# MT6252 GSM/GPRS Baseband Processor Data Sheet

Version: 1.0

Release date: 2011-01-25

# Revision History

| Revision | Date | Comments |
|---|---|---|
| 1.0 | Jan 21, 2011 | Initial Version |



MediaTek Confidential Release for Cenon WCX

# TABLE OF CONTENTS

# Preface

## Acronym for Register Type

| | |
|---|---|
| **R/W** | Capable of both read and write access |
| **RO** | Read only |
| **RC** | Read only. After reading the register bank, each bit which is HIGH(1) will be cleared to LOW(0) automatically. |
| **WO** | Write only |
| **W1S** | Write only. When writing data bits to register bank, each bit which is HIGH(1) will cause the corresponding bit to be set to 1. Data bits which are LOW(0) have no effect on the corresponding bit. |
| **W1C** | Write only. When writing data bits to register bank, each bit which is HIGH(1) will cause the corresponding bit to be cleared to 0. Data bits which are LOW(0) have no effect on the corresponding bit. |

# 1 System Overview

MT6252 is a GSM/GPRS handset chip solution which integrates RF, analog baseband, digital baseband as well as Power Management Unit (PMU) and can greatly reduce the component count and make a smaller PCB size. Besides, MT6252 is capable of Single Antenna Interference Cancellation (SAIC) and AMR speech. Based on a 32-bit ARM7EJ-S™ RISC processor, MT6252 provides an unprecedented platform with high quality modem performance. The typical application diagram is shown as in **Figure 1**.



**Platform**

MT6252 has the ARM7EJ-S™ RISC processor running at up to 104 MHz, which provides best trade-off between system performance and power consumption.

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

**Memory**

MT6252 supports serial Flash interface with various operating frequencies.

**Multimedia**

MT6252 multimedia system contains many hardware accelerators to enrich user experience, including the camera interface, display controller/resizer/rotator, etc.

MT6252 utilizes high resolution audio DAC, digital audio, and audio synthesis technology to provide superior audio features; e.g. MP3 ring tones.

**Connectivity and Storage**

MT6252 supports UART as well as a Bluetooth interface. Also, it is embedded with necessary peripheral blocks: keypad scanner with the capability to detect the multiple key presses, multiple-SIM controller, real time clock, PWM, LCD controller, USB 1.1 FS/LS, MMC/SD and general purpose programmable I/Os.

MT6252 supports USB download with battery.

**Audio**

Using a highly integrated mixed-signal Audio Front-End architecture, MT6252 allows easy audio interfacing with direct connection to the audio transducers. The audio interface integrates D/A and A/D converters for the voice band, as well as high resolution stereo D/A converters for the audio band. In addition, MT6252 provides stereo input and analog mux, and supports the AMR codec to optimize speech and audio quality.

The 700 mW class-AB amplifier is also embedded to save the BOM cost without having to adopt an external amplifier.

**Radio**

MT6252 integrates a mixed-signal baseband front-end in order to provide a radio interface with flexibility for efficient customization. It builds in gain /offset calibration mechanisms and filters with programmable coefficients for comprehensive compatibility control on RF modules. MT6252 achieves great MODEM performance by utilizing a high resolution A/D converter in the RF downlink path.

MT6252 RF is a low current transceiver to support a true quad-band GSM/GPRS cellular system. The highly integrated RF system implements the high sensitivity and channel selection receiver, high precision transmission modulator, low phase noise frequency synthesizer, and the digitally controlled crystal oscillator. The external components required for a GPRS radio design include the Rx SAWs, PA, switchplexer, X'TAL, and a few passives.

**Debug Function**

The JTAG interface enables in-circuit debugging of software program with the ARM7EJ-S™ core. With this standardized debugging interface, MT6252 provides developers with a wide set of options in choosing ARM development kits from different third party vendors.

**Low Power Features**

MT6252 offers various low power features to reduce system power consumption. Features include Pause Mode of 32 kHz clocking at Standby State, Power Down Mode for individual peripherals, and Processor Sleep Mode. In addition, MT6252 is also fabricated using the advanced low leakage CMOS process, in order to provide an ultra low leakage solution.

**Power Management**

MT6252 integrates all regulators that a voice-centric phone needs, including thirteen LDOs optimized for specific GSM/GPRS baseband sub-systems. In addition to Li-Ion battery charge function, MT6252 are equipped with a SIM-level shifter interface, one open-drain output switch controlling the KPLED and one LDO specifically driving the vibrator motor. MT6252's power management schemes also cover thermal overload protection, Under-voltage Lock-out protection (UVLO), over-voltage protection, and the power-on reset/start-up timer, etc.

**Package**

The MT6252 device is offered in an 11.6 mm×12.1 mm, 305-ball, 0.5 mm staggered pin pitch TFBGA package.

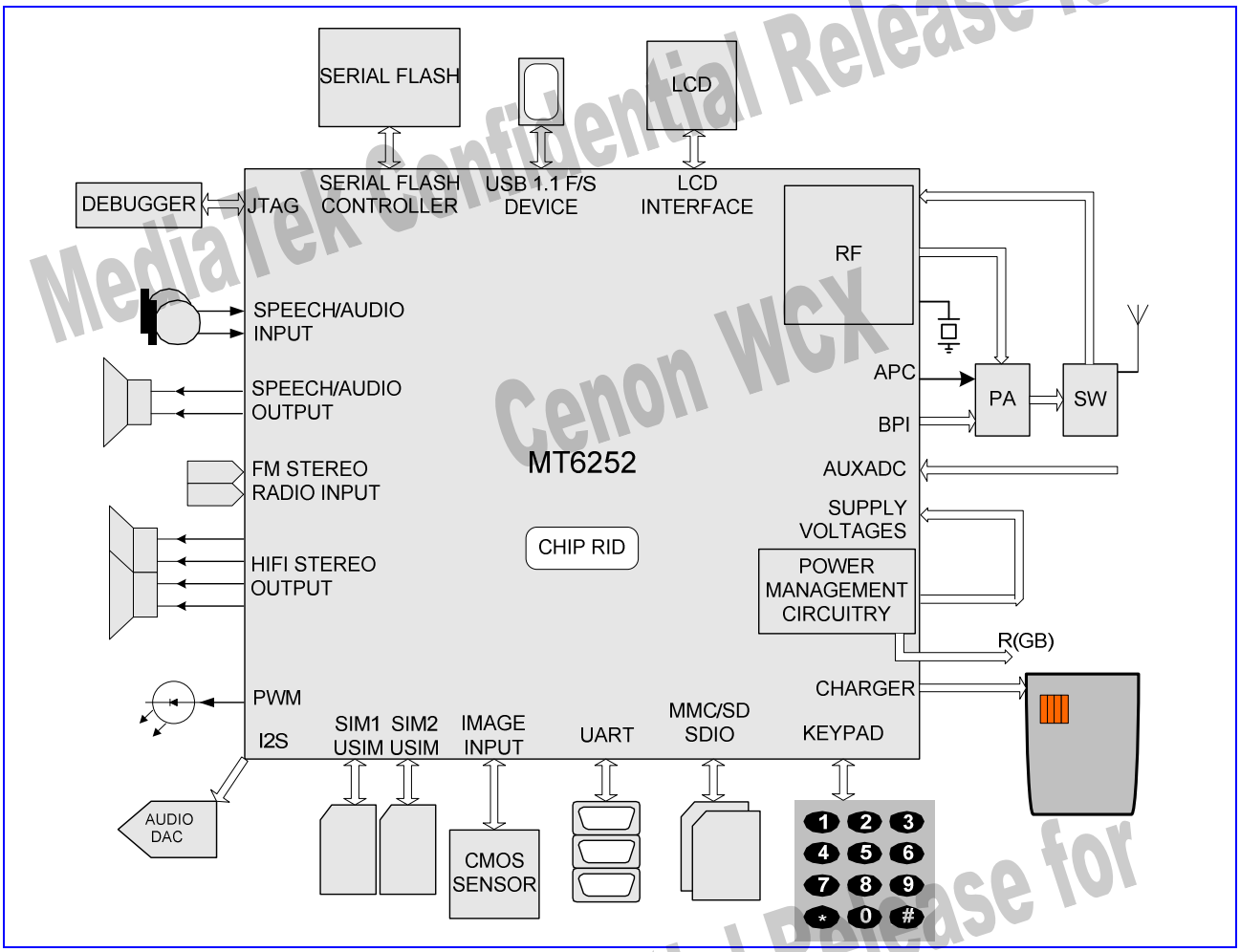



**Figure 1** Key applications of MT6252

## 1.1 Platform Features

**General**

- Integrated voice-band, audio-band and baseband analog front ends
- 11.6 mm×12.1 mm, 305-ball, 0.5 mm pitch TFBGA package

**MCU Subsystem**

- ARM7EJ-S 32-bit RISC processor
- High performance multi-layer AMBA bus
- Java hardware acceleration for fast Java-based games and applets
- Operating frequency: 104 MHz
- 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
- Frame check sequence coprocessor

**Serial Flash Interfaces**

- Supports various operating frequency combinations for Serial Flash
- Supports QPI and SPI Serial Flash

**User Interfaces**

- 8-row × 8-column keypad controller with a hardware scanner
- Supports multiple key presses for gaming
- SIM/USIM controller with hardware T=0/T=1 protocol control
- Real Time Clock (RTC) operating with a separate power supply
- General Purpose I/Os (GPIOs)
- 1 set of Pulse Width Modulation (PWM) output
- Maximum 7 external interrupt lines

**Security**

- Supports the security key and chip random ID

**Connectivity**

- Three sets of UARTs with hardware flow control and speed of up to 921600 bps
- FS/LS USB 1.1 device controller
- Multimedia Card/Secure Digital Memory Card
- DAI/PCM and I2S interface for audio applications

**Low Power Schemes**

- Power Down Mode for analog and digital circuits
- Processor Sleep Mode
- Pause Mode of 32 kHz clocking at Standby State
- 2-channel auxiliary 10-bit A/D converter for application usage other than battery monitoring

**Power and Supply Management**

- Charger Input up to 8 V
- Thirteen sets of LDOs optimized for specific GSM sub-systems
- Built-in LDO for RF transceiver
- High operation efficiency and low standby current
- Li-Ion battery charge function
- Multi- SIM Card Interface
- Four open-drain output current regulators to supply/control the LED
- LDO-type vibrator
- One NMOS switch to control R(GB) LED
- Thermal Overload Protection
- Under-voltage Lock-out Protection
- Over-voltage Protection
- Test and Debug
- Built-in digital and analog loop back modes for both audio and baseband front-ends
- DAI port complying with GSM Rec.11.10

## 1.2 Modem Features

**Integrated RF Receiver**

- Quad-band differential input LNAs
- Quadrature RF mixers
- Fully integrated channel filter

**Integrated RF Transmitter**

- Precise transmission modulator
- Low noise SAWLESS transmitter

**Integrated RF Frequency Synthesizer**

- Programmable fractional-N synthesizer
- Integrated VCO
- Integrated loop filter
- Fast settling time suitable for multi-slot GPRS applications

**Integrated Digitally controlled Crystal Oscillator (DCXO)**

- 26 MHz crystal oscillator
- On-chip programmable capacitor array for coarse tuning
- On-chip programmable capacitor array for fine-tuning

**Radio Interface and Baseband Front Ends**

- High resolution A/D converter for downlink baseband I and Q signals
- Calibration mechanism of offset and gain mismatch from analog baseband
- 10-bit D/A Converter for automatic power control
- Programmable radio Rx filter with adaptive bandwidth control
- Dedicated Rx filter for FB acquisition
- Baseband Parallel Interface (BPI) with programmable driving strength
- Multi-band support

**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
- GSM/GPRS 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 GEA1, GEA2 and GEA3 ciphering
- Programmable GSM/GPRS modem
- GSM Circuit Switch Data
- GPRS Class 12

**Voice Interface and Voice Front End**

- Two microphone inputs sharing one low noise amplifier with programmable gain and automatic gain control (AGC) mechanism
- Voice power amplifier with programmable gain
- A/D Converter for voice uplink path
- D/A Converter for voice downlink path
- Supports half-duplex hands-free operation
- Compliant with GSM 03.50

## 1.3 Multimedia Features

**LCD Interface**

- Supports an 8-/9-bit parallel interface
- Supports a serial interface

**LCD Controller**

- Supports LCD module with a maximum resolution up to 240x320 at 16 bpp
- Capable of combining display memories with up to 4 blending layers
- Supports hardware dithering function
- Supports hardware display rotation on each layer

**Camera Interface**

- Supports YUV422/RGB565 sensor interface

**Multimedia Data Path**

- Supports hardware source accumulation scaling function
- Supports hardware rotation function

**Audio Codec**

- 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 the I2S interface
- High resolution D/A converters for stereo audio playback
- Stereo analog input for stereo audio source
- Analog multiplexer for stereo audio
- FM radio recording
- Stereo-to-Mono conversion

## 1.4 General Description

Based on a dual-processor architecture, MT6252 integrates both an ARM7EJ-S core and digital signal processor cores. ARM7EJ-S is the main processor running the 2G and 2.5G protocol software. Digital signal processors are used to implement the MODEM algorithms as well as advanced audio functions. Except for some mixed-signal circuitries, the other building blocks in MT6252 are connected to either the microcontroller or one of the digital signal processors.

MT6252 consists of the following subsystems:

- Highly integrated RF transceiver for multi-band GSM and GPRS cellular systems
- Microcontroller Unit (MCU) Subsystem, including an ARM7EJ-S RISC processor and the accompanying memory management and interrupt handling logics
- Digital Signal Processor (DSP) Subsystem, including DSP cores and the accompanying memory, memory controller, and interrupt controller
- MCU/DSP Interface, where the MCU and the DSPs exchange hardware and software information
- Microcontroller Peripherals, including all user interface modules and RF control interface modules
- Microcontroller Coprocessors: Running computing-intensive processes in place of a microcontroller
- DSP Peripherals: Hardware accelerators for GSM/GPRS channel codec
- Multimedia Subsystem, including camera interface, display controller/resizer/rotator, etc.
- Voice Front End: The data path for converting analog speech from and to digital speech
- Audio Front End: The data path for converting stereo audio from stereo audio source
- Baseband Front End: The data path for converting digital signals of RF modules to analog signals or vice versa
- Timing Generator: Generating the control signals related to the TDMA frame timing
- Power, Reset and Clock subsystem: Managing the power, reset, and clock distribution inside MT6252
- LDOs, power-on sequences, switches and SIM-level shifters

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

# 2 Product Description

## 2.1 Pin Description

### 2.1.1 Ball Diagram

MT6252 is designed using 11.6 mm x 12.1 mm, 305-ball, 0.5 mm pitch TFBGA package. Pin-outs and the top view are illustrated in **Figure 2** for this package. Outline and dimensions of the package are illustrated in **Figure 19**, while the definition of the package is shown in **Table 33**.

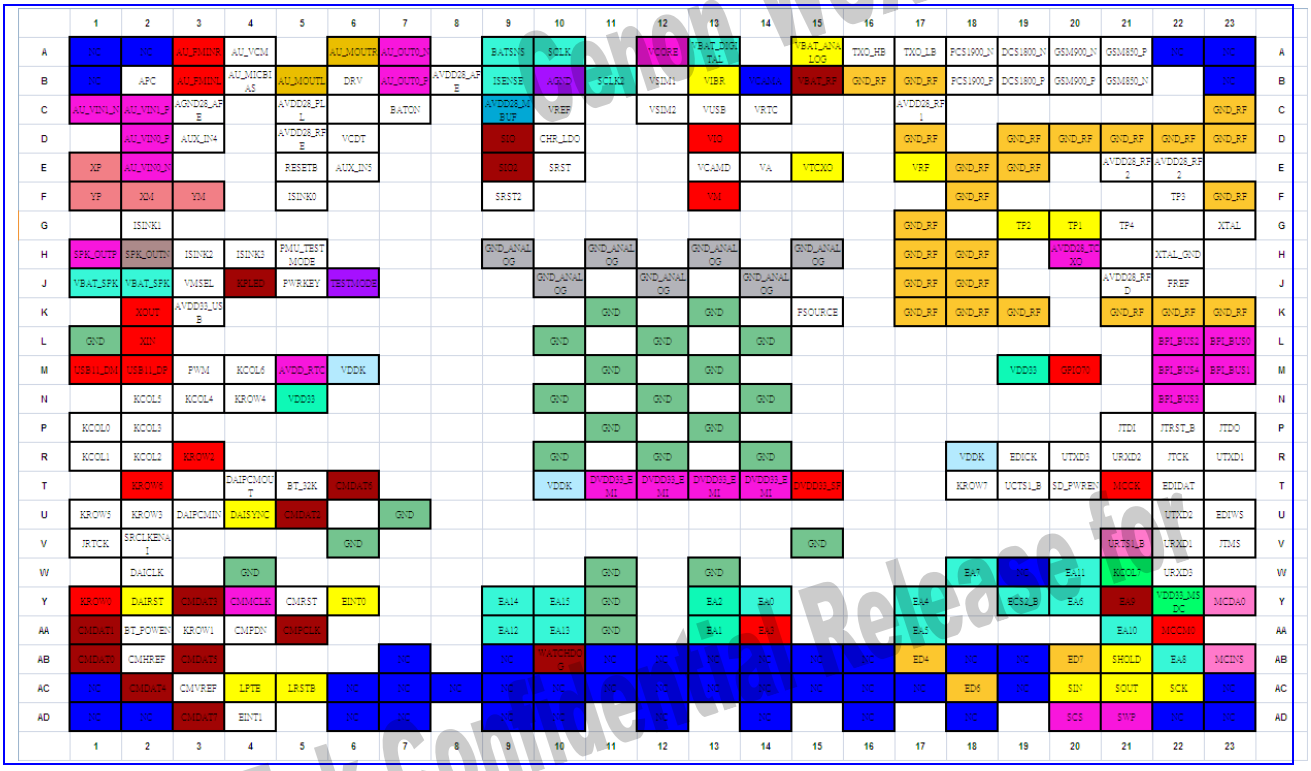


**Figure 2** Top view of MT6252 11.6 x 12.1 mm 0.5 mm pitch TFBGA package

### 2.1.2 Pin Coordination

The pin coordination is shown as in **Table 1**.

| | Pin Name | | Pin Name | | Pin Name |
|---|---|---|---|---|---|
| B10 | AGND | N12 | GND | AB7 | NC |
| C3 | AGND28_AFE | N14 | GND | AC7 | NC |
| B2 | APC | P11 | GND | AD7 | NC |
| B3 | AU_FMINL | P13 | GND | A2 | NC |
| A3 | AU_FMINR | R10 | GND | A1 | NC |
| B4 | AU_MICBIAS | R12 | GND | B1 | NC |
| B5 | AU_MOUTL | R14 | GND | AC1 | NC |
| A6 | AU_MOUTR | U7 | GND | AD1 | NC |
| A7 | AU_OUT0_N | V6 | GND | AD2 | NC |
| B7 | AU_OUT0_P | V15 | GND | W19 | NC |
| A4 | AU_VCM | W4 | GND | AC8 | NC |
| E2 | AU_VIN0_N | W11 | GND | AB9 | NC |
| D2 | AU_VIN0_P | W13 | GND | AC9 | NC |
| C1 | AU_VIN1_N | Y11 | GND | AD9 | NC |
| C2 | AU_VIN1_P | H9 | GND_ANALOG | AC10 | NC |
| D3 | AUX_IN4 | H11 | GND_ANALOG | AD10 | NC |
| E6 | AUX_IN5 | H13 | GND_ANALOG | AB11 | NC |
| M5 | AVDD_RTC | H15 | GND_ANALOG | AC11 | NC |
| B8 | AVDD28_AFE | J10 | GND_ANALOG | AB12 | NC |
| C9 | AVDD28_MBUF | J12 | GND_ANALOG | AC12 | NC |
| C5 | AVDD28_PLL | J14 | GND_ANALOG | AD12 | NC |
| C17 | AVDD28_RF1 | B16 | GND_RF | AB13 | NC |
| E21 | AVDD28_RF2 | B17 | GND_RF | AC13 | NC |
| E22 | AVDD28_RF2 | C23 | GND_RF | AB14 | NC |
| J21 | AVDD28_RFD | D17 | GND_RF | AC14 | NC |
| D5 | AVDD28_RFE | D19 | GND_RF | AD14 | NC |

| | | | | | |
|---|---|---|---|---|---|
| H20 | AVDD28_TCXO | D20 | GND_RF | A18 | PCS1900_N |
| K3 | AVDD33_USB | D21 | GND_RF | B18 | PCS1900_P |
| C7 | BATON | D22 | GND_RF | H5 | PMU_TESTMODE |
| A9 | BATSNS | D23 | GND_RF | M3 | PWM |
| L23 | BPI_BUS0 | E18 | GND_RF | J5 | PWRKEY |
| M23 | BPI_BUS1 | E19 | GND_RF | E5 | RESETB |
| L22 | BPI_BUS2 | F18 | GND_RF | AC22 | SCK |
| N22 | BPI_BUS3 | F23 | GND_RF | A10 | SCLK |
| M22 | BPI_BUS4 | G17 | GND_RF | B11 | SCLK2 |
| T5 | BT_32K | H17 | GND_RF | AD20 | SCS |
| AA2 | BT_POWEN | H18 | GND_RF | T20 | SD_PWREN |
| D10 | CHR_LDO | J17 | GND_RF | AB21 | SHOLD |
| AB1 | CMDAT0 | J18 | GND_RF | AC20 | SIN |
| AA1 | CMDAT1 | K17 | GND_RF | D9 | SIO |
| U5 | CMDAT2 | K18 | GND_RF | E9 | SIO2 |
| Y3 | CMDAT3 | K19 | GND_RF | AC21 | SOUT |
| AC2 | CMDAT4 | K21 | GND_RF | H2 | SPK_OUTN |
| AB3 | CMDAT5 | K22 | GND_RF | H1 | SPK_OUTP |
| T6 | CMDAT6 | K23 | GND_RF | V2 | SRCLKENAI |
| AD3 | CMDAT7 | M20 | GPIO70 | E10 | SRST |
| AB2 | CMHREF | B21 | GSM850_N | F9 | SRST2 |
| Y4 | CMMCLK | A21 | GSM850_P | AD21 | SWP |
| AA5 | CMPCLK | A20 | GSM900_N | J6 | TESTMODE |
| AA4 | CMPDN | B20 | GSM900_P | G20 | TP1 |
| Y5 | CMRST | B9 | ISENSE | G19 | TP2 |
| AC3 | CMVREF | F5 | ISINK0 | F22 | TP3 |
| W2 | DAICLK | G2 | ISINK1 | G21 | TP4 |

| | | | | | |
|---|---|---|---|---|---|
| U3 | DAIPCMIN | H3 | ISINK2 | A16 | TXO_HB |
| T4 | DAIPCMOUT | H4 | ISINK3 | A17 | TXO_LB |
| Y2 | DAIRST | V1 | JRTCK | T19 | UCTS1_B |
| U4 | DAISYNC | R22 | JTCK | V21 | URTS1_B |
| A19 | DCS1800_N | P21 | JTDI | V22 | URXD1 |
| B19 | DCS1800_P | P23 | JTDO | R21 | URXD2 |
| B6 | DRV | V23 | JTMS | W22 | URXD3 |
| T11 | DVDD33_EMI | P22 | JTRST_B | M1 | USB11_DM |
| T12 | DVDD33_EMI | P1 | KCOL0 | M2 | USB11_DP |
| T13 | DVDD33_EMI | R1 | KCOL1 | R23 | UTXD1 |
| T14 | DVDD33_EMI | R2 | KCOL2 | U22 | UTXD2 |
| T15 | DVDD33_SF | P2 | KCOL3 | R20 | UTXD3 |
| Y14 | EA0 | N3 | KCOL4 | E14 | VA |
| AA13 | EA1 | N2 | KCOL5 | A15 | VBAT_ANALOG |
| AA21 | EA10 | M4 | KCOL6 | A13 | VBAT_DIGITAL |
| W20 | EA11 | W21 | KCOL7 | B15 | VBAT_RF |
| AA9 | EA12 | J4 | KPLED | J1 | VBAT_SPK |
| AA10 | EA13 | Y1 | KROW0 | J2 | VBAT_SPK |
| Y9 | EA14 | AA3 | KROW1 | B14 | VCAMA |
| Y10 | EA15 | R3 | KROW2 | E13 | VCAMD |
| Y13 | EA2 | U2 | KROW3 | D6 | VCDT |
| AA14 | EA3 | N4 | KROW4 | A12 | VCORE |
| Y17 | EA4 | U1 | KROW5 | M19 | VDD33 |
| AA17 | EA5 | T2 | KROW6 | N5 | VDD33 |
| Y20 | EA6 | T18 | KROW7 | Y22 | VDD33_MSDC |
| W18 | EA7 | AC4 | LPTE | M6 | VDDK |

| | | | | | |
|---|---|---|---|---|---|
| AB22 | EA8 | AC5 | LRSTB | R18 | VDDK |
| Y21 | EA9 | T21 | MCCK | T10 | VDDK |
| Y19 | ECS2_B | AA22 | MCCM0 | B13 | VIBR |
| AB17 | ED4 | Y23 | MCDA0 | D13 | VIO |
| AC18 | ED6 | AB23 | MCINS | F13 | VM |
| AB20 | ED7 | AC6 | NC | J3 | VMSEL |
| R19 | EDICK | AD6 | NC | C10 | VREF |
| T22 | EDIDAT | AB15 | NC | E17 | VRF |
| U23 | EDIWS | AC15 | NC | C14 | VRTC |
| Y6 | EINT0 | AB16 | NC | B12 | VSIM1 |
| AD4 | EINT1 | AC16 | NC | C12 | VSIM2 |
| J22 | FREF | AD16 | NC | E15 | VTCXO |
| K15 | FSOURCE | AC17 | NC | C13 | VUSB |
| AA11 | GND | AB18 | NC | AB10 | WATCHDOG |
| K11 | GND | AB19 | NC | L2 | XIN |
| K13 | GND | AC19 | NC | F2 | XM |
| L1 | GND | AD18 | NC | K2 | XOUT |
| L10 | GND | AD22 | NC | E1 | XP |
| L12 | GND | AD23 | NC | G23 | XTAL |
| L14 | GND | AC23 | NC | H22 | XTAL_GND |
| M11 | GND | B23 | NC | F3 | YM |
| M13 | GND | A23 | NC | F1 | YP |
| N10 | GND | A22 | NC | | |

**Table 1** Pin coordination

## 2.1.3 Detailed Pin Description

| 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 2** Acronyms for pin types

| Pin Name | Type | Description | Power Domain |
|---|---|---|---|
| **System** | | | |
| TEST_MODE | DIO | Factory test mode enable input | VRTC |
| XIN | AI | 32.768 KHz crystal input | VRTC |
| XOUT | AO | 32.768 KHz crystal output | VRTC |
| SRCLKENAI | DIO | Security Enable | VDD33 |
| BT_POWEN | DIO | Bluetooth Power Enable | VDD33 |
| SD_PWREN | DIO | SD Power Enable | VDDK |
| BT_32K | DIO | Bluetooth 32KHz clk | VDD33 |
| PWM | DIO | Pulse-width modulated signal | VDD33 |
| **RF Control Circuitry** | | | |
| BPI_BUS0 | DIO | RF hard-wire control bus bit 0 | VDD33 |
| BPI_BUS1 | DIO | RF hard-wire control bus bit 1 | VDD33 |
| BPI_BUS2 | DIO | RF hard-wire control bus bit 2 | VDD33 |
| BPI_BUS3 | DIO | RF hard-wire control bus bit 3 | VDD33 |
| BPI_BUS4 | DIO | RF hard-wire control bus bit 4 | VDD33 |
| **UART Interface 1** | | | |
| URXD1 | DIO | UART 1 receive data | VDDK |
| UTXD1 | DIO | UART 1 transmit data | VDDK |
| URXD2 | DIO | UART 2 receive data | VDDK |

| Pin Name | Type | Description | Power Domain |
|---|---|---|---|
| UTXD2 | DIO | UART 2 transmit data | VDDK |
| URXD2 | DIO | UART 3 receive data | VDDK |
| UTXD2 | DIO | UART 3 transmit data | VDDK |
| URTS1_B | DIO | UART1 request to send ( active low ) | DVDD33_EMI |
| UCTS1_B | DIO | UART1 clear to send (active low ) | DVDD33_EMI |
| **Digital Audio Interface** | | | |
| DAICLK | DIO | DAI interface clock output | VDD33 |
| DAIPCMOUT | DIO | DAI PCM data output | VDD33 |
| DAIPCMIN | DIO | DAI PCM data input | VDD33 |
| DAISYNC | DIO | DAI reset signal input | VDD33 |
| DAIRST | DIO | DAI frame synchronization input | VDD33 |
| **JTAG Interface** | | | |
| JTMS | DI | JTAG test port mode switch | VDD33 |
| JTDI | DI | JTAG test port data input | VDD33 |
| JTCK | DI | JTAG test port clock input | VDD33 |
| JTRST_B | DI | JTAG test port reset input | VDD33 |
| JRTCK | DIO | JTAG test port returned clock output | VDD33 |
| JTDO | DIO | JTAG test port data output | VDD33 |
| **External Interrupt** | | | |
| EINT0 | DIO | External interrupt 0 | VDDK |
| EINT1 | DIO | External interrupt 1 | VDDK |
| **Keypad Interface** | | | |
| KCOL0 | DIO | Keypad column 0 | VDD33 |
| KCOL1 | DIO | Keypad column 1 | VDD33 |
| KCOL2 | DIO | Keypad column 2 | VDD33 |
| KCOL3 | DIO | Keypad column 3 | VDD33 |
| KCOL4 | DIO | Keypad column 4 | VDD33 |
| KCOL5 | DIO | Keypad column 5 | VDD33 |
| KCOL6 | DIO | Keypad column 6 | VDD33 |
| KCOL7 | DIO | Keypad column 7 | VDD33 |
| KROW0 | DIO | Keypad row 0 | VDD33 |
| KROW1 | DIO | Keypad row 1 | VDD33 |
| KROW2 | DIO | Keypad row 2 | VDD33 |

| Pin Name | Type | Description | Power Domain |
|---|---|---|---|
| KROW3 | DIO | Keypad row 3 | VDD33 |
| KROW4 | DIO | Keypad row 4 | VDD33 |
| KROW5 | DIO | Keypad row 5 | VDD33 |
| KROW6 | DIO | Keypad row 6 | VDD33 |
| KROW7 | DIO | Keypad row 7 | VDD33 |
| **Camera Interface** | | | |
| CMRST | DIO | CMOS sensor reset signal output | VDD33 |
| CMPDN | DIO | CMOS sensor power down control | VDD33 |
| CMVREF | DIO | CMOS sensor vertical reference signal input | VDD33 |
| CMHREF | DIO | CMOS sensor horizontal reference signal input | VDD33 |
| CMDAT0 | DIO | CMOS sensor data input 0 | VDD33 |
| CMDAT1 | DIO | CMOS sensor data input 1 | VDD33 |
| CMDAT2 | DIO | CMOS sensor data input 2 | VDD33 |
| CMDAT3 | DIO | CMOS sensor data input 3 | VDD33 |
| CMDAT4 | DIO | CMOS sensor data input 4 | VDD33 |
| CMDAT5 | DIO | CMOS sensor data input 5 | VDD33 |
| CMDAT6 | DIO | CMOS sensor data input 6 | VDD33 |
| CMDAT7 | DIO | CMOS sensor data input 7 | VDD33 |
| CMPCLK | DIO | CMOS sensor master clock output | VDD33 |
| CMMCLK | DIO | CMOS sensor master clock output | VDD33 |
| **External Memory Interface** | | | |
| ED4 | DIO | Reserved | DVDD33_EMI |
| ED6 | DIO | Reserved | DVDD33_EMI |
| ED7 | DIO | Reserved | DVDD33_EMI |
| EA0 | DIO | Reserved | DVDD33_EMI |
| EA1 | DIO | Reserved | DVDD33_EMI |
| EA2 | DIO | Reserved | DVDD33_EMI |
| EA3 | DIO | Reserved | DVDD33_EMI |
| EA4 | DIO | Reserved | DVDD33_EMI |
| EA5 | DIO | Reserved | DVDD33_EMI |
| EA6 | DIO | Reserved | DVDD33_EMI |
| EA7 | DIO | Reserved | DVDD33_EMI |
| EA8 | DIO | Reserved | DVDD33_EMI |
| EA9 | DIO | Reserved | DVDD33_EMI |

| Pin Name | Type | Description | Power Domain |
|---|---|---|---|
| EA10 | DIO | Reserved | DVDD33_EMI |
| EA11 | DIO | Reserved | DVDD33_EMI |
| EA12 | DIO | Reserved | DVDD33_EMI |
| EA13 | DIO | Reserved | DVDD33_EMI |
| EA14 | DIO | Reserved | DVDD33_EMI |
| EA15 | DIO | Reserved | DVDD33_EMI |
| ECS2_B | DIO | Reserved | DVDD33_EMI |
| WATCHDOG | DIO | Reserved | DVDD33_EMI |
| **MS/SD Card Interface** | | | |
| MCINS | DIO | SD Card Detect Input | DVDD33_EMI |
| MCDA0 | DIO | SD Serial Data IO 0/Memory Stick Serial Data IO | VDD33_MSDC |
| MCCK | DIO | SD Serial Clock/Memory Stick Serial Clock | VDD33_MSDC |
| MCCM0 | DIO | SD Command Output/Memory Stick Bus State Output | VDD33_MSDC |
| **USB Interface** | | | |
| DP | AIO | D+ data input/output | VUSB |
| DM | AIO | D- data input/output | VUSB |
| **LCD Interface** | | | |
| LPTE | DIO | Parallel display interface | DVDD33_EMI |
| LRSTB | DIO | Parallel display interface Reset Signal | DVDD33_EMI |
| **SIM Card Interface** | | | |
| SIO | DIO | SIM Data Input / Outputs | VSIM1 |
| SRST | DO | SIM card reset output | VSIM1 |
| SCLK | DO | SIM card clock output | VSIM1 |
| SIO2 | DIO | SIM 2 Data Input / Outputs | VSIM2 |
| SRST2 | DIO | SIM 2 card reset output | VSIM2 |
| SCLK2 | DIO | SIM 2 card clock output | VSIM2 |
| **Serial Flash Interface** | | | |
| SCS | DIO | Serial Flash Chip Select | DVDD33_SF |
| SIN | DIO | Serial Flash Data input / output | DVDD33_SF |
| SOUT | DIO | Serial Flash Data input / output | DVDD33_SF |
| SHOLD | DIO | Serial Flash Data input / output | DVDD33_SF |
| SWP | DIO | Serial Flash Data input / output | DVDD33_SF |
| SCK | DIO | Serial Flash Clock | DVDD33_SF |

| Pin Name | Type | Description | Power Domain |
|---|---|---|---|
| **PMIC Miscellaneous** | | | |
| PMU_TESTMODE | AIO | PMU test mode | |
| VBAT_RF | AIO | RF used battery voltage input | - |
| VBAT_Analog | AIO | Analog used battery voltage input | - |
| VBAT_Digital | AIO | Digital used battery voltage input | - |
| VREF | AIO | Reference voltage for PMIC | - |
| VBAT_BACKUP | AIO | Backup battery for RTC | - |
| BATSNS | AIO | Battery sense input | - |
| ISENSE | AIO | Current sense input | - |
| VMSEL | AIO | Memory supply voltage level select input | - |
| BATON | AIO | Battery insertion test | - |
| PWRKEY | AIO | Power key press input (low active) | - |
| RESETB | AIO | Power on reset (low active) | - |
| VCORE | AIO | Voltage output to digital core | - |
| SPK_OUTP | AIO | Speaker positive output | - |
| SPK_OUTN | AIO | Speaker negative output | - |
| **PMU LDO Outputs** | | | |
| VTCXO | AIO | Crystal or VCTCXO LDO output | |
| VRF | AIO | RF LDO output | |
| VCAMA | AIO | Camera module analog/io power | |
| VA | AIO | Analog LDO output | |
| VM | AIO | External memory LDO output | |
| VIO | AIO | Digital I/O voltage LDO output | |
| VCAMD | AIO | Camera module core/io power | |
| VUSB | AIO | USB power | |
| VSIM1 | AIO | LDO output to SIM card | |
| VSIM2 | AIO | LDO output to SIM 2 card | |
| **PMU Charger and LED Driving Interface** | | | |
| DRV | AIO | Charging driving | |
| CHRIN | AIO | Charger-In voltage detection | |
| CHR_LDO | AIO | Charger regulator | |
| VIBR | AIO | Vibrator driving output | |
| KP_LED | AIO | Keypad LED driver input | |
| ISINK1 | AIO | Current sink1 for B/L LED | |

| Pin Name | Type | Description | Power Domain |
|---|---|---|---|
| ISINK3 | AIO | Current sink3 for B/L LED | |
| ISINK2 | AIO | Current sink2 for B/L LED | |
| ISINK4 | AIO | Current sink4 for B/L LED | |
| **RF Interface** | | | |
| GSM850_N | AIO | Differential RF input ball for RX GSM850 band | AVDD28_RF1 |
| GSM850_P | AIO | Differential RF input ball for RX GSM850 band | AVDD28_RF1 |
| GSM900_N | AIO | Differential RF input ball for RX GSM900 band | AVDD28_RF1 |
| GSM900_P | AIO | Differential RF input ball for RX GSM900 band | AVDD28_RF1 |
| DCS1800_N | AIO | Differential RF input ball for RX DCS1800 band | AVDD28_RF1 |
| DCS1800_P | AIO | Differential RF input ball for RX DCS1800 band | AVDD28_RF1 |
| PCS1900_N | AIO | Differential RF input ball for RX PCS1900 band | AVDD28_RF1 |
| PCS1900_P | AIO | Differential RF input ball for RX PCS1900 band | AVDD28_RF1 |
| TXO_LB | AIO | RF output ball for TX LB (GSM900/GSM850) | AVDD28_RF1 |
| TXO_HB | AIO | RF output ball for TX HB (DCS/PCS) | AVDD28_RF1 |
| FREF | AIO | Monitor ball for DCXO output | AVDD28_TCXO |
| XTAL | AIO | Input ball for DCXO crystal | AVDD28_TCXO |
| XTAL_GND | AIO | Input ball for DCXO crystal ground | AVDD28_TCXO |
| TP1 | AIO | RF test ball 1 | AVDD28_TCXO |
| TP2 | AIO | RF test ball 2 | AVDD28_TCXO |
| TP3 | AIO | RF test ball 3 | AVDD28_TCXO |
| TP4 | AIO | RF test ball 4 | AVDD28_TCXO |
| **Analog Baseband Interface** | | | |
| AU_MOUTL | A | Audio analog output left channel | |
| AU_MOUTR | A | Audio analog output right channel | |
| AU_FMINR | A | FM radio analog input right channel | |
| AU_FMINL | A | FM radio analog input left channel | |
| AU_OUT0_N | A | Earphone 0 amplifier output (-) | |
| AU_OUT0_P | A | Earphone 0 amplifier output (+) | |
| AU_VIN1_P | A | Microphone 1 amplifier input (+) | |
| AU_VIN1_N | A | Microphone 1 amplifier input (-) | |
| AU_VIN0_N | A | Microphone 0 amplifier input (+) | |
| AU_VIN0_P | A | Microphone 0 amplifier input (-) | |
| AUX_IN4 | A | Auxiliary ADC input | |
| AUX_IN5 | A | Auxiliary ADC input | |

| Pin Name | Type | Description | Power Domain |
|---|---|---|---|
| APC | A | Automatic power control DAC output | |
| XP | A | Touch panel X-axis positive input | |
| XM | A | Touch panel X-axis negative input | |
| YP | A | Touch panel Y-axis positive input | |
| YM | A | Touch panel Y-axis negative input | |
| AU_VCM | A | Clean VCM for reference buffer | |
| AU_MICBIAS | A | Microphone bias source P | |
| **Digital Power** | | | |
| VDDK | P | Supply voltage of digital core | |
| VDD33 | P | Supply voltage of digital IO | |
| DVDD33_SF | P | Supply voltage of digital Serial Flash IO | |
| VDD33_MSDC | P | Supply voltage of digital SD/MS IO | |
| DVDD33_EMI | P | Supply voltage of external memory interface | |
| **RF Power and Ground** | | | |
| AVDD28_RF1 | P | AVDD 2.8V for RFSYS front end circuit block | |
| AVDD28_RF2 | P | AVDD 2.8V for RFSYS RFVCO | |
| AVDD28_TCXO | P | AVDD 2.8V for RFSYS DCXO, SX and RX IF | |
| AVDD28_RFD | P | AVDD 2.8V for RFSYS Digital circuit block | |
| GND_RF | G | RF Ground | |
| **Analog Power and Ground** | | | |
| AGND28_AFE | A | AFE AGND (clean reference ground) | - |
| VUSB | A | Supply voltage of USB analog | - |
| VRTC | A | Supply voltage of real time clock circuitry | - |
| VBAT_SPK | A | Supply voltage of Speaker | |
| GND_ANALOG | A | Supply ground of analog | - |
| AVDD28_PLL | A | Supply voltage of PLL | |
| AVDD28_RFE | A | Supply voltage of RFE | |
| AVDD28_MBUF | A | Audio buffer AVDD | |
| AVDD28_AFE | A | AFE AVDD | |
| AGND | A | Clean GND for ANALOG Circuitry | |
| GND | A | Ground for MT6252 | |

