	Mnemonic	Name	Description
24	**CMD**	CMD	**Memory card/SDIO card/MMC card command lines**
23:16	**DAT**	DAT	**Memory card/SDIO card/MMC card data lines**
15:12	**CDDEBOUNCE**	CDDEBOUNCE	**Specifies the time interval for card detection de-bounce** Default value: 0. It means the de-bounce interval is 32-cycle time at 32kHz. The interval can extend one cycle time at 32kHz by increasing the counter by 1.
4	**PINCHG**	PINCHG	**Pin change** The register bit indicates the status of card insertion/removal. If the memory card is inserted or removed, the register bit will be set to 1 no matter pin change interrupt is enabled or not. It will be cleared when the register is read. 0: Otherwise 1: Card is inserted or removed.
3	**PIN0**	PIN0	**Shows the value of input pin for card detection** 0: The value of input pin for card detection is logic low. 1: The value of input pin for card detection is logic high.
2	**POEN0**	POEN0	**Controls output of input pin for card detection** 0: Output of input pin for card detection is disabled. 1: Output of input pin for card detection is enabled.

Bit(s)	Mnemonic	Name	Description
1	**PIEN0**	PIEN0	**Controls input pin for card detection** 0: Input pin for card detection is disabled. 1: Input pin for card detection is enabled.
0	**CDEN**	CDEN	**Enables card detection** The register bit is used to enable or disable card detection. 0: Card detection is disabled. 1: Card detection is enabled.

A0270010 MSDC_DAT SD Memory Card Controller Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to read/write data from/to FIFO inside the SD controller. Data access unit: 32 bits.

Bit(s)	Mnemonic	Name	Description
31:0	**DATA**	DATA	**Reads/Writes data from/to FIFO inside SD controller** Data access unit: 32 bits

A0270014 MSDC_IOCON SD Memory Card Controller IO Control Register 010000C3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							SAMPLEDLY		FIXDLY		SAMPON	CRCDIS	CMDSEL	INTLH		DSW
Type							RW		RW		RW	RW	RW	RW		RW
Reset							0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDRE					HIGH_SPEED	DMABURST		SRCFG1	SRCFG0	ODCCFG1			ODCCFG0		
Type	RW					RW	RW		RW	RW	RW			RW		
Reset	0					0	0	0	1	1	0	0	0	0	1	1

Overview: The register specifies the output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current depends on the PAD type selected for the chip.

Bit(s)	Mnemonic	Name	Description
25:24	**SAMPLEDLY**	SAMPLEDLY	**Used for SW to select the turn-around delay cycle between write data end bit and CRC status for SD card** 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
23:22	**FIXDLY**	FIXDLY	**Used for SW to select the delay cycle after clock fix high for the host controller to SD card** 00: 0-T delay

Bit(s)	Mnemonic	Name	Description
			01: 1-T delay 10: 2-T delay 11: 3-T delay
21	**SAMPON**	SAMPON	**Data sample enabling always on** The bit is suggested to be set to 1 when the feedback clock is used and to 0 when the multiple phase clock is used. 0: Data sample enabling not always on 1: Data sample enabling always on
20	**CRCDIS**	CRCDIS	**Switches off data CRC check for SD read data** 0: CRC check is on. 1: CRC check is off.
19	**CMDSEL**	CMDSEL	**Determines whether the host should delay 1-T to latch response from card** 0: Host latches response without 1-T delay 1: Host latches response with 1-T delay.
18:17	**INTLH**	INTLH	**Selects latch timing for SDIO multi-block read interrupt** 00: Host latches INT at the second backend clock after the end bit of the current data block from card is received. (Default) 01: Host latches INT at the first backend clock after the end bit of the current data block from card is received. 10: Host latches INT at the second backend clock after the end bit of the current data block from card is received. 11: Host latches INT at the third backend clock after the end bit of the current data block from card is received.
16	**DSW**	DSW	**Determines whether the host should latch data with 1-T delay or not** For SD card, this bit is suggested to be 0. For MSPRO cards, it is suggested to be 1. 0: Host latches the data with 1-T delay. 1: Host latches the data without 1-T delay.
15	**CMDRE**	CMDRE	**Determines whether the host should latch response token (sent from card on CMD line) at rising edge or falling edge of serial clock** (T.B.D this bit is un-useful) 0: Host latches response at rising edge of serial clock. 1: Host latches response at falling edge of serial clock.
10	**HIGH_SPEED**	HIGH_SPEED	**For high-speed mode when internal sample clock is used** High-speed mode means the SD/MMC serial bus clock rate is bigger than 25MHz. The default speed mode means that the SD/MMC serial bus clock rate is bigger than 25MHz. 0: Default speed 1: High speed
9:8	**DMABURST**	DMABURST	**Used for SW to select burst type when data are transferred by DMA** *Note: Only single mode can support non-4N bytes data transfer in read operation.* 00: Single mode 01: 4-beat incrementing burst 10: 8-beat incrementing burst 11: Reserved
7	**SRCFG1**	SRCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 0: Fast slew rate 1: Slow slew rate
6	**SRCFG0**	SRCFG0	**Output driving capability for pins CMD/BS and SCLK** 0: Fast slew rate 1: Slow slew rate

Bit(s)	Mnemonic	Name	Description
5:3	**ODCCFG1**	ODCCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 000: 4mA 001: 8mA 010: 12mA 011: 16mA
2:0	**ODCCFG0**	ODCCFG0	**Output driving capability for pins CMD/BS and SCLK** 000: 4mA 001: 8mA 010: 12mA 011: 16mA

A0270018 MSDC_IOCON1 SD Memory Card Controller IO Control Register 1 00022022

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														PRCFG_RST/WP	PRVAL_RST/WP	
Type														RW	RW	
Reset														0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		PRCFG_CK	PRVAL_CK			PRCFG_CM	PRVAL_CM			PRCFG_DA	PRVAL_DA			PRCFG_INS	PRVAL_INS	
Type		RW	RW			RW	RW			RW	RW			RW	RW	
Reset		0	1	0		0	0	0		0	1	0		0	1	0

Bit(s)	Mnemonic	Name	Description
18	**PRCFG_RST/WP**	PRCFG_RST_WP	**Pull-up/down register configuration for pin RST/WP** Default value: 0 0: Pull-up resistor in the I/O pad of pin WP is enabled. 1: Pull-down resistor in the I/O pad of pin WP is enabled.
17:16	**PRVAL_RST/WP**	PRVAL_RST_WP	**Pull-up/down register value for pin RST/WP** Default value: 10 00: Pull–up/down resistor in the I/O pad of pin WP are all disabled. 01: Pull-up/down resistor in the I/O pad of pin WP value is 47k. 10: Pull-up/down resistor in the I/O pad of pin WP value is 47k. 11: Pull-up/down resistor in the I/O pad of pin WP value is 23.5k.
14	**PRCFG_CK**	PRCFG_CK	**Pull-up/down register configuration for pin CK** Default value: 0 0: Pull-up resistor in the I/O pad of pin CK is enabled. 1: Pull-down resistor in the I/O pad of pin CK is enabled.
13:12	**PRVAL_CK**	PRVAL_CK	**Pull-up/down register value for pin CLK** Default value: 10 00: Pull-up/down resistor in the I/O pad of pin CLK are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CLK value is 23.5k.
10	**PRCFG_CM**	PRCFG_CM	**Pull-up/down register configuration for pin CM** Default value is 0. 0: Pull-up resistor in the I/O pad of pin CM is enabled. 1: Pull-down resistor in the I/O pad of pin CM is enabled.
9:8	**PRVAL_CM**	PRVAL_CM	**Pull-up/down register value for pin CMD/BS** Default value: 00 00: Pull-up/down resistor in the I/O pad of pin CMD/BS are all disabled.

Bit(s)	Mnemonic	Name	Description
			01: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 23.5k.
6	**PRCFG_DA**	PRCFG_DA	**Pull-up/down register configuration for pin DAT0, DAT1, DAT2 and DAT3** Default value: 0 0: Pull-up resistor in the I/O pad of pin DAT is enabled. 1: Pull-down resistor in the I/O pad of pin DAT is enabled.
5:4	**PRVAL_DA**	PRVAL_DA	**Pull-up/down register value for pin DAT0, DAT1, DAT2 and DAT3** Default value: 10 00: Pull-up/ down resistor in the I/O pad of pin DAT are all disabled. 01: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 10: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 11: Pull-up/down resistor in the I/O pad of pin DAT value is 23.5k.
2	**PRCFG_INS**	PRCFG_INS	**Pull-up/down register configuration for pin INS** Default value: 0 0: Pull-up resistor in the I/O pad of pin WP is enabled. 1: Pull-down resistor in the I/O pad of pin WP is enabled.
1:0	**PRVAL_INS**	PRVAL_INS	**Pull-up/down register value for pin INS** Default value: 10 00: Pull-up/down resistor in the I/O pad of pin INS are all disabled. 01: Pull-up/down resistor in the I/O pad of pin INS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin INS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin INS value is 23.5k.