**Table 3** Pin function description and power domain

| Name | Reset | | | Output Drivability | Termination When Not Used | IO Type |
|---|---|---|---|---|---|---|
| | State | Aux | PU/PD | | | |
| **Analog IO** | | | | | | |
| APC | - | - | - | - | - | ANALOG |
| AU_FMINL | - | - | - | - | - | ANALOG |
| AU_FMINR | - | - | - | - | - | ANALOG |
| AU_MICBIAS | - | - | - | - | - | ANALOG |
| AU_MOUTL | - | - | - | - | - | ANALOG |
| AU_MOUTR | - | - | - | - | - | ANALOG |
| AU_OUT0_N | - | - | - | - | - | ANALOG |
| AU_OUT0_P | - | - | - | - | - | ANALOG |
| AU_VCM | - | - | - | - | - | ANALOG |
| AU_VIN0_N | - | - | - | - | - | ANALOG |
| AU_VIN0_P | - | - | - | - | - | ANALOG |
| AU_VIN1_N | - | - | - | - | - | ANALOG |
| AU_VIN1_P | - | - | - | - | - | ANALOG |
| AUX_IN4 | - | - | - | - | - | ANALOG |
| AUX_IN5 | - | - | - | - | - | ANALOG |
| BATON | - | - | - | - | - | ANALOG |
| DCS1800_N | - | - | - | - | - | ANALOG |
| DCS1800_P | - | - | - | - | - | ANALOG |
| DRV | - | - | - | - | - | ANALOG |
| FREF | - | - | - | - | - | ANALOG |
| FSOURCE | - | - | - | - | - | ANALOG |
| GSM850_N | - | - | - | - | - | ANALOG |
| GSM850_P | - | - | - | - | - | ANALOG |
| GSM900_N | - | - | - | - | - | ANALOG |
| GSM900_P | - | - | - | - | - | ANALOG |
| ISENSE | - | - | - | - | - | ANALOG |
| ISINK0 | - | - | - | - | - | ANALOG |
| ISINK1 | - | - | - | - | - | ANALOG |
| ISINK2 | - | - | - | - | - | ANALOG |
| ISINK3 | - | - | - | - | - | ANALOG |
| KPLED | - | - | - | - | - | ANALOG |
| PCS1900_N | - | - | - | - | - | ANALOG |
| PCS1900_P | - | - | - | - | - | ANALOG |

| Name | Reset | | | Output Drivability | Termination When Not Used | IO Type |
|---|---|---|---|---|---|---|
| | State | Aux | PU/PD | | | |
| PMU_TESTMODE | - | - | - | - | VSS | ANALOG |
| PWRKEY | - | - | - | - | - | ANALOG |
| RESETB | - | - | - | - | - | ANALOG |
| SCLK | - | - | - | - | - | ANALOG |
| SCLK2 | - | - | - | - | - | ANALOG |
| SIO | - | - | - | - | - | ANALOG |
| SIO2 | - | - | - | - | - | ANALOG |
| SPK_OUTN | - | - | - | - | - | ANALOG |
| SPK_OUTP | - | - | - | - | - | ANALOG |
| SRST | - | - | - | - | - | ANALOG |
| SRST2 | - | - | - | - | - | ANALOG |
| TP1 | - | - | - | - | VSS | ANALOG |
| TP2 | - | - | - | - | VSS | ANALOG |
| TP3 | - | - | - | - | VSS | ANALOG |
| TP4 | - | - | - | - | VSS | ANALOG |
| TXO_HB | - | - | - | - | - | ANALOG |
| TXO_LB | - | - | - | - | - | ANALOG |
| USB11_DM | - | - | - | - | - | ANALOG |
| USB11_DP | - | - | - | - | - | ANALOG |
| VCDT | - | - | - | - | - | ANALOG |
| VMSEL | - | - | - | - | - | ANALOG |
| VREF | - | - | - | - | - | ANALOG |
| XIN | - | - | - | - | - | ANALOG |
| XM | - | - | - | - | - | ANALOG |
| XOUT | - | - | - | - | - | ANALOG |
| XP | - | - | - | - | - | ANALOG |
| XTAL | - | - | - | - | - | ANALOG |
| YM | - | - | - | - | - | ANALOG |
| YP | - | - | - | - | - | ANALOG |
| **PWR/GND** | | | | | | |
| AVDD_RTC | - | - | - | - | - | PWR |
| AVDD28_AFE | - | - | - | - | - | PWR |
| AVDD28_MBUF | - | - | - | - | - | PWR |
| AVDD28_PLL | - | - | - | - | - | PWR |

| Name | Reset | | | Output Drivability | Termination When Not Used | IO Type |
|---|---|---|---|---|---|---|
| | State | Aux | PU/PD | | | |
| AVDD28_RF1 | - | - | - | - | - | PWR |
| AVDD28_RF2 | - | - | - | - | - | PWR |
| AVDD28_RFD | - | - | - | - | - | PWR |
| AVDD28_RFE | - | - | - | - | - | PWR |
| AVDD28_TCXO | - | - | - | - | - | PWR |
| AVDD33_USB | - | - | - | - | - | PWR |
| BATSNS | - | - | - | - | - | PWR |
| CHR_LDO | - | - | - | - | - | PWR |
| VA | - | - | - | - | - | PWR |
| VBAT_ANALOG | - | - | - | - | - | PWR |
| VBAT_DIGITAL | - | - | - | - | - | PWR |
| VBAT_RF | - | - | - | - | - | PWR |
| VBAT_SPK | - | - | - | - | - | PWR |
| VCAMA | - | - | - | - | - | PWR |
| VCAMD | - | - | - | - | - | PWR |
| VCORE | - | - | - | - | - | PWR |
| VDD33 | - | - | - | - | - | PWR |
| VDD33_MSDC | - | - | - | - | - | PWR |
| DVDD33_EMI | - | - | - | - | - | PWR |
| DVDD33_SF | - | - | - | - | - | PWR |
| VDDK | - | - | - | - | - | PWR |
| VIBR | - | - | - | - | - | PWR |
| VIO | - | - | - | - | - | PWR |
| VM | - | - | - | - | - | PWR |
| VRF | - | - | - | - | - | PWR |
| VSIM1 | - | - | - | - | - | PWR |
| VSIM2 | - | - | - | - | - | PWR |
| VTCXO | - | - | - | - | - | PWR |
| VUSB | - | - | - | - | - | PWR |
| VRTC | - | - | - | - | - | PWR |
| AGND28_AFE | - | - | - | - | - | GND |
| AGND | - | - | - | - | - | GND |
| GND | - | - | - | - | - | GND |
| GND_ANALOG | - | - | - | - | - | GND |

| Name | Reset | | | Output Drivability | Termination When Not Used | IO Type |
|---|---|---|---|---|---|---|
| | State | Aux | PU/PD | | | |
| GND_RF | - | - | - | - | - | GND |
| XTAL_GND | - | - | - | - | - | GND |
| **TESTMODE** | | | | | | |
| TESTMODE | I | X | PD | 12mA | No Need | IO Type 2 |
| **RF Control Circuitry** | | | | | | |
| BPI_BUS4 | LO | X | - | 2,4,6,8mA | No Need | IO Type 2 |
| BPI_BUS3 | LO | X | - | 2,4,6,8mA | No Need | IO Type 2 |
| BPI_BUS2 | LO | X | - | 2,4,6,8mA | No Need | IO Type 2 |
| BPI_BUS1 | LO | X | - | 2,4,6,8mA | No Need | IO Type 2 |
| BPI_BUS0 | LO | X | - | 2,4,6,8mA | No Need | IO Type 2 |
| **UART Interface 1** | | | | | | |
| URXD1 | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| UTXD1 | HO | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| UCTS1_B | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| URTS1_B | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| **UART Interface 2** | | | | | | |
| URXD2 | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| UTXD2 | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| **UART Interface 3** | | | | | | |
| URXD3 | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| UTXD3 | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| **Digital Audio Interface** | | | | | | |
| DAICLK | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| DAIPCMOUT | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| DAIPCMIN | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| DAISYNC | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| DAIRST | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| **JTAG Interface** | | | | | | |
| JTMS | I | X | PU | N/A | No Need | IO Type 3 |
| JTDI | I | X | PU | N/A | No Need | IO Type 3 |
| JTCK | I | X | PU | N/A | No Need | IO Type 3 |
| JTRST_B | I | X | PD | N/A | No Need | IO Type 3 |
| JTDO | XO | X | - | 2,4,6,8mA | No Need | IO Type 2 |
| JRTCK | LO | 4 | PU | 2,4,6,8mA | No Need | IO Type 2 |

| Name | Reset | | | Output Drivability | Termination When Not Used | IO Type |
|---|---|---|---|---|---|---|
| | State | Aux | PU/PD | | | |
| **PWM** | | | | | | |
| PWM | I | 2 | - | 4,8,12,16mA | VSS | IO Type 2 |
| **External Interrupt** | | | | | | |
| EINT0 | I | 1 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| EINT1 | I | 1 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| **Keypad Interface** | | | | | | |
| KCOL7 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KCOL6 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KCOL5 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KCOL4 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KCOL3 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KCOL2 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KCOL1 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KCOL0 | I | 0 | PU | 2,4,6,8mA | No Need | IO Type 2 |
| KROW7 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| KROW6 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| KROW5 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| KROW4 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| KROW3 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| KROW2 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| KROW1 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| KROW0 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| **System** | | | | | | |
| SRCLKENA | I | 1 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| **Camera Interface** | | | | | | |
| CMRST | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| CMPDN | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMVREF | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMHREF | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMDAT7 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMDAT6 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMDAT5 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMDAT4 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMDAT3 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |

| Name | Reset | | | Output Drivability | Termination When Not Used | IO Type |
|---|---|---|---|---|---|---|
| | State | Aux | PU/PD | | | |
| CMDAT2 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMDAT1 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMDAT0 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMPCLK | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| CMMCLK | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| **LCD Interface** | | | | | | |
| LPTE | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| LRSTB | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| **I2S** | | | | | | |
| EDICK | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| EDIDAT | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| EDIWS | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| **External Memory Interface** | | | | | | |
| EA15 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA14 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA13 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA12 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA11 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA10 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA9 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA8 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA7 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA6 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA5 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA4 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA3 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA2 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA1 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| EA0 | LO | 0 | | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| ECS2_B | HO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| ED7 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| ED6 | LO | 0 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| ED4 | LO | X | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| WATCHDOG | HO | X | PD | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |

| Name | Reset | | | Output Drivability | Termination When Not Used | IO Type |
|---|---|---|---|---|---|---|
| | State | Aux | PU/PD | | | |
| **Serial Flash Interface** | | | | | | |
| SCK | LO | 1 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| SWP | LO | 1 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| SHOLD | LO | 1 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| SCS | HO | 1 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| SIN | LO | 1 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| SOUT | LO | 1 | - | 2,4,6,8,10,12,14,16mA | No Need | IO Type 2 |
| **MS/SD Card Interface** | | | | | | |
| MCINS | I | 0 | PU | 4,8,12,16mA | No Need | IO Type 2 |
| MCCK | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| MCDA0 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| MCCM0 | I | 0 | PD | 4,8,12,16mA | No Need | IO Type 2 |
| SD_PWREN | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| **MISC/GPIO** | | | | | | |
| BT_POWEN | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| BT_32K | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |
| GPIO70 | I | 0 | PD | 2,4,6,8mA | No Need | IO Type 2 |

**Table 4** Default pin state drivability

| Abbreviation | Description |
|---|---|
| I | Input |
| LO | Low output |
| HO | High output |
| XO | Low or high output |
| PU | Pull-up |
| PD | Pull-down |
| - | No PU/PD |
| 0~N | Aux. function number |
| X | Delicate function pin |

**Table 5** State of Pins

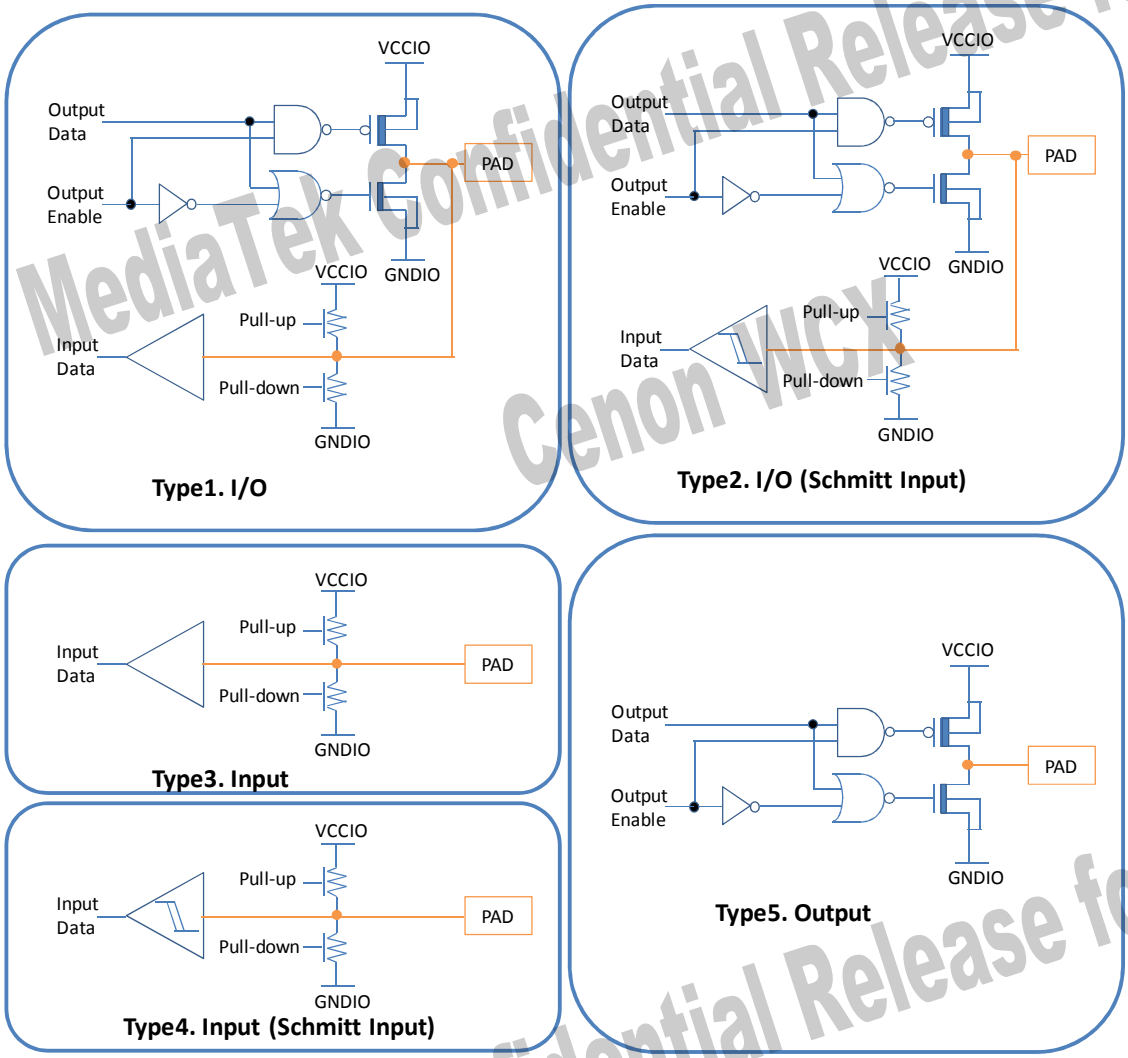



**Figure 3** IO Types in state of pins

## 2.1.4 Pin Multiplexing, Capability and Settings

| Abbreviation | Description |
|---|---|
| PU | Pull-up, not controllable |

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

**Table 6** Acronyms for pull-up/down typess

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| PWM | 0 | GPIO0 | IO | CU, CD | 4mA, 8mA, 12mA, 16mA | 0 |
| | 1 | PWM1 output | O | CU, CD | 4mA, 8mA, 12mA, 16mA | 0 |
| | 2 | CLKSQ_SEL | I | CU, CD | - | 0 |
| KCOL6 | 0 | GPIO1 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL6 | I | CU, CD | - | 0 |
| | 2 | EINT4 | I | CU, CD | - | 0 |
| KCOL5 | 0 | GPIO2 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL5 | I | CU, CD | - | 0 |
| | 3 | CAM_SDA | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KCOL4 | 0 | GPIO3 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL4 | I | CU, CD | - | 0 |
| | 3 | CAM_SCL | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KCOL3 | 0 | GPIO4 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL3 | I | CU, CD | - | 0 |
| | 2 | UART1 CTS | I | CU, CD | - | 0 |
| | 3 | BSI clock | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KCOL2 | 0 | GPIO5 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL2 | I | CU, CD | - | 0 |
| | 2 | UART1 CTS | I | CU, CD | - | 0 |
| | 3 | BSI clock | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KCOL1 | 0 | GPIO6 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL1 | I | CU, CD | - | 0 |
| | 3 | BSI clock | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 5 | EDICK | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KCOL0 | 0 | GPIO7 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL0 | I | CU, CD | - | 0 |
| KROW5 | 0 | GPIO8 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW5 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 2 | EINT5 | I | CU, CD | - | 0 |
| | 5 | EDIDAT | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KROW4 | 0 | GPIO9 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW4 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 3 | SRCLKENA | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 5 | EDIWS | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KROW3 | 0 | GPIO10 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW3 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 2 | URST1 RTS_B | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| | 4 | LSA0 | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KROW2 | 0 | GPIO11 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW2 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 4 | LSCK | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 5 | URTD3 | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KROW1 | 0 | GPIO12 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW1 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 4 | LSDA | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KROW0 | 0 | GPIO13 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW0 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 4 | LSDI | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| ECS2_B | 0 | GPIO14 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| | 1 | ECS2_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| | 2 | LPCE0_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| | 3 | LSCE0_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| | 6 | SEN2LCM_CS_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| DAICLK | 0 | GPIO15 | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | DAICLK | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 3 | EDICLK | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| DAIPCMOUT | 0 | GPIO16 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | DAIPCMOUT | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 3 | EDI_DATA | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| DAIPCMIN | 0 | GPIO17 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | DAIPCMIN | I | CU, CD | - | 0 |
| DAIRST | 0 | GPIO18 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | DAIRST | I | CU, CD | - | 0 |
| DAISYNC | 0 | GPIO19 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | DAISYNC | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 3 | EDI_WS | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| URXD3 | 0 | GPIO20 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | URXD3 | I | CU, CD | - | 0 |
| | 2 | UCTS2_B | I | CU, CD | - | 0 |
| UTXD3 | 0 | GPIO21 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | UTXD3 | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 2 | URTS2_B | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| URXD2 | 0 | GPIO22 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | URXD2 | I | CU, CD | - | 0 |
| | 2 | UCTS1_B | I | CU, CD | - | 0 |
| | 3 | CAM_SCL | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| UTXD2 | 0 | GPIO23 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | UTXD2 | O | CU, CD | - | 0 |
| | 2 | URTS1_B | O | CU, CD | - | 0 |
| | 3 | CAM_SDA | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| UCTS1_B | 0 | GPIO24 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | UCTS1_B | I | CU, CD | - | 0 |

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| | 2 | CAM_SCL | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 3 | NLD13 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 5 | EINT5 | I | CU, CD | - | 0 |
| URTS1_B | 0 | GPIO25 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | UCTS1_B | O | CU, CD | - | 0 |
| | 2 | CAM_SDA | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 3 | NLD14 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 5 | EINT6 | I | CU, CD | - | 0 |
| EINT0 | 0 | GPIO26 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | EINT0 | I | CU, CD | - | 0 |
| EINT1 | 0 | GPIO27 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | EINT1 | I | CU, CD | - | 0 |
| BPI_BUS4 | 0 | GPIO28 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | BPI_BUS4 | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| PAD_IO_SIM2 | 0 | GPIO29 | IO | X | 4mA | 0 |
| | 1 | SIO2 | IO | X | 4mA | 0 |
| PAD_CLK_SIM2 | 0 | GPIO30 | IO | X | 4mA | 0 |
| | 1 | SCLK2 | O | X | 4mA | 0 |
| PAD_RST_SIM2 | 0 | GPIO31 | IO | X | 4mA | 0 |
| | 1 | SRST2 | O | X | 4mA | 0 |
| BT_POWEN | 0 | GPIO32 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| SD_PWREN | 0 | GPIO33 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | CLK32K | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| BT_32K | 0 | GPIO34 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 2 | SRCLKENA | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| SRCLKENAI | 0 | GPIO35 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | SRCLKENAI | I | CU, CD | - | 0 |
| MCINS | 0 | GPIO36 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | MCINS | I | CU, CD | - | 0 |
| | 3 | NLD15 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 5 | EINT6 | I | CU, CD | - | 0 |
| MCCK | 0 | GPIO37 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | MCCK | I | CU, CD | - | 0 |
| MCDA0 | 0 | GPIO38 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | MCDA0 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| MCCM0 | 0 | GPIO39 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | MCCM0 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| LPTE | 0 | GPIO40 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | LPTE | I | CU, CD | - | 0 |
| LRSTB | 0 | GPIO41 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | LRSTB | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| ECS3_B | 0 | GPIO42 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| | 2 | LPCE1_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| | 3 | LSCE1_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0 |
| | 6 | SEN2LCM_CS_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, | 0 |

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| | | | | | 12mA , 14mA, 16mA, 18mA | |
| | 7 | EINT2 | I | CU, CD | - | 0 |
| KCOL7 | 0 | GPIO43 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KCOL7 | I | CU, CD | - | 0 |
| | 2 | EINT2 | I | CU, CD | - | 0 |
| | 4 | LSCK | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KROW6 | 0 | GPIO44 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW6 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 4 | LSDA | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| KROW7 | 0 | GPIO45 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 1 | KROW6 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 2 | EINT3 | I | CU, CD | - | 0 |
| | 4 | CAM_SDA | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| CMDAT0 | 0 | GPIO47 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT0 | I | CU, CD | - | 0 |
| | 2 | CAM_CSD | I | CU, CD | - | 0 |
| CMDAT1 | 0 | GPIO48 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT1 | I | CU, CD | - | 0 |
| | 2 | LSDA | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| CMDAT2 | 0 | GPIO49 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT2 | I | CU, CD | - | 0 |
| CMDAT3 | 0 | GPIO50 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT3 | I | CU, CD | - | 0 |
| CMDAT4 | 0 | GPIO51 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT4 | I | CU, CD | - | 0 |
| CMDAT5 | 0 | GPIO52 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT5 | I | CU, CD | - | 0 |
| CMDAT6 | 0 | GPIO53 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT6 | I | CU, CD | - | 0 |
| CMDAT7 | 0 | GPIO54 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMDAT7 | I | CU, CD | - | 0 |
| CMHREF | 0 | GPIO55 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMHREF | I | CU, CD | - | 0 |
| CMVREF | 0 | GPIO56 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMVREF | I | CU, CD | - | 0 |
| CMPDN | 0 | GPIO57 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMPDN | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 2 | LSCK | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| CMMCLK | 0 | GPIO58 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMMCLK | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| CMPCLK | 0 | GPIO59 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMPCLK | I | CU, CD | - | 0 |
| | 2 | CAM_CSK | I | CU, CD | - | 0 |
| CMRST | 0 | GPIO60 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | CMRST | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| EDICK | 0 | GPIO61 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | EDICK | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 3 | BPI_BUS5 | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| EDIDAT | 0 | GPIO62 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| | 1 | EDIDAT | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 3 | BPI_BUS6 | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| EDI_WS | 0 | GPIO63 | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 1 | EDI_WS | IO | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| | 3 | BPI_BUS6 | O | CU, CD | 4mA,8mA,12mA,16mA | 0 |
| SCK | 0 | GPIO64 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | SCK | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSCK | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| SWP | 0 | GPIO65 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | SWP | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSCK | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| SHOLD | 0 | GPIO66 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | SHOLD | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSCE0_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| SCS | 0 | GPIO67 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | SCS | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSCE1_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| SIN | 0 | GPIO68 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | SIN | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSDI | I | CU, CD | - | 0/1 |
| | 3 | EINT3 | I | CU, CD | - | 0/1 |
| SOUT | 0 | GPIO69 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | SOUT | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSDA | I | CU, CD | - | 0/1 |
| | 3 | EINT3 | I | CU, CD | - | 0/1 |
| GPIO70 | 0 | GPIO70 | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 2 | EINT4 | I | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 3 | CLK32K | O | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| | 4 | CAM_SCL | IO | CU, CD | 2mA,4mA,6mA,8mA | 0 |
| EA0 | 0 | EA0 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | LPA0 | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | SEN2LCM_A0 | O | CU, CD | 2mA, 6mA, 8mA, 10mA, | 0/1 |

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| | | | | | 12mA , 14mA, 16mA, 18mA | |
| EA1 | 0 | EA0 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | LRD_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| EA2 | 0 | EA0 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | LWR_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | SEN2LCM_WR_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| EA3 | 0 | EA3 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD0 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSCK | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CAMDAT0 | I | CU, CD | - | 0/1 |
| EA4 | 0 | EA4 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD1 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSA0 | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CAMDAT1 | I | CU, CD | - | 0/1 |
| EA5 | 0 | EA5 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD2 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSDA | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CAMDAT2 | I | CU, CD | - | 0/1 |
| EA6 | 0 | EA6 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD3 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | LSDI | I | CU, CD | - | 0/1 |
| | 3 | CAMDAT3 | I | CU, CD | - | 0/1 |
| EA7 | 0 | EA7 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD4 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CMDAT4 | I | CU, CD | - | 0/1 |
| EA8 | 0 | EA8 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD5 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CMDAT5 | I | CU, CD | - | 0/1 |
| EA9 | 0 | EA9 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD6 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, | 0/1 |

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| | | | | | 12mA , 14mA, 16mA, 18mA | |
| | 3 | CMDAT6 | I | CU, CD | - | 0/1 |
| EA10 | 0 | EA10 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD7 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CMDAT7 | I | CU, CD | - | 0/1 |
| EA11 | 0 | EA11 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD8 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| EA12 | 0 | EA12 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD9 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | CMMCLK | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CMMCLK | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| EA13 | 0 | EA13 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD10 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | CAM_CSK | I | CU, CD | - | 0/1 |
| | 3 | CMPCLK | I | CU, CD | - | 0/1 |
| EA14 | 0 | EA14 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD11 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | CAM_CSD | I | CU, CD | - | 0/1 |
| | 3 | CAMHSYNC | I | CU, CD | - | 0/1 |
| EA15 | 0 | EA15 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 1 | NLD12 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | CMPDN | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | CAMVSYNC | I | CU, CD | - | 0/1 |
| ED6 | 0 | ED6 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 2 | CMRST | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | LSCE1_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 4 | SEN2LCM_CS_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 5 | CAM_SDA | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 6 | LPCE1_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| ED7 | 0 | ED7 | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, | 0/1 |

| Name | Aux. Function | Aux. Name | Aux. Type | PU/PD/ CU/CD | Driving | SMT |
|---|---|---|---|---|---|---|
| | | | | | 12mA , 14mA, 16mA, 18mA | |
| | 2 | CMPDN | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 3 | LPCE1_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 4 | SEN2LCM_CS_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 5 | CAM_SCL | IO | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| | 6 | LSCE1_B | O | CU, CD | 2mA, 6mA, 8mA, 10mA, 12mA , 14mA, 16mA, 18mA | 0/1 |
| URXD1 | 0 | URXD1 | I | PU | - | 0 |
| | 1 | EINT2 | I | PU | - | 0 |
| | 2 | LSCK | O | PU | 4mA,8mA,12mA,16mA | 0 |
| UTXD1 | 0 | UTXD1 | O | PU | 4mA,8mA,12mA,16mA | 0 |
| | 1 | EINT3 | I | PU | 4mA,8mA,12mA,16mA | 0 |
| | 2 | LSDA | IO | PU/CD | 4mA,8mA,12mA,16mA | 0 |

**Table 7** Pin multiplexing, capability and settings

## 2.2 Electrical Characteristics

### 2.2.1 Absolute Maximum Ratings

Prolonged exposure to absolute maximum ratings may reduce device reliability. Functional operation at these maximum ratings is not implied.

| Pin Name | Description | Min | Max | Unit |
|---|---|---|---|---|
| VBAT_DIGITAL, VBAT_ANALOG, VBAT_SPK, VBAT_RF | Battery regulator supply voltage | -0.3 | +4.73 | V |
| VDD33 | Regulated digital I/O supply voltage | -0.3 | +3.63 | V |
| VUSB | Regulated USB supply voltage | -0.3 | +3.63 | V |
| DVDD33_EMI | Regulated memory supply voltage | -0.3 | +3.63 | V |
| DVDD33_SF | Regulated serial flash supply voltage | -0.3 | +3.63 | V |

| | | | | |
|---|---|---|---|---|
| VDD33_MSDC | Regulated memory card supply voltage | -0.3 | +3.63 | V |
| VDDK | Regulated digital circuit supply voltage | -0.3 | +1.65 | V |
| AVDD28_MBUF, AVDD28_RFE, AVDD28_AFE, AVDD28_PLL, AVDD_RTC | Regulated analog circuit supply voltage | -0.3 | +3.63 | V |
| AVDD28_RF1, AVDD28_RF2, AVDD28_TCXO, AVDD28_RFD | Regulated RF circuit supply voltage | -0.3 | +3.63 | V |

**Warning: Stressing the device beyond the Absolute Maximum Ratings may cause permanent damage. These are stress ratings only.**

| Symbol | Description | Min | Max | Unit |
|---|---|---|---|---|
| Tstg | Storage temperature | -55 | +125 | Celsius |

## 2.2.2 Recommended Operating Conditions

| Pin Name | Description | Min | Max | Unit |
|---|---|---|---|---|
| VBAT_DIGITAL, VBAT_ANALOG, VBAT_SPK, VBAT_RF | Battery regulator supply voltage | | +4.3 | V |
| VDD33 | Regulated digital I/O supply voltage | +2.7 | +2.9 | V |
| VUSB | Regulated USB supply voltage | +2.97 | +3.63 | V |
| DVDD33_EMI | Regulated supply voltage for memory | +1.7 | +1.95 | V |
| DVDD33_SF | Regulated supply voltage for serial flash | +1.7(+1.65?) | +3.6 | V |
| VDD33_MSDC | Regulated supply voltage for SD card | +2.7 | +3.6 | V |
| VDDK | Regulated digital circuit supply voltage | +1.08 | +1.32 | V |
| AVDD28_MBUF, AVDD28_RFE, AVDD28_AFE, AVDD28_PLL, AVDD_RTC | Regulated analog supply voltage | +2.7 | +2.9 | V |

| | | | | |
|---|---|---|---|---|
| AVDD28_RF1, AVDD28_RF2, AVDD28_TCXO, AVDD28_RFD | Regulated RF circuit supply voltage | +2.7 | +2.9 | V |

**Warning: Operation beyond the Operating Conditions is not recommended. Extended exposure may affect device reliability.**

| Symbol | Description | Min | Max | Unit |
|---|---|---|---|---|
| Topr | Operating temperature | -20 | +80 | Celsius |



## 2.2.3 Serial Flash Interface

| Symbol | Description | | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| fSCLK | Clock frequency for supporting serial flash | | | | 104 | MHz |
| tSLCH | /CS active setup time (relative to SCLK) | | | 5 | | ns |
| tSHSL | /CS deselect time | | | 10 | | ns |
| tCHSL | /CS not active hold time (relative to SCLK) | | | 5 | | ns |
| tDVCH | Data in setup time | | | 3 | | ns |
| tCHSH | /CS active hold time (relative to SCLK) | | | 5 | | ns |
| tSHCH | /CS not active setup time (relative to SCLK) | | | 5 | | ns |
| tCLQV | Clock Low to Output Valid Loading: 30pF / 15pF | Loading :30pF | | | 8 | ns |
| | | Loading :15pF | | | 6 | ns |
| tCH | Clock high time | | | 4.5 | | ns |
| tCL | Clock low time | | | 4.5 | | ns |

**Table 8** Serial Flash AC Characteristics

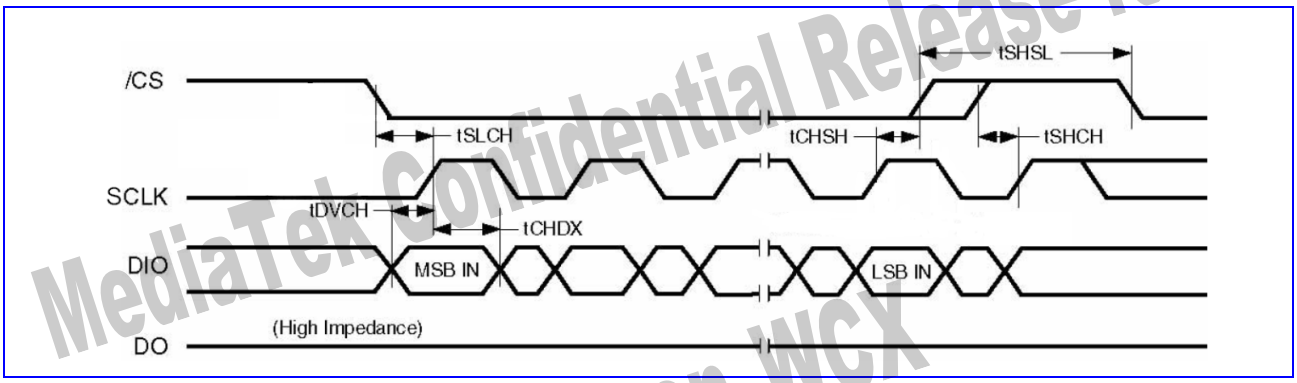



**Figure 4** Serial Flash write timing

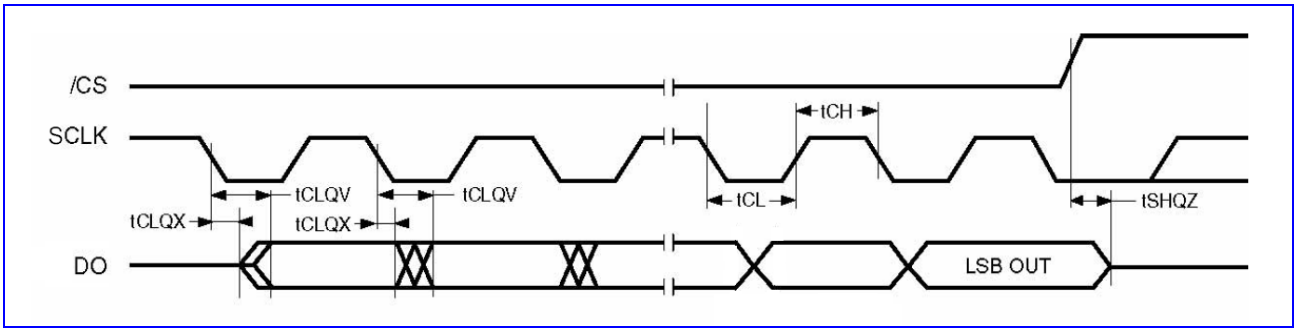



**Figure 5** Serial Flash read timing

## 2.3 System Configurations

### 2.3.1 Strapping Resistors

| Pin Name | Description | Trapping Condition |
|---|---|---|
| DIASYNC | Pull-up with 100K resistor | Reset |
| PWM | Pull-down with 100K resistor | (GPIO_MODE0==2) |
| DAIRST | Pull-up with 100K resistor | Reset |

**Table 9** Strapping table

### 2.3.2 Constant Tied Pins

| Pin Name | Tie Value |
|---|---|
| TESTMODE | GND |
| PMU_TESTMODE | GND |

## 2.4 Power-on Sequence

The power-on/off sequence which is controlled by “Control” and “Reset Generator” is shown as follows,

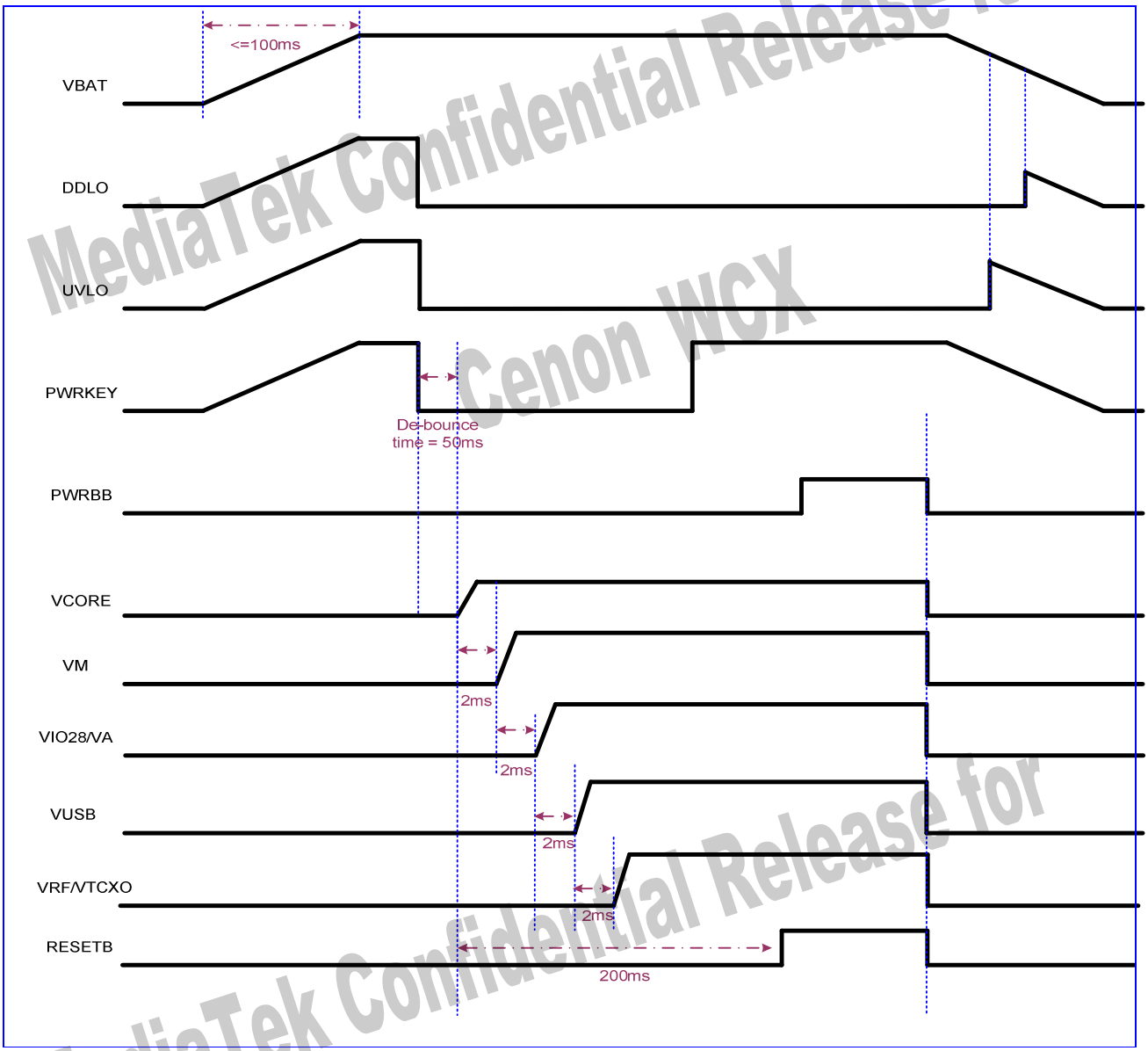



**Figure 6** Power-on/off Control Sequence

Note that the above figure only shows one power-on/off condition. The MT6252 handles the powering ON and OFF of the handset. Use the following three ways to switch on the handset (When VBAT ≥ 3.2 V):

- Pulling PWRKEY low (User push PWRKEY)
- Pulling PWRBB high (Baseband BB_WakeUp)
- Valid charger plug-in

Pulling PWRKEY low is a normal way to turn on the handset. That will turn on VCORE, VIO, VM, VA and VUSB as long as the PWRKEY is kept low. The VRF and VTCXO are turned on when SRCLKEN is high. The microprocessor then starts and pulls PWRBB high. After that, PWRKEY can be released. Pulling PWRBB high will also turn on the handset. This is the case when the alarm in the RTC expires.