A0270020 SDC_CFG SD Memory Card Controller Configuration Register 00008000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DTOC								WDOD				SDIO		MDLEN	SIEN
Type	RW								RW				RW		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BSYDLY				BLKLEN											
Type	RW				RW											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to configure the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
31:24	**DTOC**	DTOC	**Data time-out counter** The period from finish of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit of 65,536 serial clock. See the register field descriptions of register bit RDINT for reference. 00000000: Extend 65,536 more serial clock cycles 00000001: Extend 65,536x2 more serial clock cycles 00000010: Extend 65,536x3 more serial clock cycles 00000011~11111110: ... 11111111: Extend 65,536x 256 more serial clock cycles

Bit(s)	Mnemonic	Name	Description
23:20	**WDOD**	WDOD	**Write data output delay** The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
19	**SDIO**	SDIO	**Enables SDIO** 0: Disable SDIO mode 1: Enable SDIO mode
17	**MDLEN**	MDLEN	**Enables multiple data line** The register can be enabled only when SD memory card is applied and detected by software application. It is the responsibility of the application to program the bit correctly when an multi-media card is applied. If an multi-media card is applied and 4-bit data line is enabled, the 4 bits will be output every serial clock. Therefore, data integrity will fail. 0: Disable 4-bit data line 1: Enable 4-bit data line
16	**SIEN**	SIEN	**Enables serial interface** It should be enabled as soon as possible before any command. 0: Disable serial interface for SD 1: Enable serial interface for SD
15:12	**BSYDLY**	BSYDLY	**Only valid for the commands with R1b response** If the command has a response of R1b type, the SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if the operation in SD memory card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, the controller will abandon the detection. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
11:0	**BLKLEN**	BLKLEN	**Block length** The register field is used to define the length of one block in unit of byte in a data transaction. The maximum value of block length is 2,048 bytes. 000000000000: Reserved 000000000001: Block length is 1 byte. 000000000010: Block length is 2 bytes. 000000000011~011111111110: ... 011111111111: Block length is 2,047 bytes. 100000000000: Block length is 2,048 bytes.

A0270024 SDC_CMD SD Memory Card Controller Command Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																CMDFAIL

Type																RW
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTC	STOP	RW	DTYPE		IDRT	RSPTYP			BREAK	CMD					
Type	RW	RW	RW	RW		RW	RW			RW	RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto the SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.

Bit(s)	Mnemonic	Name	Description
16	**CMDFAIL**	CMDFAIL	**If 4-bit SDIO mode is enabled and when CMD/DAT error occurrs, set this bit to select whether to "wait stop command" or "wait data state machine idle".** 0: Wait stop command 1: Wait data state machine idle
15	**INTC**	INTC	**Indicates if the command is GO_IRQ_STATE** If the command is GO_IRQ_STATE, the period between command token and response token will not be limited. 0: The command is not GO_IRQ_STATE. 1: The command is GO_IRQ_STATE.
14	**STOP**	STOP	**Indicates if the command is a stop transmission command** 0: The command is not a stop transmission command. 1: The command is a stop transmission command.
13	**RW**	RW	**Defines the command is a read command or write command** The register bit is valid only when the command causes a transaction with data token. 0: The command is a read command. 1: The command is a write command.
12:11	**DTYPE**	DTYPE	**Defines data token type for the command** 00: No data token for the command 01: Single block transaction 10: Multiple block transaction, i.e. the command is a multiple block read or write command. 11: Stream operation. It can only be used when an multi-media card is applied.
10	**IDRT**	IDRT	**Identification response time** The register bit indicates if the command has a response with NID (i.e. 5 serial clock cycles as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 1.0) response time. The register bit is valid only when the command has a response token. Thus the register bit must be set to 1 for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD). 0: Otherwise 1: The command has a response with NID response time.
9:7	**RSPTYP**	RSPTYP	**Defines response type for the command** For commands with R1 and R1b response, register SDC_CSTA (not SDC_STA) updates after response token is received. This register SDC_CSTA contains the status of the SD, and it can be used as a response interrupt source. *Note: If CMD7 is used with all 0's RCA, then RSPTYP must be "000". Command "GO_TO_IDLE" also has RSPTYP='000'.* 000: There is no response for the command, e.g. broadcast command without response and GO_INACTIVE_STATE command.

Bit(s)	Mnemonic	Name	Description
			001: The command has R1 response. R1 response token is 48-bit. 010: The command has R2 response. R2 response token is 136-bit. 011: The command has R3 response. Even though R3 is 48-bit response, it does not contain CRC checksum. 100: The command has R4 response. R4 response token is 48-bit. (only for MMC) 101: The command has R5 response. R5 response token is 48-bit. (only for MMC) 110: The command has R6 response. R6 response token is 48-bit. 111: The command has R1b response. If the command has a response of R1b type, SD controller must monitor the data line 0 for card busy status from the bit time that is 2 or 4 serial clock cycles after the command end bit to check if the operation in SD memory card has finished. There are two cases for detection of card busy status. The first case is that the host stops the data transmission during an active write data transfer. The card will assert busy signal after the stop transmission command end bit followed by 4 serial clock cycles. The second case is that the card is in idle state or receiving a stop transmission command between data blocks when multiple block write command is in progress. The register bit will be valid only when the command has a response token.
6	**BREAK**	BREAK	**Aborts pending MMC GO_IRQ_MODE command** It is only valid for a pending GO_IRQ_MODE command waiting for MMC interrupt response. 0: Other fields are valid. 1: Break a pending MMC GO_IRQ_MODE command in the controller. Other fields are invalid.
5:0	**CMD**	CMD	**SD memory card command** Total 6 bits.

A0270028 SDC_ARG SD Memory Card Controller Argument Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ARG[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ARG[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains the argument of the SD memory card command.

Bit(s)	Mnemonic	Name	Description
31:0	**ARG**	ARG	Contains argument of the SD memory card command.

A027002C SDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WP											FEDATBUSY	FECMDBUSY	BEDATBUSY	BECMDBUSY	BESDCBUSY
Type	RO											RO	RO	RO	RO	RO

Reset	0											0	0	0	0	0

Overview: The register contains various statuses of SD controller as the controller is configured as the host of SD memory card.

Bit(s)	Mnemonic	Name	Description
15	**WP**	WP	**Detects the status of write protection switch on SD memory card** The register bit shows the status of write protection switch on SD memory card. There is no default reset value. Pin WP (Write Protection) is only useful when the controller is configured for SD memory card. 1: Write protection switch on, i.e. memory card is desired to be write-protected. 0: Write protection switch off, i.e. memory card is writable.
4	**FEDATBUSY**	FEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** This bit indicates directly the CMD line at card clock domain. For those commands without data but still involving DAT line, the register bit is useless. For example, if an erase command is issued, checking if the register bit is 0 before issuing the next command with data will not guarantee that the controller is idle. In this case, use register bit BESDCBUSY. 0: No transmission is going on DAT line on SD bus. 1: There is transmission going on DAT line on SD bus.
3	**FECMDBUSY**	FECMDBUSY	**Indicates if there is any transmission going on CMD line on SD bus** This bit indicates directly the CMD line at card clock domain. 0: No transmission is going on CMD line on SD bus. 1: There is transmission going on CMD line on SD bus.
2	**BEDATBUSY**	BEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** 0: Backend SDC controller gets the info that no transmission is going on DAT line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on DAT line on SD bus.
1	**BECMDBUSY**	BECMDBUSY	**Indicates if there is any transmission going on CMD line on SD bus** This bit shows backend controller's CMD busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SDC controller gets the info that no transmission is going on CMD line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on CMD line on SD bus.
0	**BESDCBUSY**	BESDCBUSY	**Indicates if SD controller is busy, i.e. is there any transmission going on CMD or DAT line on SD bus** This bit shows backend controller's SDC busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SD controller is idle. 1: Backend SD controller is busy.