In addition, applying a valid external supply on CHRIN will also turn on the handset. However, If the battery is in UV state (VBAT < 3.2 V), the handset cannot be turned on.

The UVLO function of MT6252 prevents system startup when the initial voltage of the main battery is below the 3.2 V threshold. When the battery voltage is greater than 3.2 V, the UVLO comparator switches and the threshold is reduced to 2.9 V. This allows the handset to start smoothly unless the battery decays to 2.9 V and below.

Once the MT6252 enters the UVLO state, it draws a very low quiescent current. The VRTC LDO is still active until the DDLO disables it.

# 2.5 Analog Baseband

## 2.5.1 Audio Mixed-signal Blocks

### 2.5.1.1 Block Descriptions

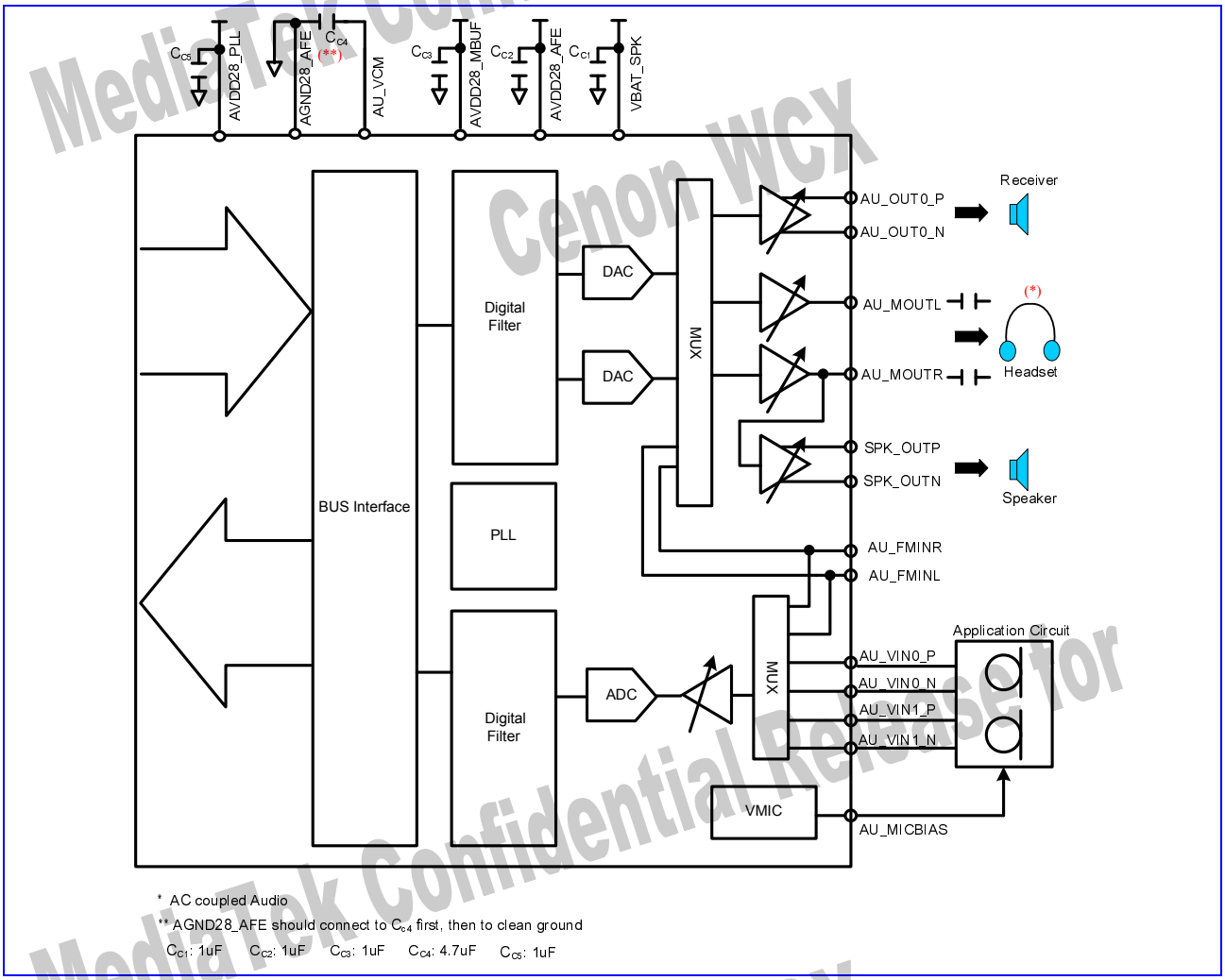


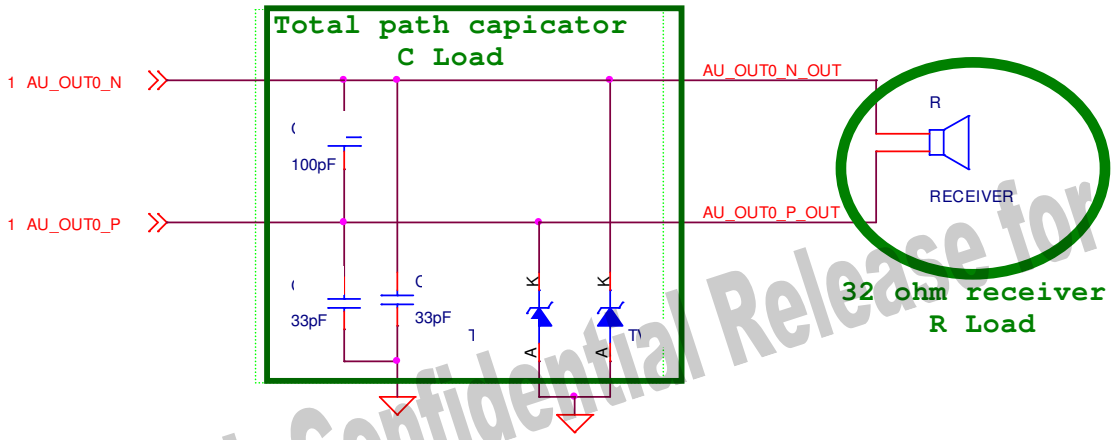

**Figure 7** Block diagram of audio/speech part

### 2.5.1.2 Functional Specification

The following table provides functional specifications of voice-band uplink/downlink blocks.

| Symbol | Description | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| AVDD | Power Supply (AVDD28_MBUF, AVDD28_AFE, AVDD28_PLL) | 2.7 | 2.8 | 2.9 | V |
| VMIC | Microphone Biasing Voltage | | 1.9 | 2.2 | V |
| Uplink Path | | | | | |
| SINAD | Signal to Noise and Distortion Ratio<br>Input Level: -40 dbm0<br>Input Level: 0 dbm0 | <br>29 | <br>69 | | <br>dB<br>dB |
| RIN | Input Impedance (Differential) | 13 | 20 | 27 | KΩ |
| Downlink Path | | | | | |
| SINAD | Signal to Noise and Distortion Ratio<br>Input Level: -40 dBm0<br>Input Level: 0 dBm0 | 29 | <br>69 | | <br>dB<br>dB |
| RLOAD | Output Resistor Load (Differential) | | 32 | | Ω |
| CLOAD | Output Capacitor Load | | | 200 | pF |

**Table 10** Functional specifications of analog voice blocks




Functional specifications of the audio blocks are described in the following.

| Symbol | Description | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Fs | Sampling Rate | 32 | 44.1 | 48 | KHz |
| AVDD | Power Supply (AVDD28_MBUF, AVDD28_AFE, | 2.6 | 2.8 | 3.0 | V |

| | | | | | |
|---|---|---|---|---|---|
| | AVDD28_PLL) | | | | |
| PSNR | Peak Signal to Noise Ratio | | 88 | | dB |
| VOUT | Output Swing for 0dBFS Input Level | | 0.64 | | Vrms |
| THD | Total Harmonic Distortion<br>22mW at 32 Ω Load | | -70 | | dB |
| RLOAD | Output Resistor Load (Single-ended) | | 32 | | Ω |
| CLOAD | Output Capacitor Load | | | 200 | pF |

**Table 11** Functional specifications of the analog audio blocks

## 2.5.2 BBRX

### 2.5.2.1 Functional Specification

The receiver (RX) performs base-band I/Q channels downlink analog-to-digital conversion:

1. *Analog input multiplexer:* For each channel, a 4-input multiplexer that supports offset and gain calibration is included.
2. *A/D converter:* Two 14-bit sigma-delta ADCs perform I/Q digitization for further digital signal processing.

| Symbol | Description | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| N | Resolution | | 14 | | Bit |
| FC | Clock Rate | | 26 | | MHz |
| FS | Output Sampling Rate | | 13/12 | | MSPS |
| SINAD | Signal to Noise and Distortion Ratio<br>- 45kHz sine wave in [0:90] kHz bandwidth<br>- 145kHz sine wave in [10:190] kHz bandwidth | <br>65<br>65 | | | <br>dB<br>dB |
| ICN | Idle channel noise<br>- [0:90] kHz bandwidth<br>- [10:190] kHz bandwidth | | | <br>-74<br>-70 | <br>dB<br>dB |
| DR | Dynamic Range<br>- [0:90] kHz bandwidth<br>- [10:190] kHz bandwidth | <br>74<br>70 | | | <br>dB<br>dB |
| RIN | Input Resistance | 75 | | | kΩ |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply (AVDD_RFE) | 2.7 | 2.8 | 2.9 | V |
| T | Operating Temperature | -20 | | 80 | ℃ |
| | Current Consumption<br>Power-up<br>Power-down | | <br>5<br>5 | | <br>mA<br>μA |

**Table 12** Base-band Downlink Specifications

## 2.5.3 Auxiliary ADC / Touch Screen Controller

### 2.5.3.1 Block Descriptions

The auxiliary ADC includes the following functional blocks:

1. *Analog Multiplexer:* The analog multiplexer selects signal from one of the several auxiliary input pins. Real world messages to be monitored, such as the temperature, should be transferred to the voltage domain.
2. *10-bit A/D Converter:* The ADC converts the multiplexed input signal to 10-bit digital data.

Touch Screen Interface can control input pads (XP, XM, YP, and YM) to obtain X/Y-position on the external touch screen device. Touch Screen Interface contains three main blocks; which are touch screen pads control logic, ADC interface logic and interrupt generation logic.

| AuxADC Channel ID | Description |
|---|---|
| Channel 0 | Internal use |
| Channel 1 | Internal use |
| Channel 2 | Internal use |
| Channel 3 | Internal use |
| Channel 4 | External (AUX_IN4) |
| Channel 5 | External (AUX_IN5) |
| Channel 6 | Internal use |
| Channel 7 | Internal use |
| Channel 8 | Internal use |
| Channel 9 | Internal use |
| Channel 10 | Internal use |
| Channel 11 | Internal use |
| Channel 12 | Internal use |
| Channel 13 | Internal use |
| Channel 14 | Internal use |
| Channel 15 | Internal use |

**Table 13** AuxADC channel list

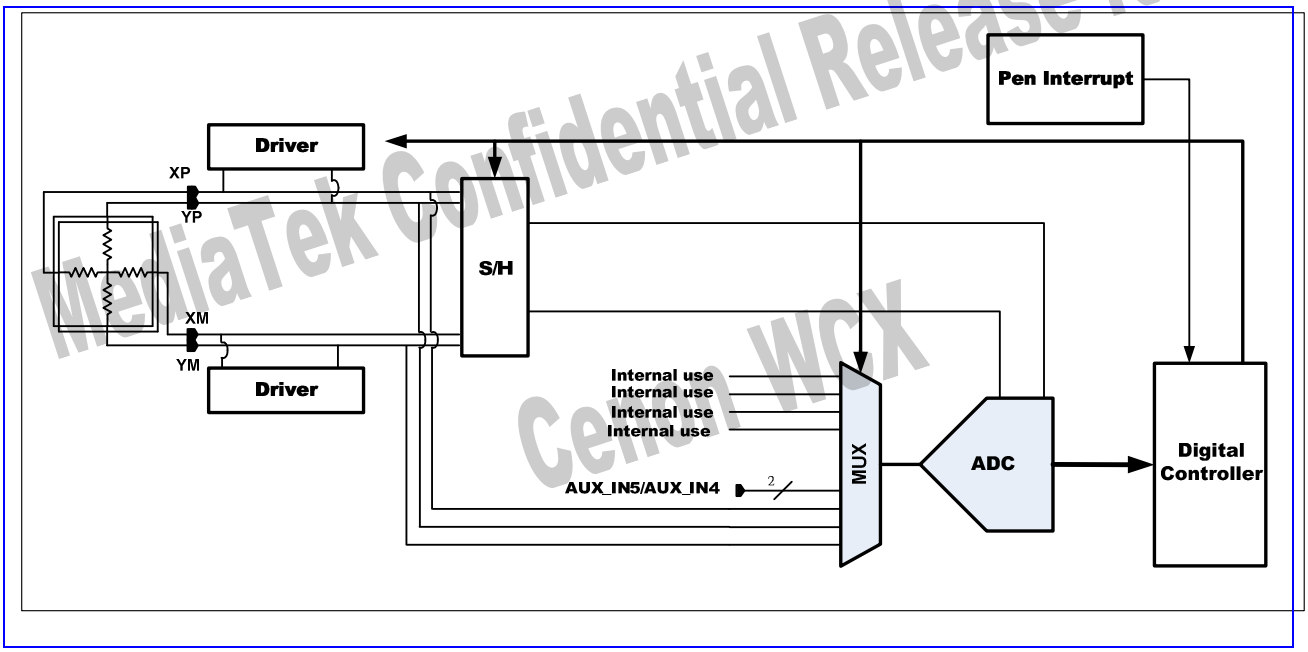



**Figure 8** Block diagram of AuxADC and touch screen

### 2.5.3.2 Function Specifications

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

| Symbol | Description | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| N | Resolution | | 10 | | Bit |
| FC | Clock Rate | | 1.0833 | | MHz |
| FS | Sampling Rate @ N-Bit | | 1.0833/(N+1) | | MSPS |
| | Input Swing | 0 | | AVDD | V |
| CIN | Input Capacitance<br>Unselected Channel<br>Selected Channel | | | 50<br>1.2 | <br>fF<br>pF |
| RIN | Input Resistance<br>Unselected Channel<br>Selected Channel | <br>10<br>1.8 | | | <br>MΩ<br>MΩ |
| | Clock Latency | | 11 | | 1/FC |
| DNL | Differential Nonlinearity | | +1.0/-1.0 | | LSB |

| | | | | | |
|---|---|---|---|---|---|
| INL | Integral Nonlinearity | | +1.0/-1.0 | | LSB |
| DVDD | Digital Power Supply | 1..08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply (AVDD_RFE) | 2.7 | 2.8 | 2.9 | V |
| T | Operating Temperature | -20 | | 80 | ℃ |
| | Current Consumption<br>Power-up<br>Power-Down | | <br>150<br>1 | | <br>μA<br>μA |
| Ztp | Supported Touch Panel Impedance | 200 | | 2K | Ohm |

**Table 14** The Functional specification of Auxiliary ADC

## 2.5.4 32-KHz Crystal Oscillator (RTC)

The low-power 32-KHz crystal oscillator XOSC32 is designed to work with an external piezoelectric 32.768KHz crystal and a load composed of two functional capacitors, as shown in the following figure.

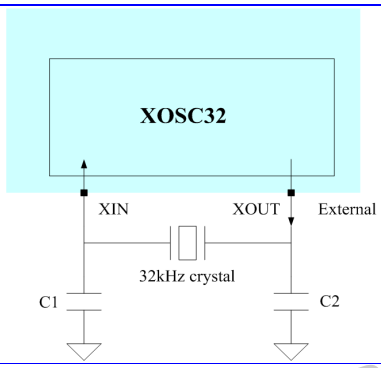



**Figure 9** Block diagram of XOSC32

The functional specification of XOSC32 is shown in the following table.

| Symbol | Description | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| AVDDRTC | Analog power supply | 1.0 | 2.8 | 3.0 | V |
| Tosc | Start-up time | | | 1 | sec |
| Dcyc | Duty cycle | 20 | 50 | 80 | % |
| | Current consumption | | | 5 | μA |
| | Leakage current | | 1 | | μA |
| T | Operating temperature | -20 | | 80 | ℃ |

**Table 15** Functional specification of XOSC32

Following are a few recommendations for the crystal parameters to be used with XOSC32.

| Symbol | Description | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| F | Frequency range | | 32768 | | Hz |
| GL | Drive level | | | 5 | uW |
| Δf/f | Frequency tolerance | | +/- 20 | | ppm |
| ESR | Series resistance | | | 50 | KΩ |
| C0 | Static capacitance | | | 1.6 | pF |
| C1/C2 | Load capacitance | 6 | | 12.5 | pF |

**Table 16** Recommended parameters of the 32 kHz crystal

## 2.5.5 APC DAC

### 2.5.5.1 Block Descriptions

The APC-DAC is a 10-bit DAC with an output buffer aiming for automatic power control. Following is the functional specification table.

### 2.5.5.2 Function Specifications

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| N | Resolution | | 10 | | Bit |
| FS | Sampling Rate | | | 1.0833 | MSPS |
| | Output Swing | 0 | | AVDD | V |
| | Drive Capacitance | | 200 | | pF |
| | Drive Resistance | | 10 | | kΩ |
| DNL | Differential Nonlinearity | | +/- 1.0 | | LSB |
| INL | Integral Nonlinearity | | +/- 1.0 | | LSB |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply (AVDD_RFE) | 2.7 | 2.8 | 2.9 | V |

| | | | | | |
|---|---|---|---|---|---|
| T | Operating Temperature | -20 | | 80 | ℃ |
| | Current Consumption<br>Power-up<br>Power-Down | | <br>200<br>1 | | <br>μA<br>μA |

**Table 17** APC-DAC specifications

## 2.6 PMU

Power management unit (PMU), is integrated into analog part. Following is the PMU block diagram.

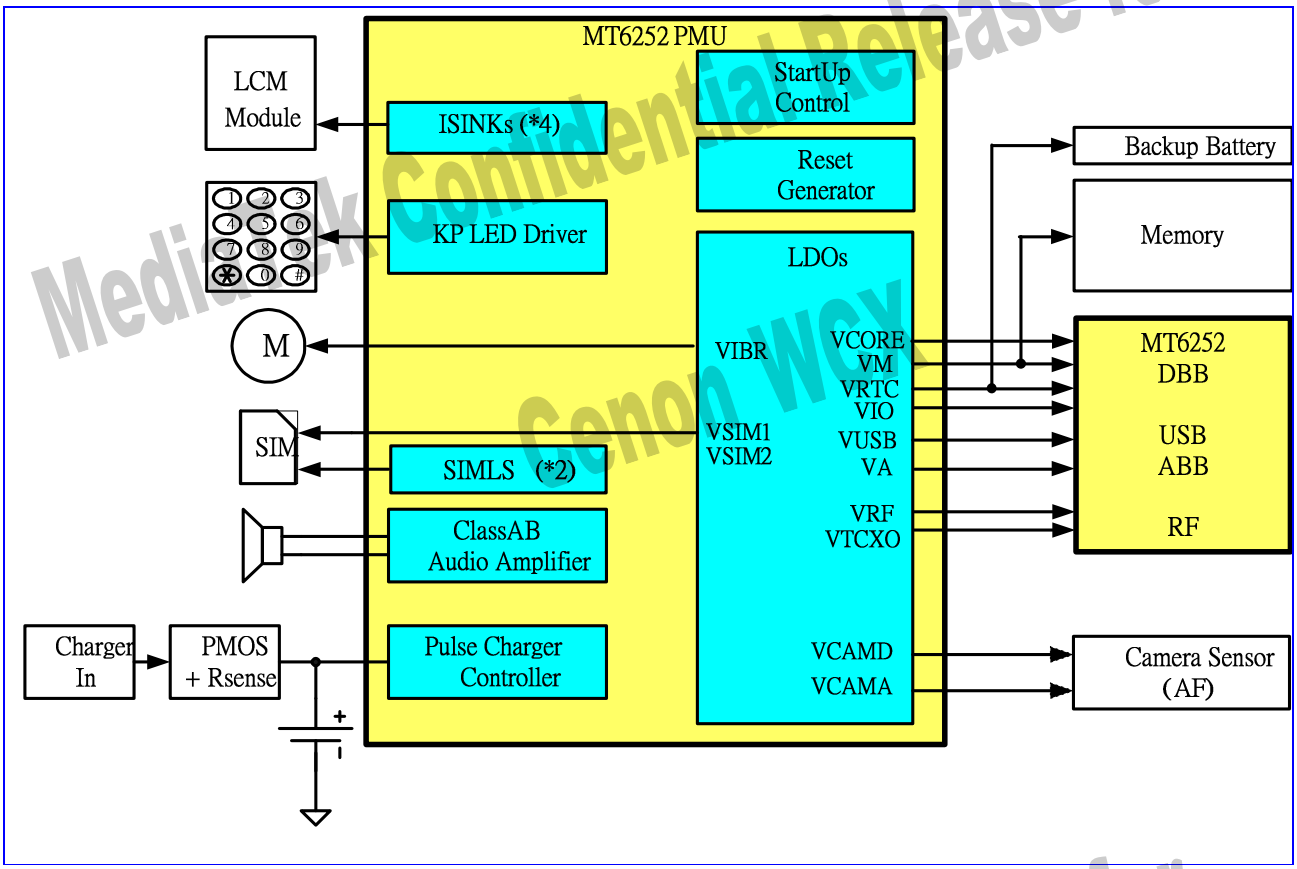



**Figure 10** PMU system block diagram

## 2.6.1 Low Dropout Regulators ( LDOs ), and Reference

The PMU Integrates 13 LDOs that are optimized for their given functions by balancing quiescent current, dropout voltage, line/load regulation, ripple rejection, and output noise.

### 2.6.1.1 Block Description

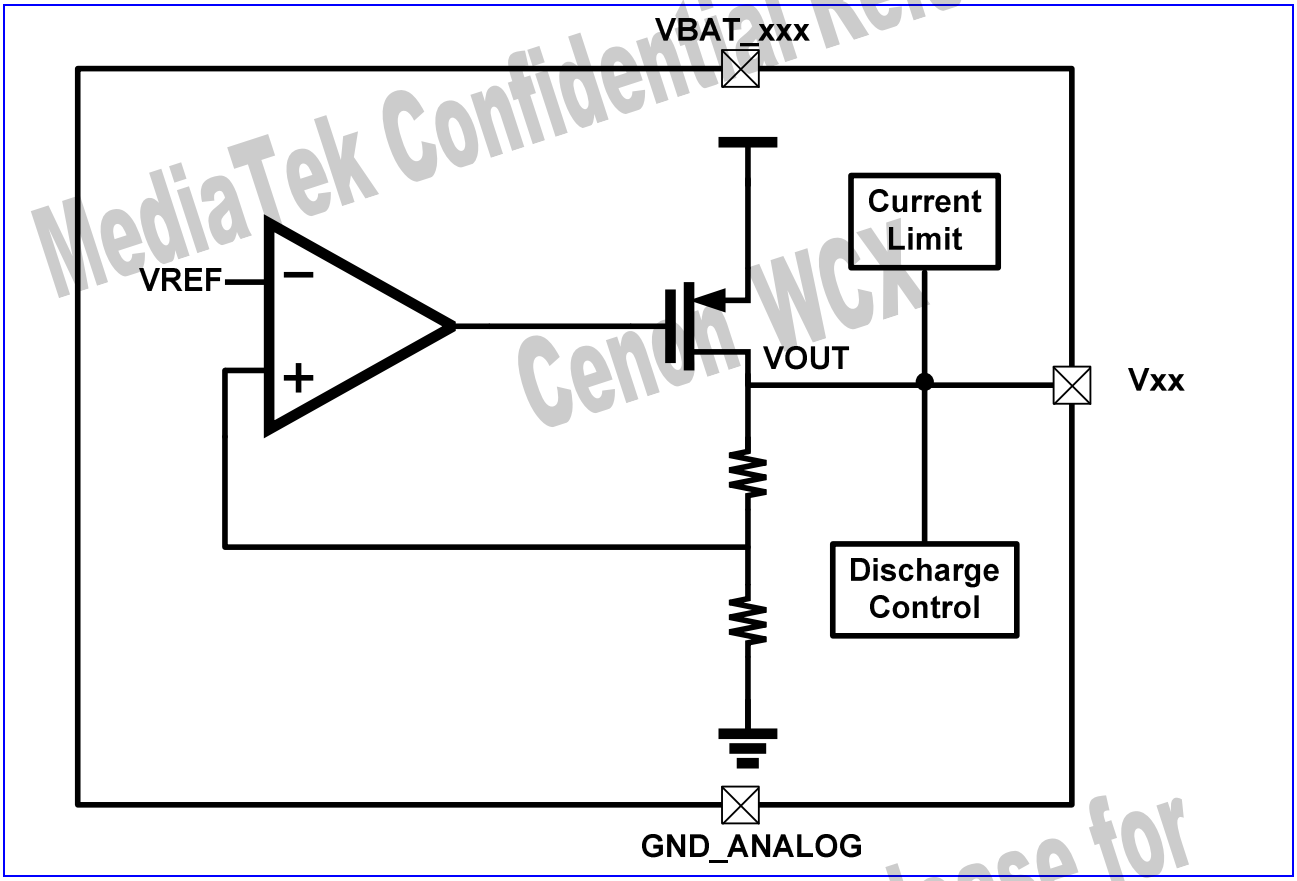



**Figure 11** LDO block diagram

### 2.6.1.2 LDO Types

| Type | LDO Name | Vout | Imax | Description |
|---|---|---|---|---|
| ALDO | VRF | 2.8 | 150 | RF circuit |
| ALDO | VA | 2.8 | 100 | Analog baseband |
| ALDO | VTCXO | 2.8 | 40 | 13/26 MHz reference clock |

| Type | LDO Name | Vout | Imax | Description |
|---|---|---|---|---|
| ALDO | VCAMA | 1.5/1.8/2.5/2.8 | 150 | Analog camera power |
| DLDO | VCORE | 0.8V~1.35V(25mv/step) | 200 | Digital core |
| DLDO | VIO | 2.8 | 200 | Digital IO |
| DLDO | VM | 1.8/2.9 | 150 | External memory, selectable |
| DLDO | VSIM1 | 1.8/3.0 | 30 | SIM card, selectable |
| DLDO | VSIM2 | 1.3/1.5/1.8/2.5/2.8/3.0/3.3 | 30 | SIM2 card, selectable |
| DLDO | VUSB | 3.3 | 50 | USB |
| DLDO | VIBR | 1.3/1.5/1.8/2.5/2.8/3.0/3.3 | 150 | Vibrator |
| DLDO | VCAMD | 1.3/1.5/1.8/2.5/2.8/3.0/3.3 | 100 | Digital camera power |
| RTCLDO | VRTC | 2.8 | 2 | Real-time clock |

**Table 18** LDO types and brief specification

### 2.6.1.3 Functional Specification

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | Load capacitor(VA/VTCXO) | | | 1 | | µF |
| | Load capacitor(VRF/VCAMA) | | | 2.2 | | µF |
| | Current limit | | 1.2*Imax | | 5*Imax | mA |
| | Vout | Including load regulation, line regulation, and temperature coefficient | max(-5%, -0.1V) | | max(+5% , +0.1V) | V |
| | Transient response | Slew: 15mA/us | -0.1 | | +0.1 | V |
| | Temperature coefficient | | | | 100 | ppm/C |
| | PSRR | Iout<0.5*Imax 217<f<3K Hz | 65 | | | dB |
| | | Iout<0.5*Imax 3K<f<30K Hz | 45 | | | dB |
| | Output noise | 10 to 80K Hz | | 90 | | uVrms |
| | Quiescent current | Iout=0 | | | 55 | µA |
| | Turn-on overshoot | Iout=0 | | | max(+10 %, +0.1V) | V |
| | Turn-on settling time | Iout=0 | | | 360 | µS |

**Table 19** Analog LDO Specification

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | Load capacitor(except VIO) | | | 1 | | µF |
| | Load capacitor(VIO) | | | 2.2 | | µF |
| | Current limit(except VCAMD) | | 1.2*Imax | | 5*Imax | mA |
| | Current limit(VCAMD) | | 2*Imax | | 8*Imax | mA |
| | Vout | Include 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 |
| | Output noise | 10 to 80K Hz | | 250 | 500 | uVrms |
| | Quiescent current | Iout=0 | | | 15 | µA |
| | Turn-on overshoot | Iout=0 | | | max(+10 %, +0.1V) | V |
| | Turn-on settling time(except VIO) | Iout=0 | | | 360 | µs |
| | Turn-on settling time(VIO) | Iout=0 | | | 40 | µs |

**Table 20** Digital LDO Specification

## 2.6.2 SIM Interface

### 2.6.2.1 Block Description

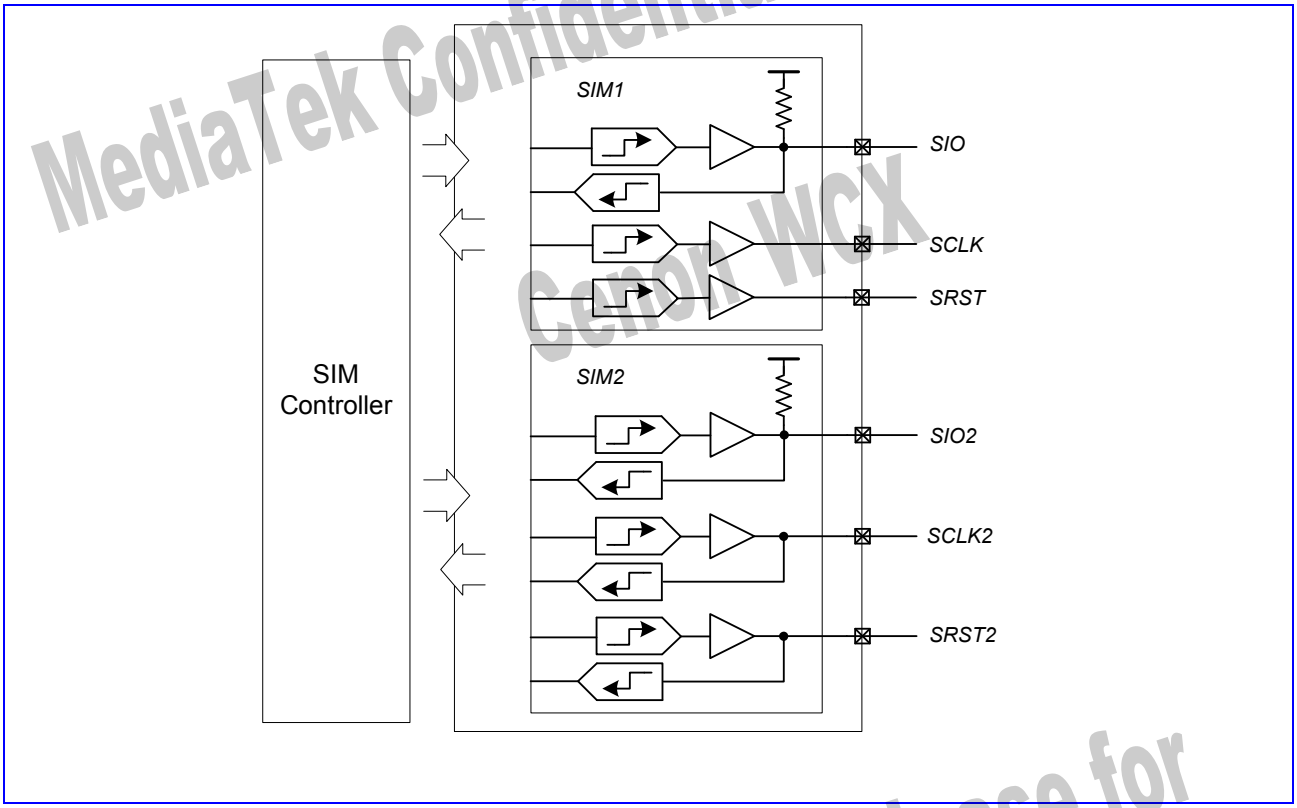



**Figure 12** SIM interface block diagram

There are two SIM card interface modules to support two SIM cards simultaneously. The SIM card interface circuitry of PMU meets all ETSI and IMT-2000 SIM interface requirements. It provides level shifting needs for the low voltage GSM controller to communicate with either 1.8 V or 3 V SIM cards. All SIM cards contain a clock input, a reset input, and a bi-directional data input/output. The clock and reset inputs to SIM cards are level shifted from the supply of digital baseband to the SIM supply (Vsim). The bi-directional data bus is internal pull high to Vsim via 5 kΩ resistor.

The 2nd SIM card interface can be used for supporting another SIM card or mobile TV. The interface pins such as SIO2, SRST2, SCLK2, can be configured as GPIO when there is no need to use the 2nd SIM card interface.