A0270030 SDC_RESP0 SD Memory Card Controller Response Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[31:0][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	RESP[31:0][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[31:0]**	RESP_31_0	

A0270034 SDC_RESP1 SD Memory Card Controller Response Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[63:32][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[63:32][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[63:32]**	RESP_63_32	

A0270038 SDC_RESP2 SD Memory Card Controller Response Register 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[95:64][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[95:64][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[95:64]**	RESP_95_64	

A027003C SDC_RESP3 SD Memory Card Controller Response Register 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[127:96][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[127:96][15:0]															
Type	RO															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, i.e. response of commands ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[127:96]**	RESP_127_96	

A0270040 SDC_CMDSTA SD Memory Card Controller Command Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**RSPCRCERR**	**CMDTO**	**CMDRDY**
Type														RC	RC	RC
Reset														0	0	0

Overview: The register contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register can also be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
2	**RSPCRCERR**	RSPCRCERR	**CRC error on CMD detected** '1' indicates the SD controller detects a CRC error after reading a response from the CMD line. 0: Otherwise 1: SD controller detects a CRC error after reading a response from the CMD line.
1	**CMDTO**	CMDTO	**Time-out on CMD detected** '1' indicates the SD controller detects a time-out condition while waiting for a response on the CMD line. 0: Otherwise 1: SD controller detects a timeout condition while waiting for a response on the CMD line.
0	**CMDRDY**	CMDRDY	**For command without response, the register bit will be 1 once the command is completed on SD bus** For command with response, the register bit will be 1 whenever the command is issued onto the SD bus and its corresponding response is received without CRC error. 0: Otherwise 1: Command with/without response is finished successfully without CRC error.

A0270044 SDC_DATSTA SD Memory Card Controller Data Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DATCRCERR								DATTO	BLKDONE
Type							RC								RC	RC
Reset							0	0	0	0	0	0	0	0	0	0

Overview: The register contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register can be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
9:2	**DATCRCERR**	DATCRCERR	**CRC error on DAT detected** '1' indicates that the SD controller detects a CRC error for bit n after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. 0: Otherwise 1: SD controller detects a CRC error after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. *Note: n is 7 ~ 1 for 8-bits mode. Each bit is read and cleared individually.*
1	**DATTO**	DATTO	**Time-out on DAT detected** A '1' indicates that the SD controller detects a time-out condition while waiting for data token on the DAT line. 0: Otherwise 1: SD controller detects a time-out condition while waiting for data token on the DAT line.
0	**BLKDONE**	BLKDONE	**Indicates the status of data block transfer** 0: Otherwise 1: A data block is successfully transferred.

A0270048 SDC_CSTA SD Memory Card Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSTA [31:0][31:16]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSTA [31:0][15:0]															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: After commands with R1 and R1b respond this register containing the status of the SD card, it will be used as a response interrupt source. In all register fields, logic high indicates error, and logic low indicates no error. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
31:0	**CSTA [31:0]**	CSTA_31_0	CSTA31: OUT_OF_RANGE. The command's argument is out of the

allowed range for this card.

CSTA30: ADDRESS_ERROR. A misaligned address that does not match the block length is used in the command.

CSTA29: BLOCK_LEN_ERROR. The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.

CSTA28: ERASE_SEQ_ERROR. An error in the sequence of erase commands occurs.

CSTA27: ERASE_PARAM. An invalid selection of write-blocks for erase occurs.

CSTA26: WP_VIOLATION. Attempt to program a write-protected block.

CSTA25: Reserved. Return to 0.

CSTA24: LOCK_UNLOCK_FAILED. Set when a sequence or password error is detected in lock/unlock card command or if there is an attempt to access a locked card.

CSTA23: COM_CRC_ERROR. The CRC check of the previous command fails.

CSTA22: ILLEGAL_COMMAND. Command not legal for the card state.

CSTA21: CARD_ECC_FAILED. Card internal ECC is applied but fails to correct the data.

CSTA20: CC_ERROR. Internal card controller error.

CSTA19: ERROR. A general or unknown error occurs during the operation.

CSTA18: UNDERRUN. The card cannot sustain data transfer in stream read mode.

CSTA17: OVERRUN. The card cannot sustain data programming in stream write mode.

CSTA16: CID/CSD_OVERWRITE. It can be either one of the following errors: 1) The CID register has been already written and cannot be overwritten; 2) The read-only section of the CSD does not match the card; 3) An attempt to reverse the copy (set as the original) or permanent WP (unprotected) bits is made.

CSTA[15: 4]: Reserved. Return to 0.

CSTA3: AKE_SEQ_ERROR. Error in the sequence of authentication process

CSTA[2: 0]: Reserved. Return to 0.

A027004C SDC_IRQMASK0 SD Memory Card IRQ Mask Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [31:0][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [31:0][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. See the register descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from registers SDC_CMDSTA and SDC_DATSTA.

Bit(s)	Mnemonic	Name	Description
31:0	**IRQMASK [31:0]**	IRQMASK_31_0	

A0270050 SDC_IRQMASK1 SD Memory Card IRQ Mask Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [63:32][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [63:32][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then the interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.

Bit(s)	Mnemonic	Name	Description
31:0	**IRQMASK [63:32]**	IRQMASK_63_32	

A0270054 SDIO_CFG SDIO Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**DISSEL**		**INTCSEL**	**DSBSEL**		**INTEN**
Type											RW		RW	RW		RW
Reset											0		0	0		0

Overview: The register is used to configure functions for SDIO.

Bit(s)	Mnemonic	Name	Description
5	**DISSEL**	DISSEL	**Selects data block interrupt source** 0: The host detects SDIO interrupt during interrupt period between two data blocks of multiple block data access. 1: The host ignores SDIO interrupt during interrupt period between two data blocks of multiple block data access.
3	**INTCSEL**	INTCSEL	**Selects interrupt control** 0: The host detects DAT1 low as SDIO interrupt. 1: The host detects DAT3/DAT2/DAT1/DAT0 4'b1101 as SDIO interrupt.
2	**DSBSEL**	DSBSEL	**Selects data block start bit** 0: Use data line 0 as start bit of data block. Other data lines are ignored.

Bit(s)	Mnemonic	Name	Description
			1: Start bit of a data block is received only when all data line 0-3 become low.
0	**INTEN**	INTEN	**Enables interrupt for SDIO** 0: Disable 1: Enable

A0270058 SDIO_STA SDIO Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**IRQ**
Type																RO
Reset																0

Overview: This register is used to identify if there is SDIO interrupt during the interrupt period on data line.

Bit(s)	Mnemonic	Name	Description
0	**IRQ**	IRQ	**SDIO interrupt exists on the data line.** For example, when in the interrupt period or the 1 -bit data line mode and DAT1/5 goes low from high, this bit will become 1 from 0. If DAT1/5 goes high from low, this bit will become 0 from 1. 0: There is no SDIO interrupt existing on the data line. 1: There is SDIO interrupt existing on the data line.

A0270080 CLK_RED CLK Latch Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**CMD_RED**													
Type			RW													
Reset			0													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**DAT_RED**						**CLKPAD_RED**	**CLK_LATCH**						
Type			RW						RW	RW						
Reset			0						0	0						

Overview: The register is used to configure the MSDC sample data/response clock. Note that only when MSDC_IOCON[19] = 1 will the host latch response; otherwise MSDC FSM will handle the response from PAD directly.

Bit(s)	Mnemonic	Name	Description
29	**CMD_RED**	CMD_RED	**Determines the command response from card output is latched at falling edge or rising edge of internal clock** Only effective when CLK_LATCH = 1 0: Internal clock rising edge to latch response 1: Internal clock falling edge to latch response
13	**DAT_RED**	DAT_RED	**Determines the input data from card output is latched at falling edge or rising edge of internal sample clock** Only effective when CLK_LATCH = 1

Bit(s)	Mnemonic	Name	Description
			0: Internal clock rising edge to latch data 1: Internal clock falling edge to latch data
7	**CLKPAD_RED**	CLKPAD_RED	**Determines the input data from card is latched at falling edge or rising edge of the feedback clock from pad** The suggested setting is 0 when SD serial clock is lower than 25MHz. The suggestion setting is 1 when SD serial clock is higher than 25MHz. The suggestion setting is 0 for MMC card no matter the serial clock rate is high speed or default speed. Only effective when CLK_LATCH = 0. 0: Internal feedback clock rising edge to latch data/response 1: Internal feedback clock falling edge to latch data/response
6	**CLK_LATCH**	CLK_LATCH	**Determines which clock to latch data from card** The suggested setting is 1 if SCLKF in register field MSDC_CFG is 0x0. Otherwise, the suggested setting is 0. 0: Internal feedback clock is used to latch data/response from card. 1: Internal clock is used to latch data/response from card.