All pins that connect to the SIM card (Vsim, SRST, SCLK, SIO) withstand over 2kV HBM (human body mode) ESD. In order to ensure proper ESD protection, careful board layout is required.

### 2.6.2.2 Functional Specification

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Interface to 3 V SIM Card | | | | | | |
| | Output Low of SRST | I = 200 µA | | | 0.36 | V |
| | Output High of SRST | I = -200 µA | 0.9*VSIM | | | V |
| | Output Low of SCLK | I = 200 µA | | | 0.4 | V |
| | Output High of SCLK | I =-100 µA | 0.9*VSIM | | | V |
| | Input/Output High of SIO | I = ±20 µA | VSIM-0.4 | | | V |
| | (Iil)Pull high current of SIO | Vil = 0 V | | | -1 | mA |
| | (Vol)Input/Output Low of SIO | Iol = 1 mA | | | 0.4 | V |
| Interface to 1.8 V SIM Card | | | | | | |
| | Output Low of SRST | I = 200 µA | | | 0.2*VSIM | V |
| | Output High of SRST | I = -200 µA | 0.9*VSIM | | | V |
| | Output Low of SCLK | I = 200 µA | | | 0.12*VSIM | V |
| | Output High of SCLK | I = -100 µA | 0.9*VSIM | | | V |
| | Input/Output Low of SIO | | | | 0.15*VSIM | V |
| | Input/Output High of SIO | I = ±20 µA | VSIM-0.4 | | | V |
| | (Iil)Pull high current of SIO | Vil = 0 V | | | -1 | mA |
| | (Vol)Input/Output Low of SIO | Iol = 1 mA | | | 0.15*VSIM | V |
| SIM Card Interface Timing | | | | | | |
| | SIO pull-up resistance to VSIM | | 4 | 5 | 6 | kΩ |
| | SRST, SIO rise/fall times | VSIM = 3, 1.8 V, load with 30 pF | | | 1 | µs |
| | SCLK rise/fall times | VSIM = 3 V, CLK load with 30 pF | | | 18 | ns |
| | | VSIM = 1.8 V, CLK load with 30 pF | | | 50 | ns |
| | SCLK frequency | CLK load with 30 pF | | | 5 | MHz |
| | SCLK duty cycle | SIMCLK Duty = 50%, fsimclk = 5 MHz | 47 | | 53 | % |

**Table 21** SIM Level Shifter

## 2.6.3 Current Sink Driver and Keypad LED Switches

### 2.6.3.1 Block Description

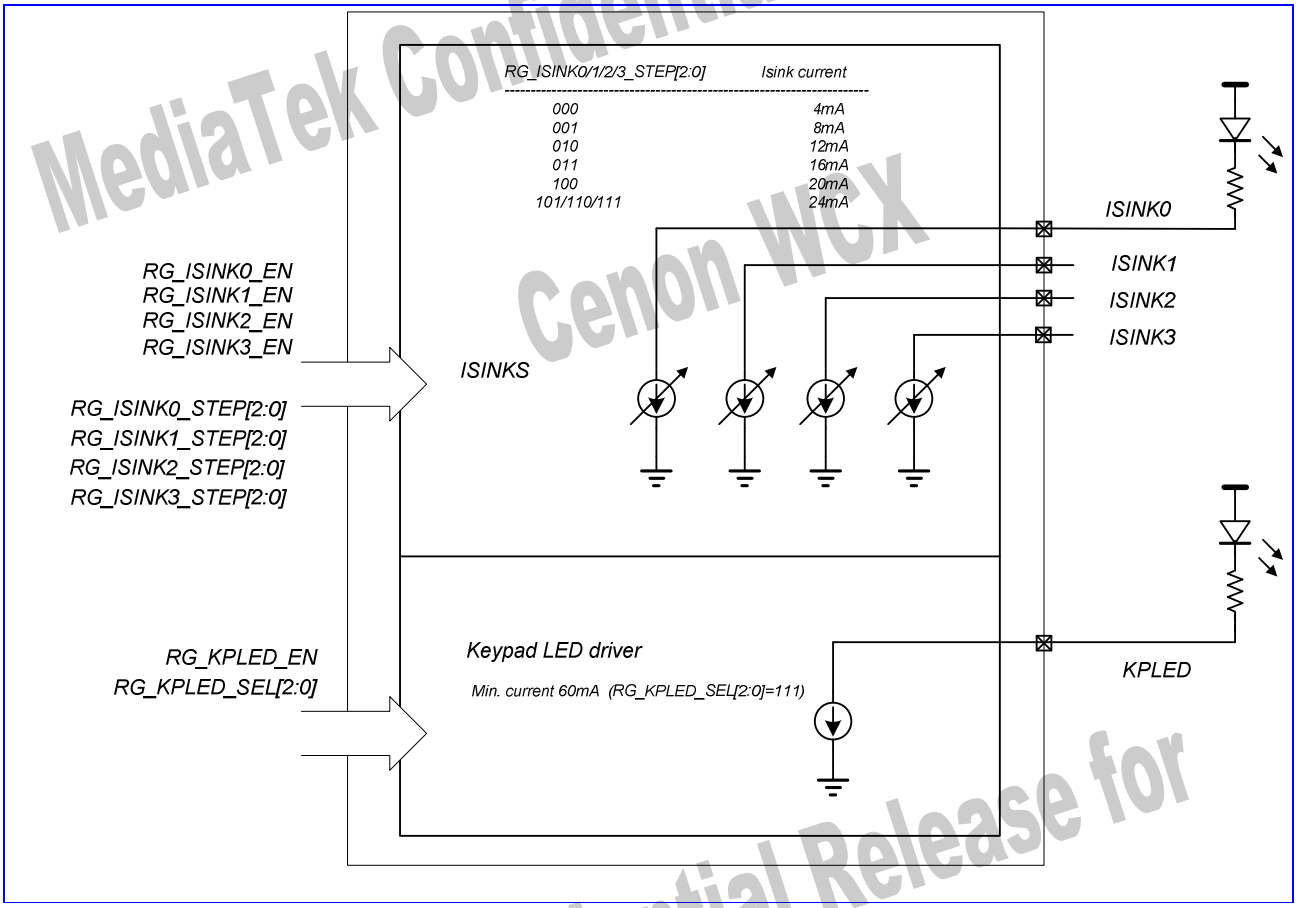



**Figure 13** Current sink and keypad LED driver block diagram

### 2.6.3.2 Functional Specification

Four current controlled open drain drivers (Isink0~3) are also implemented to drive LCM backlight module and each provides 6 current level steps up to 24 mA. These current sinks are controlled by enabling registers (RG_ISINK0~3_EN). The impedance output is high when disabled. The sink current can be set by RG_ISINK0~3_STEP[2:0] from 000 (4 mA)

to 101 (24 mA) at 4 mA per step. Also, the drivers can be switching on/off through dimming control. The dimming frequency and duty can be programmed by registers. For details, please refer to the "PWM" section.

Built-in open-drain output switches drive the Keypad LED in the handset. This driver is controlled by baseband with enable registers (RG_KPLED_EN), and the output is high impedance when disabled. The Keypad LED connects its anode to VBAT and its cathode to the ballast resistor. The other terminal of the ballast resistor connects to the driver of MT6252. The Keypad LED driver is a low Ron switch which allows 60mA current.

The brightness of the Keypad LED can be controlled by changing the external ballast resistor or switching on/off the driver through dimming control. The dimming frequency and duty can be programmed by registers. For the details, please refer to the "PWM" section.



| 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<0.3V, 100% dimming duty | | 24 | | mA |

**Table 22** KP LED functional specification

## 2.6.4 Battery Charge

### 2.6.4.1 Block Description

The charger circuit in MT6252 is shown as in the following block diagram.

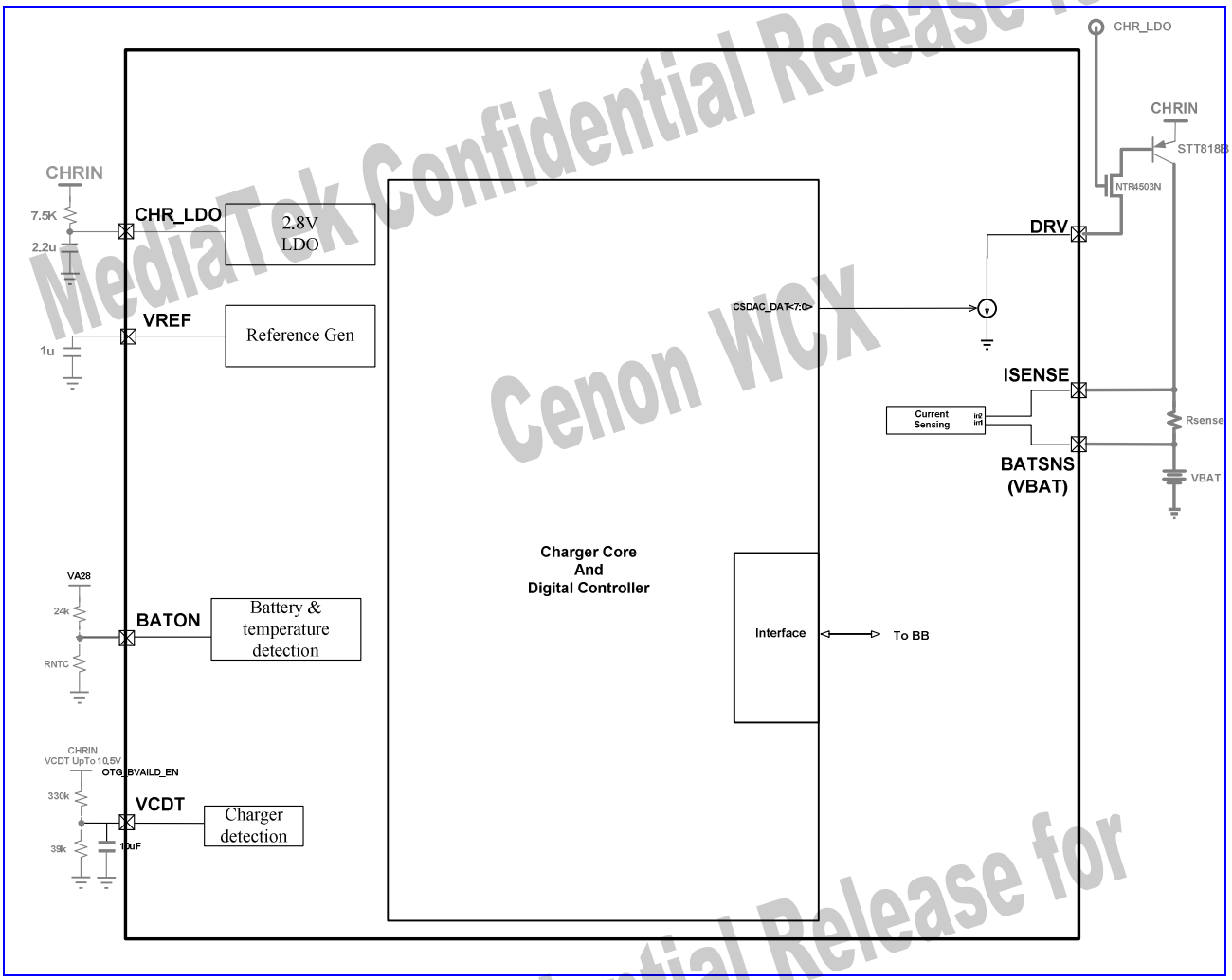



**Figure 14** Battery charger block diagram

## 2.6.4.2 Functional Specification

The charger controller senses the charger input voltage (CHRIN) 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 is activated. This detector can resist higher input voltages than other parts of the PMU. Therefore, if 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.

Also if the charger-in level is not high enough (<4.3V), the charger will also be disabled to avoid improper charging behavior.

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

**Table 23** Charger Specifications

### 2.6.4.3 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 in MT6252 supports pre-charge mode (VBAT<3.2V, PMU power-off state), CC mode (constant current mode or fast charging mode within the range of 3.2V<VBAT<4.2V) and Top-Off mode to optimize the charging procedure for the Li-ion battery.

### 2.6.4.4 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. While the battery voltage is deeply discharged below 2.2V, PRECC0 trickle charging current applies to the battery.

The PRECC0 trickle charging current is about 56 mA when VBAT is under 2.2 V.

When the battery voltage exceeds 2.2 V, the so-called PRECC1 stage, the closed-loop pre-charge is enabled. The voltage drop across the external RSENSE is kept around 40 mV (AC Charger) or 14 mV (USB Host). The closed-loop pre-charge current can be calculated using the following formulas:

$$I_{PRECC1,AC\ adapter} = \frac{V_{SENSE}}{Rsense} = \frac{40mV}{Rsense}$$

$$I_{PRECC1,USB\ HOST} = \frac{V_{SENSE}}{Rsense} = \frac{14mV}{Rsense}$$

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | **IUNIT with 500ms Pulse** | **VBAT<2.2V** | | **56** | | **mA** |
| | Pre-charging current | VBAT<2.2V | | 56 | | mA |
| | | VBAT>=2.2V(USB HOST) | | $14/R_{sense}$ | | mA |
| | | VBAT>=2.2V(AC Adapter) | | $40/R_{sense}$ | | mA |

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | **IUNIT with 500ms Pulse** | **VBAT<2.2V** | | **56** | | **mA** |
| | Pre-charging off threshold | CHR_EN=L | | 3.3 | | V |
| | Pre-charging off hysteresis | | | 0.4 | | V |

**Table 24** Pre-charge Specifications

### 2.6.4.5 Constant Current Mode

As the battery is charged up and over 3.2 V, it can switch to the CC mode (CHR_EN should be high). In 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 0.2 ohm, the CC mode charging current can be set between 70 mA and 800 mA. It can accommodate the battery charger to the various charger inputs with different current capabilities.

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | CC mode charging current ( CS_VTH ) | RG_CS_VTH [2:0]=000 | | $14/R_{sense}$ | | mA |
| | | RG_CS_VTH [2:0]=001 | | $40/R_{sense}$ | | mA |
| | | RG_CS_VTH [2:0]=010 | | $80/R_{sense}$ | | mA |
| | | RG_CS_VTH [2:0]=011 | | $90/R_{sense}$ | | mA |
| | | RG_CS_VTH [2:0]=100 | | $110/R_{sense}$ | | mA |
| | | RG_CS_VTH [2:0]=101 | | $130/R_{sense}$ | | mA |
| | | RG_CS_VTH [2:0]=110 | | $140/R_{sense}$ | | mA |
| | | RG_CS_VTH [2:0]=111 | | $160/R_{sense}$ | | mA |
| | Current sensing resistor | RSENSE | | 0.2 | | ohm |

**Table 25** Constant Current Specifications

### 2.6.4.6 Top-Off Mode and Over-voltage Protection

While the battery voltage reaches about 4.2 V, a constant current with a much shorter period is used for charging. It allows more frequent full battery detections in the non-charging period. This is called full voltage charging mode or constant voltage charging mode in correspondence to a linear charger. While the battery voltage reaches 4.2 V, more

than the pre-setting times within the limited charging cycles, the end-of-charging process starts. It may prolong the charging and detecting period to get the optimized the full charging volume. This end-of-charging process is fully controlled by the baseband and can be easily optimized for different battery packs. Once the battery voltage exceeds 4.35 V, a hardware Over-voltage protection (OV) should be activated and the charger will be turned off immediately.

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | Charging complete threshold | | 4.15 | 4.2 | 4.25 | V |
| | Battery Over voltage protection threshold (OV) | | | 4.35 | | V |

**Table 26** Top-Off Specifications

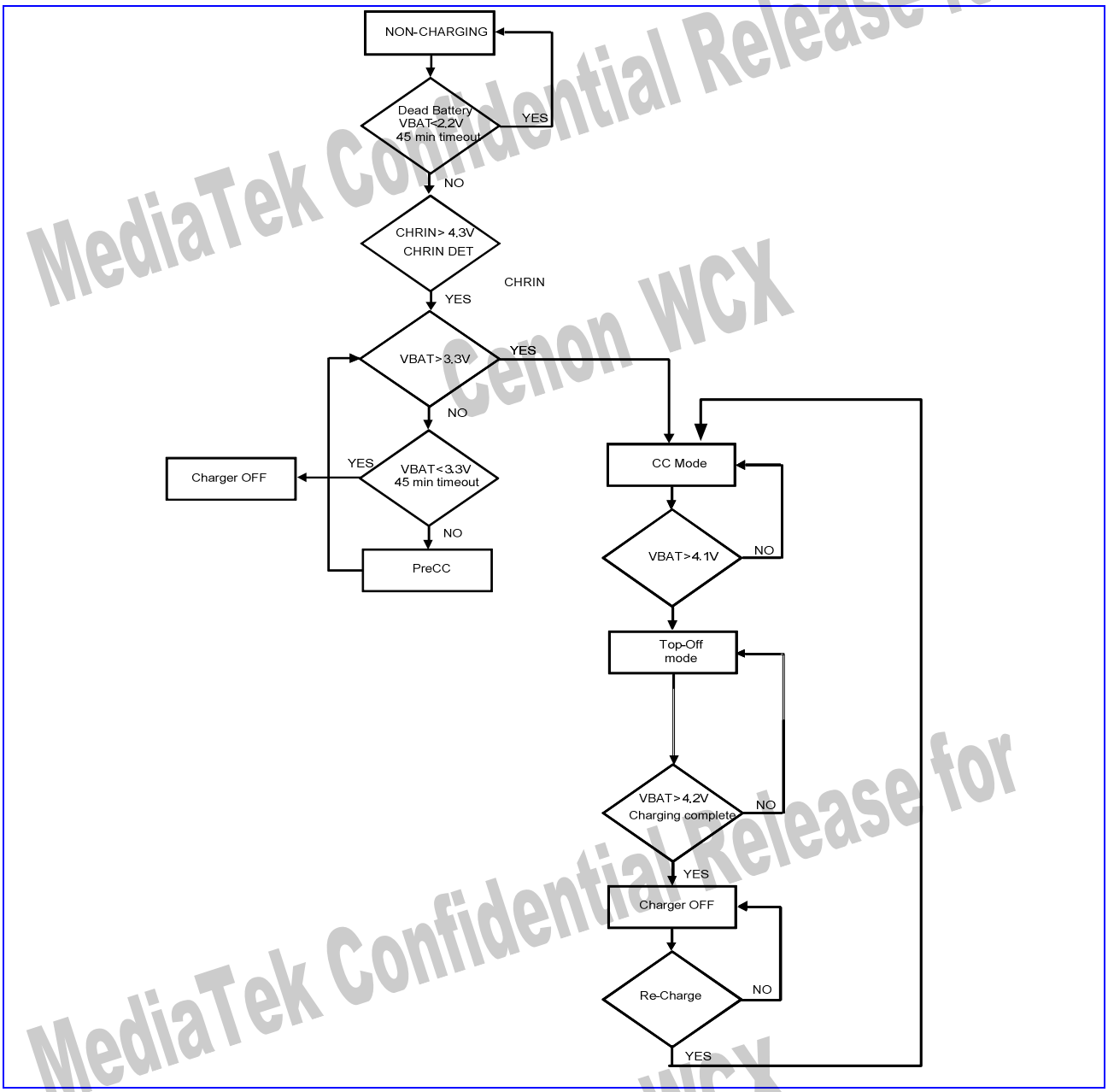



**Figure 15** Charging States Diagram

CHR_EN is a register programmed by the register setting, which is controlled by BB or SW. In other words, BB and SW are the masters to manage the charging process at VBAT > 3.2 V. It can implement any charging profile; e.g. trickling charging, simply by programming CHR_EN.

Pin BATON turns off the charger immediately if it goes high (> 2.5±0.1 V). This function is designated to stop CC or Top-Off charging mode in case the battery is accidentally removed

### 2.6.4.7 Charging Profile

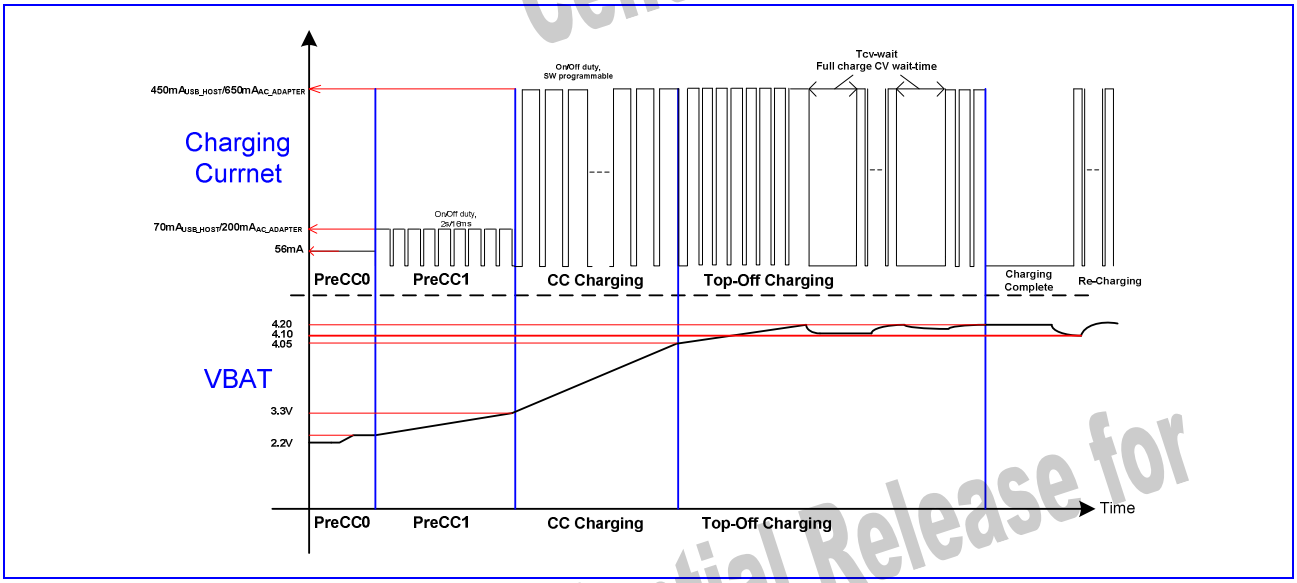



**Figure 16** I-V curve of Li-Ion battery charging

## 2.6.5 Class-AB Audio Amplifier

### 2.6.5.1 Block Description

MT6252 has a built-in high fidelity class AB audio power amplifier. It is capable of delivering 1 watt of power to an 8 ohm BTL load with less than 10% distortion (THD+N) from a 4.2 V battery supply.

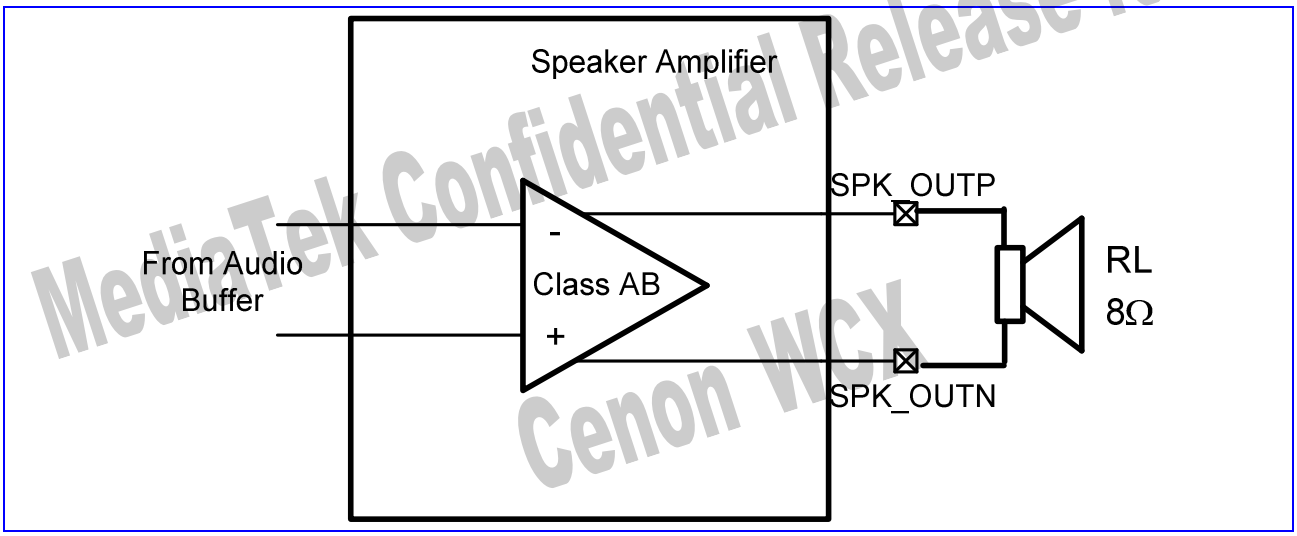



**Figure 17** Class-D audio amplifier block diagram

### 2.6.5.2 Functional Specification

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | RMS Power | 8Ω load, VBAT = 4.2 V | | 900 | | mW |
| | | 8Ω load, VBAT = 3.4 V | | 550 | | mW |
| | THD+N | 1KHz, Po=0.8Wrms, 4.2V | | 0.01 | | % |
| | PSRR | 20 Hz ~ 1 kHz, diff. mode | | 75 | | dB |
| | Shutdown current | SPKL_EN = SPKR_EN = 0 | | 0.03 | | μA |
| | Quiescent power supply current | VBAT = 4.2 V, no input | | 3 | | mA |
| | Gain adjustment | | 0 | | 18 | dB |
| | Gain adjustment steps | | | 3 | | dB |

# 2.7 GSM/GPRS RF

## 2.7.1 Block Description

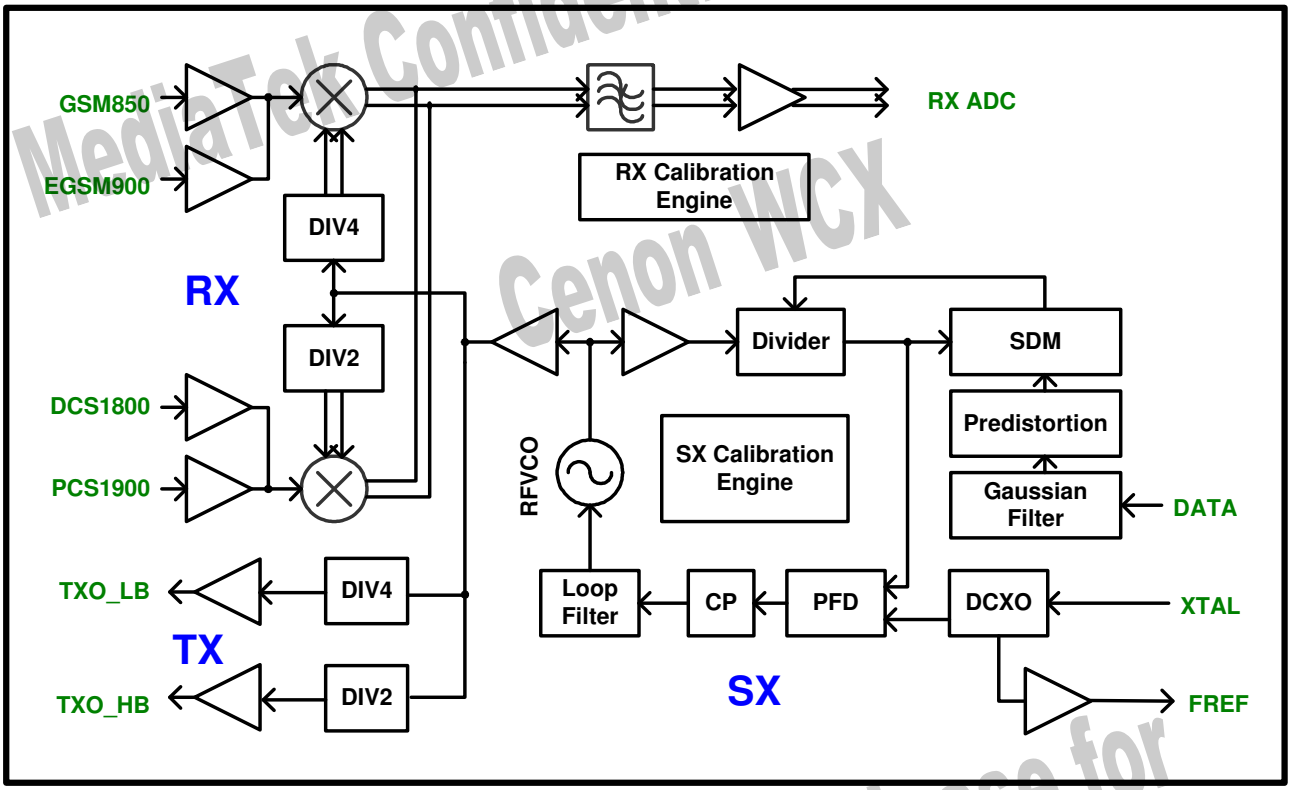



**Figure 18** RFSYS Block Diagram

MT6252 RFSYS is a low current transceiver to support a true quad-band GSM/GPRS cellular system. The highly integrated RF system implements a high sensitivity and channel selection receiver, a high precision transmission modulator, a low phase noise frequency synthesizer, and a digitally controlled crystal oscillator. The external components required for the GPRS radio design are the Rx SAWs, PA, switchplexer, X'TAL, and a few passives.

### 2.7.1.1 Receiver

The receiver section integrates a low noise amplifier (LNA), down-converting mixers, and baseband amplifier/channel filters. The input of the receiver are from four LNAs to support quad-band applications. The LNAs are used to amplify

the received signal from the antenna, and then output to the mixer. While receiving large input signals, the LNAs can be switched to the low gain mode. The LO of quadrature mixers are generated from divide-by-4 and divide-by-2 of Fractional-N SX output. The sebsequent baseband amplifier is then used to amplify the down-converted signal. The gain of the baseband amplifier is programmable. DC offset correction is built in, and the DC correction is automatically triggered for each receiver slot. To provide the necessary adjacent channel and blocking filtering, the RX IF channel filter is built in.

#### 2.7.1.2 Transmitter

The transmit section is a frequency modulator. The input DATA stream of transmitter is Gaussian filtered from baseband, and is fed to a pre-distortion filter. SX calibration engine are built in and are triggered by the Warm-up state before a TX slot. The pre-distorted DATA stream together with the channel word is then input to SD modulator for DIV-N control and frequency modulation of RFVCO.

The output of RFVCO is then divided by four for the applications of GSM850 and EGSM bands and by two for the applications of DCS and PCS bands. TX output buffers amplify the divided signals for PA input.

#### 2.7.1.3 Synthesizer

The synthesizer section is a Phase-locked Loop (PLL) based fractional-N frequency synthesizer with a fully integrated set of RF VCO and loop filter. It provides the Local Oscillator (LO) signals for both the receiver and the transmitter. Other blocks in PLL are also included, such as the divider, Phase Frequency Detector (PFD), Charge Pump (CP), Sigma-delta Modulator (SDM), and X'TAL oscillator. The X'TAL oscillator is used to generate a 26 MHz clock as a reference signal for the synthesizer and the baseband. It is a digitally controlled X'TAL oscillator. A coarse capacitor array is built in for the factory 26 MHz clock frequency calibration, and a fine capacitor array is for the dynamic frequency control of the GSM system. The SX PLL takes the 26 MHz clock as a reference signal, and synthesize the desired LO frequency for both the receiver and transmitter. The synthesizer frequency programming is detailed as in Section 2.7.3.3.

### 2.7.2 Functional Specification

| RFSYS MODE | AVDD28_RFD | AVDD28_TCXO | AVDD28_RF1<br>AVDD28_RF2 | RFSYS Total | Unit |
|---|---|---|---|---|---|
| Deep Sleep (DCXO is off) | 8.1 | 2.3 | 2.5 | 13 | uA |
| Sleep (DCXO is on) | 0.5 | 2.3 | 0.0025 | 2.8 | mA |
| RX (GSM850/EGSM) | 0.8 | 14 | 48 | 62 | mA |
| RX (DCS/PCS) | 0.8 | 14 | 51 | 65 | mA |
| TX (GSM850/EGSM) | 3.9 | 6.6 | 44 | 55 | mA |
| TX (DCS/PCS) | 3.9 | 6.6 | 39 | 50 | mA |

**Table 27** DC Characteristic (TA=25°C, VDD=2.8V unless otherwise stated)

| Symbol | Description | Band | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $F_{RX}$ | Input frequency | GSM850 | | 869 | | 894 | MHz |
| | | GSM900 | | 925 | | 960 | MHz |
| | | DCS1800 | | 1805 | | 1880 | MHz |
| | | PCS1900 | | 1930 | | 1990 | MHz |
| $G_{max}$ | Differential max voltage gain | GSM850 | LNA = high gain. PGA = 54 dB | 92[1] | 95 | | dB |
| | | GSM900 | | 92[2] | 95 | | dB |
| | | DCS1800 | | 92[3] | 95 | | dB |
| | | PCS1900 | | 92[4] | 95 | | dB |
| $G_{step,LNA}$ | Front-end LNA gain step | GSM850 | LNA = high gain to low gain | 23[1] | 26 | 29[1] | dB |
| | | GSM900 | | 23[1] | 26 | 29[1] | dB |
| | | DCS1800 | | 23[1] | 26 | 29[1] | dB |
| | | PCS1900 | | 23[1] | 26 | 29[1] | dB |
| $NF_{25}$ | Noise figure at 25℃ | GSM850 | LNA = high gain. PGA = 54 dB | | 2.5 | 4.5[1] | dB |
| | | GSM900 | | | 2.5 | 4.5[2] | dB |
| | | DCS1800 | | | 2.5 | 4.5[3] | dB |
| | | PCS1900 | | | 2.5 | 4.5[4] | dB |
| $NF_{85}$ | Noise figure at 85℃ | GSM850 | LNA = high gain. PGA = 54 dB | | 3.5 | 4[5] | dB |
| | | GSM900 | | | 3.5 | 4[5] | dB |
| | | DCS1800 | | | 3.5 | 4[5] | dB |
| | | PCS1900 | | | 3.5 | 4[5] | dB |
| IIP2 | 2nd-order input intercept point | GSM850 | LNA = high gain. PGA = 54 dB | 31[1] | 43 | | dBm |
| | | GSM900 | | 31[2] | 43 | | dBm |
| | | DCS1800 | | 31[3] | 43 | | dBm |
| | | PCS1900 | | 31[4] | 43 | | dBm |
| IIP3 | 3rd-order input intercept point | GSM850 | LNA = high gain. PGA = 0 dB | -14[1] | -3 | | dBm |
| | | GSM900 | | -14[2] | -3 | | dBm |
| | | DCS1800 | | -14[3] | -3 | | dBm |
| | | PCS1900 | | -14[4] | -3 | | dBm |
| $IIP3_{-20}$ | 3rd-order input intercept point @ -20 ℃ | GSM850 | LNA = high gain. PGA = 0 dB | | -5 | | dBm |
| | | GSM900 | | | -5 | | dBm |
| | | DCS1800 | | | -5 | | dBm |
| | | PCS1900 | | | -5 | | dBm |
| $IP_{1dB}$ | Input 1 dB compression point | GSM850 | LNA = high gain. PGA = 0dB | | -20 | | dBm |
| | | GSM900 | | | -20 | | dBm |
| | | DCS1800 | | | -20 | | dBm |
| | | PCS1900 | | | -20 | | dBm |
| $SN_{3M}$ | Receiver S/N with 3 MHz blocker | GSM850 | Blocker = –23 dBm. Noise power is calculated within 130 kHz bandwidth | 11[1] | 14 | | dB |
| | | GSM900 | | 11[2] | 14 | | dB |
| | | DCS1800 | Blocker = –26 dBm. Noise power is calculated within 130 kHz bandwidth | 11[3] | 14 | | dB |
| | | PCS1900 | | 11[4] | 14 | | dB |
| $PGA_{linear}$ | PGA gain linearity | ALL | INL | | 0.2 | 1[5] | dB |
| | | | DNL | | 0.1 | 0.5[5] | dB |
| $PGA_{step}$ | PGA gain step | ALL | | | 2 | | dB |
| $PGA_{step}$ | PGA dynamic range | ALL | PGA = 0 dB to 60 dB | | 60 | | dB |

**Table 28** RX AC Characteristic (TA=25°C, VDD=2.8V unless otherwise stated)

| Symbol | Description | Band | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $F_{TX}$ | Frequency | GSM850 | | 824 | | 849 | MHz |
| | | GSM900 | | 880 | | 915 | MHz |
| | | DCS1800 | | 1710 | | 1785 | MHz |
| | | PCS1900 | | 1850 | | 1910 | MHz |
| $PE_{rms}$ | RMS phase error | GSM850<br>GSM900 | | | 1 | 2.5[1,2] | degrees |
| | | DCS1800<br>PCS1900 | | | 1 | 2.5[3,4] | degrees |
| ORFS | Output modulation spectrum | GSM850<br>GSM900 | 400 kHz offset<br>(RBW = 30 kHz bandwidth) | | -70 | -64[1,2] | dBc |
| | | DCS1800<br>PCS1900 | | | -67 | -64[3,4] | dBc |
| | | GSM850<br>GSM900 | 1.8 MHz offset<br>(RBW = 30 kHz bandwidth) | | | -75[5] | dBc |
| | | DCS1800<br>PCS1900 | | | | -75[5] | dBc |
| $TX_{NOISE}$ | Tx noise in Rx band | GSM850 | 20 MHz Offset | | -165 | -164[5] | dBc/Hz |
| | | | 35 MHz Offset | | -168 | -167[5] | dBc/Hz |
| | | GSM900 | 20 MHz Offset | | -165 | -164[5] | dBc/Hz |
| | | | 35 MHz Offset | | -168 | -167[5] | dBc/Hz |
| | | DCS1800 | 20 MHz Offset | | -160 | -156[5] | dBc/Hz |
| | | PCS1900 | 20 MHz Offset | | -160 | -156[5] | dBc/Hz |
| $P_{out}$ | Output power level | GSM850<br>GSM900 | PA driver amplifier.<br>$R_{load}$ = 50 Ω | 1[1,2] | 3 | 6[1,2] | dBm |
| | | DCS1800<br>PCS1900 | | 1[3,4] | 3 | 6[3,4] | dBm |
| $TX_{HARM}$ | Output 3rd harmonics | ALL | PA driver amplifier. | | -10 | | dBc |

**Table 29** TX AC Characteristic (TA=25°C, VDD=2.8V unless otherwise stated)

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $F_{range}$ | Frequency range | | 3296 | | 3980 | MHz |
| $F_{ref}$ | Reference frequency | | | 26 | | MHz |
| $F_{res}$ | Frequency step resolution | | | 3 | | Hz |
| $PN_{10k}$ | Phase noise | @ 10 kHz offset | | -83 | | dBc/Hz |
| $PN_{400k}$ | | @ 400 kHz offset | | -116 | | dBc/Hz |
| $PN_{3M}$ | | @ 3 MHz offset | | -136 | | dBc/Hz |
| $T_{lock\_rx}$ | Lock time of RX Burst | Frequency error<± 0.1ppm | | 150 | 200[5] | us |
| $T_{lock\_tx}$ | Lock time of TX Burst | Frequency error<± 0.1ppm | | 200 | 300[5] | us |
| $RFVCO_{PS}$ | Pushing figure | Supply Pushing from VDD28_RF2 | | 400 | | kHz/V |

**Table 30** DCXO AC Characteristic (TA=25°C, VDD=2.8V unless otherwise stated)

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $F_{ref}$ | Operating frequency | | | 26 | | MHz |
| $C_L$ | Crystal C Load | | | 7.5 | | pF |
| $T_S$ | Crystal tuning sensitivity | | 27.5 | 32.3 | | ppm/pF |
| SR | Static range | CDAC from 0 to 100 | ± 22 | ± 30 | | ppm |
| DR | Dynamic range | CAFC from 0 to 8191 | 36 | 42 | | ppm |
| TC | Tempe Characteristic | $T^A$ from -20°C to 65°C | | +2 | | ppm |
| $F_{res\text{-}AFC}$ | AFC tuning step | | | 0.01 | | ppm/DA |

| Symbol | Description | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| | | | | | | C |
| $T_{AFC}$ | AFC settling time | CAFC from 0 to 8191<br>CAFC from 8191 to 0<br>Frequency error < 0.1ppm | | 100 | 200[5] | us |
| $T_{DCXO}$ | Start-up time | Frequency error < 1ppm<br>Amplitude > 90 % | | | 2[5] | ms |
| $DCXO_{PS}$ | Pushing figure | Supply Pushing from VDD28_TCXO | | 0.5 | | ppm/V |
| $V_{Fref}$ | Fref buffer output level | Loading = C3pF//R2Kohm | 600[5] | 800 | | $mV_{p-p}$ |
| $XO_{PN}$ | Fref buffer output phase noise<br>CDAC=50<br>CAFC=6400 | @ 1kHz offset | | -135 | | dBc/Hz |
| | | @ 10 kHz offset | | -145 | | dBc/Hz |
| | | @ 100 kHz offset | | -155 | | dBc/Hz |
| | | @ 1 MHz offset | | -155 | | dBc/Hz |

**Table 31** DCXO AC Characteristic (TA=25°C, VDD=2.8V unless otherwise stated)

1. *Tested at channel 190 of GSM 850 band*
2. *Tested at channel 70 of E-GSM 900 band*
3. *Tested at channel 700 of DCS 1800 band*
4. *Tested at channel 660 of PCS 1900 band*
5. *Not subject to production test – verified by characterization and design*
6. *AVDD includes AVDD28_RF1, AVDD28_RF2, AVDD28_TCXO, and AVDD28_RFD*

## 2.7.3 Software Programming

The RFSYS transceiver operation commands are sent to control words via 3-wire serial interface, so that the transceiver works at the defined state and setting, which are described in Section 2.7.3.