A0270098 DAT_CHECKSUM MSDC Rx Data Checksum Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAT_CHECKSUM[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT_CHECKSUM[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to compete the checksum value of Rx read data

Bit(s)	Mnemonic	Name	Description
31:0	**DAT_CHECKSUM**	DAT_CHECKSUM	The checksum algorithm is 32-bit XOR.

3.14 BTIF

3.14.1 General Description

Bluetooth Interface (BTIF) is designed in SOC (BT+GSM) in order to be instead of the UART interface between the BT chip and the baseband chip. As in the UART design, BTIF is an APB slave which transmits or receives data by MCU access or through DMA/VFIFO.

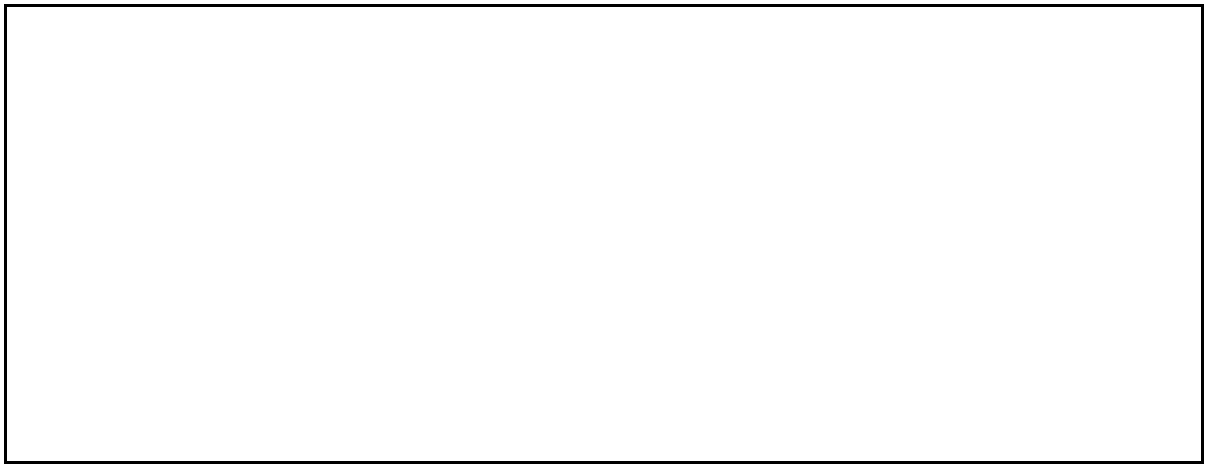


Figure 1: Interface connection between BT and Baseband system

3.14.2 Register Definition

BTIF+0000h Rx buffer register BTIF_RBR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RBR[7:0]							
Type									RO							

RBR Rx buffer register. A read-only register. The received data can be read by accessing this register. This register is valid only when BTIF_FAKELCR[7] (0x0C) is 0.

BTIF+0000h Tx holding register BTIF_THR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									THR[7:0]							
Type									WO							

THR Tx holding register. A write-only register. The data to be transmitted are written to this register and sent to the Bluetooth via BTIF. This register is valid only when BTIF_FAKELCR[7] (0x0C) is 0.

BTIF+0004h Interrupt enable register BTIF_IER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															TXEEN	RXFEN
Type															W/R	W/R
Reset															0	0

This register is valid only when BTIF_FAKELCR[7] is 0.

TXEEN Enables Tx empty interrupt. When set to 1, an interrupt will be generated if the Tx holding register is empty.

0 No interrupt will be generated if the Tx holding register is empty.

1 An interrupt will be generated if the Tx holding register is empty

RXFEN Enables Rx full interrupt. When set to 1, an interrupt will be generated if the Rx buffer contains data.

0 No interrupt will be generated if the Rx buffer contains data.

1 An interrupt will be generated if the Rx buffer contains data.

BTIF+0008h Interrupt identification register BTIF_IIR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													ID2	ID1	ID0	NINT
Type													RO	RO	RO	RO
Reset													0	0	0	1

This register is valid only when BTIF_FAKELCR (0x0C) is not 0xBF.

IIR Identifies if there are pending interrupts. The following table lists the IIR[5:0] codes associated with the possible interrupts:

R[3:0]	Priority level	Interrupt	Source
0001	-	No pending interrupt	
0100	1	Rx data received	Rx data received
1100	2	Rx data time-out	Time-out on character in Rx buffer
0010	3	Tx holding register empty	Tx holding register empty.

Table 49: IIR[5:0] codes associated with the possible interrupts

Rx data received interrupt

A Rx received interrupt (IIR[3:0] = 0x04) is generated when RXFEN (IER[0]) is set and Rx data are placed in the Rx buffer register. The interrupt is cleared by reading the Rx buffer register.

Rx data time-out interrupt

The Rx data time-out interrupt will be generated if all of the following conditions are applied:

1. Rx buffer is empty.
2. The most recent character is received longer than (RTOCNT*bclk period*4).
3. RXFEN (IER[0]) is set to 1.

The time-out timer is restarted upon receipt of a new byte from the Rx shift register. This interrupt is only valid while VFIFO is used. This register is cleared by reading the VFIFO status register (0x4C).

Tx holding register empty

A Tx holding register empty interrupt (IIR[3:0] = 0x02) is generated when TXEEN(IER[1]) is set and no data are placed in the Tx holding register. This interrupt is cleared by writing data into BTIF_THR (0x00).

BTIF+0008h FIFO_CTRL BTIF_FIFOCTRL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														CLRT	CLRR	
Type														WO	WO	
Reset														0	0	

This register is valid only when BTIF_FAKELCR (0x0C) is not 0xBF.

CLRT Clears transmit FIFO. This bit is self-clearing.
- **0** Leave Tx FIFO intact.
- **1** Clear all the bytes in Tx FIFO.

CLRR Clears receive FIFO. This bit is self-clearing.
- **0** Leave Rx FIFO intact.
- **1** Clear all the bytes in Rx FIFO.

BTIF+000Ch FAKE LCR BTIF_FAKELCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FAKELCR[7:0]							
Type									R/W							
Reset									0	0	0	0	0	0	0	0

FAKELCR This register is added to synchronize the software control method of UART. When FAKELCR[7] is 1, RBR(0x00), THR(0x00) and IER(0x04) will not be readable/writable. When FAKELCR is 0xBF, RBR(0x00), THR(0x00), IER(0x04), IIR(0x08) and LSR(0x14) will not be readable/writable.

BTIF+0014h Line status register BTIF_LSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										TEMT	THRE					DR
Type										RO	RO					RO
Reset										1	1					0

LSR Line status register. Readable when LCR != 0xBF.

TEMT Tx holding register is empty.
- **0** Empty conditions are not met.
- **1** This bit is set when the Tx holding register is empty.

THRE Indicates if Tx FIFO is reduced to its trigger level

0 Reset whenever the contents of Tx FIFO are more than its trigger level (FIFOs are enabled)

1 Set whenever the contents of Tx FIFO are reduced to its trigger level (FIFOs are enabled)

DR Data Ready

0 Cleared by reading the Rx buffer.

1 Set by the Rx buffer becoming full.

BTIF+0048h Sleep enable register BTIF_SLEEP_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																R/W
Reset																0

SLEEP_EN For sleep mode issue

0 Does not deal with sleep mode indication signal

1 Activate flow control according to software initial settings when the chip enters the sleep mode. Release hardware flow when the chip wakes up.

BTIFn+004Ch DMA enable register BTIF_DMA_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type														R/W	R/W	R/W
Reset														0	0	0

RX_DMA_EN RX_DMA mechanism enable signal

0 Does not use DMA in Rx.

1 Use DMA in Rx. When this register is enabled, the flow control is based on the DMA threshold and generates a time-out interrupt

TX_DMA_EN TX_DMA mechanism enable signal

0 Does not use DMA in Tx.

1 Use DMA in Tx. When this register is enabled, the flow control is based on the DMA threshold and generates a time-out interrupt for DMA.

TO_CNT_AUTORST Time-out counter auto reset register

0 After Rx time-out takes place, the software shall reset the interrupt by reading BTIF 0x4C.

1 The time-out counter will be auto reset.

BTIF+0054h Rx time-out count BTIF_RTOCNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RTOCNT[7:0]							
Type									R/W							
Reset									0x40							

RTOCNT Used for Rx time-out interrupt. The Rx time-out interrupt will be generated when:

1. RXFEN (0x04[0]) is set to 1.
2. Rx buffer is empty.
3. The most recent character is received longer than (RTOCNT*bclk period*4).