### 2.7.3.1 Control Word Registers Description

Several control word registers are used for normal transceiver operation mode and SX frequency plan as listed below. These control word registers can be set automatically to default values after software or hardware Power-on-Reset.

| Reg | Bit(s) | Name | Description |
|---|---|---|---|
| 00h | 19 | SOR | **Software Register Rest**<br>0 = Disable (default)<br>1 = Enable<br>Note(s) :<br>1. Used to restore the default values of registers. |
| 01h | 17:10 | N_INT[7:0] | **Integer Part of N Counter**<br>Note(s) :<br>1. Used to set output frequency of SX |
| 01h<br>02h | 9:0<br>12:0 | N_FRAC[9:0]<br>N_FRAC[22:10] | **Fractional Part of N Counter**<br>Note(s) :<br>1. Used to set output frequency of SX |

| Reg | Bit(s) | Name | Description |
|---|---|---|---|
| 02h | 19:18 | TRX[1:0] | **Mode Selection between TX and RX**<br>00 = RX (default)<br>01 = TX<br>1X = Reserved<br>Note(s) :<br>1. Used as a flag before TX or RX mode. |
| 02h | 17:16 | BAND[1:0] | **Band Selection**<br>00 = GSM 850 band<br>01 = EGSM 900 band (default)<br>10 = DCS 1800 band<br>11 = PCS 1900 band |
| 02h | 15:13 | MODE[2:0] | **Mode Selection**<br>000 = Sleep mode (default)<br>010 = Standby mode<br>001 = Warmup mode<br>011 = RX mode<br>100 = TX mode |
| 60h | 5:0 | RX_GC[5:0] | **RX PGA Gain Setting**<br>111111 = Maximum gain (default)<br>Note(s) :<br>1. Sending CW96 is to enter RX mode, too. |
| 39h | 12:0 | CAFC[12:0] | **DCXO Fine Frequency Adjustment for Digital AFC**<br>0000000000000 = Lowest frequency<br>1100100000000 = Midscale frequency (default)<br>1111111111111 = Highest frequency |

**Table 32** Description of Control Registers and Power-On-Reset Values of MT6252 RFSYS 3wire tablecontrol registers

### 2.7.3.2 Operation Control System Descriptions

There are seven operation states in the RFSYS, which are reset, deep sleep, sleep, stand-by, warm-up, receive and transmit state. The detailed descriptions are as follows.

**Reset States** (Power-on Reset)

To ensure the RFSYS in a known state after power-on, a power-on reset circuit is included. In this state, the supply voltage should be ready for RFSYS. Then, RFSYS will do power-on reset to ensure the known state. There are two reset methods for RFSYS: Power-on-Reset by the internal hardware power-on reset circuit. Software can be reset by setting the CW0:SOR = 1

The effect of the reset is to load default values to all the RFSYS control bits.

**Deep Sleep State**

In the Deep Sleep State, the RFSYS circuit blocks are inactive, including the 26 MHz DCXO.

**Sleep State**

In the Sleep State, the RFSYS circuit blocks are inactive, with the exception of DCXO. Programming [MODE] = [000] of CW2 will cause the RFSYS to enter the Sleep State. The Sleep State is the default state after either a power-on reset or a software reset.

**Stand-by State**

In the Standby State, Sx related blocks will be active. Programming [MODE] = [010] of CW2 will cause the RFSYS to enter the Standby State.

**Warm-up State**

In the Warm-up State, all circuit blocks except the Rx and Tx are active. Programming CW1 or programming [MODE]=[001] will cause the RFSYS to enter the Warm-up State. Entering the warm-up mode begins Sx calibrations before Tx and Rx slots.

**Rx State**

In the Rx State only the Tx circuit blocks are inactive. Programming [MODE] = [011] of CW2 or sending CW96 will cause the RFSYS to enter the Rx State..

**Tx State**

In the Tx State, only the Rx circuit blocks are inactive. Programming [MODE] = [100] of CW2 or sending CW133 will cause the RFSYS to enter the Tx State.

### 2.7.3.3 Synthesizer Frequency Programming

This section describes the synthesizer frequency planning in RX and TX modes.

The channel frequency ranges of Rx mode and TX mode are shown below:

| | |
|---|---|
| RX-GSM850 | 869 MHz ~ 894 MHz |
| RX-EGSM900 | 925 MHz ~ 960 MHz |
| RX-DCS1800 | 1805 MHz ~ 1880 MHz |
| RX-PCS1900 | 1930 MHz ~ 1990 MHz |
| TX-GSM850 | 824 MHz ~ 849 MHz |
| TX-EGSM900 | 880 MHz ~ 915 MHz |
| TX-DCS1800 | 1710 MHz ~ 1785 MHz |
| TX-PCS1900 | 1850 MHz ~ 1910 MHz |

The SX divider number N is decided by the following procedure in the RX/TX mode.

1. Calculate LO frequency $f_{VCO}$ from RX/TX channel frequency $f_{CH}$
   $f_{VCO} = 4 * f_{CH}$ for GSM850 and EGSM900
   $f_{VCO} = 2 * f_{CH}$ for DCS1800 and PCS1900
2. Calculate $N_{int}$ and $N_{frac}$.
   $N = f_{VCO}/26M = N\_INT + N\_FRAC/2^{23}$ $64 \le N\_INT \le 255$, $0 \le N\_FRAC < 2^{23}-1$
3. N_INT and N_FRAC are integers, which use the binary equivalents of N_INT and N_FRAC to program registers CW1:N_INT[7:0], CW1:N_FRAC[9:0] and CW2:N_FRAC[22:10].

#### 2.7.3.4 RFSYS Power-on Sequence

##### 2.7.3.4.1 BSI Programming of TX Burst

1. Initialization of RFSYS control registers
2. Stand-by (CW2/CW57, BSI)
   Set CW2 TRX[1:0]=01, MODE[2:0]=010, BAND[1:0] (00: GSM850, 01:EGSM900, 10:DCS, 11:PCS) and N_FRAC based on channel information
   Set CW57 XO_CAFC[12:0]
3. Warm-up (CW1, BSI)
   Set CW1 N_INT, N_FRAC based on channel information
4. TX-mode (CW133, BSI)
5. Sleep-mode (CW2, BSI)

##### 2.7.3.4.2 BSI Programming of RX Burst

1. Initialization of RFSYS control registers
2. Stand-by (CW2/CW57, BSI)
   Set CW2 TRX[1:0]=00, MODE[2:0]=010, BAND[1:0] (00: GSM850, 01:EGSM900, 10:DCS, 11:PCS) and N_FRAC based on channel information
   Set CW57 XO_CAFC[12:0]
3. Warm-up (CW1, BSI)

Set CW1 N_INT, N_FRAC based on channel information

4. RX-mode (CW96, BSI)
   Set RX_GC[5:0] based on RX gain setting

6. Sleep-mode (CW2, BSI)



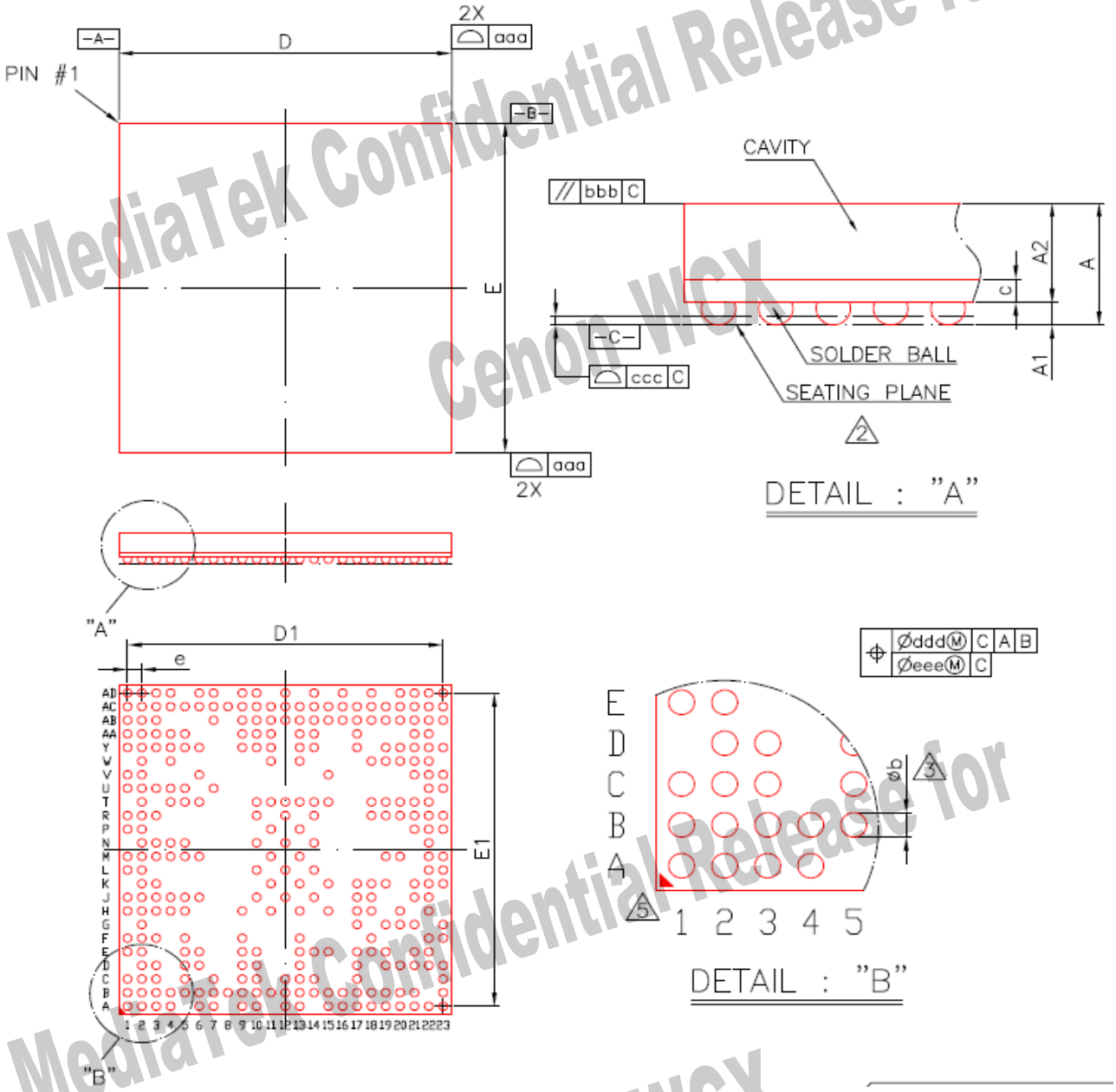
MediaTek Confidential Release for Cenon WCX

## 2.8 Package Information

### 2.8.1 Package Outline



MediaTek Confidential Release for Cenon WCX



| Symbol | Dimension in mm | | | Dimension in inch | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | --- | --- | 1.20 | --- | --- | 0.047 |
| A1 | 0.16 | 0.21 | 0.26 | 0.006 | 0.008 | 0.010 |
| A2 | 0.86 | 0.91 | 0.96 | 0.034 | 0.036 | 0.038 |
| c | 0.17 | 0.21 | 0.25 | 0.007 | 0.008 | 0.010 |
| D | 11.50 | 11.60 | 11.70 | 0.453 | 0.457 | 0.461 |
| E | 12.00 | 12.10 | 12.20 | 0.472 | 0.476 | 0.480 |
| D1 | --- | 11.00 | --- | --- | 0.433 | --- |
| E1 | --- | 11.50 | --- | --- | 0.453 | --- |
| e | --- | 0.50 | --- | --- | 0.020 | --- |
| b | 0.25 | 0.30 | 0.35 | 0.010 | 0.012 | 0.014 |
| aaa | 0.10 | | | 0.004 | | |
| bbb | 0.10 | | | 0.004 | | |
| ccc | 0.08 | | | 0.003 | | |
| ddd | 0.15 | | | 0.006 | | |
| eee | 0.05 | | | 0.002 | | |
| MD/ME | 23/24 | | | 23/24 | | |

NOTE :

1. CONTROLLING DIMENSION : MILLIMETER.
2. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
3. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
4. SPECIAL CHARACTERISTICS C CLASS: bbb, ccc
5. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY.

**Figure 19** Outlines and Dimension of MT6252 TFBGA 11.6mm * 12.1mm 0.5mm pitch package

| Body Size | Ball Count | Ball Pitch | Ball Dia. | Package Thk. |
|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 11.6 | 12.1 | 305 | 0.5 | 0.3 | 1.2 |

**Table 33** MT6252 definition of TFBGA 11.6 x 12.1 mm 0.5 mm pitch package (unit: mm)

### 2.8.2 Thermal Operating Specifications

| Symbol | Description | Value | Unit | Notes |
|---|---|---|---|---|
| $\Theta_{JA}$ | Thermal resistance from device junction to package case | 27.67 | ‘C/W | |

### 2.8.3 Lead-free Packaging

MT6252 is provided in a lead-free package which meets RoHS requirements.

## 2.9 Ordering Information

### 2.9.1 Top Marking Definition

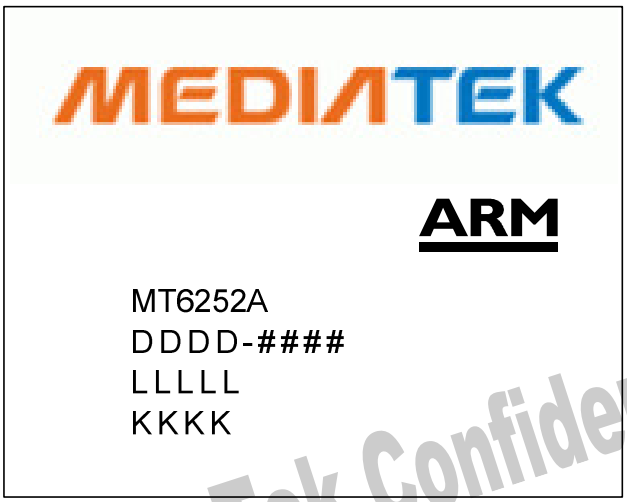



**Figure 20** Top marking of MT6252

# 3 Micro-Controller Unit Subsystem

**Figure 21** illustrates the block diagram of the Micro-Controller Unit Subsystem in MT6252. The subsystem utilizes a main 32-bit ARM7EJ-S RISC processor, which plays the role of the main bus master controlling the whole subsystem. All processor transactions go to code cache first. The code cache controller accesses TCM (memory dedicated to ARM7EJS core), cache memory, or bus according to the processor's request address. If the requested content is found in TCM or in cache, no bus transaction is required. If the code cache hit rate is high enough, bus traffic can be effectively reduced and processor core performance maximized.

The bus comprises of two-level system buses: Advanced High-Performance Bus (AHB) and Advanced Peripheral Bus (APB). All bus transactions originate from bus masters, while slaves can only respond to requests from bus masters. Before data transfer can be established, the bus master must ask for bus ownership, accomplished by request-grant handshaking protocol between masters and arbiters.

Two levels of bus hierarchy are designed to provide optimum usage for different performance requirements. Specifically, AHB Bus, the main system bus, is tailored toward high-speed requirements and provides 32-bit data path with multiplex scheme for bus interconnections. The APB Bus, on the other hand, is designed to reduce interface complexity for lower data transfer rate, and so it is isolated from high bandwidth AHB Bus by APB Bridge. APB Bus supports 16-bit addressing and both 16-bit and 32-bit data paths. APB Bus is also optimized for minimal power consumption by turning off the clock when there is no APB bus activity.

During operation, if the target slave is located on AHB Bus, the transaction is conducted directly on AHB Bus. However, if the target slave is a peripheral and is attached to the APB bus, then the transaction is conducted between AHB and APB bus through the use of APB Bridge.

In order to off-load the processor core, a DMA Controller is designated to act as a master and share the bus resources on AHB Bus to do fast data movement between modules. This controller comprises thirteen DMA channels.

The Interrupt Controller provides a software interface to manipulate interrupt events. It can handle up to 32 interrupt sources asserted at the same time. In general, it generates 2 levels of interrupt requests, FIQ and IRQ, to the processor.

A SRAM is provided for acting as system memory for high-speed data access. For factory programming purpose, a Boot ROM module is used. These two modules use the same Internal Memory Controller to connect to AHB Bus.

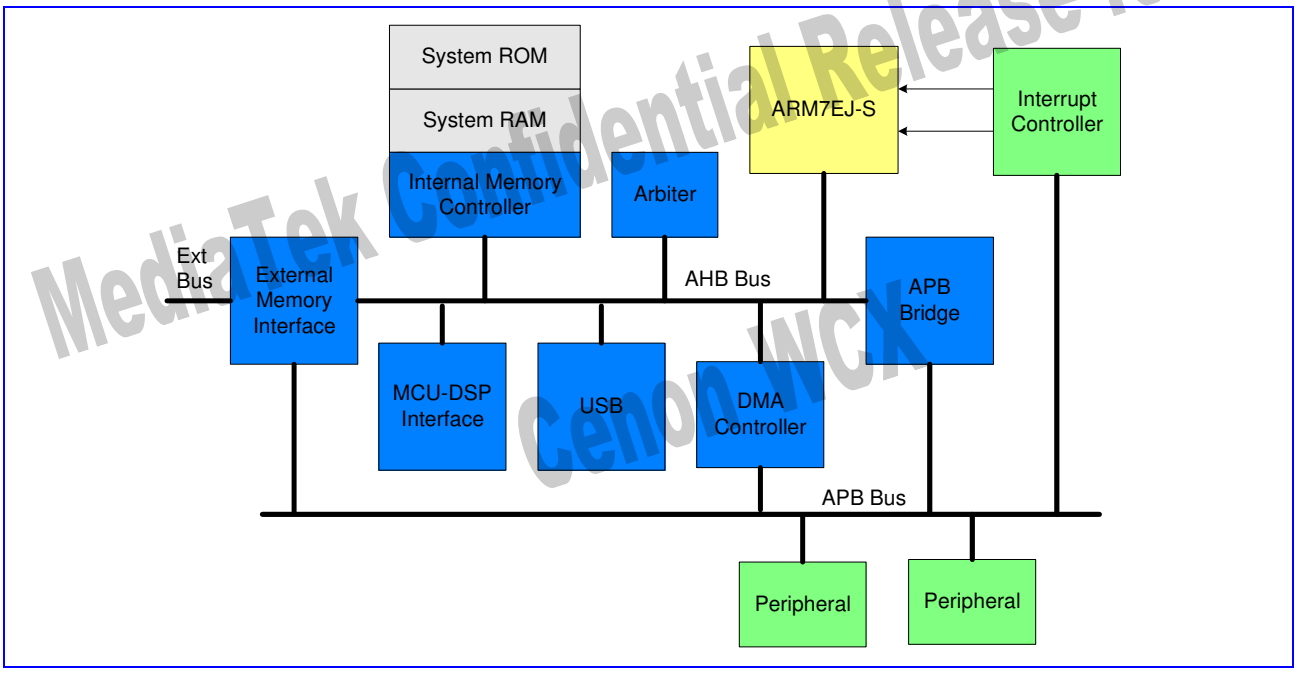



**Figure 21** Block Diagram of the Micro-Controller Unit Subsystem in MT6252

# 3.1 Direct Memory Access

## 3.1.1 General Description

A generic DMA Controller is placed on Layer 2 AHB Bus to support fast data transfers and to off-load the processor. With this controller, specific devices on AHB or APB buses can benefit greatly from quick completion of data movement from or to memory modules such as Internal System RAM or External SRAM, excluding TCM. TCM is invisible for DMA engine. Such Generic DMA Controller can also be used to connect any two devices other than memory module as long as they can be addressed in memory space. **Figure 22** illustrates the system connections.

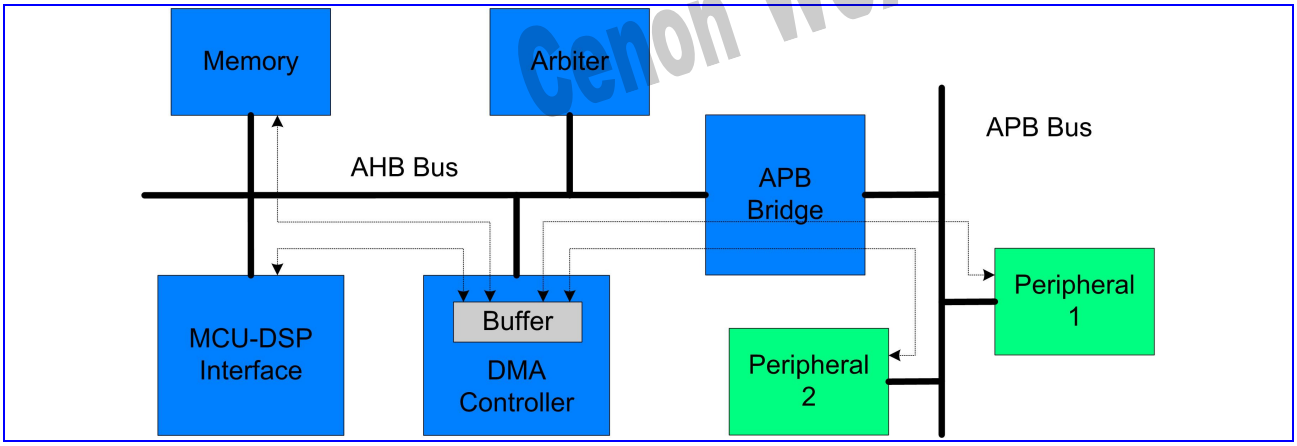



**Figure 22** Variety Data Paths of DMA Transfers

Up to fourteen channels of simultaneous data transfers are supported. Each channel has a similar set of registers to be configured to different scheme as desired. If more than fourteen devices are requesting the DMA resources at the same time, software based arbitration should be employed. Once the service candidate is decided, the responsible device driver should configure the Generic DMA Controller properly in order to conduct DMA transfers. Both Interrupt and Polling based schemes in handling the completion event are supported. The block diagram of such generic DMA Controller is illustrated in **Figure 23**

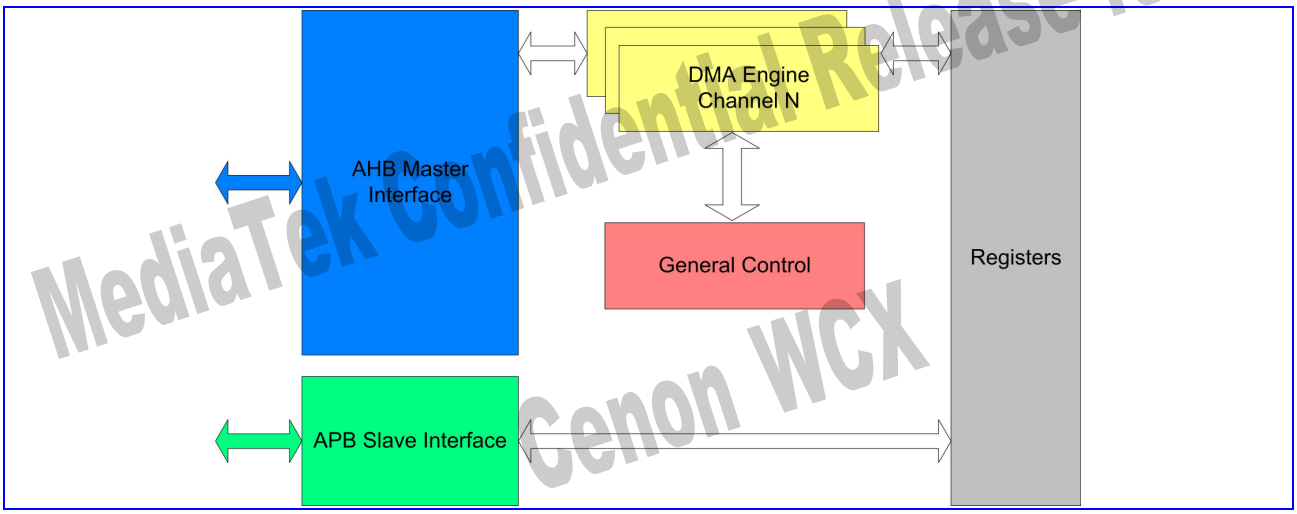



**Figure 23** DMA Block Diagram

### 3.1.1.1 Full-Size & Half-Size DMA Channels

There are three types of DMA channels in the DMA controller. The first one is called a full-size DMA channel, the second one is called a half-size DMA channel, and the last is Virtual FIFO DMA. Channel 1 is full-size DMA channel; channels 2 through 9 are half-size ones; and channels 10 through 15 are Virtual FIFO DMA channels. The difference between the first two types of DMA channels is that both source and destination address are programmable in full-size DMA channels, but only the address of one side can be programmed in half-size DMA channel. In half-size channels, only either the source or destination address can be programmed, while the addresses of the other side are preset. Which preset address is used depends on the setting of MAS in DMA Channel Control Register. Refer to the Register Definition section for more detail.

### 3.1.1.2 Ring Buffer & Double Buffer Memory Data Movement

DMA channels 1 through 9 support ring-buffer and double-buffer memory data movement. This can be achieved by programming DMA_WPPT and DMA_WPTO, as well as setting WPEN in DMA_CON register to enable. **Figure 24** illustrates how this function works. Once the transfer counter reaches WPPT, the next address will jump to WPTO address after completing the WPPT data transfer. Note that only one side can be configured as ring-buffer or double-buffer memory, and this is controlled by WPSD in DMA_CON register.

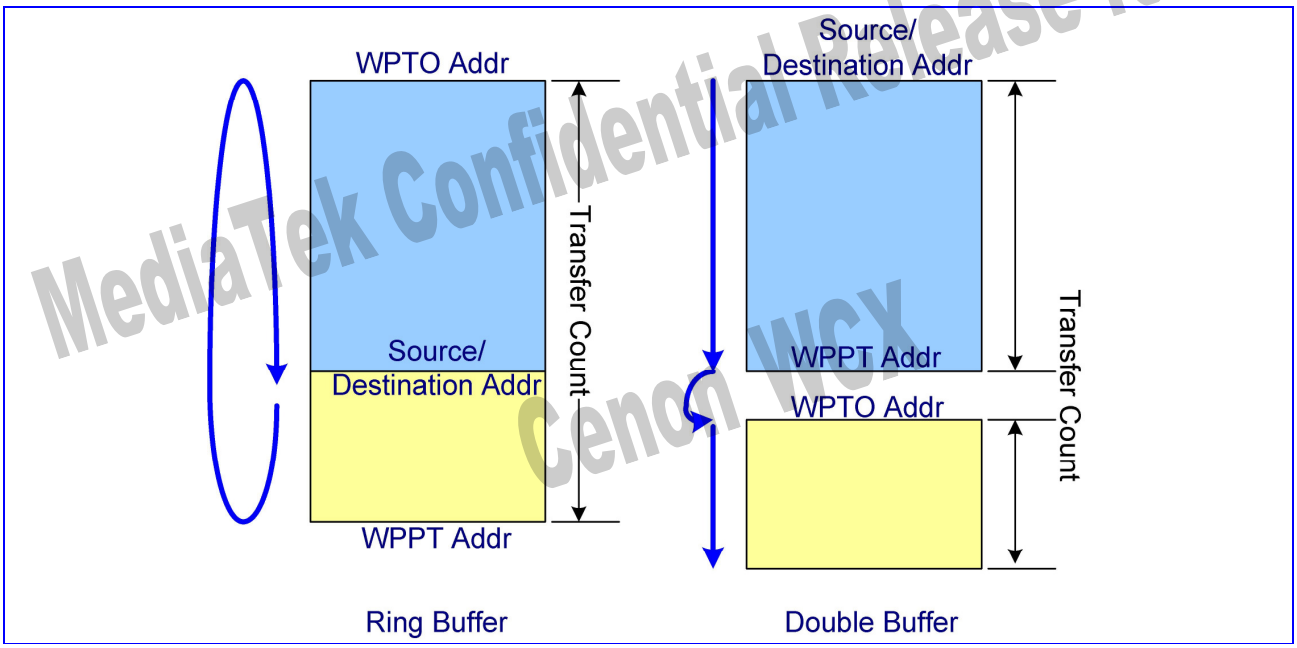



**Figure 24** Ring Buffer and Double Buffer Memory Data Movement

### 3.1.1.3 Unaligned Word Access

The address of word access on AHB bus must be aligned to word boundary, or the 2 LSB is truncated to 00b. If programmers do not notice this, it may cause an incorrect data fetch. In the case where data is to be moved from unaligned addresses to aligned addresses, the word is usually first split into four bytes and then moved byte by byte. Thus four read and four write transfers will be appeared on the bus.

To improve bus efficiency, unaligned-word access is provided in DMA2~9. While this function is enabled, DMAs move data from unaligned address to aligned address by executing four continuous byte-read access and one word-write access, reducing the number of transfers on the bus by three.

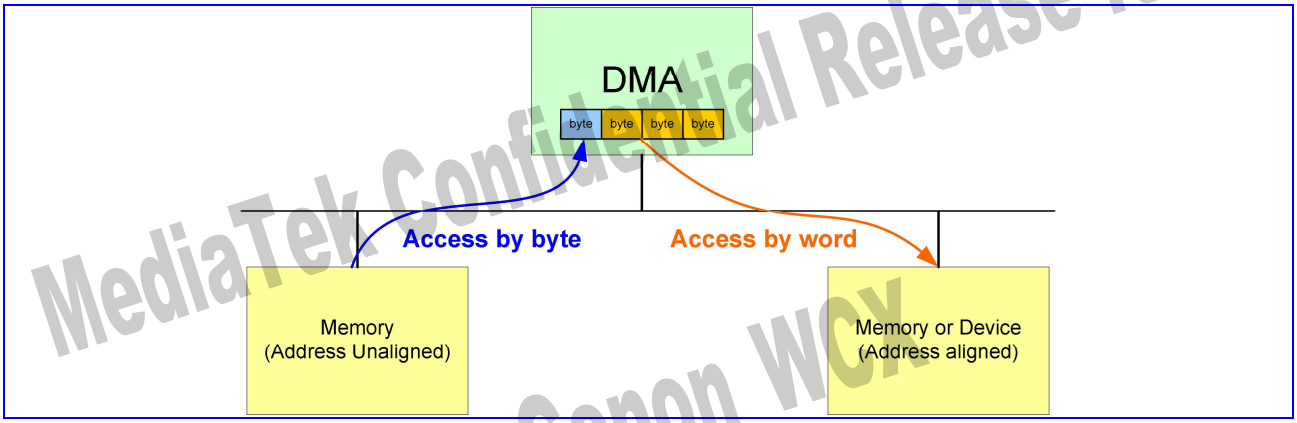



**Figure 25** Unaligned Word Accesses

### 3.1.1.4 Virtual FIFO DMA

Virtual FIFO DMA is used to ease UART control. The difference between the Virtual FIFO DMAs and the ordinary DMAs is that Virtual FIFO DMA contains additional FIFO controller. The read and write pointers are kept in the Virtual FIFO DMA. During a read from the FIFO, the read pointer points to the address of the next data. During a write to the FIFO, the write pointer moves to the next address. If the FIFO is empty, a FIFO read is not allowed. Similarly, data is not written into the FIFO if the FIFO is full. Due to UART flow control requirements, an alert length is programmed. Once the FIFO Space is less than this value, an alert signal is issued to enable UART flow control. The type of flow control performed depends on the setting in UART.

Each Virtual FIFO DMA can be programmed as RX or TX FIFO. This depends on the setting of DIR in DMA_CON register. If DIR is “0” (READ), it means TX FIFO. On the other hand, if DIR is “1” (WRITE), the Virtual FIFO DMA is specified as a RX FIFO.

Virtual FIFO DMA provides an interrupt to MCU. This interrupt informs MCU that there is data in the FIFO, and the amount of data is over or under the value defined in DMA_COUNT register. With this, MCU does not need to poll DMA to know when data must be removed from or put into the FIFO.

Note that Virtual FIFO DMAs cannot be used as generic DMAs, i.e. DMA1~9.

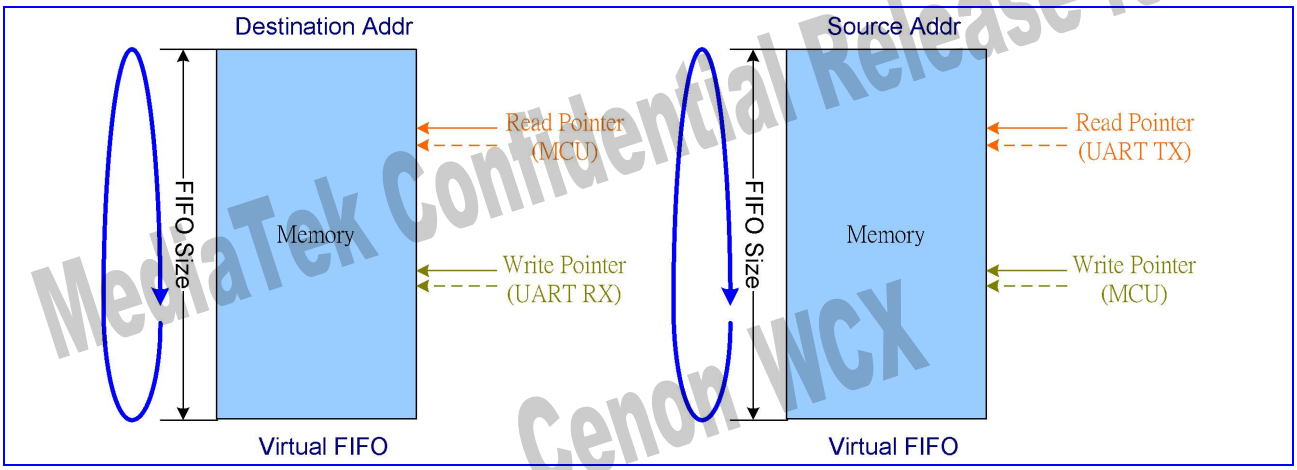



**Figure 26** Virtual FIFO DMA

| DMA number | Address of Virtual FIFO Access Port | Reference UART |
|---|---|---|
| DMA9 | 7800_0000h | UART1 RX |
| DMA10 | 7800_0100h | UART2 RX |
| DMA11 | 7800_0200h | UART 3 RX |
| DMA12 | 7800_0300h | UART1 TX |
| DMA13 | 7800_0400h | UART2 TX |
| DMA14 | 7800_0500h | UART3 TX |

**Table 34** Virtual FIFO Access Port

| DMA number | Type | Ring Buffer | Double Buffer | Burst Mode | Unaligned Word Access |
|---|---|---|---|---|---|
| DMA1 | Full Size | ● | ● | ● | |
| DMA2 | Half Size | ● | ● | ● | ● |

| | | | | | |
|---|---|---|---|---|---|
| DMA3 | Half Size | ● | ● | ● | ● |
| DMA4 | Half Size | ● | ● | ● | ● |
| DMA5 | Half Size | ● | ● | ● | ● |
| DMA6 | Half Size | ● | ● | ● | ● |
| DMA7 | Half Size | ● | ● | ● | ● |
| DMA8 | Half Size | ● | ● | ● | ● |
| DMA9 | Half Size | ● | ● | ● | ● |
| DMA10 | Virtual FIFO | ● | | | |
| DMA11 | Virtual FIFO | ● | | | |
| DMA12 | Virtual FIFO | ● | | | |
| DMA13 | Virtual FIFO | ● | | | |
| DMA14 | Virtual FIFO | ● | | | |
| DMA15 | Virtual FIFO | ● | | | |

**Table 35** Function List of DMA channels

| Register Address | Register Function | Acronym |
|---|---|---|
| 0x8102_0000h | DMA Global Status Register | **DMA_GLBSTA** |
| 0x8102_0028h | DMA Global Bandwidth Limiter Register | **DMA_GLBLIMITER** |
| 0x8102_0100h | DMA Channel 1 Source Address Register | **DMA1_SRC** |

| | | |
|---|---|---|
| 0x8102_0104h | DMA Channel 1 Destination Address Register | **DMA1_DST** |
| 0x8102_0108h | DMA Channel 1 Wrap Point Address Register | **DMA1_WPPT** |
| 0x8102_010Ch | DMA Channel 1 Wrap To Address Register | **DMA1_WPTO** |
| 0x8102_0110h | DMA Channel 1 Transfer Count Register | **DMA1_COUNT** |
| 0x8102_0114h | DMA Channel 1 Control Register | **DMA1_CON** |
| 0x8102_0118h | DMA Channel 1 Start Register | **DMA1_START** |
| 0x8102_011Ch | DMA Channel 1 Interrupt Status Register | **DMA1_INTSTA** |
| 0x8102_0120h | DMA Channel 1 Interrupt Acknowledge Register | **DMA1_ACKINT** |
| 0x8102_0124h | DMA Channel 1 Remaining Length of Current Transfer | **DMA1_RLCT** |
| 0x8102_0128h | DMA Channel 1 Bandwidth Limiter Register | **DMA1_LIMITER** |
| 0x8102_0208h | DMA Channel 2 Wrap Point Address Register | **DMA2_WPPT** |
| 0x8102_020Ch | DMA Channel 2 Wrap To Address Register | **DMA2_WPTO** |
| 0x8102_0210h | DMA Channel 2 Transfer Count Register | **DMA2_COUNT** |
| 0x8102_0214h | DMA Channel 2 Control Register | **DMA2_CON** |
| 0x8102_0218h | DMA Channel 2 Start Register | **DMA2_START** |
| 0x8102_021Ch | DMA Channel 2 Interrupt Status Register | **DMA2_INTSTA** |
| 0x8102_0220h | DMA Channel 2 Interrupt Acknowledge Register | **DMA2_ACKINT** |
| 0x8102_0224h | DMA Channel 2 Remaining Length of Current Transfer | **DMA2_RLCT** |
| 0x8102_0228h | DMA Channel 2 Bandwidth Limiter Register | **DMA2_LIMITER** |
| 0x8102_022Ch | DMA Channel 2 Programmable Address Register | **DMA2_PGMADDR** |
| 0x8102_0308h | DMA Channel 3 Wrap Point Address Register | **DMA3_WPPT** |
| 0x8102_030Ch | DMA Channel 3 Wrap To Address Register | **DMA3_WPTO** |
| 0x8102_0310h | DMA Channel 3 Transfer Count Register | **DMA3_COUNT** |
| 0x8102_0314h | DMA Channel 3 Control Register | **DMA3_CON** |

| | | |
|---|---|---|
| 0x8102_0318h | DMA Channel 3 Start Register | **DMA3_START** |
| 0x8102_031Ch | DMA Channel 3 Interrupt Status Register | **DMA3_INTSTA** |
| 0x8102_0320h | DMA Channel 3 Interrupt Acknowledge Register | **DMA3_ACKINT** |
| 0x8102_0324h | DMA Channel 3 Remaining Length of Current Transfer | **DMA3_RLCT** |
| 0x8102_0328h | DMA Channel 3 Bandwidth Limiter Register | **DMA3_LIMITER** |
| 0x8102_032Ch | DMA Channel 3 Programmable Address Register | **DMA3_PGMADDR** |
| 0x8102_0408h | DMA Channel 4 Wrap Point Address Register | **DMA4_WPPT** |
| 0x8102_040Ch | DMA Channel 4 Wrap To Address Register | **DMA4_WPTO** |
| 0x8102_0410h | DMA Channel 4 Transfer Count Register | **DMA4_COUNT** |
| 0x8102_0414h | DMA Channel 4 Control Register | **DMA4_CON** |
| 0x8102_0418h | DMA Channel 4 Start Register | **DMA4_START** |
| 0x8102_041Ch | DMA Channel 4 Interrupt Status Register | **DMA4_INTSTA** |
| 0x8102_0420h | DMA Channel 4 Interrupt Acknowledge Register | **DMA4_ACKINT** |
| 0x8102_0424h | DMA Channel 4 Remaining Length of Current Transfer | **DMA4_RLCT** |
| 0x8102_0428h | DMA Channel 4 Bandwidth Limiter Register | **DMA4_LIMITER** |
| 0x8102_042Ch | DMA Channel 4 Programmable Address Register | **DMA4_PGMADDR** |
| 0x8102_0508h | DMA Channel 5 Wrap Point Address Register | **DMA5_WPPT** |
| 0x8102_050Ch | DMA Channel 5 Wrap To Address Register | **DMA5_WPTO** |
| 0x8102_0510h | DMA Channel 5 Transfer Count Register | **DMA5_COUNT** |
| 0x8102_0514h | DMA Channel 5 Control Register | **DMA5_CON** |
| 0x8102_0518h | DMA Channel 5 Start Register | **DMA5_START** |
| 0x8102_051Ch | DMA Channel 5 Interrupt Status Register | **DMA5_INTSTA** |
| 0x8102_0520h | DMA Channel 5 Interrupt Acknowledge Register | **DMA5_ACKINT** |
| 0x8102_0524h | DMA Channel 5 Remaining Length of Current Transfer | **DMA5_RLCT** |

| | | |
|---|---|---|
| 0x8102_0528h | DMA Channel 5 Bandwidth Limiter Register | **DMA5_LIMITER** |
| 0x8102_052Ch | DMA Channel 5 Programmable Address Register | **DMA5_PGMADDR** |
| 0x8102_0608h | DMA Channel 6 Wrap Point Address Register | **DMA6_WPPT** |
| 0x8102_060Ch | DMA Channel 6 Wrap To Address Register | **DMA6_WPTO** |
| 0x8102_0610h | DMA Channel 6 Transfer Count Register | **DMA6_COUNT** |
| 0x8102_0614h | DMA Channel 6 Control Register | **DMA6_CON** |
| 0x8102_0618h | DMA Channel 6 Start Register | **DMA6_START** |
| 0x8102_061Ch | DMA Channel 6 Interrupt Status Register | **DMA6_INTSTA** |
| 0x8102_0620h | DMA Channel 6 Interrupt Acknowledge Register | **DMA6_ACKINT** |
| 0x8102_0624h | DMA Channel 6 Remaining Length of Current Transfer | **DMA6_RLCT** |
| 0x8102_0628h | DMA Channel 6 Bandwidth Limiter Register | **DMA6_LIMITER** |
| 0x8102_062Ch | DMA Channel 6 Programmable Address Register | **DMA6_PGMADDR** |
| 0x8102_0708h | DMA Channel 7 Wrap Point Address Register | **DMA7_WPPT** |
| 0x8102_070Ch | DMA Channel 7 Wrap To Address Register | **DMA7_WPTO** |
| 0x8102_0710h | DMA Channel 7 Transfer Count Register | **DMA7_COUNT** |
| 0x8102_0714h | DMA Channel 7 Control Register | **DMA7_CON** |
| 0x8102_0718h | DMA Channel 7 Start Register | **DMA7_START** |
| 0x8102_071Ch | DMA Channel 7 Interrupt Status Register | **DMA7_INTSTA** |
| 0x8102_0720h | DMA Channel 7 Interrupt Acknowledge Register | **DMA7_ACKINT** |
| 0x8102_0724h | DMA Channel 7 Remaining Length of Current Transfer | **DMA7_RLCT** |
| 0x8102_0728h | DMA Channel 7 Bandwidth Limiter Register | **DMA7_LIMITER** |
| 0x8102_072Ch | DMA Channel 7 Programmable Address Register | **DMA7_PGMADDR** |
| 0x8102_0808h | DMA Channel 8 Wrap Point Address Register | **DMA8_WPPT** |
| 0x8102_080Ch | DMA Channel 8 Wrap To Address Register | **DMA8_WPTO** |

| | | |
|---|---|---|
| 0x8102_0810h | DMA Channel 8 Transfer Count Register | **DMA8_COUNT** |
| 0x8102_0814h | DMA Channel 8 Control Register | **DMA8_CON** |
| 0x8102_0818h | DMA Channel 8 Start Register | **DMA8_START** |
| 0x8102_081Ch | DMA Channel 8 Interrupt Status Register | **DMA8_INTSTA** |
| 0x8102_0820h | DMA Channel 8 Interrupt Acknowledge Register | **DMA8_ACKINT** |
| 0x8102_0824h | DMA Channel 8 Remaining Length of Current Transfer | **DMA8_RLCT** |
| 0x8102_0828h | DMA Channel 8 Bandwidth Limiter Register | **DMA8_LIMITER** |
| 0x8102_082Ch | DMA Channel 8 Programmable Address Register | **DMA8_PGMADDR** |
| 0x8102_0908h | DMA Channel 9 Wrap Point Address Register | **DMA9_WPPT** |
| 0x8102_090Ch | DMA Channel 9 Wrap To Address Register | **DMA9_WPTO** |
| 0x8102_0910h | DMA Channel 9 Transfer Count Register | **DMA9_COUNT** |
| 0x8102_0914h | DMA Channel 9 Control Register | **DMA9_CON** |
| 0x8102_0918h | DMA Channel 9 Start Register | **DMA9_START** |
| 0x8102_091Ch | DMA Channel 9 Interrupt Status Register | **DMA9_INTSTA** |
| 0x8102_0920h | DMA Channel 9 Interrupt Acknowledge Register | **DMA9_ACKINT** |
| 0x8102_0924h | DMA Channel 9 Remaining Length of Current Transfer | **DMA9_RLCT** |
| 0x8102_0928h | DMA Channel 9 Bandwidth Limiter Register | **DMA9_LIMITER** |
| 0x8102_092Ch | DMA Channel 9 Programmable Address Register | **DMA9_PGMADDR** |
| 0x8102_0C10h | DMA Channel 10 Transfer Count Register | **DMA10_COUNT** |
| 0x8102_0A14h | DMA Channel 10 Control Register | **DNA10_CON** |
| 0x8102_0A18h | DMA Channel 10 Start Register | **DNA10_START** |
| 0x8102_0A1Ch | DMA Channel 10 Interrupt Status Register | **DNA10_INTSTA** |
| 0x8102_0A20h | DMA Channel 10 Interrupt Acknowledge Register | **DNA10_ACKINT** |
| 0x8102_0A28h | DMA Channel 10 Bandwidth Limiter Register | **DNA10_LIMITER** |

| | | |
|---|---|---|
| 0x8102_0A2Ch | DMA Channel 10 Programmable Address Register | **DNA10_PGMADDR** |
| 0x8102_0A30h | DMA Channel 10 Write Pointer | **DNA10_WRPTR** |
| 0x8102_0A34h | DMA Channel 10 Read Pointer | **DNA10_RDPTR** |
| 0x8102_0A38h | DMA Channel 10 FIFO Count | **DNA10_FFCNT** |
| 0x8102_0A3Ch | DMA Channel 10 FIFO Status | **DNA10_FFSTA** |
| 0x8102_0A40h | DMA Channel 10 Alert Length | **DNA10_ALTLEN** |
| 0x8102_0A44h | DMA Channel 10 FIFO Size | **DNA10_FFSIZE** |
| 0x8102_0B10h | DMA Channel 11 Transfer Count Register | **DMA11_COUNT** |
| 0x8102_0B14h | DMA Channel 11 Control Register | **DMA11_CON** |
| 0x8102_0B18h | DMA Channel 11 Start Register | **DMA11_START** |
| 0x8102_0B1Ch | DMA Channel 11 Interrupt Status Register | **DMA11_INTSTA** |
| 0x8102_0B20h | DMA Channel 11 Interrupt Acknowledge Register | **DMA11_ACKINT** |
| 0x8102_0B28h | DMA Channel 11 Bandwidth Limiter Register | **DMA11_LIMITER** |
| 0x8102_0B2Ch | DMA Channel 11 Programmable Address Register | **DMA11_PGMADDR** |
| 0x8102_0B30h | DMA Channel 11 Write Pointer | **DMA11_WRPTR** |
| 0x8102_0B34h | DMA Channel 11 Read Pointer | **DMA11_RDPTR** |
| 0x8102_0B38h | DMA Channel 11 FIFO Count | **DMA11_FFCNT** |
| 0x8102_0B3Ch | DMA Channel 11 FIFO Status | **DMA11_FFSTA** |
| 0x8102_0B40h | DMA Channel 11 Alert Length | **DMA11_ALTLEN** |
| 0x8102_0B44h | DMA Channel 11 FIFO Size | **DMA11_FFSIZE** |
| 0x8102_0C10h | DMA Channel 12 Transfer Count Register | **DMA12_COUNT** |
| 0x8102_0C14h | DMA Channel 12 Control Register | **DMA12_CON** |
| 0x8102_0C18h | DMA Channel 12 Start Register | **DMA12_START** |
| 0x8102_0C1Ch | DMA Channel 12 Interrupt Status Register | **DMA12_INTSTA** |

| | | |
|---|---|---|
| 0x8102_0C20h | DMA Channel 12 Interrupt Acknowledge Register | **DMA12_ACKINT** |
| 0x8102_0C28h | DMA Channel 12 Bandwidth Limiter Register | **DMA12_LIMITER** |
| 0x8102_0C2Ch | DMA Channel 12 Programmable Address Register | **DMA12_PGMADDR** |
| 0x8102_0C30h | DMA Channel 12 Write Pointer | **DMA12_WRPTR** |
| 0x8102_0C34h | DMA Channel 12 Read Pointer | **DMA12_RDPTR** |
| 0x8102_0C38h | DMA Channel 12 FIFO Count | **DMA12_FFCNT** |
| 0x8102_0C3Ch | DMA Channel 12 FIFO Status | **DMA12_FFSTA** |
| 0x8102_0C40h | DMA Channel 12 Alert Length | **DMA12_ALTLEN** |
| 0x8102_0C44h | DMA Channel 12 FIFO Size | **DMA12_FFSIZE** |
| 0x8102_0D10h | DMA Channel 13 Transfer Count Register | **DMA13_COUNT** |
| 0x8102_0D14h | DMA Channel 13 Control Register | **DMA13_CON** |
| 0x8102_0D18h | DMA Channel 13 Start Register | **DMA13_START** |
| 0x8102_0D1Ch | DMA Channel 13 Interrupt Status Register | **DMA13_INTSTA** |
| 0x8102_0D20h | DMA Channel 13 Interrupt Acknowledge Register | **DMA13_ACKINT** |
| 0x8102_0D28h | DMA Channel 13 Bandwidth Limiter Register | **DMA13_LIMITER** |
| 0x8102_0D2Ch | DMA Channel 13 Programmable Address Register | **DMA13_PGMADDR** |
| 0x8102_0D30h | DMA Channel 13 Write Pointer | **DMA13_WRPTR** |
| 0x8102_0D34h | DMA Channel 13 Read Pointer | **DMA13_RDPTR** |
| 0x8102_0D38h | DMA Channel 13 FIFO Count | **DMA13_FFCNT** |
| 0x8102_0D3Ch | DMA Channel 13 FIFO Status | **DMA13_FFSTA** |
| 0x8102_0D40h | DMA Channel 13 Alert Length | **DMA13_ALTLEN** |
| 0x8102_0D44h | DMA Channel 13 FIFO Size | **DMA13_FFSIZE** |
| 0x8102_0E10h | DMA Channel 14 Transfer Count Register | **DMA14_COUNT** |
| 0x8102_0E14h | DMA Channel 14 Control Register | **DMA14_CON** |

| | | |
|---|---|---|
| 0x8102_0E18h | DMA Channel 14 Start Register | **DMA14_START** |
| 0x8102_0E1Ch | DMA Channel 14 Interrupt Status Register | **DMA14_INTSTA** |
| 0x8102_0E20h | DMA Channel 14 Interrupt Acknowledge Register | **DMA14_ACKINT** |
| 0x8102_0E28h | DMA Channel 14 Bandwidth Limiter Register | **DMA14_LIMITER** |
| 0x8102_0E2Ch | DMA Channel 14 Programmable Address Register | **DMA14_PGMADDR** |
| 0x8102_0E30h | DMA Channel 14 Write Pointer | **DMA14_WRPTR** |
| 0x8102_0E34h | DMA Channel 14 Read Pointer | **DMA14_RDPTR** |
| 0x8102_0E38h | DMA Channel 14 FIFO Count | **DMA14_FFCNT** |
| 0x8102_0E3Ch | DMA Channel 14 FIFO Status | **DMA14_FFSTA** |
| 0x8102_0E40h | DMA Channel 14 Alert Length | **DMA14_ALTLEN** |
| 0x8102_0E44h | DMA Channel 14 FIFO Size | **DMA14_FFSIZE** |
| 0x8102_0F10h | DMA Channel 15 Transfer Count Register | **DMA15_COUNT** |
| 0x8102_0F14h | DMA Channel 15 Control Register | **DMA15_CON** |
| 0x8102_0F18h | DMA Channel 15 Start Register | **DMA15_START** |
| 0x8102_0F1Ch | DMA Channel 15 Interrupt Status Register | **DMA15_INTSTA** |
| 0x8102_0F20h | DMA Channel 15 Interrupt Acknowledge Register | **DMA15_ACKINT** |
| 0x8102_0F28h | DMA Channel 15 Bandwidth Limiter Register | **DMA15_LIMITER** |
| 0x8102_0F2Ch | DMA Channel 15 Programmable Address Register | **DMA15_PGMADDR** |
| 0x8102_0F30h | DMA Channel 15 Write Pointer | **DMA15_WRPTR** |
| 0x8102_0F34h | DMA Channel 15 Read Pointer | **DMA15_RDPTR** |
| 0x8102_0F38h | DMA Channel 15 FIFO Count | **DMA15_FFCNT** |
| 0x8102_0F3Ch | DMA Channel 15 FIFO Status | **DMA15_FFSTA** |
| 0x8102_0F40h | DMA Channel 15 Alert Length | **DMA15_ALTLEN** |
| 0x8102_0F44h | DMA Channel 15 FIFO Size | **DMA15_FFSIZE** |

**Table 36** DMA Registers



MediaTek Confidential Release for Cenon WCX

### 3.1.2 Register Definitions

Register programming tips:

- Start registers shall be cleared, when associated channels are being programmed.
- PGMADDR, i.e. programmable address, only exists in half-size DMA channels. If DIR in Control Register is high, PGMADDR represents Destination Address. Conversely, If DIR in Control Register is low, PGMADDR represents Source Address.
- Functions of ring-buffer and double-buffer memory data movement can be activated on either source side or destination side by programming DMA_WPPT & and DMA_WPTO, as well as setting WPEN in DMA_CON register high. WPSD in DMA_CON register determines the activated side.



**DMA+0000h DMA Global Status Register DMA_GLBSTA**

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | IT14 | RUN14 | IT13 | RUN13 | IT12 | RUN12 | IT11 | RUN11 | IT10 | RUN10 | IT9 | RUN9 |
| Type | | | | | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | | | | | 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 | IT8 | RUN8 | IT7 | RUN7 | IT6 | RUN6 | IT5 | RUN5 | IT4 | RUN4 | IT3 | RUN3 | IT2 | RUN2 | IT1 | RUN1 |
| Type | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register helps software program keep track of the global status of DMA channels.

**$RUN_N$** DMA channel n status
- **0** Channel n is stopped or has completed the transfer already.
- **1** Channel n is currently running.

**$IT_N$** Interrupt status for channel n
- **0** No interrupt is generated.
- **1** An interrupt is pending and waiting for service.

**DMA+0028h DMA Global Bandwidth limiter Register DMA_GLBLIMITER**

| 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 | | | | | | | | | GLBLIMITER | | | | | | | |
| Type | | | | | | | | | WO | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

Please refer to the expression in DMAn_LIMITER for detailed note. The value of DMA_GLBLIMITER is set to all DMA channels, from 1 to 15.



## DMA+0n00h DMA Channel n Source Address Register DMAn_SRC

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

The above registers contain the base or current source address that the DMA channel is currently operating on. Writing to this register specifies the base address of transfer source for a DMA channel. Before programming these registers, the software program should make sure that STR in DMAn_START is set to 0; that is, the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. Reading this register returns the address value from which the DMA is reading.

Note that n is 1 and SRC can't be TCM address. TCM is not accessible by DMA..

**SRC** **SRC**[31:0] specifies the base or current address of transfer source for a DMA channel, i.e. channel 1.

**WRITE** Base address of transfer source

**READ** Address from which DMA is reading



## DMA+0n04h DMA Channel n Destination Address Register DMAn_DST

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

| Reset | 0 |
|---|---|

The above registers contain the base or current destination address that the DMA channel is currently operating on.. Writing to this register specifies the base address of the transfer destination for a DMA channel. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0'; that is, the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. Reading this register returns the address value to which the DMA is writing.

Note that n is 1 and DST can't be TCM address. TCM is not accessible by DMA.

**DST** **DST**[31:0] specifies the base or current address of transfer destination for a DMA channel, i.e. channel 1

**WRITE** Base address of transfer destination.

**READ** Address to which DMA is writing.



## DMA+0n08h DMA Channel n Wrap Point Count Register DMAn_WPPT

| 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 | WPPT[15:0] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

The above registers are to specify the transfer count required to perform before the jump point. This can be used to support ring buffer or double buffer style memory accesses. To enable this function, two control bits, WPEN and WPSD, in DMA control register must be programmed. See the following register description for more details. If the transfercounter in the DMA engine matches this value, an address jump occurs, and the next address is the address specified in DMAn_WPTO. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. To enable this function, WPEN in DMA_CON is set. Note that the total size of data specify in the wrap point count in a DMA channel is determined by LEN together with the SIZE in DMAn_CON, i.e. WPPT x SIZE.

Note that n is from 1 to 9.

**WPPT** **WPPT**[15:0] specifies the amount of the transfer count from start to jumping point for a DMA channel, i.e. channel 1 – 9.
**WRITE** Wrap point transfer count.
**READ** Value set by the programmer.

## DMA+0n0Ch DMA Channel n Wrap To Address Register DMAn_WPTO

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

The above registers specify the address of the jump destination of a given DMA transfer to support ring buffer or double buffer style memory accesses. To enable this function, set the two control bits, WPEN and WPSD, in the DMA control register . See the following register description for more details. Before programming these registers, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order. To enable this function, WPEN in DMA_CON should be set.

Note that n is from 1 to 9.

**WPTO** **WPTO**[31:0] specifies the address of the jump point for a DMA channel, i.e. channel 1 – 9.
**WRITE** Address of the jump destination.
**READ** Value set by the programmer.

## DMA+0n10h DMA Channel n Transfer Count Register DMAn_COUNT

| 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 | LEN | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

This register specifies the amount of total transfer count that the DMA channel is required to perform. Upon completion, the DMA channel generates an interrupt request to the processor while ITEN in DMAn_CON is set as '1'.

Note that the total size of data being transferred by a DMA channel is determined by LEN together with the SIZE in DMAn_CON, i.e. LEN x SIZE.

For virtual FIFO DMA, this register is used to configure the RX threshold and TX threshold. Interrupt is triggered while FIFO count >= RX threshold in RX path or FIFO count =< TX threshold in TX path. Note that ITEN bit in DMA_CON register shall be set, or no interrupt is issued.

Note that n is from 1 to 15.

**LEN** The amount of total transfer count

## DMA+0n14h DMA Channel n Control Register DMAn_CON

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | MAS | | | | | | DIR | WPEN | WPSD |
| 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 | ITEN | | | | | BURST | | | | | B2W | DRQ | DINC | SINC | SIZE | |
| Type | R/W | | | | | R/W | | | | | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | | | | | 0 | | | | | 0 | 0 | 0 | 0 | 0 | |

This register contains all the available control schemes for a DMA channel that is ready for software programmer to configure. Note that all these fields cannot be changed while DMA transfer is in progress or an unexpected situation may occur.

Note that n is from 1 to 15.

**SIZE** Data size within the confine of a bus cycle per transfer.
These bits confines the data transfer size between source and destination to the specified value for individual bus cycle. The size is in terms of byte and has maximum value of 4 bytes. It is mainly decided by the data width of a DMA master.
**00** Byte transfer/1 byte
**01** Half-word transfer/2 bytes
**10** Word transfer/4 bytes
**11** Reserved

**SINC** Incremental source address. Source addresses increase every transfer. If the setting of SIZE is Byte, Source addresses increase by 1 every single transfer. If Half-Word, increase by 2; and if Word, increase by 4.
**0** Disable
**1** Enable

**DINC** Incremental destination address. Destination addresses increase every transfer. If the setting of SIZE is Byte, Destination addresses increase by 1 every single transfer. If Half-Word, increase by 2; and Iif Word, increase by 4.

**0** Disable

**1** Enable

**DREQ** Throttle and handshake control for DMA transfer

**0** No throttle control during DMA transfer or transfers occurred only between memories

**1** Hardware handshake management

The DMA master is able to throttle down the transfer rate by way of request-grant handshake.

**B2W** Word to Byte or Byte to Word transfer for the applications of transferring non-word-aligned-address data to word-aligned-address data. Note that BURST is set to 4-beat burst while enabling this function, and the SIZE is set to Byte.

NO effect on channel 1 & 10 - 15.

**0** Disable

**1** Enable

**BURST** Transfer Type. Burst-type transfers have better bus efficiency. Mass data movement is recommended to use this kind of transfer. However, note that burst-type transfer does not stop until all of the beats in a burst are completed or transfer length is reached. FIFO threshold of peripherals must be configured carefully while being used to move data from/to the peripherals.

What transfer type can be used is restricted by the SIZE. If SIZE is 00b, i.e. byte transfer, all of the four transfer types can be used. If SIZE is 01b, i.e. half-word transfer, 16-beat incrementing burst cannot be used. If SIZE is 10b, i.e. word transfer, only single and 4-beat incrementing burst can be used.

NO effect on channel 10 - 15.

**000** Single

**001** Reserved

**010** 4-beat incrementing burst

**011** Reserved

**100** 8-beat incrementing burst

**101** Reserved

**110** 16-beat incrementing burst

**111** Reserved

**ITEN** DMA transfer completion interrupt enable.

**0** Disable

**1** Enable

**WPSD** The side using address-wrapping function. Only one side of a DMA channel can activate address-wrapping function at a time.

NO effect on channel 10 - 15.

**0** Address-wrapping on source .

**1** Address-wrapping on destination.

**WPEN** Address-wrapping for ring buffer and double buffer. The next address of DMA jumps to WRAP TO address when the current address matches WRAP POINT count.

NO effect on channel 10 - 15.

**0** Disable

**1** Enable

**DIR** Directions of DMA transfer for half-size and Virtual FIFO DMA channels, i.e. channels 2~15. The direction is from the perspective of the DMA masters. WRITE means read from master and then write to the address specified in DMA_PGMADDR, and vice versa.

NO effect on channel 1

**0** Read

**1** Write

**MAS** Master selection. Specifies which master occupies this DMA channel. Once assigned to certain master, the corresponding DREQ and DACK are connected. For half-size and Virtual FIFO DMA channels, i.e. channels 2 ~ 15, a predefined address is assigned as well.

**00000** SIM
**00001** MSDC
**00010** Reserved
**00011** Reserved
**00100** USB1 TX
**00101** USB1 RX
**00110** USB2 TX
**00111** USB2 RX
**01000** UART1 TX
**01001** UART1 RX
**01010** UART2 TX
**01011** UART2 RX
**01100** UART3 TX
**01101** UART3 RX
**01110** DSP-DMA 1
**01111** Reserved
**10000** Reserved
**10001** I2C dual TX
**10010** I2C dual RX

**10011** SIM2
**10011** DSP-DMA 2

**OTHERS** Reserved

## DMA+0n18h DMA Channel n Start Register DMAn_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 | STR | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

This register controls the activity of a DMA channel. Note that prior to setting STR to "1", all the configurations should be done by giving proper value to the registers. Note also that once the STR is set to "1", the hardware does not clear it automatically no matter if the DMA channel accomplishes the DMA transfer or not. In other works, the value of **STR** stays "1" regardless of the completion of DMA transfer. Therefore, the software program should be sure to clear **STR** to "0" before restarting another DMA transfer. If this bit is cleared to "0" during DMA transfer is active, software should polling MDDMA_GLBSTA **RUN$_N$** after this bit is cleared to ensure current DMA transfer is terminated by DMA engine.

Note that n is from 1 to 15.

**STR** Start control for a DMA channel.
**0** The DMA channel is stopped.
**1** The DMA channel is started and running.

## DMA+0n1Ch DMA Channel n Interrupt Status Register DMAn_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 | INT | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

This register shows the interrupt status of a DMA channel. It has the same value as DMA_GLBSTA.

Note that n is from 1 to 14.

**INT** Interrupt Status for DMA Channel

- **0** No interrupt request is generated.
- **1** One interrupt request is pending and waiting for service.

## DMA+0n20h DMA Channel n Interrupt Acknowledge Register DMAn_ACKINT

| 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 | ACK | | | | | | | | | | | | | | | |
| Type | WO | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

This register is used to acknowledge the current interrupt request associated with the completion event of a DMA channel by software program. Note that this is a write-only register, and any read to it returns a value of "0".

Note that n is from 1 to 15.

**ACK** Interrupt acknowledge for the DMA channel

- **0** No effect
- **1** Interrupt request is acknowledged and should be relinquished.

## DMA+0n24h DMA Channel n Remaining Length of Current Transfer DMAn_RLCT

| 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 | RLCT | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

This register is to reflect the left count of the transfer. Note that this value is transfer count not the transfer data size.

Note that n is from 1 to 9.

## DMA+0n28h DMA Bandwidth limiter Register DMAn_LIMITER

| 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 | | | | | | | | | LIMITER | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

This register is to suppress the Bus utilization of the DMA channel. The value is from 0 to 255. 0 means no limitation, and 255 means totally banned. The value between 0 and 255 means certain DMA can have permission to use AHB every (4 X n) AHB clock cycles.

Note that it is not recommended to limit the Bus utilization of the DMA channels because this increases the latency of response to the masters, and the transfer rate decreases as well. Before using it, programmer must make sure that the bus masters have some protective mechanism to avoid entering the wrong states.

Note that n is from 1 to 15.

**LIMITER** from 0 to 255. 0 means no limitation, 255 means totally banned, and others mean Bus access permission every (4 X n) AHB clock.

## DMA+0n2Ch DMA Channel n Programmable Address Register DMAn_PGMADDR

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

The above registers specify the address for a half-size DMA channel. This address represents a source address if DIR in DMA_CON is set to 0, and represents a destination address if DIR in DMA_CON is set to 1. Before being able to program these register, the software should make sure that STR in DMAn_START is set to '0', that is the DMA channel is stopped and disabled completely. Otherwise, the DMA channel may run out of order.

Note that n is from 2 to 15 and PGMADDR can't be TCM address. TCM is not accessible by DMA.

**PGMADDR** **PGMADDR**[31:0] specifies the addresses for a half-size or a Virtual FIFO DMA channel, i.e. channel 2 – 15.

**WRITE** Base address of transfer source or destination according to DIR bit
**READ** Current address of the transfer.



## DMA+0n30h DMA Channel n Virtual FIFO Write Pointer Register DMAn_WRPTR

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

Note that n is from 10 to 15.

**WRPTR** Virtual FIFO Write Pointer.

## DMA+0n34h DMA Channel n Virtual FIFO Read Pointer Register DMAn_RDPTR

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

Note that n is from 10 to 15.

**RDPTR** Virtual FIFO Read Pointer.

## DMA+0n38h DMA Channel n Virtual FIFO Data Count Register DMAn_FFCNT

| 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 | FFCNT | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |

Note that n is from 10 to 15.

**FFCNT** To display the number of data stored in FIFO. 0 means FIFO empty, and FIFO is full if FFCNT is equal to FFSIZE.

## DMA+0n3Ch DMA Channel n Virtual FIFO Status Register DMAn_FFSTA

| 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 | | | | | | | | | | | | | | ALT | EMPTY | FULL |
| Type | | | | | | | | | | | | | | RO | RO | RO |
| Reset | | | | | | | | | | | | | | 1 | 1 | 1 |

Note that n is from 10 to 15.

**FULL** To indicate FIFO is full.
- **0** Not Full
- **1** Full

**EMPTY** To indicate FIFO is empty.
- **0** Not Empty
- **1** Empty

**ALT** To indicate FIFO Count is larger than ALTLEN. DMA issues an alert signal to UART to enable UART flow control.
- **0** Not reach alert region.
- **1** Reach alert region.

## DMA+0n40h DMA Channel n Virtual FIFO Alert Length Register DMAn_ALTLEN

| 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 | | | | | | | | | | | ALTLEN | | | | | |
| Type | | | | | | | | | | | R/W | | | | | |
| Reset | | | | | | | | | | | 0 | | | | | |

Note that n is from 10 to 15.

**ALTLEN** Specifies the Alert Length of Virtual FIFO DMA. Once the remaining FIFO space is less than ALTLEN, an alert signal is issued to UART to enable flow control. Normally, ALTLEN shall be larger than 16 for UART application.

## DMA+0n44h DMA Channel n Virtual FIFO Size Register DMAn_FFSIZE

| 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 | FFSIZE | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

Note that n is from 10 to 15.

**FFSIZE** Specifies the FIFO Size of Virtual FIFO DMA.

## 3.2 Interrupt Controller

### 3.2.1 General Description

**Figure 27** outlines the major functionality of the MCU Interrupt Controller. The interrupt controller processes all interrupt sources coming from external lines and internal MCU peripherals. Since ARM7EJ-S core supports two levels of interrupt latency, this controller generates two request signals: FIQ for fast, low latency interrupt request and IRQ for more general interrupts with lower priority.

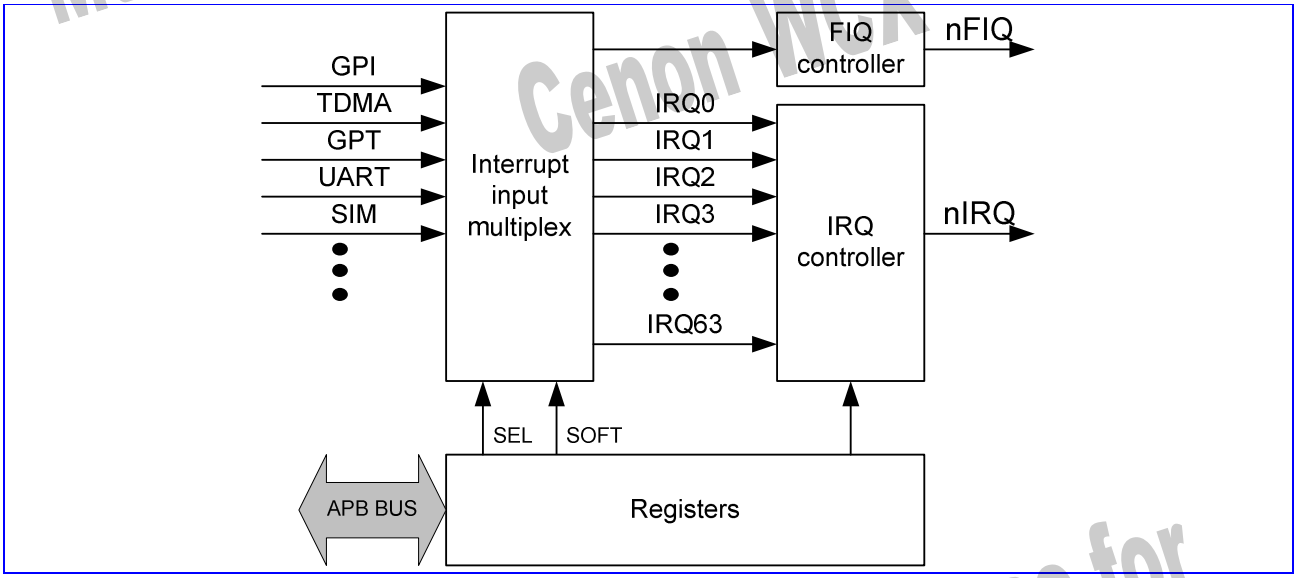



**Figure 27** Block Diagram of the Interrupt Controller

One and only one of the interrupt sources can be assigned to FIQ Controller and have the highest priority in requesting timing critical service. All the others share the same IRQ signal by connecting them to IRQ Controller. The IRQ Controller manages up 64 interrupt lines of IRQ0 to IRQ63 with fixed priority in descending order.

The Interrupt Controller provides a simple software interface by mean of registers to manipulate the interrupt request shared system. IRQ Selection Registers and FIQ Selection Register determine the source priority and connecting relation among sources and interrupt lines. IRQ Source Status Register allows software program to identify the source of interrupt that generates the interrupt request. IRQ Mask Register provides software to mask out undesired sources some time. End of Interrupt Register permits software program to indicate to the controller that a certain interrupt service routine has been finished.

Binary coded version of IRQ Source Status Register is also made available for software program to helpfully identify the interrupt source. Note that while taking advantage of this feature, it should also take the binary coded version of End of Interrupt Register coincidently.