BTIF+0060h TRX_TRIGGER_LEVEL BTIF_TRI_LVL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BTIF_LOOP	RX_TRI_LVL			TX_TRI_LVL			
Type									R/W	R/W			R/W			
Reset									0x0	0x5			0x0a			

TX_TRI_LVL Used for Tx FIFO trigger threshold. THRE(0x14[5]) will be set if the data in the TXFIFO are less than TX_TRI_LVL.

RX_TRI_LVL Used for Rx FIFO trigger threshold. A Rx trigger interrupt (IIR(0x08]) = 4) might be set if the data in the RXFIFO are more than RX_TRI_LVL. The output flow control signal will also be set if the data in the RXFIFO are more than RX_TRI_LVL.

BTIF_LOOP This register is used to enable BTIF loop back mode. The data output from Tx will be received by Rx.

BTIF+0064h SLEEP_WAKEUP BTIF_WAK

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_WAKE
Type																WO
Reset																1

SLEEP_WAKE ARM9 side btif_sleep_wakeup_in_b is connected to eint[16] (ARM9 has eint[19:0].
ARM7 side btif_sleep_wakeup_in_b is connected to eint[0] (ARM7 has eint[3:0])

BTIF+0068h ASYNC_WAIT_TIME BTIF_WAT_TIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name											WAT_TIME_2	WAT_TIME_1
Type											R/W	R/W
Reset											0X2	0x2

ASYNC_WAIT_TIME This register is used to set the waiting time of RX read out.

WAT_TIME_1 The first level of wait time.

WAT_TIME_2 The second level of wait time.

Notes: The value of WAT_TIME_1/ WAT_TIME_1 cant not be less than 0x2.

BTIF+006Ch NEW_HANDSHAKE BTIF_HANDSHAKE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RTO_EXT	HIGH_SPEED_EN	HANDSHAKE_EN
Type														R/W	R/W	R/W
Reset														0	0	1

NEW_HANDSHAKE The default value of handshake is zero. And the function of handshake is disable. The function of BTIF has its limitation. The ratio of bclk is not larger than two. Otherwise, two system data transmission will be wrong.

If the value of handshake is one. The ratio of bclk is free.

HANDSHAKE_EN This register is used to enable handshake mode.

high_speed_en This register is used to enable high speed mode. It is reserved.

RTO_EXT This register is used to extend the value of RX time out counter(16*rto_time).

3.15 FM FSPI Master

1.1.1. General Description

FSPI is an internal SPI master that provides access to the FMSYS register files. The protocol is proprietary and previously defined between the FSPI master controller and the FMSYS slave controller. All FMSYS register reads/writes are operated based on this controller which provides the following features:

- Basic single read or write access
- Supports multiple reads of up to 8 data reads within 1 SPI read transfer

The multiple-read accesses can be utilized to off-load the MCU's control effort when accessing large data buffers such as RDS content from FMSYS. The internal read buffer is a FIFO with maximum depth of 5 entries.

1.1.2. Single RD/WR SPI Timing Diagram

1.1.3. Multiple Read Configuration

For the multiple-read access mode, the parameters can be set as the following:

3.15.1 Register Map

Address	Name	Width	Register function
0xA0260000	**CTRL**	32	Main control
0xA0260004	**ADDR**	32	Address + Command
0xA0260008	**WR DATA**	32	Write data port
0xA026000C	**RD DATA**	32	Read data port
0xA0260010	**CONFIG1**	32	Config set 1
0xA0260014	**CONFIG2**	32	Config set 2
0xA0260020	**MODE SEL**	32	FMSYS HCI mode select
0xA0260030	**RESET**	32	Soft reset
0xA0260034	**DEBUG1**	32	Debug 1
0xA0260038	**DEBUG2**	32	Debug 2
0xA026003C	**DEBUG3**	32	Debug 3

Address	Name	Width	Register function
0xA0260040	**DEBUG4**	32	Debug 4
0xA0260048	**DEBUG5**	32	Debug 5
0xA0260050	**RRCCAL**	32	R/RC calibration value from BT
0xA0260054	**RESERVED1**	32	FM2MCU_RESERVED
0xA0260058	**RESERVED2**	32	MCU2FM_RESERVED

3.15.2 Register Definition

0xA0260000 CTRL Main Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																**START**
Type																R/W
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**START**	START	**Setting up this bit will start the SPI command sequence.** The bit will be auto-cleared upon completion of the SPI commands. Therefore this bit can also be polled as a completion indicator.

0xA0260004 ADDR Address + Command 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne								**RD_WR**	**ADDR**							
Type								R/W	R/W							
Reset								0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**ADDR**	FMSYS ADDR	**Sets up FMSYS register address**
8	**RD_WR**	RD_WR CMD	**Indicates if the transaction is a master read or master write transaction** 0: Master write 1: Master read

0xA0260008 WR DATA Write Data Port 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Mne	WRDATA															
Type	R/W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**WRDATA**	WR_DATA	Write data

0xA026000C RD DATA Read Data Port 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RDDATA															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**RDDATA**	RD_DATA	Read port of the 8-entry read buffer

0xA0260010 CONFIG1 Config Set 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DIV								GUARD TIME							EXTRA CLK EN
Type	R/W								R/W							R/W
Reset	0	0	0	0	0	0	0	0	0	0	0	0				0

Bit(s)	Mnemonic	Name	Description
0	**EXTRA CLK EN**	RGF_EXTRA_CLK_EN	**For debugging only. Enables extra SPI CLKS to be generated even after CS is de-asserted.** 0: Disable 1: Enable *Note: This bit is effective only when EN_MAINCON = 0.*
7:4	**GUARD TIME**	RGF_GUARD_TIME	**Configures the amount of guard steps to insert between consecutive commands**
15:8	**DIV**	RGF_DIV	**Sets up the final output SPI CLK period** The clock period is calculated as: 1/(bclk_ck freq/div/2)

0xA0260014 CONFIG2 Config Set 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														TRAN LEN		

Type														R/W		
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**TRAN LEN**	TRANS LEN	**Allows MCU to set up the number of data reads within 1 SPI transfer to execute automatically** Actual data reads = (TRAN LEN + 1)*16b

0xA0260020 MODE SEL FMSYS HCI Mode Select 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																**MODE_SEL**
Type																R/W
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**MODE_SEL**	HCI_MODE_SEL	**Controls the top FMSYS HCI signal mux** The allowed selections are: 1'b0: Select internal fspi_mas control 1'b1: Select external master SPI control

0xA0260030 RESET Soft Reset 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															**FIFO ADDR CLR**	**SOFT RESET**
Type															R/W	R/W
Reset															0	0

Bit(s)	Mnemonic	Name	Description
0	**SOFT RESET**	SOFT_RESET	**For debugging only. Clears the fspi_mas state to reset state** 0: De-assert soft reset 1: Assert soft reset
1	**FIFO ADDR CLR**	FIFO_ADDR_CLR	**For debugging only** 0: Normal operation 1. Clear FIFO addresses

0xA0260034 DEBUG1 Debug 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne		RD_ADDR				WR_ADDR			FSPI_STAT		WR_FULL	RD_EMPTY		OFFSET		
Type		R				R			R		R	R		R		
Reset		0	0	0		0	0	0	0	0	0	0		0	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**OFFSET**	FIFO OFFSET	
4	**RD_EMPTY**	READ EMPTY	
5	**WR_FULL**	WRITE FULL	
7:6	**FSPI_STAT**	FSPI STATE	
10:8	**WR_ADDR**	WR ADDR	
14:12	**RD_ADDR**	RD ADDR	

0xA0260038 DEBUG2 Debug 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	CUR_RDDATA															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**CUR_RDDATA**	RDDATA	Read data shift register content

0xA026003C DEBUG3 Debug 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne									AUTO_ADDR							
Type									R							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	**AUTO_ADDR**	AUTO ADDR	Auto address increment read back

0xA0260040 DEBUG4 Debug 4 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Mne										RD_ADDR				WR_ADDR		
Type										R				R		
Reset										0	0	0		0	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**WR_ADDR**	WR ADDR	
6:4	**RD_ADDR**	RD ADDR	

0xA0260048 DEBUG5 Debug 5 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															FSPI_RGF_RSTB	FSPI_CORE_RSTB
Type															R/W	R/W
Reset															0	0

Bit(s)	Mnemonic	Name	Description
0	**FSPI_CORE_RSTB**	FSPI CORE RSTB	**When set to 1, it resets the core logic inside FMSYS.** Set to 0 to de-assert.
1	**FSPI_RGF_RSTB**	FSPI RGF RSTB	**When set to 1, it resets the RGF inside FMSYS.** Set to 0 to de-assert.