The essential Interrupt Table of ARM7EJ-S core is shown as **Table 37**.

| Address | Description |
|---|---|
| 00000000h | System Reset |
| 00000018h | IRQ |
| 0000001Ch | FIQ |

**Table 37** Table of ARM7EJ-S

### 3.2.1.1 Interrupt Source Masking

Interrupt controller provides the function of Interrupt Source Masking by the way of programming MASK register. Any of them can be masked individually.

However, because of the bus latency, the masking takes effect no earlier than 3 clock cycles later. In this time, the to-be-masked interrupts could come in and generate an IRQ pulse to MCU, and then disappear immediately. This IRQ forces MCU going to Interrupt Service Routine and polling Status Register (IRQ_STA2), but the register shows there is no interrupt. This might cause MCU malfunction.

There are two ways for programmer to protect their software.

1. Return from ISR (Interrupt Service Routine) immediately while the Status register shows no interrupt.
2. Set I bit of MCU before doing Interrupt Masking, and then clear it after Interrupt Masking done.

Both avoid the problem, but the first item is recommended in the ISR.

### 3.2.1.2 External Interrupt

This interrupt controller also integrates an External Interrupt Controller that can support up to 10 interrupt requests coming from external sources, the EINT0~6+10~12, and 3 WakeUp interrupt requests, i.e. EINT7~9, coming from peripherals used to inform system to resume the system clock. All EINT sources can wake up system.

The external interrupts can be used for different kind of applications, mainly for event detections: detection of hand free connection, detection of hood opening, detection of battery charger connection.

Since the external event may be unstable in a certain period, a de-bounce mechanism is introduced to ensure the functionality. The circuitry is mainly used to verify that the input signal remains stable for a programmable number of periods of the clock. When this condition is satisfied, for the appearance or the disappearance of the input, the output

of the de-bounce logic changes to the desired state. Note that, because it uses the 32768Hz slow clock for performing the de-bounce process, the parameter of de-bounce period and de-bounce enable takes effect no sooner than one 32768Hz clock cycle (~30.52us) after the software program sets them. When the sources of External Interrupt Controller are used to resume the system clock in sleep mode, the de-bounce mechanism must be enabled. However, the polarities of EINTs are clocked with the system clock. Any changes to them take effect immediately.

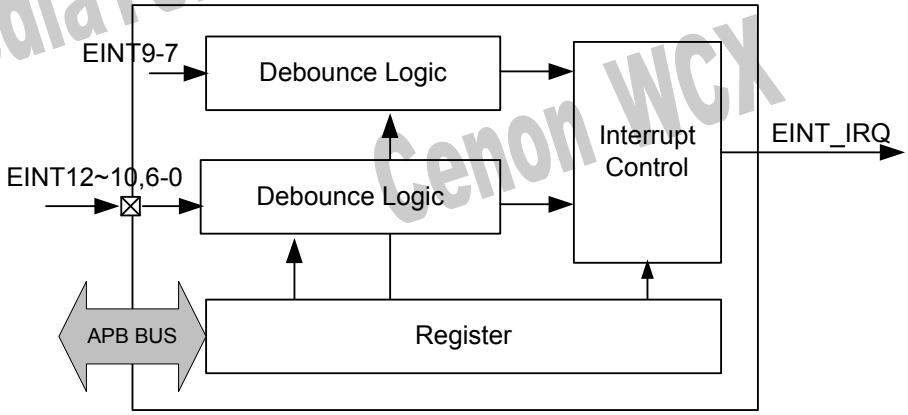



**Figure 28** Block Diagram of External Interrupt Controller

### 3.2.1.3 External Interrupt Input Pins

| EINT | Edge / Level HW Debounce | SOURCE PIN | SUPPLEMENT |
|---|---|---|---|
| EINT0 | Edge / Level Yes | if(GPIO26_M==1) then EINT0=GPIO26(EINT0) else EINT0=1 | 1. GPIOs should be in the input mode and are effected by GPIO data input inversion registers. 2. GPIOxx_M is the GPIO mode control registers, please refer to GPIO segment. |
| EINT1 | Edge / Level Yes | if(GPIO27_M==1) then EINT1=GPIO27(EINT1) else EINT1=1 | |
| EINT2 | Edge / Level Yes | if(GPIO43_M==2) then EINT2=GPIO43(KCOL7) else if(GPIO42_M==7) then EINT2=GPIO42(ECS3_B) else if(urxd1_spmode==1) then EINT2=URXD1 else EINT2=1 | |
| EINT3 | Edge / Level Yes | if(GPIO45_M==2) then EINT3=GPIO45(KROW7) else if(utxd1_spmode==1) then EINT3=UTXD1 else EINT3=1 | |
| EINT4 | Edge / Level Yes | if(GPIO1_M==2) then EINT4=GPIO1(KCOL6) else if(GPIO70_M==2) then EINT4=GPIO70 else EINT4=1 | |
| EINT5 | Edge / Level Yes | if(GPIO8_M==2) then EINT5=GPIO8(KROW5) else if(GPIO24_M==5) then EINT5=GPIO24(UCTS1_B) else EINT5=1 | |

| | | | |
|---|---|---|---|
| EINT6 | Edge / Level<br>Yes | if(GPIO36_M==5) then EINT6=GPIO36(MCINS) else<br>if(GPIO25_M==5) then EINT6=GPIO25(URTS1_B) else EINT6=1 | |
| EINT7 | Edge / Level<br>Yes | Charger detect interrupt (Active low) | |
| EINT8 | Edge / Level<br>Yes | PMU OC interrupt (Active low) | |
| EINT9 | Edge / Level<br>Yes | PMU OV protection interrupt (Active low) | |
| EINT10 | Edge / Level<br>Yes | URXD1 | |
| EINT11 | Edge / Level<br>Yes | if(GPIO22_M==1) then EINT11=GPIO22(URXD2) else EINT11=1 | |
| EINT12 | Edge / Level<br>Yes | if(GPIO20_M==1) then EINT12=GPIO20(URXD3) else EINT12=1 | |
| EINT13 | Edge / Level<br>Yes | Reserved | |

**Table 38** EINT source

| REGISTER ADDRESS | REGISTER NAME | SYNONYM |
|---|---|---|
| 0x8101_0000h | IRQ Selection 0 Register | IRQ_SEL0 |
| 0x8101_0004h | IRQ Selection 1 Register | IRQ_SEL1 |
| 0x8101_0008h | IRQ Selection 2 Register | IRQ_SEL2 |
| 0x8101_000ch | IRQ Selection 3 Register | IRQ_SEL3 |
| 0x8101_0010h | IRQ Selection 4 Register | IRQ_SEL4 |
| 0x8101_0014h | IRQ Selection 5 Register | IRQ_SEL5 |
| 0x8101_0018h | IRQ Selection 6 Register | IRQ_SEL6 |
| 0x8101_001ch | IRQ Selection 7 Register | IRQ_SEL7 |
| 0x8101_0020h | IRQ Selection 8 Register | IRQ_SEL8 |
| 0x8101_0024h | IRQ Selection 9 Register | IRQ_SEL9 |
| 0x8101_0028h | IRQ Selection 10 Register | IRQ_SEL10 |
| 0x8101_002ch | IRQ Selection 11 Register | IRQ_SEL11 |
| 0x8101_0030h | IRQ Selection 12 Register | IRQ_SEL12 |
| 0x8101_0034h | IRQ Selection 13 Register | IRQ_SEL13 |
| 0x8101_0038h | IRQ Selection 14 Register | IRQ_SEL14 |

| | | |
|---|---|---|
| 0x8101_003ch | IRQ Selection 15 Register | IRQ_SEL15 |
| 0x8101_006ch | FIQ Selection Register | FIQ_SEL |
| 0x8101_0070h | IRQ Mask Register (low) | IRQ_MASKL |
| 0x8101_0074h | IRQ Mask Register (high) | IRQ_MASKH |
| 0x8101_0080h | IRQ Mask Clear Register (low) | IRQ_MASK_CLRL |
| 0x8101_0084h | IRQ Mask Clear Register (high) | IRQ_MASK_CLRH |
| 0x8101_0090h | IRQ Mask Set Register (low) | IRQ_MASK_SETL |
| 0x8101_0094h | IRQ Mask Set Register (high) | IRQ_MASK_SETH |
| 0x8101_00a0h | IRQ End of Interrupt Register (low) | IRQ_EOIL |
| 0x8101_00a4h | IRQ End of Interrupt Register (high) | IRQ_EOIH |
| 0x8101_00b0h | IRQ Sensitive Register (low) | IRQ_SENSL |
| 0x8101_00b4h | IRQ Sensitive Register (high) | IRQ_SENSH |
| 0x8101_00c0h | IRQ Software Interrupt Register (low) | IRQ_SOFTL |
| 0x8101_00c4h | IRQ Software Interrupt Register (high) | IRQ_SOFTH |
| 0x8101_00d0h | FIQ Control Register | FIQ_CON |
| 0x8101_00d4h | FIQ End of Interrupt Register | FIQ_EOI |
| 0x8101_00d8h | Binary Coded Value of IRQ_STATUS | IRQ_STA2 |
| 0x8101_00dch | Binary Coded Value of IRQ_EOI | IRQ_EOI2 |
| 0x8101_0100h | EINT Status Register | EINT_STA |
| 0x8101_0104h | EINT Mask Register | EINT_MASK |
| 0x8101_0108h | EINT Mask Clear Register | EINT_MASK_CLR |
| 0x8101_010ch | EINT Mask Set Register | EINT_MASK_SET |
| 0x8101_0110h | EINT Interrupt Acknowledge Register | EINT_INTACK |
| 0x8101_0114h | EINT Sensitive Register | EINT_SENS |
| 0x8101_0118h | EINT Software Interrupt Register | EINT_SOFT |
| 0x8101_0120h | EINT0 De-bounce Control Register | EINT0_CON |
| 0x8101_0130h | EINT1 De-bounce Control Register | EINT1_CON |
| 0x8101_0140h | EINT2 De-bounce Control Register | EINT2_CON |
| 0x8101_0150h | EINT3 De-bounce Control Register | EINT3_CON |
| 0x8101_0160h | EINT4 De-bounce Control Register | EINT4_CON |
| 0x8101_0170h | EINT5 De-bounce Control Register | EINT5_CON |
| 0x8101_0180h | EINT6 De-bounce Control Register | EINT6_CON |

| | | |
|---|---|---|
| 0x8101_0190h | EINT7 De-bounce Control Register | EINT7_CON |
| 0x8101_01a0h | EINT8 De-bounce Control Register | EINT8_CON |
| 0x8101_01b0h | EINT9 De-bounce Control Register | EINT9_CON |
| 0x8101_01c0h | EINT10 De-bounce Control Register | EINT10_CON |
| 0x8101_01d0h | EINT11 De-bounce Control Register | EINT11_CON |
| 0x8101_01e0h | EINT12 De-bounce Control Register | EINT12_CON |
| 0x8101_01f0h | EINT13 De-bounce Control Register | EINT13_CON |

**Table 39** Interrupt Controller Register Map

## 3.2.2 Register Definitions

### 0x8101_0000 IRQ Selection 0 Register — IRQ_SEL0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ3 | | | | | | | | IRQ2 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 3 | | | | | | | | 2 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ1 | | | | | | | | IRQ0 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 1 | | | | | | | | 0 | | | | | |

### 0x8101_0004 IRQ Selection 1 Register — IRQ_SEL1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ7 | | | | | | | | IRQ6 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 7 | | | | | | | | 6 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ5 | | | | | | | | IRQ4 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 5 | | | | | | | | 4 | | | | | |

### 0x8101_0008 IRQ Selection 2 Register — IRQ_SEL2

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQB | | | | | | | | IRQA | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0xb | | | | | | | | 0xa | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ9 | | | | | | | | IRQ8 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 9 | | | | | | | | 8 | | | | | |

## 0x8101_000c IRQ Selection 3 Register — IRQ_SEL3

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQF | | | | | | | | IRQE | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0xf | | | | | | | | 0xe | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQD | | | | | | | | IRQC | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0xd | | | | | | | | 0xc | | | | | |

## 0x8101_0010 IRQ Selection 4 Register — IRQ_SEL4

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ13 | | | | | | | | IRQ12 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x13 | | | | | | | | 0x12 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ11 | | | | | | | | IRQ10 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x11 | | | | | | | | 0x10 | | | | | |

## 0x8101_0014 IRQ Selection 5 Register — IRQ_SEL5

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ17 | | | | | | | | IRQ16 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x17 | | | | | | | | 0x16 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ15 | | | | | | | | IRQ14 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x15 | | | | | | | | 0x14 | | | | | |

## 0x8101_0018 IRQ Selection 6 Register — IRQ_SEL6

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ1B | | | | | | | | IRQ1A | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x1b | | | | | | | | 0x1a | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ19 | | | | | | | | IRQ18 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x19 | | | | | | | | 0x18 | | | | | |

## 0x8101_001c IRQ Selection 7 Register — IRQ_SEL7

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ1F | | | | | | | | IRQ1E | | | | | |

| Type | | | R/W | | | | | | | | R/W | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset | | | 0x1f | | | | | | | | 0x1e | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ1D | | | | | | | | IRQ1C | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x1d | | | | | | | | 0x1c | | | | | |

## 0x8101_0020 IRQ Selection 8 Register IRQ_SEL8

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ23 | | | | | | | | IRQ22 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x23 | | | | | | | | 0x22 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ21 | | | | | | | | IRQ20 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x21 | | | | | | | | 0x20 | | | | | |

## 0x8101_0024 IRQ Selection 9 Register IRQ_SEL9

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ27 | | | | | | | | IRQ26 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x27 | | | | | | | | 0x26 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ25 | | | | | | | | IRQ24 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x25 | | | | | | | | 0x24 | | | | | |

## 0x8101_0028 IRQ Selection 10 Register IRQ_SEL10

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ2B | | | | | | | | IRQ2A | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x2b | | | | | | | | 0x2a | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ29 | | | | | | | | IRQ28 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x29 | | | | | | | | 0x28 | | | | | |

## 0x8101_002c IRQ Selection 11 Register IRQ_SEL11

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ2F | | | | | | | | IRQ2E | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x2f | | | | | | | | 0x2e | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ2D | | | | | | | | IRQ2C | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |

| Reset | | | 0x2d | | | | | | 0x2c | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

## 0x8101_0030 IRQ Selection 12 Register IRQ_SEL12

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ33 | | | | | | | | IRQ32 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x33 | | | | | | | | 0x32 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ31 | | | | | | | | IRQ30 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x31 | | | | | | | | 0x30 | | | | | |

## 0x8101_0034 IRQ Selection 13 Register IRQ_SEL13

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ37 | | | | | | | | IRQ36 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x37 | | | | | | | | 0x36 | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ35 | | | | | | | | IRQ34 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x35 | | | | | | | | 0x34 | | | | | |

## 0x8101_0038 IRQ Selection 14 Register IRQ_SEL14

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ3B | | | | | | | | IRQ3A | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x3b | | | | | | | | 0x3a | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ39 | | | | | | | | IRQ38 | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x39 | | | | | | | | 0x38 | | | | | |

## 0x8101_003c IRQ Selection 15 Register IRQ_SEL15

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | IRQ3F | | | | | | | | IRQ3E | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x3f | | | | | | | | 0x3e | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | IRQ3D | | | | | | | | IRQ3C | | | | | |
| Type | | | R/W | | | | | | | | R/W | | | | | |
| Reset | | | 0x3d | | | | | | | | 0x3c | | | | | |

## 0x8101_006c FIQ Selection Register FIQ_SEL

| 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 | | | | | | | | | | | FIQ | | | | | | |
| Type | | | | | | | | | | | R/W | | | | | | |
| Reset | | | | | | | | | | | 0 | | | | | | |

The IRQ/FIQ Selection Registers provide system designers with a flexible routing scheme to make various mappings of priority among interrupt sources possible. The registers allow the interrupt sources to be mapped onto interrupt requests of either FIQ or IRQ. While only one interrupt source can be assigned to FIQ, the other ones share IRQs by mapping them onto IRQ0 to IRQ3F connected to IRQ controller. The priority sequence of IRQ0~IRQ3F is fixed, i.e. IRQ0 > IRQ1 > IRQ2 > … > IRQ1E > IRQ3F. During the software configuration process, the Interrupt Source Code of desired interrupt source should be written into source field of the corresponding IRQ_SEL0-IRQ_SEL15/FIQ_SEL. 6-bit Interrupt Source Codes for all interrupt sources are fixed and defined.

| Interrupt Source | STA2 (Hex) | STA |
|---|---|---|
| GPI_FIQ | 0 | 00000001 |
| TDMA_CTIRQ1 | 1 | 00000002 |
| TDMA_CTIRQ2 | 2 | 00000004 |
| DSP12CPU | 3 | 00000008 |
| SIM | 4 | 00000010 |
| DMA | 5 | 00000020 |
| TDMA | 6 | 00000040 |
| UART1 | 7 | 00000080 |
| KP | 8 | 00000100 |
| UART2 | 9 | 00000200 |
| GPTimer | A | 00000400 |
| EINT | B | 00000800 |
| USB | C | 00001000 |
| MSDC | D | 00002000 |
| RTC | E | 00004000 |
| Serial-Flash | F | 00008000 |
| LCD | 10 | 00010000 |
| UART3 | 11 | 00020000 |

| | | |
|---|---|---|
| GPI | 12 | 00040000 |
| WDT | 13 | 00080000 |
| DSP22CPU | 14 | 00100000 |
| RESIZER | 15 | 00200000 |
| Reserved | 16 | 00400000 |
| Reserved | 17 | 00800000 |
| IRDBG1 | 18 | 01000000 |
| MSDC_CD | 19 | 02000000 |
| I2C | 1A | 04000000 |
| IRDBG2 | 1B | 08000000 |
| SIM2 | 1C | 10000000 |
| Reserved | 1D | 20000000 |
| CAM | 1E | 40000000 |
| Touch Panel | 1F | 80000000 |
| Sysram/rom | 20 | 100000000 |
| ROT_DMA | 21 | 200000000 |
| PAD2CAM | 22 | 400000000 |
| CHR_BATOFF | 23 | 800000000 |
| CHR_LDO | 24 | 1000000000 |
| PMU_THR[0] | 25 | 2000000000 |
| PMU_THR[1] | 26 | 4000000000 |
| PMU_THR[2] | 27 | 8000000000 |
| Reserved | 28~3F | |

**Table 40** Interrupt Source Code

**FIQ, IRQ0-3F** The 6-bit content of this field corresponds to an Interrupt Source Code shown above.

### 0x8101_0070 IRQ Mask Register (low) IRQ_MASKL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ1F | IRQ1E | IRQ1D | IRQ1C | IRQ1B | IRQ1A | IRQ19 | IRQ18 | IRQ17 | IRQ16 | IRQ15 | IRQ14 | IRQ13 | IRQ12 | IRQ11 | IRQ10 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Name | IRQF | IRQE | IRQD | IRQC | IRQB | IRQA | IRQ9 | IRQ8 | IRQ7 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | IRQ2 | IRQ1 | IRQ0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

## 0x8101_0074 IRQ Mask Register (high) IRQ_MASKH

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ3F | IRQ3E | IRQ3D | IRQ3C | IRQ3B | IRQ3A | IRQ39 | IRQ38 | IRQ37 | IRQ36 | IRQ35 | IRQ34 | IRQ33 | IRQ32 | IRQ31 | IRQ30 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQ2F | IRQ2E | IRQ2D | IRQ2C | IRQ2B | IRQ2A | IRQ29 | IRQ28 | IRQ27 | IRQ26 | IRQ25 | IRQ24 | IRQ23 | IRQ22 | IRQ21 | IRQ20 |
| 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 |

This register contains a mask bit for each interrupt line in IRQ Controller. The register allows each interrupt source IRQ0 to IRQ3F to be disabled or masked separately under software control. After a system reset, all bit values are set to 1 to indicate that interrupt requests are prohibited.

**IRQ0-3F** Mask control for the associated interrupt source in the IRQ controller

- **0** Interrupt is enabled
- **1** Interrupt is disabled

## 0x8101_0080 IRQ Mask Clear Register (low) IRQ_MASK_CLRL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ1F | IRQ1E | IRQ1D | IRQ1C | IRQ1B | IRQ1A | IRQ19 | IRQ18 | IRQ17 | IRQ16 | IRQ15 | IRQ14 | IRQ13 | IRQ12 | IRQ11 | IRQ10 |
| Type | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQF | IRQE | IRQD | IRQC | IRQB | IRQA | IRQ9 | IRQ8 | IRQ7 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | IRQ2 | IRQ1 | IRQ0 |
| Type | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C |

## 0x8101_0084 IRQ Mask Clear Register (high) IRQ_MASK_CLRH

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ3F | IRQ3E | IRQ3D | IRQ3C | IRQ3B | IRQ3A | IRQ39 | IRQ38 | IRQ37 | IRQ36 | IRQ35 | IRQ34 | IRQ33 | IRQ32 | IRQ31 | IRQ30 |
| Type | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQ2F | IRQ2E | IRQ2D | IRQ2C | IRQ2B | IRQ2A | IRQ29 | IRQ28 | IRQ27 | IRQ26 | IRQ25 | IRQ24 | IRQ23 | IRQ22 | IRQ21 | IRQ20 |
| Type | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C |

This register is used to clear bits in IRQ Mask Register. When writing to this register, the data bits that are HIGH cause the corresponding bits in IRQ Mask Register to be cleared. Data bits that are LOW have no effect on the corresponding bits in IRQ Mask Register.

**IRQ0-3F** Clear corresponding bits in IRQ Mask Register.

**0** No effect

**1** Disable the corresponding MASK bit

## 0x8101_0090 IRQ Mask SET Register (low) IRQ_MASK_SETL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ1F | IRQ1E | IRQ1D | IRQ1C | IRQ1B | IRQ1A | IRQ19 | IRQ18 | IRQ17 | IRQ16 | IRQ15 | IRQ14 | IRQ13 | IRQ12 | IRQ11 | IRQ10 |
| Type | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQF | IRQE | IRQD | IRQC | IRQB | IRQA | IRQ9 | IRQ8 | IRQ7 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | IRQ2 | IRQ1 | IRQ0 |
| Type | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S |

## 0x8101_0094 IRQ Mask SET Register (high) IRQ_MASK_SETH

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ3F | IRQ3E | IRQ3D | IRQ3C | IRQ3B | IRQ3A | IRQ39 | IRQ38 | IRQ37 | IRQ36 | IRQ35 | IRQ34 | IRQ33 | IRQ32 | IRQ31 | IRQ30 |
| Type | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQ2F | IRQ2E | IRQ2D | IRQ2C | IRQ2B | IRQ2A | IRQ29 | IRQ28 | IRQ27 | IRQ26 | IRQ25 | IRQ24 | IRQ23 | IRQ22 | IRQ21 | IRQ20 |
| Type | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S |

This register is used to set bits in the IRQ Mask Register. When writing to this register, the data bits that are HIGH cause the corresponding bits in IRQ Mask Register to be set. Data bits that are LOW have no effect on the corresponding bits in IRQ Mask Register.

**IRQ0-3F** Set corresponding bits in IRQ Mask Register.

**0** No effect

**1** Enable corresponding MASK bit

## 0x8101_00a0 IRQ End of Interrupt Register (low) IRQ_EOIL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ1F | IRQ1E | IRQ1D | IRQ1C | IRQ1B | IRQ1A | IRQ19 | IRQ18 | IRQ17 | IRQ16 | IRQ15 | IRQ14 | IRQ13 | IRQ12 | IRQ11 | IRQ10 |

| Type | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | 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 | IRQF | IRQE | IRQD | IRQC | IRQB | IRQA | IRQ9 | IRQ8 | IRQ7 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | IRQ2 | IRQ1 | IRQ0 |
| Type | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 0x8101_00a4 IRQ End of Interrupt Register (high) IRQ_EOIH

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ3F | IRQ3E | IRQ3D | IRQ3C | IRQ3B | IRQ3A | IRQ39 | IRQ38 | IRQ37 | IRQ36 | IRQ35 | IRQ34 | IRQ33 | IRQ32 | IRQ31 | IRQ30 |
| Type | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | 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 | IRQ2F | IRQ2E | IRQ2D | IRQ2C | IRQ2B | IRQ2A | IRQ29 | IRQ28 | IRQ27 | IRQ26 | IRQ25 | IRQ24 | IRQ23 | IRQ22 | IRQ21 | IRQ20 |
| Type | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register provides a mean for software to relinquish and to refresh the interrupt controller. Writing a 1 to a specific bit position results in an End of Interrupt command issued internally to the corresponding interrupt line.

**IRQ0-3F** End of Interrupt command for the associated interrupt line.

- **0** No service is currently in progress or pending
- **1** Interrupt request is in-service

## 0x8101_00b0 IRQ Sensitive Register (low) IRQ_SENSL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ1F | IRQ1E | IRQ1D | IRQ1C | IRQ1B | IRQ1A | IRQ19 | IRQ18 | IRQ17 | IRQ16 | IRQ15 | IRQ14 | IRQ13 | IRQ12 | IRQ11 | IRQ10 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQF | IRQE | IRQD | IRQC | IRQB | IRQA | IRQ9 | IRQ8 | IRQ7 | IRQ6 | IRQ5 | IRQ4 | IRQ3 | IRQ2 | IRQ1 | IRQ0 |
| 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 |

## 0x8101_00b4 IRQ Sensitive Register (high) IRQ_SENSH

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQ3F | IRQ3E | IRQ3D | IRQ3C | IRQ3B | IRQ3A | IRQ39 | IRQ38 | IRQ37 | IRQ36 | IRQ35 | IRQ34 | IRQ33 | IRQ32 | IRQ31 | IRQ30 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Name | IRQ2F | IRQ2E | IRQ2D | IRQ2C | IRQ2B | IRQ2A | IRQ29 | IRQ28 | IRQ27 | IRQ26 | IRQ25 | IRQ24 | IRQ23 | IRQ22 | IRQ21 | IRQ20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

All interrupt lines of IRQ Controller, IRQ0~IRQ3F can be programmed as either edge or level sensitive. By default, all the interrupt lines are edge sensitive and should be active LOW. Once a interrupt line is programmed as edge sensitive, an interrupt request is triggered only at the falling edge of interrupt line, and the next interrupt is not accepted until the EOI command is given. However, level sensitive interrupts trigger is according to the signal level of the interrupt line. Once the interrupt line become from HIGH to LOW, an interrupt request is triggered, and another interrupt request is triggered if the signal level remain LOW after an EOI command. Note that in edge sensitive mode, even if the signal level remains LOW after EOI command, another interrupt request is not triggered. That is because edge sensitive interrupt is only triggered at the falling edge.

**IRQ0-3F** Sensitivity type of the associated Interrupt Source

- **0** Edge sensitivity with active LOW
- **1** Level sensitivity with active LOW

## 0x8101_00c0 IRQ Software Interrupt Register (low) IRQ_SOFTL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | INT_SRC1F | INT_SRC1E | INT_SRC1D | INT_SRC1C | INT_SRC1B | INT_SRC1A | INT_SRC19 | INT_SRC18 | INT_SRC17 | INT_SRC16 | INT_SRC15 | INT_SRC14 | INT_SRC13 | INT_SRC12 | INT_SRC11 | INT_SRC10 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | INT_SRCF | INT_SRCE | INT_SRCD | INT_SRCC | INT_SRCB | INT_SRCA | INT_SRC9 | INT_SRC8 | INT_SRC7 | INT_SRC6 | INT_SRC5 | INT_SRC4 | INT_SRC3 | INT_SRC2 | INT_SRC1 | INT_SRC0 |
| 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 |

## 0x8101_00c4 IRQ Software Interrupt Register (high) IRQ_SOFTH

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | INT_SRC3F | INT_SRC3E | INT_SRC3D | INT_SRC3C | INT_SRC3B | INT_SRC3A | INT_SRC39 | INT_SRC38 | INT_SRC37 | INT_SRC36 | INT_SRC35 | INT_SRC34 | INT_SRC33 | INT_SRC32 | INT_SRC31 | INT_SRC30 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | INT_SRC2F | INT_SRC2E | INT_SRC2D | INT_SRC2C | INT_SRC2B | INT_SRC2A | INT_SRC29 | INT_SRC28 | INT_SRC27 | INT_SRC26 | INT_SRC25 | INT_SRC24 | INT_SRC23 | INT_SRC22 | INT_SRC21 | INT_SRC20 |
| 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 |

Setting "1" to the specific bit position generates a software interrupt for corresponding interrupt line before interrupt input multiplex. This register is used for debug purpose.

**INT_SRC0-3F** Software Interrupt

### 0x8101_00d0 FIQ Control Register FIQ_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 | | | | | | | | | | | | | | | SENS | MASK |
| Type | | | | | | | | | | | | | | | R/W | R/W |
| Reset | | | | | | | | | | | | | | | 0 | 1 |

This register provides a means for software program to control the FIQ controller.

**MASK** Mask control for the FIQ Interrupt Source
- **0** Interrupt is enabled
- **1** Interrupt is disabled

**SENS** Sensitivity type of the FIQ Interrupt Source
- **0** Edge sensitivity with active LOW
- **1** Level sensitivity with active LOW

### 0x8101_00d4 FIQ End of Interrupt Register FIQ_EOI

| 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 | | | | | | | | | | | | | | | | EOI |
| Type | | | | | | | | | | | | | | | | WO |
| Reset | | | | | | | | | | | | | | | | 0 |

This register provides a means for software to relinquish and to refresh the FIQ controller. Writing a '1' to the specific bit position results in an End of Interrupt command issued internally to the corresponding interrupt line.

**EOI** End of Interrupt command

### 0x8101_00d8 Binary Coded Value of IRQ_STATUS IRQ_STA2

| 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 | | | | | | | | NOIRQ | | | STA | | | | | |
| Type | | | | | | | | RO | | | RO | | | | | |
| Reset | | | | | | | | 0 | | | 0 | | | | | |

This Register is a binary coded version of IRQ_STA. It is used by the software program to poll which interrupt line has generated the IRQ interrupt request in a much easier way. Any read to it has the same result as reading IRQ_STA. The IRQ_STA2 is also read-only; write access has no effect on the content. Note that IRQ_STA2 should be coupled with IRQ_EOI2 while using it.

**STA** Binary coded value of IRQ_STA

**NOIRQ** Indicating if there is an IRQ or not. If there is no IRQ, this bit is HIGH, and the value of STA is 0_0000b.

## 0x8101_00dc Binary Coded Value of IRQ_EOI IRQ_EOI2

| 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 | | | | | | | | | | | EOI | | | | | |
| Type | | | | | | | | | | | WO | | | | | |
| Reset | | | | | | | | | | | 0 | | | | | |

This register is a binary coded version of IRQ_EOI. It provides an easier way for software program to relinquish and to refresh the interrupt controller. Writing a specific code results in an End of Interrupt command issued internally to the corresponding interrupt line. Note that IRQ_EOI2 should be coupled with IRQ_STA2 while using it.

**EOI** Binary coded value of IRQ_EOI

## 0x8101_0100 EINT Interrupt Status Register EINT_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 | | | EINT13 | EINT12 | EINT11 | EINT10 | EINT9 | EINT8 | EINT7 | EINT6 | EINT5 | EINT4 | EINT3 | EINT2 | EINT1 | EINT0 |
| Type | | | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

This register keeps up with current status of which EINT Source generated the interrupt request. If EINT sources are set to edge sensitive, EINT_IRQ is de-asserted while the corresponding EINT_INTACK is programmed by 1.

**EINT0-EINT13** Interrupt Status

- **0** No interrupt request is generated
- **1** Interrupt request is pending

## 0x8101_0104 EINT Interrupt Mask Register — EINT_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 | | | EINT13 | EINT12 | EINT11 | EINT10 | EINT9 | EINT8 | EINT7 | EINT6 | EINT5 | EINT4 | EINT3 | EINT2 | EINT1 | EINT0 |
| 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 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

This register controls whether or not EINT Source is allowed to generate an interrupt request. Setting a "1" to the specific bit position prohibits the external interrupt line from becoming active.

**EINT0-EINT13** Interrupt Mask

- **0** Interrupt request is enabled.
- **1** Interrupt request is disabled.

## 0x8101_0108 EINT Interrupt Mask Clear Register — EINT_MASK_CLR

| 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 | | | EINT13 | EINT12 | EINT11 | EINT10 | EINT9 | EINT8 | EINT7 | EINT6 | EINT5 | EINT4 | EINT3 | EINT2 | EINT1 | EINT0 |
| Type | | | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C | W1C |

This register is used to clear individual mask bits. Only the bits set to 1 are in effect, and interrupt masks for which the mask bit is set are cleared (set to 0). Otherwise the interrupt mask bit retains its original value.

**EINT0-EINT13** Disable mask for the associated external interrupt source

- **0** No effect.
- **1** Disable the corresponding MASK bit.

## 0x8101_010C EINT Interrupt Mask Set Register — EINT_MASK_SET

| 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 | | | EINT13 | EINT12 | EINT11 | EINT10 | EINT9 | EINT8 | EINT7 | EINT6 | EINT5 | EINT4 | EINT3 | EINT2 | EINT1 | EINT0 |
| Type | | | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S | W1S |

This register is used to set individual mask bits. Only the bits set to 1 are in effect, and interrupt masks for which the mask bit is set are set to 1. Otherwise the interrupt mask bit retains its original value.

**EINT0-EINT13** Disable mask for the associated external interrupt source.

**0** No effect.

**1** Enable corresponding MASK bit.

## 0x8101_0110 EINT Interrupt Acknowledge Register EINT_INTACK

| 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 | | | EINT13 | EINT12 | EINT11 | EINT10 | EINT9 | EINT8 | EINT7 | EINT6 | EINT5 | EINT4 | EINT3 | EINT2 | EINT1 | EINT0 |
| Type | | | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO | WO |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Writing “1” to the specific bit position acknowledges the interrupt request correspondingly to the external interrupt line source.

**EINT0-EINT13** Interrupt acknowledgement

**0** No effect.

**1** Interrupt Request is acknowledged.

## 0x8101_0114 EINT Sensitive Register EINT_SENS

| 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 | | | EINT13 | EINT12 | EINT11 | EINT10 | EINT9 | EINT8 | EINT7 | EINT6 | EINT5 | EINT4 | EINT3 | EINT2 | EINT1 | EINT0 |
| 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 | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Sensitivity type of external interrupt source.

**EINT0-EINT13** Sensitive type of the associated external interrupt source

- **0** Edge sensitivity.
- **1** Level sensitivity.

## 0x8101_0118 EINT Software Interrupt Register — EINT_SOFT

| 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 | | | EINT13 | EINT12 | EINT11 | EINT10 | EINT9 | EINT8 | EINT7 | EINT6 | EINT5 | EINT4 | EINT3 | EINT2 | EINT1 | EINT0 |
| 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 |

Setting "1" to the specific bit position generates a software interrupt for corresponding interrupt line before mask. This register is used for debug purpose.

**EINT0-EINT13** Software Interrupt

## 0x8101_0120+ n*0x10 EINTn De-bounce Control Register — EINTn_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 | EN | PRESCALER | | | POL | CNT | | | | | | | | | | |
| Type | R/W | R/W | | | R/W | R/W | | | | | | | | | | |
| Reset | 0 | 0 | | | 0 | 0 | | | | | | | | | | |

These registers control the de-bounce logic for external interrupt sources in order to minimize the possibility of false activations.

When the external interrupt sources is used to resume the system clock from the sleep mode, the De-bounce control circuit must be enabled.

Note that n is from 0 to 13

**CNT** De-bounce duration in terms of numbers of 32768Hz clock cycles. The cycle length is determined by PRESCALER

**POL** Activation type of the EINT source

- **0** Negative polarity

**1** Positive polarity

**PRESCALER** Determine the clock cycle period for debounce count.

**000** 32768Hz, max: 0.0625sec

**001** 16384Hz

**010** 8192Hz

**011** 4096Hz

**100** 2048Hz, max: 1sec

**101** 1024Hz

**110** 512Hz

**111** 256Hz, max: 8secs

**EN** De-bounce control circuit

**0** Disable

**1** Enable

If you have to change the debounce setting of some EINT. Please follow the steps:

1. mask the EINT which you want to change the debounce setting
2. disable debounce (EN=0)
3. delay at least 5 32K cycles
4. Enable the debounce (EN=1) and change the debounce setting
5. unmask the EINT

# 4 Microcontroller Peripherals

Microcontroller (MCU) Peripherals are devices that are under direct control of the Microcontroller. Most of the devices are attached to the Advanced Peripheral Bus (APB) of the MCU subsystem, and serve as APB slaves. Each MCU peripheral must be accessed as a memory-mapped I/O device; that is, the MCU or the DMA bus master reads from or writes to the specific peripheral by issuing memory-addressed transactions.

## 4.1 Pulse-Width Modulation Outputs

### 4.1.1 General Description

Three generic pulse-width modulators are implemented to generate pulse sequences with programmable frequency and duty cycle for LCD backlight or charging purpose. The duration of the PWM output signal is LOW as long as the internal counter value is greater than or equal to the threshold value. The waveform is shown in **Figure 29**.

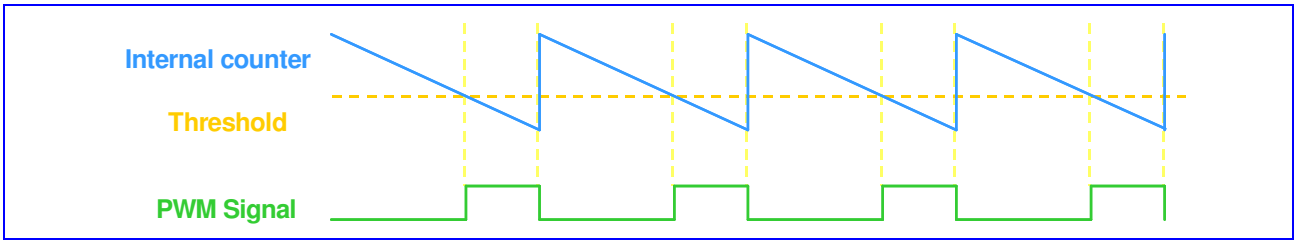



**Figure 29** PWM waveform

The frequency and volume of PWM output signal are determined by these registers: PWM_COUNT, PWM_THRES, PWM_CON. The POWERDOWN (pdn_pwm) signal is applied to power-down the PWM module. When PWM is deactivated (POWERDOWN=1), the output is in LOW state.