0xA0260050 RRCCAL R/RC Calibration value from BT 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne						RCCAL_VAL	RCCAL NOBIAS					RCAL_VAL	RCAL NOBIAS			
Type						R/W	R/W					R/W	R/W			
Reset						0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
3:0	**RCAL_NOBIAS**	RCAL_NOBIAS	BT R calibration result
4	**RCAL_VAL**	RCAL_VAL	BT R calibration valid
9:5	**RCCAL_NOBIAS**	RCCAL_NOBIAS	BT RC calibration result
4	**RCCAL_VAL**	RCCAL_VAL	BT RC calibration valid

0xA0260054 RESERVED1 RESERVED1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																

Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	FM2MCU_RESERVED															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**FM2MCU_RESERVED**	FM2MCU_RESERVED	Reserved for FM2MCU

0xA0260058 RESERVED2 RESERVED2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	MCU2FM_RESERVED															
Type	R/W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**MCU2FM_RESERVED**	MCU2FM_RESERVED	Reserved for MCU2FM

3.16 HIF

3.16.1 General Description

This module is used to help peripheral DMA to access the external device which is connected to the host interface (NLD[7:0], LWRB, LRDB, LPA0, LPCE1B, LPCE2B). There is one HIF module in MT6260D. HIF's base address is 0xA0190000.

The data path of writing to the external device is:

1. DMA moves data from memory to HIF internal buffer through dedicated HIF data port for DMA (HIF_DMA_DATA).
2. HIF moves data from its internal buffer to the external device through the host interface. nli_arbiter is used to arbitrate the ownership of shared host interface, which is shared among LCM, NAND, Wi-Fi and TD-modem.

Data path of write to external device

The data path of reading from the external device is:

1. HIF reads data from the external device and puts data into HIF internal buffer.
2. DMA moves data from HIF internal buffer to memory through dedicated HIF data port for DMA (HIF_DMA_DATA).

Data path of read from external device

3.16.2 Programming Guide

HIF provides two access modes to transfer data from/to the external device. One is MCU mode, and the other is DMA mode.

In the MCU mode, MCU transfers data from/to the external device by reading/writing the two ports, HIF_MCU_DATA0 or HIF_MCU_DATA1. If MCU accesses HIF_MCU_DATA0, the LPA0 pin will be 0 during transferring. If MCU accesses HIF_MCU_DATA1, the LPA0 pin will be 1 during the transfer. Note that the two ports are only for MCU, and DMA is forbidden to access them.

Programming guides of DMA mode

1. Enable HIF interrupt
2. Setup HIF for transfer width: 8 bits or 16 bits, direction = write or read, LPA0 value during transfer and transfer priority on the host interface. All settings are in HIF_CON.
3. Chip selects that the external device is connected: LPCE1B or LPCE2B. It is set by NLI_ARB_CS (0x811A_0014).
4. Setup HIF timing in HIF_TIMING0 and HIF_TIMING1. They only need to be set once, because for an external device, its access timing should be always the same.
5. Set up transfer data amount in HIF_DAMOUNT.
6. Set up peripheral DMA required setting for HIF. Refer to the functional spec of PDMA.
7. Start HIF and peripheral DMA.
8. After the transfer is completed, HIF issues an interrupt.

Table 50. HIF register map

Register address	Register function	Acronym
A019_0000h	HIF status register	HIF_STA
A019_0004h	HIF interrupt enable register	HIF_INTEN
A019_0008h	HIF interrupt status register	HIF_INTSTA
A019_000Ch	HIF start register	HIF_START
A019_0010h	HIF software reset register	HIF_SWRST

Register address	Register function	Acronym
A019_0014h	HIF timing control register 0	HIF_TIME0
A019_0018h	HIF timing control register 1	HIF_TIME1
A019_0020h	HIF control register	HIF_CON
A019_0024h	HIF transfer data amount register	HIF_DAMOUNT
A019_0030h	HIF MCU/DMA access arbitration register	HIF_ACS_ARB
A019_0040h	HIF DMA side data counter register	HIF_DMA_CNT
A019_0044h	HIF PIF side data counter register	HIF_PIF_CNT
A019_0048h	HIF FIFO pointer register	HIF_FIFOPTR
A019_0200h	HIF data port for DMA	HIF_DMA_DATA
A019_0300h	HIF data port 0 for MCU	HIF_MCU_DATA0
A019_0310h	HIF data port 1 for MCU	HIF_MCU_DATA1

3.16.3 Register Definition

A019_0000h HIF Status Register HIF_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																BUSY
Type																R
Reset																0

BUSY HIF busy flag

0 HIF is idle.

1 HIF is transferring data.

A019_0004h HIF Interrupt Enable Register HIF_INTEN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																CPL
Type																R/W
Reset																0

CPL Enables HIF transfer completion interrupt

0 Disable this interrupt

1 Enable this interrupt. HIF will issue an interrupt when a transfer is completed.

A019_0008h HIF Interrupt Status Register HIF_INTSTA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																CPL
Type																W0CLR
Reset																0

CPL HIF transfer completion interrupt status

0 This interrupt is not issued.

1 This interrupt is issued.

A019_000Ch HIF Start Register HIF_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																START
Type																WO
Reset																0

START Starts HIF transfer

0 Stop HIF transfer. If this bit is cleared when HIF is still transferring data, HIF will continue to complete all transfers of data amount specified in HIF_DAMOUNT.

1 Start HIF transfer. HIF will start transfer when this bit has a transition from 0 to 1.

A019_0010h HIF Software Reset Register HIF_SWRST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SWRST
Type																R/W
Reset																0

SWRST Software reset

0 Does not reset HIF

1 Reset HIF

A019_0014h HIF Timing Register 0 HIF_TIME0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Name	C2RH				C2RS						RLT					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	C2WH				C2WS						WST					
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0			0	0	0	0	0	0

WST Write wait state time. See Figure 46.

C2WS Chip Select (LPCEB0) to Write Strobe (LWRB) setup time. See Figure 46. C2WS must <= WST.

C2WH Chip Select (LPCEB0) to Write Strobe (LWRB) hold time. See Figure 46.

RLT Read latency time. See Figure 3.

C2RS Chip Select (LPCEB0) to Read Strobe (LRDB) setup time. See Figure 3. C2RS must <= RLT.

C2RH Chip Select (LPCEB0) to Read Strobe (LRDB) hold time. See Figure 3.

A019_0018h HIF Timing Register 1 HIF_ TIME1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CHW			
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

CHW Chip select high width

Figure 46. Interface write timing diagram

Figure 47. Interface read timing diagram

A019_0020h HIF Control Register — HIF_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**ULTRA**	**A0**	**WRITE**	**HIFW**	
Type												R/W	R/W	R/W	R/W	R/W
Reset												0	0	0	0	0

HIFW Host interface width
0 8 bits
1 16 bits
Others Reserved
Note: The 8-bit width is only used in MT6260D.

WRITE Write transfer flag
0 Read transfer
1 Write transfer

A0 Sets up the value of LPA0 pin during transfer
0 LPA0 = 0 during transfer
1 LPA0 = 1 during transfer

ULTRA Enables ultra high signal to nli_arbiter
0 Does not issue ultra high signal
1 Issue ultra high signal

A019_0024h HIF Data Amount Register — HIF_DAMOUNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	DAMOUNT															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DAMOUNT Specifies the data amount of this transfer. Unit: bytes. 0 means to transferring 1 byte, 1 means to transferring 2 bytes, and N means to transferring N+1 bytes. DAMOUNT+1 must be multiple of HIF width, e.g., if HIF width is 16bit, DAMOUNT+1 must be multiple of 2 bytes.

A019_0030h HIF MCU/DMA Access Arbitration Register HIF_ACS_ARB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														MCU_ACS_STR	PIF_BUSY	MCU_ACS_REQ
Type														W	R	R/W
Reset														0	0	0

MCU_ACS_REQ MCU access request. MCU sets this bit to be 1'b1 to request for MCU data access via the NLD pin. This bit will block the next 8-bit/16-bit DMA data transfer that is intended to be transferred after the current one is completed. Thus, MCU shall poll the PIF_BUSY bit and wait until PIF_BUSY = 0 before proceeding following MCU access steps (set up MCU_ACS_STR and initiate MCU access).

0 MCU access request de-asserted

1 MCU access request asserted

PIF_BUSY Busy/idle status of the NLD pin data-transfer control FSM

0 Idle

1 Busy

MCU_ACS_STR Starts MCU access. MCU sets this bit to 1'b1 to switch data path for MCU access. MCU must wait PIF_BUSY = 0 before setting up this bit. MCU should clear the two bits MCU_ACS_STR and MCU_ACS_REQ after MCU accesses. DMA transfer will auto resume after the two bits are cleared if DMA is originally running.

0 DMA access path is selected.

1 MCU access path is selected.

A019_0040h HIF DMA Side Data Counter Register HIF_DMA_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA_CNT															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	CMD_CNT															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMD_CNT This is a down counter in unit of byte to indicate the amount of data (command phase) waiting to be accessed by DMA. It counts down from {HIF_DAMOUNT[15:2],2'b0} to 0. This value is always the multiple of 4 bytes due to one DMA access is 4 bytes. For debugging.

DATA_CNT This is a down counter in unit of byte to indicate the amount of data (data phase) waiting to be accessed by DMA. It counts down from {HIF_DAMOUNT[15:2],2'b0} to 0. This value is always the multiple of 4 bytes due to one DMA access is 4 bytes. For debugging.

A019_0044h HIF PIF Side Data Counter Register HIF_PIF_CNT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA_CNT															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMD_CNT															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMD_CNT This is a down counter in unit of byte to indicate the amount of data (command phase) waiting to be transferred through the host interface. It counts down from HIF_DAMOUNT to 0. For debugging.

DATA_CNT This is a down counter in unit of byte to indicate the amount of data (data phase) waiting to be transferred through the host interface. It counts down from HIF_DAMOUNT to 0. For debugging.

A019_0044h HIF FIFO Pointer Register HIF_FIFOPTR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										FIFO_RPTR				FIFO_WPTR		
Type										R	R	R		R	R	R
Reset										0	0	0		0	0	0

FIFO_WPTR FIFO write pointer. For debugging.

FIFO_RPTR FIFO read pointer. For debugging.

A019_0200h HIF Data Port for DMA

HIF_DMA_DATA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DATA Data port of HIF for DMA mode. This port is only for DMA, and MCU is forbidden to access it.

A019_0300h HIF Data Port 0 for MCU

HIF_MCU_DATA0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DATA Data port of HIF for MCU mode. When MCU accesses this port, the LPA0 pin is 0 when transferring data. This port is only for MCU, and DMA is forbidden to access it.

A019_0310h HIF Data Port 1 for MCU

HIF_MCU_DATA1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DATA Data port of HIF for MCU mode. When MCU accesses this port, the LPA0 pin is 1 when transferring data. This port is only for MCU, and DMA is forbidden to access it.

3.17 NLI_ARBITER

3.17.1 General Description

This module is used to arbitrate the ownership of the host interface. There are maximal four external devices hooked on the host interface, which are LCM, NAND flash, TD-modem and Wi-Fi. Each external device has its own controller. The LCD controller controls LCM, NAND flash controller controls NAND flash, HIF0 controls TD-modem and HIF1 controls Wi-Fi. These controllers issue requests to nli_arbiter, and nli_arbiter will give grant according to the priority and then send data and control signals to the chip IO pins.

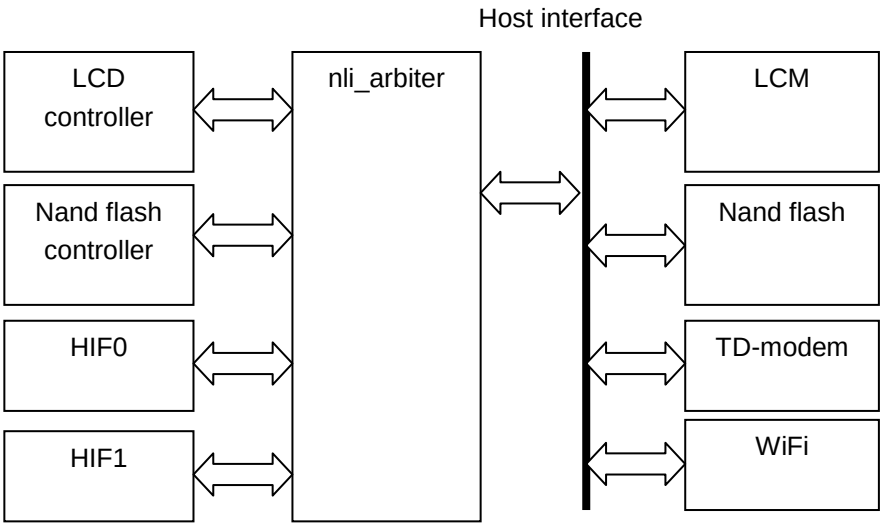


Figure 48. Block diagram of NLI_ARBITER

Table 51. NLI_ARBITER register map

Register address	Register name	Synonym
A01B_0000h	NLI_ARB status	NLI_ARB_STA
A01B_0010h	NLI_ARB software reset register	NLI_ARB_SWRST
A01B_0014h	NLI_ARB chip select	NLI_ARB_CS
A01B_0018h	NLI_ARB continuous grant count	NLI_ARB_CONT_GRANT
A01B_001Ch	NLI_ARB handover count	NLI_ARB_HANDOVER
A01B_0020h	NLI_ARB share-pin control	NLI_ARB_SHRPIN_CON
A01B_0040h	NLI_ARB monitor source select	NLI_ARB_MON_SRC
A01B_0044h	NLI_ARB monitor clear	NLI_ARB_MON_CLR
A01B_0048h	NLI_ARB monitor start	NLI_ARB_MON_START
A01B_0050h	NLI_ARB monitor read counter	NLI_ARB_MON_RD
A01B_0054h	NLI_ARB monitor write counter	NLI_ARB_MON_WR
A01B_0058h	NLI_ARB monitor total cycle counter	NLI_ARB_MON_CYC

3.17.2 Register Definition

A01B_0000h NLI_ARB Status NLI_ARB_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name												ULTRA_ARB				
Type												R	R	R	R	R
Reset												0	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				NORM_ARB								HIF1_REQ	NF_REQ	LCD_REQ	HIF0_REQ	
Type				R	R	R	R	R				R	R	R	R	
Reset				0	0	0	0	1				0	0	0	0	

This register is for debugging.

HIF0_REQ HIF0 request flag
LCD_REQ LCD request flag
NF_REQ NAND flash controller request flag
HIF1_REQ HIF1 request flag

NORM_ARB Normal priority arbiter state
Bit 0 Idle state
Bit 1 HIF0 state
Bit 2 LCD state
Bit 3 NF state
Bit 4 HIF1 state

ULTRA_ARB Ultra high priority arbiter state
Bit 0 Idle state
Bit 1 HIF0 state
Bit 2 LCD state
Bit 3 NF state
Bit 4 HIF1 state

A01B_0010h NLI_ARB Software Reset Register — NLI_ARB_SWRST

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SWRST
Type																R/W
Reset																0

SWRST Software reset
0 Does not reset NLI_ARB
1 Reset NLI_ARB

A01B_0014h NLI_ARB Chip Select — NLI_ARB_CS

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															FAVOR_ULTRA	FAVOR_NORMAL
Type															R/W	R/W
Reset															0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		LPCE3B_SEL				LPCE2B_SEL				LPCE1B_SEL				LPCE0B_SEL		

Type		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	1		1	0	0		0	1	1		0	0	0

LPCE0B_SEL Selects pin LPCE0B source

0 Pin LPCE0B is controlled by LCD chip select 0

1 Pin LPCE0B is controlled by LCD chip select 1

2 Pin LPCE0B is controlled by LCD chip select 2

3 Pin LPCE0B is controlled by HIF0

4 Pin LPCE0B is controlled by HIF1

Others Reserved, LPCE0B is always high.

LPCE1B_SEL Selects pin LPCE1B source

0 Pin LPCE1B is controlled by LCD chip select 0

1 Pin LPCE1B is controlled by LCD chip select 1

2 Pin LPCE1B is controlled by LCD chip select 2

3 Pin LPCE1B is controlled by HIF0

4 Pin LPCE1B is controlled by HIF1

Others Reserved, LPCE1B is always high.

LPCE2B_SEL Selects pin LPCE2B source

0 Pin LPCE2B is controlled by LCD chip select 0

1 Pin LPCE2B is controlled by LCD chip select 1

2 Pin LPCE2B is controlled by LCD chip select 2

3 Pin LPCE2B is controlled by HIF0

4 Pin LPCE2B is controlled by HIF1

Others Reserved, LPCE2B is always high.

LPCE3B_SEL Selects pin LPCE3B source

0 Pin LPCE3B is controlled by LCD chip select 0

1 Pin LPCE3B is controlled by LCD chip select 1

2 Pin LPCE3B is controlled by LCD chip select 2

3 Pin LPCE3B is controlled by HIF0

4 Pin LPCE3B is controlled by HIF1

Others Reserved, LPCE3B is always high.

FAVOR_NORMAL Makes nli_arbiter favor normal requests

0 Make nli_arbiter serves only ultra requests if ultra requests are issued. It serves normal requests only when there is no ultra request.

1 Make nli_arbiter serves normal requests when ultra requests are issued. Its behavior depends on FAVOR_ULTRA.

FAVOR_ULTRA Makes nli_arbiter favor ultra requests when FAVOR_NORMAL = 1.

The chip select pin selection circuit is described in Figure 49. The user must set LPCE*B_SEL and GPIO MUX properly.

Figure 49. Chip selection circuit

The block diagram of nli_arbiter is described in Figure 3. There are two levels of arbiters. Level 1 arbiter is for normal requests and level 2 arbiter is for ultra requests.

If FAVOR_NORMAL = 0, only ultra requests are served when ultra requests are issued. If FAVOR_NORMAL = 1, the winner of level 1 arbiter will be regarded as an ultra request and be arbitrated in the level 2 arbiter. Therefore, normal requests can also be served if FAVOR_NORMAL = 1.

When FAVOR_NORMAL = 1, FAVOR_ULTRA can change the behavior of level 1 arbiter. If FAVOR_ULTRA = 0, only normal requests will be arbitrated in the level 1 arbiter. If FAVOR_ULTRA = 1, all normal requests and ultra requests will be arbitrated in the level 1 arbiter.

Figure 50. NLI_ARBITER block diagram

An example is given in Table 46 to show how FAVOR_NORMAL and FAVOR_ULTRA affect nli_arbiter's behavior.

Assume all the four masters issue requests, HIF0 and LCD controller issue ultra high requests, and HIF1 and NAND flash controller issue normal requests. The grant sequence is shown in Table 46. The red texts mean ultra requests.

*Table 52. **NLI_ARBITER grant example***

FAVOR_NORMAL	FAVOR_ULTRA	Grant sequence
0	0	**HIF0, LCD, HIF0, LCD, HIF0, LCD, HIF0, LCD**…
0	1	Same as FAVOR_NORMAL = 0 and FAVOR_ULTRA = 0
1	0	**HIF0, LCD**, NAND, **HIF0, LCD**, HIF1, **HIF0, LCD**, NAND, **HIF0, LCD**, HIF1…
1	1	**HIF0, LCD**, NAND, **HIF0, LCD**, HIF1, **HIF0, LCD**, HIF0, **HIF0, LCD**, LCD, **HIF0, LCD**, NAND, **HIF0, LCD**, HIF1, **HIF0, LCD**, HIF0, **HIF0, LCD**, LCD…

A01B_0018h NLI_ARB Continuous Grant Count

NLI_ARB_CONT_GRANT

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CONT_GRANT_CNT															CONT_GRANT_EN
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W								R/W
Reset	0	0	0	0	0	0	0	0								0

CONT_GRANT_EN Enables continuous grant. When this function is enabled, nli_arbiter will grant each master continuous transfer of CONT_GRANT_CNT times.

0 Disable continuous grant

1 Enable continuous grant

CONT_GRANT_CNT Continuous grant count

A01B_001Ch NLI_ARB Handover Count

NLI_ARB_HANDOVER

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name					HANDOVER_CYC				LEAVE_HIF1_EN	LEAVE_HIF0_EN	LEAVE_NF_EN	LEAVE_LCD_EN	ENTER_HIF1_EN	ENTER_HIF0_EN	ENTER_NF_EN	ENTER_LCD_EN
Type					R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset					0	0	0	0	0	0	0	0	0	0	0	0

ENTER_LCD_EN Adds handover cycles when the bus grant is switched to LCD from another master

0 Does not add handover cycles

1 Add handover cycles

ENTER_NF_EN Adds handover cycles when the bus grant is switched to NFI from another master

0 Does not add handover cycles

1 Add handover cycles

ENTER_HIF0_EN Adds handover cycles when the bus grant is switched to HIF0 from another master

0 Does not add handover cycles

1 Add handover cycles.

ENTER_HIF1_EN Adds handover cycles when the bus grant is switched to HIF1 from another master

0 Does not add handover cycles

1 Add handover cycles

LEAVE_LCD_EN Adds handover cycles when the bus grant is switched from LCD to another master

0 Does not add handover cycles

1 Add handover cycles

LEAVE _NF_EN Adds handover cycles when the bus grant is switched from NFI to another master

0 Does not add handover cycles

1 Add handover cycles

LEAVE _HIF0_EN Adds handover cycles when the bus grant is switched from HIF0 to another master

0 Does not add handover cycles

1 Add handover cycles

LEAVE _HIF1_EN Adds handover cycles when the bus grant is switched from HIF1 to another master

0 Does not add handover cycles

1 Add handover cycles

HANDOVER_CYC Idle cycles added when switching bus grant

The following waveform gives an example of how handover_cyc works.

A01B_0020h NLI_ARB Share-pin Control

NLI_ARB_SHRPIN_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													NREB_LPA0_SHR_EN	NWEB_LPA0_SHR_EN	NCLE_LRDB_SHR_EN	NALE_LWRB_SHR_EN
Type													R/W	R/W	R/W	R/W
Reset													0	0	0	0

NALE_LWRB_SHR_EN Enables control of NF ALE pin shared with LWRB signal

0 Disabled. NF ALE pin is exclusively for NFI.

1 Enabled. NF ALE pin is shared with LWRB signal. NLI_ARB will arbitrate ownership of this pin.

NCLE_LRDB_SHR_EN Enables control of NF CLE pin shared with LRDB signal

0 Disabled. NF CLE pin is exclusively for NFI.

1 Enabled. NF CLE pin is shared with LRDB signal. NLI_ARB will arbitrate ownership of this pin.

NWEB_LPA0_SHR_EN Enables control of NF WEB pin shared with LPA0 signal

0 Disabled. NF WEB pin is exclusively for NFI.

1 Enabled. NF WEB pin is shared with LPA0 signal. NLI_ARB will arbitrate ownership of this pin.

NREB_LPA0_SHR_EN Enables control of NF RDB pin shared with LPA0 signal

0 Disabled. NF RDB pin is exclusively for NFI.

1 Enabled. NF RDB pin is shared with LPA0 signal. NLI_ARB will arbitrate ownership of this pin.

A01B_0040h NLI_ARB Monitor Source Select

NLI_ARB_MON_SRC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															MON_SRC_SEL	
Type															R/W	R/W
Reset															0	0

MON_SRC_SEL Selects NLI_ARB monitor source

00 LCD

01 Nand flash

10 HIF0

11 HIF1

A01B_0044h NLI_ARB Monitor Clear

NLI_ARB_MON_CLR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																CLR
Type																R/W
Reset																0

CLR Clears NLI_ARB monitor counters, including NLI_ARB_MON_RD, NLI_ARB_MON_WR, and NLI_ARB_MON_CYC. Before starting monitor counters, NLI_ARB_MON_CLR should be set to 1 then cleared to 0, in order to clear monitor counters.

0 Does not clear

1 Clear

A01B_0048h NLI_ARB Monitor Start

NLI_ARB_MON_START

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																START
Type																R/W
Reset																0

START NLI_ARB monitor starts to collect information

0 Stop collecting information

1 Start collecting information

A01B_0050h NLI_ARB Monitor Read Counter

NLI_ARB_MON_RD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RD_CNT[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RD_CNT[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RD_CNT Accumulated read counts when NLI_ARB_MON_START = 1

A01B_0054h NLI_ARB Monitor Write Counter

NLI_ARB_MON_WR

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WR_CNT[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WR_CNT[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WR_CNT Accumulated write counts when NLI_ARB_MON_START = 1

A01B_0058h NLI_ARB Monitor Total Cycle Counter

NLI_ARB_MON_CYC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CYC_CNT[31:16]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CYC_CNT[15:0]															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CYC_CNT Accumulated cycles when NLI_ARB_MON_START = 1