The output PWM frequency is determined by:

$$\frac{CLK}{CLOCK\_DIV \times (PWM\_COUNT + 1)}$$ CLK = 13000000 when CLKSEL = 0, $CLK = 32000 when CLKSEL = 1$

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: $\frac{PWM\_THRES}{PWM\_COUNT+1}$

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

## 4.1.2 Register Definitions

### 0x81080000h PWM1 Control register PWM1_CON

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

**CLK** Select PWM1 clock prescaler scale.

**00** CLK Hz

**01** CLK/2 Hz

**10** CLK/4 Hz

**11** CLK/8 Hz

Note: When PWM1 module is disabled, its output should be kept in the LOW state.

**CLKSEL** Select PWM1 clock

**0** CLK=13M Hz

**1** CLK=32K Hz

### 0x81080004h PWM1 max counter value register PWM1_COUNT

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | PWM1_COUNT [12:0] | | | | | | | | | | | | |
| Type | | | | R/W | | | | | | | | | | | | |
| Reset | | | | 1FFFh | | | | | | | | | | | | |

**PWM1_COUNT** PWM1 max counter value. This value is the initial value for the internal counter. Regardless of the operation mode, if PWM1_COUNT is written while the internal counter is counting backwards, the new initial value does not take effect until the internal counter counts down to zero, i.e. a complete period.

### 0x81080008h PWM1 Threshold Value register PWM1_THRES

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | PWM1_THRES [12:0] | | | | | | | | | | | | |
| Type | | | | R/W | | | | | | | | | | | | |

| Reset | | | | 0 |
|---|---|---|---|---|

**PWM1_THRES** Threshold value. When the internal counter value is greater than or equal to PWM1_THRES, the PWM1 output signal is 0; when the internal counter is less than PWM1_THRES, the PWM1 output signal is 1.

## 0x8108000Ch PWM2 Control register PWM2_CON

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

**CLK** Select PWM2 clock prescaler scale.

**00** CLK Hz

**01** CLK/2 Hz

**10** CLK/4 Hz

**11** CLK/8 Hz

Note: When PWM2 module is disabled, its output should be kept in the LOW state.

**CLKSEL** Select PWM2 clock

**0** CLK=13M Hz

**1** CLK=32K Hz

## 0x81080010h PWM2 max counter value register PWM2_COUNT

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | PWM2_COUNT [12:0] | | | | | | | | | | | | |
| Type | | | | R/W | | | | | | | | | | | | |
| Reset | | | | 1FFFh | | | | | | | | | | | | |

**PWM2_COUNT** PWM2 max counter value. This value is the initial value for the internal counter. Regardless of the operation mode, if PWM2_COUNT is written while the internal counter is counting backwards, the new initial value does not take effect until the internal counter counts down to zero, i.e. a complete period.

## 0x81080014h PWM2 Threshold Value register PWM2_THRES

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | PWM2_THRES [12:0] | | | | | | | | | | | | |
| Type | | | | R/W | | | | | | | | | | | | |
| Reset | | | | 0 | | | | | | | | | | | | |

**PWM2_THRES** Threshold value. When the internal counter value is greater than or equal to PWM2_THRES, the PWM2 output signal is 0; when the internal counter is less than PWM2_THRES, the PWM2 output signal is 1.

### 0x81080018h PWM3 Control register PWM3_CON

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

**CLK** Select PWM3 clock prescaler scale.

**02** CLK Hz

**03** CLK/2 Hz

**10** CLK/4 Hz

**11** CLK/8 Hz

Note: When PWM3 module is disabled, its output should be kept in the LOW state.

**CLKSEL** Select PWM3 clock

**0** CLK=13M Hz

**1** CLK=32K Hz

### 0x8108001Ch PWM3 max counter value register PWM3_COUNT

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | PWM3_COUNT [12:0] | | | | | | | | | | | | |
| Type | | | | R/W | | | | | | | | | | | | |
| Reset | | | | 1FFFh | | | | | | | | | | | | |

**PWM3_COUNT** PWM3 max counter value. This value is the initial value for the internal counter. Regardless of the operation mode, if PWM3_COUNT is written while the internal counter is counting backwards, the new initial value does not take effect until the internal counter counts down to zero, i.e. a complete period.

### 0x81080020h PWM3 Threshold Value register PWM3_THRES

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | PWM3_THRES [12:0] | | | | | | | | | | | | |
| Type | | | | R/W | | | | | | | | | | | | |
| Reset | | | | 0 | | | | | | | | | | | | |

**PWM3_THRES** Threshold value. When the internal counter value is greater than or equal to PWM3_THRES, the PWM3 output signal is 0; when the internal counter is less than PWM3_THRES, the PWM3 output signal is 1.

**Figure 30** shows the PWM waveform with the indicated register values.

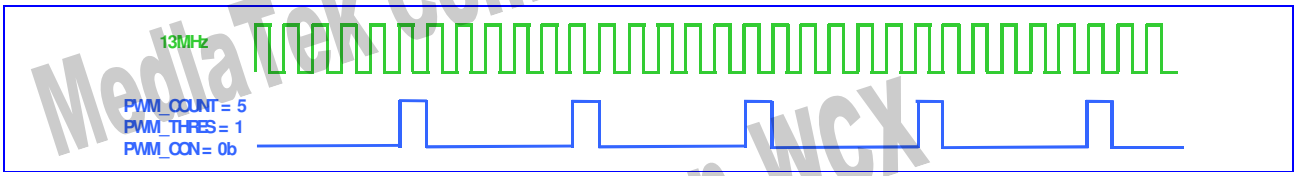



**Figure 30** PWM waveform with register values

## 4.2 SIM Interface

The MT6252 contains two dedicated smart card interfaces to allow the MCU to access the two SIM cards. Each interface can operate via 5 terminals. As shown is the Figure 31, SIMVCC, SIMSEL, SIMRST, SIMCLK and SIMDATA are for one SIM interface, while SIM2VCC, SIM2SEL, SIM2RST, SIM2CLK and SIM2DATA are for the other one.

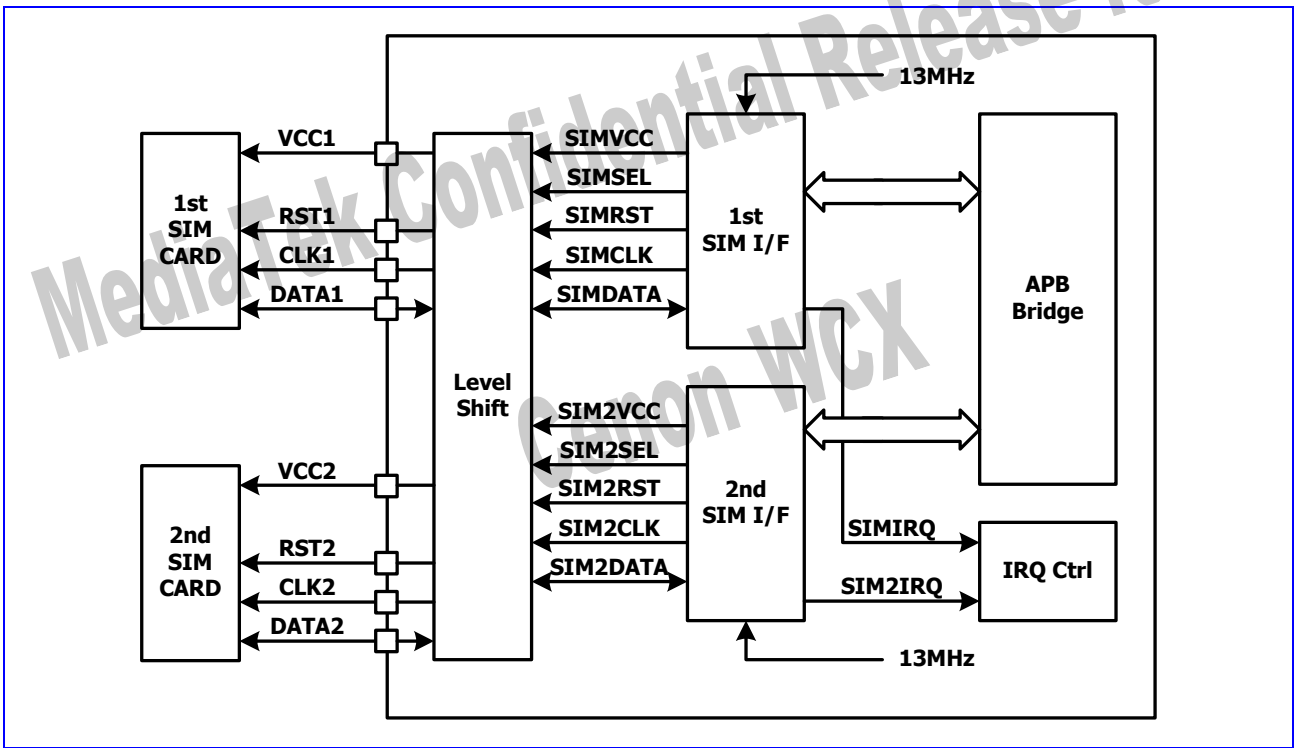



Figure 31 **SIM Interface Block Diagram**

The functions of the two SIM interfaces are identical; therefore, only first SIM interface will be described in this document. The 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 purpose.

Basically, the SIM interface acts as 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 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 first and logic level ONE is in state Low)

***PB***: Odd Parity Check Bit

If the receiver gets a wrong parity bit, it will respond by pulling the SIMDATA Low to inform the transmitter and the transmitter will retransmit the character.

When the receiver is a SIM Card, the error response starts 0.5 bits after the PB and it may last for 1~2 bit periods.

When the receiver is the SIM interface, the error response starts 0.5 bits after the PB and lasts for 1.5 bit period.

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

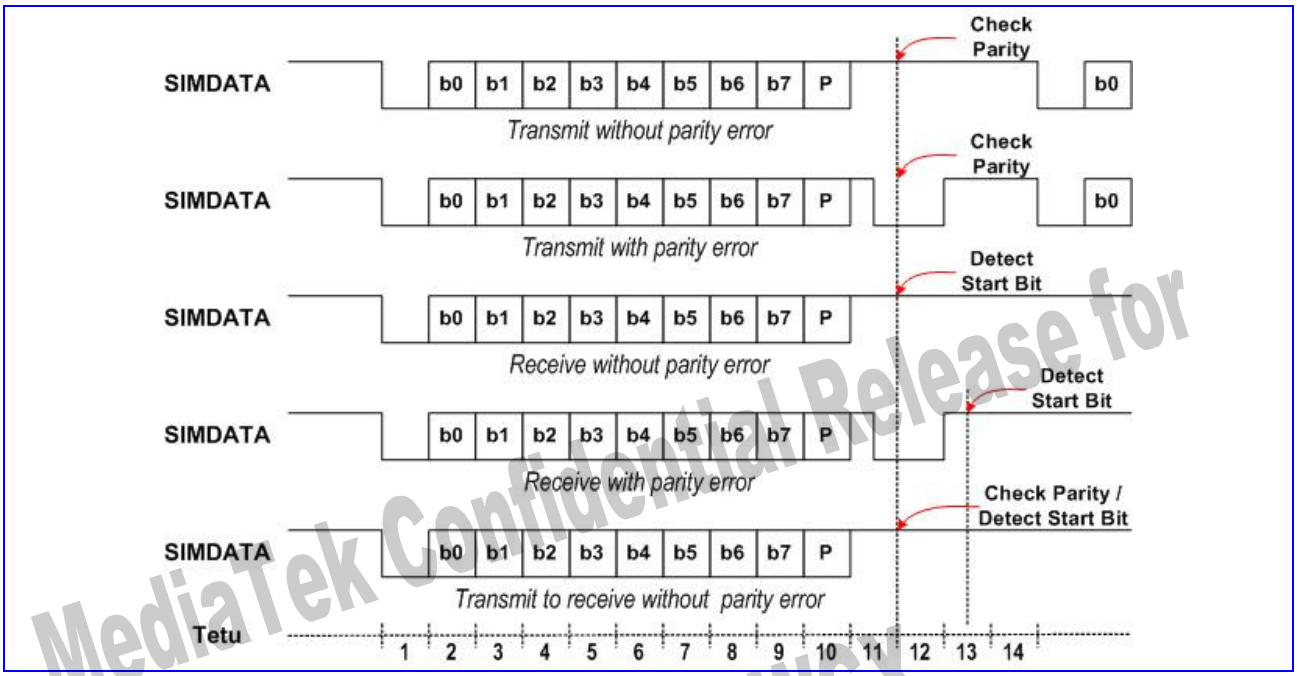



Figure 32 **SIM Interface Timing Diagram**

### 4.2.1 Register Definitions

For MCU to control two SIM card interface, all registers are duplicated to two copies but with different base address. In the following, n = "1" is for 1st SIM card interface, while n=2 is for 2nd SIM card interface. For example, address SIM1+0000h is mapped to SIM1_SIM_CTRL register, while address SIM2+0000h is mapped to SIM2_SIM_CTRL register.

#### 4.2.1.1 Register Overview

| MCU Register Address (hex) | Acronym | Discription |
|---|---|---|
| 1st SIM card Interface | | |
| SIM1+0000h | SIM1_SIM_CTRL | Control register |
| SIM1+0004h | SIM1_SIM_CONF | Configuration register |
| SIM1+0008h | SIM1_SIM_BRR | Baud rate register |
| SIM1+0010h | SIM1_SIM_IRQEN | Interrupt enable register |
| SIM1+0014h | SIM1_SIM_STS | Status register |
| SIM1+0020h | SIM1_SIM_RETRY | Retry limit register |
| SIM1+0024h | SIM1_SIM_TIDE | FIFO tide mark register |
| SIM1+0030h | SIM1_SIM_DATA | TX/RX data register |
| SIM1+0034h | SIM1_SIM_COUNT | FIFO count register |
| SIM1+0040h | SIM1_SIM_ATIME | Activation time register |
| SIM1+0044h | SIM1_SIM_DTIME | Deactivation time register |
| SIM1+0048h | SIM1_SIM_TOUT | Character to character waiting time register |
| SIM1+004Ch | SIM1_SIM_GTIME | Block to block guard time register |
| SIM1+0050h | SIM1_SIM_ETIME | Block to error signal time register |
| SIM1+0054h | SIM1_SIM_EXT_TIME | Extend data I/O state switch time register |
| SIM1+0060h | SIM1_SIM_INS | Command header register : INS |

| | | |
|---|---|---|
| SIM1+0064h | SIM1_SIM_IMP3 | Command header register : P3 |
| SIM1+0068h | SIM1_SIM_SW1 | Procedure byte register : SW1 |
| SIM1+006Ch | SIM1_SIM_SW2 | Procedure byte register : SW2 |
| SIM1+0070h | SIM1_SIM_ATRSTA | ATR state register |
| SIM1+0074h | SIM1_SIM_STATUS | Protocol state register |
| SIM1+0080h | SIM1_SIM_DMADATA | TX/RX data register for DMA |
| 2nd SIM card Interface | | |
| SIM2+0000h | SIM2_SIM_CTRL | Control register |
| SIM2+0004h | SIM2_SIM_CONF | Configuration register |
| SIM2+0008h | SIM2_SIM_BRR | Baud rate register |
| SIM2+0010h | SIM2_SIM_IRQEN | Interrupt enable register |
| SIM2+0014h | SIM2_SIM_STS | Status register |
| SIM2+0020h | SIM2_SIM_RETRY | Retry limit register |
| SIM2+0024h | SIM2_SIM_TIDE | FIFO tide mark register |
| SIM2+0030h | SIM2_SIM_DATA | TX/RX data register |
| SIM2+0034h | SIM2_SIM_COUNT | FIFO count register |
| SIM2+0040h | SIM2_SIM_ATIME | Activation time register |
| SIM2+0044h | SIM2_SIM_DTIME | Deactivation time register |
| SIM2+0048h | SIM2_SIM_TOUT | Character to character waiting time register |
| SIM2+004Ch | SIM2_SIM_GTIME | Block to block guard time register |
| SIM2+0050h | SIM2_SIM_ETIME | Block to error signal time register |
| SIM2+0054h | SIM2_SIM_EXT_TIME | Extend data I/O state switch time register |
| SIM2+0060h | SIM2_SIM_INS | Command header register : INS |
| SIM2+0064h | SIM2_SIM_IMP3 | Command header register : P3 |

| | | |
|---|---|---|
| SIM2+0068h | SIM2_SIM_SW1 | Procedure byte register : SW1 |
| SIM2+006Ch | SIM2_SIM_SW2 | Procedure byte register : SW2 |
| SIM2+0070h | SIM2_SIM_ATRSTA | ATR state register |
| SIM2+0074h | SIM2_SIM_STATUS | Protocol state register |
| SIM2+0080h | SIM2_SIM_DMADATA | TX/RX data register for DMA |

### 4.2.1.2 Register Description

#### SIMn+0000h SIM module control register SIMN_SIM_CTRL

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | WRST | CSTOP | SIMON |
| Type | | | | | | | | | | | | | | W | R/W | R/W |
| Reset | | | | | | | | | | | | | | 0 | 0 | 0 |

**SIMON** SIM card power-up/power-down control

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

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

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

**0** Enable the SIMCLK output.

**1** Disable the SIMCLK output

**WRST** SIM card warm reset control

#### SIMn+0004h SIM module configuration register SIMN_SIM_CON

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

**RXACK** SIM card reception error handshake control

**0** Disable character receipt handshaking

**1** Enable character receipt handshaking

**TXACK** SIM card transmission error handshake control

**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** Data Transfer Direction

0 LSB is transmitted and received first

1 MSB is transmitted and received first

**ODD** Select odd or even parity

0 Even parity

1 Odd parity

**SIMSEL** SIM card supply voltage select

0 SIMSEL pin is set to LOW level, 1.8V

1 SIMSEL pin is set to HIGH level, 3V

**TOUT** SIM work waiting time counter control

0 Disable Time-Out counter

1 Enable Time-Out counter

**T1EN** T=1 protocol controller control

0 Disable T=1 protocol controller

1 Enable T=1 protocol controller

**T0EN** T=0 protocol controller control

0 Disable T=0 protocol controller

1 Enable T=0 protocol controller

**HFEN** Hardware flow control

0 Disable hardware flow control

1 Enable hardware flow control

## SIMn +0008h SIM Baud Rate Register SIMN_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** Set SIMCLK frequency

00 Reserved

**01** 13/4 MHz
**10** 13/8 MHz
**11** 13/12 MHz

**ETU** Determines the duration of elementary time unit in unit of SIMCLK

## SIMn +0010h SIM interrupt enable register SIMN_SIM_IRQEN

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

For all these bits
**0** Interrupt is disabled
**1** Interrupt is enabled

## SIMn +0014h SIM module status register SIMN_SIM_STS

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | 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 | R |
| Reset | | | | | | — | — | — | — | — | — | — | — | — | — | — |

**TXTIDE** The interrupt occurs when number of transmitted data in the FIFO is less than transmitted tide.
**RXTIDE** The interrupt occurs when number of received data in the FIFO is larger than received tide.
**OVRUN** Receive FIFO overflow interrupt occurred
**TOUT** Between characters timeout interrupt occurred
**TXERR** Character transmission error interrupt occurred
**ATRERR** ATR start time-out interrupt occurred
**SIMOFF** Card deactivation complete interrupt occurred
**T0END** Data Transfer handled by T=0 Controller completed interrupt occurred
**RXERR** Character reception error interrupt occurred
**T1END** Data Transfer handled by T=1 Controller completed interrupt occurred
**EDCERR** T=1 Controller CRC error occurred

## SIMn +0020h SIM retry limit register SIMN_SIM_RETRY

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | TXRETRY | | | | | | | | RXRETRY | | |
| Type | | | | | | R/W | | | | | | | | R/W | | |
| Reset | | | | | | 3h | | | | | | | | 3h | | |

**RXRETRY** Specify the maximum numbers of receive retries that are allowed when parity error has occurred.
**TXRETRY** Specify the maximum numbers of transmit retries that are allowed when parity error has occurred.

### SIMn +0024h SIM FIFO tide mark register SIMN_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 SIMN_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. These correspond to the character being read or written

### SIMn +0034h SIM FIFO count register SIMN_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 SIMN_SIM_ATIME

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATIME[15:0] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | AFC7h | | | | | | | | | | | | | | | |

**ATIME** The register defines the duration, in SIM clock cycles, of the time taken for each of the three stages of the card activation process, from SIMON transits to high to turn on VCC, from turn on VCC to pull DATA high and then from pull DATA high to turn on CLK.

### SIMn +0044h SIM deactivation time register SIMN_SIM_DTIME

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | DTIME[11:0] | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | | | | | R/W | | | |
| Reset | | | | | | | | | | | | | 3E7h | | | |

**DTIME** The register defines the duration, in 13MHz clock cycles, of the time taken for each of the three stages of the card deactivation sequence, from pull RST low to turn of CLK, from turn off CLK to pull DATA low, from pull DATA low to turn off VCC.

## SIMn +0048h Character to character waiting time register — SIMN_SIM_TOUT

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | WTIME[24:0] | | | | | | | | |
| Type | | | | | | | | R/W | | | | | | | | |
| Reset | | | | | | | | 983h | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | WTIME[24:0] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 983h | | | | | | | | | | | | | | | |

**WTIME** Maximum interval between the leading edge of two consecutive characters in 4 ETU unit

## SIMn +004Ch Block to block guard time register — SIMN_SIM_GTIME

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

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

## SIMn +0054h Active High Period Control register — SIMN_SIM_EXT_TIME

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

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

## SIMn +0060h SIM command header register: INS — SIMN_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 will be initiated.

**INSD** [Description for this register field]

**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 — SIMN_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. While the data transfer is going on, this field shows the no. of the remaining data to be sent or to be received

## SIMn +0068h SIM procedure byte register: SW1 — SIMN_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 debug purpose. When the T0END interrupt occurred, it keeps the SW1 procedure byte.

## SIMn +006Ch SIM procedure byte register: SW2 — SIMN_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

**SIMN_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 |

Initial, SIM card is 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** SIM card's ON/OFF indicator

**IR** SIM card is IR (internal reset) card

**AL** SIM card is AL (active low reset) card

### SIMn +0074h SIM protocol state register

**SIMN_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 |

Initial, SIM card's T0 or T1 protocol is turned off. When T0 or T1 protocol is turned on, SIMn_SIM_STATUS will transit between ONE or ALL according to SIM card's procedure byte.

**IDLE** SIM card's T0 or T1 protocol is active or idle.

**ONE** SIM card will send next byte

**ALL** SIM card will send all remained bytes.

## 4.2.2 SIM Card Insertion and Removal

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

## 4.2.3 Card Activation and Deactivation

The card activation and deactivation sequence both are controlled by H/W. The MCU initiates the activation sequence by writing a "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 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 contact deactivation in order that the card is not electrically damaged. The deactivation sequence is initiated by writing a "0" to bit 0 of the SIM_CTRL register, and then the interface performs the following deactivation sequence:

- Assert SIMRST LOW
- Set SCIMCLK at LOW level
- Set SIMDATA at LOW level
- Set SIMVCC at LOW level

### 4.2.4 Answer to Reset Sequence

After card activation, a reset operation results in an answer from the card consisting of the initial character TS, followed by at most 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, MCU should read this character, establish the respective required convention and reprogram the related registers. These processes should be completed prior to the completion of reception of the next character. And then, the remainder of the ATR sequence is 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 33.

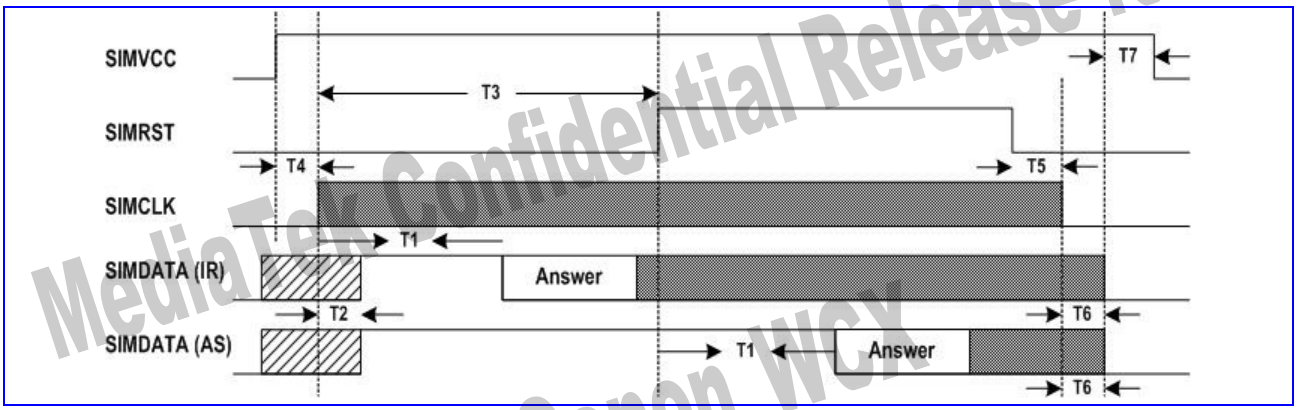



Figure 33 **Answer to Reset Sequence**

| Time | Value | Comment |
|---|---|---|
| T1 | > 400 SIMCLK | SIMCLK start to ATR appear |
| T2 | < 200 SIMCLK | SIMCLK start to SIMDATA in reception mode |
| T3 | > 40000 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 41 **Answer to Reset Sequence Time-Out Condition**

## 4.2.5 SIM Data Transfer

Two transfer modes are provided, either 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 could be enabled to monitor the elapsed time between two consecutive bytes.

### 4.2.5.1 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 Character**

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 character-receive handshaking is disabled, the received-character is written into the SIM FIFO and the SIM_COUNT register is increased by one. Otherwise, the SIMDATA line is held low at 0.5 etu after detecting the parity error for 1.5 etus, and the character is re-received. If a character fails to be received correctly for the RXRETRY times, the receive-handshaking is aborted and the last-received character is written into the SIM FIFO, the SIM_COUNT is increased by one and the RXERR interrupt is generated

When the number of characters held in the receive FIFO exceeds the level defined in the SIM_TIDE register, a RXTIDE interrupt is 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 Character**

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-transmit handshaking is enabled, the SIMDATA line is sampled at 1 etu after the parity bit. If the card indicates that it did not receive the character correctly, the character is retransmitted a maximum of TXRETRY times before a TXERR interrupt is generated and the transmission is aborted. Otherwise, the succeeding byte in the SIM FIFO is transmitted.

If a character fails to be transmitted and a TXERR interrupt is generated, the interface needs to be reset by flushing the SIM FIFO before any subsequent transmit or receive operation.

When the number of characters held in the SIM FIFO falls below the level defined in the SIM_TIDE register, a TXTIDE interrupt is 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.

### 4.2.5.2 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 like in byte transfer mode if necessary and thus the T=0 protocol should be controlled by software.

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

SIM_INS, SIM_P3

SIM_SW1, SIM_SW2

During characters 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 debug purpose.

### Data Receive Instruction

Data Receive Instructions receive data from the SIM card. It is instantiated as the following procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in 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 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 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 SIM_CONF register
8. Start the DMA controller by writing 0x8000 into the DMA*n*_START register

Upon completion of the Data Receive Instruction, T0END interrupt will be generated and then the Time-out counter should be disabled by setting the TOUT bit back to 0 in SIM_CONF register.

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

### Data Send Instruction

Data Send Instructions send data to the SIM card. It is instantiated as the following procedure.

1. Enable the T=0 protocol controller by setting the T0EN bit to 1 in 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 SIM_DATA register
   DMA*n*_COUNT : identical to P3
   DMA*n*_CON : 0x0074
6. Write P3 into 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 SIM_CONF register
8. Start the DMA controller by writing 0x8000 into the DMA*n*_START register

Upon completion of the Data Send Instruction, T0END interrupt will be generated and then the Time-out counter should be disabled by setting the TOUT bit back to 0 in SIM_CONF register.

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

## 4.3 Keypad Scanner

### 4.3.1 General Description

The keypad can be divided into two parts: one is the keypad interface including 8 columns and 8 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 8 x 8 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.

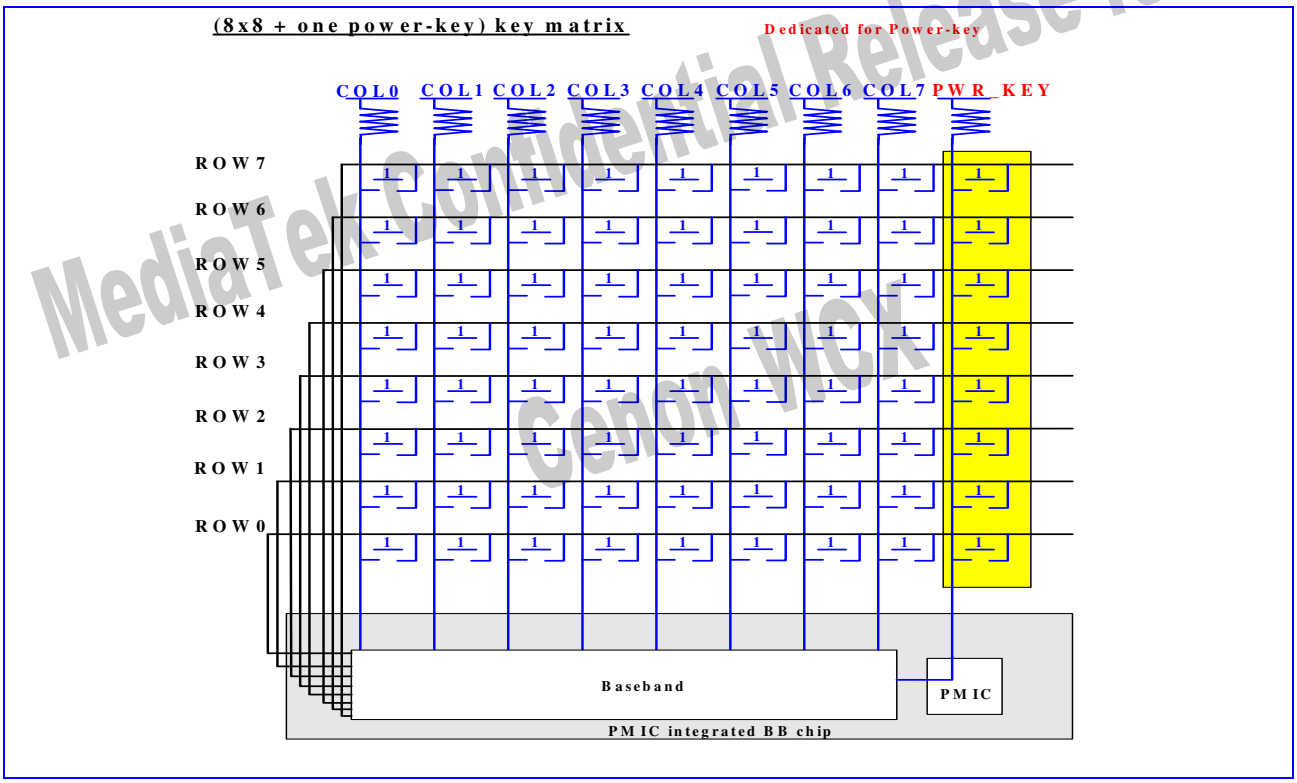



Fig. 1 8x8 matrix with one power-key

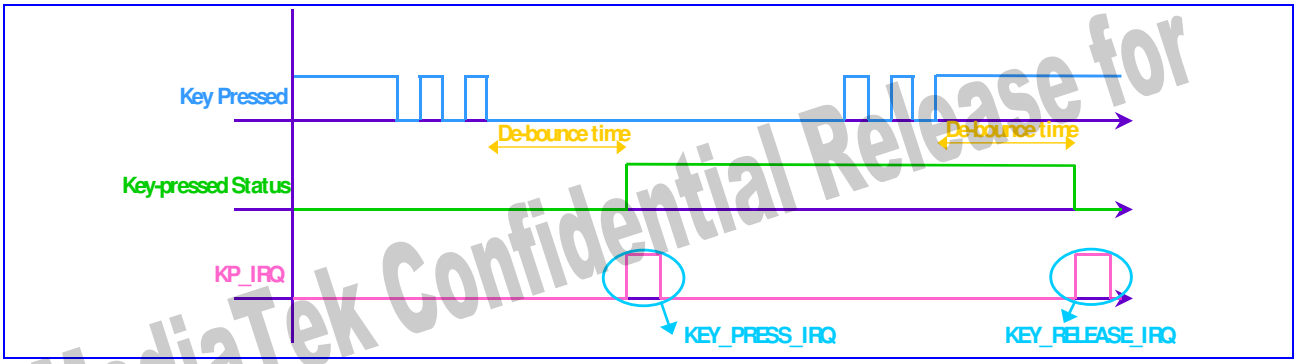



Fig. 2. One key pressed with de-bounce mechanism denoted

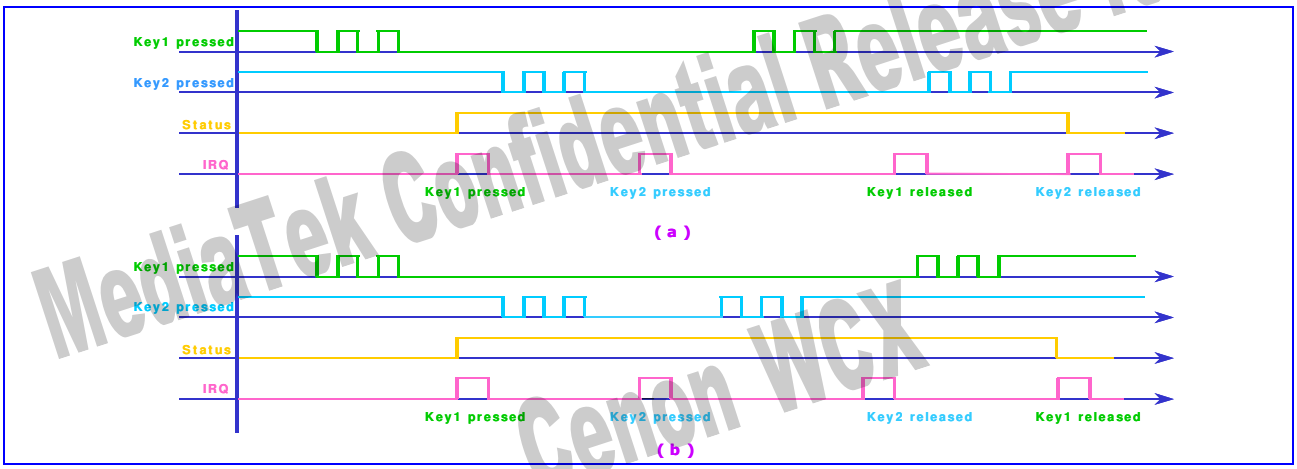



**Fig. 3. (a) Two keys pressed, case 1 (b) Two keys pressed, case 2**

## 4.3.2 Register Definitions

### KP +0000h Keypad status KP_STA

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

**STA** This register indicates the keypad status. The register is not cleared by the read operation.

**0** No key pressed

**1** Key pressed

### KP +0004h Keypad scanning output Register KP_MEM1

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

The register shows up the key-press status of key0(LSB)~key15. Please reference Table 42.

### KP +0008h Keypad scanning output Register KP_MEM2

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

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

The register shows up the key-press status of key16(LSB)~key31. Please reference Table 42.

## KP +000Ch Keypad scanning output Register KP_MEM3

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | KEY 47 | KEY 46 | KEY 45 | KEY 44 | KEY 43 | KEY 42 | KEY 41 | KEY 40 | KEY 39 | KEY 38 | KEY 37 | KEY 36 | KEY 35 | KEY 34 | KEY 33 | KEY 32 |
| 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 |

The register shows up the key-press status of key32(LSB)~key47. Please reference Table 42.

## KP +0010h Keypad scanning output Register KP_MEM4

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | KEY 63 | KEY 62 | KEY 61 | KEY 60 | KEY 59 | KEY 58 | KEY 57 | KEY 56 | KEY 55 | KEY 54 | KEY 53 | KEY 52 | KEY 51 | KEY 50 | KEY 49 | KEY 48 |
| 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 |

The register shows up the key-press status of key48(LSB)~key63. Please reference Table 42.

## KP +0014h Keypad scanning output Register KP_MEM5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | KEY 71 | KEY 70 | KEY 69 | KEY 68 | KEY 67 | KEY 66 | KEY 65 | KEY 64 |
| Type | | | | | | | | | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | | | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

The register shows up the key-press status of key64(LSB)~key71. Please reference Table 42.

## KP +0010h Keypad scanning output Register KP_MEM4

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | KEYS [15:0] | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |
| Reset | 16'hFFFF | | | | | | | | | | | | | | | |

The register shows up the key-press status of key48(LSB)~key63. Please reference Table 42.

## KP +0014h Keypad scanning output Register KP_MEM5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | KEYS [7:0] | | | | | | | |
| Type | | | | | | | | | RO | | | | | | | |
| Reset | | | | | | | | | 8'hFF | | | | | | | |

The register shows up the key-press status of key64(LSB)~key71. Please reference Table 42.

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. 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.

## KP +00018h De-bounce period setting KP_DEBOUNCE

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | DEBOUNCE [13:0] | | | | | | | | | | | | | |
| Type | | | R/W | | | | | | | | | | | | | |
| Reset | | | 400h | | | | | | | | | | | | | |

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.

**DEBOUNCE** De-bounce time = KP_DEBOUNCE/32 ms.

| | 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 42 KEY's order number in COL/ROW matrix.

## 4.4 General Purpose Inputs/Outputs

MT6252 offers 71 general-purpose I/O pins. By setting the control registers, 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 functionalities to reduce the pin count. In addition, all GPO pins are removed. To facilitate application use, software can configure which clock to send outside the chip. There are 6 clock-out ports embedded in 71 GPIO pins, and each clock-out can be programmed to output appropriate clock source.

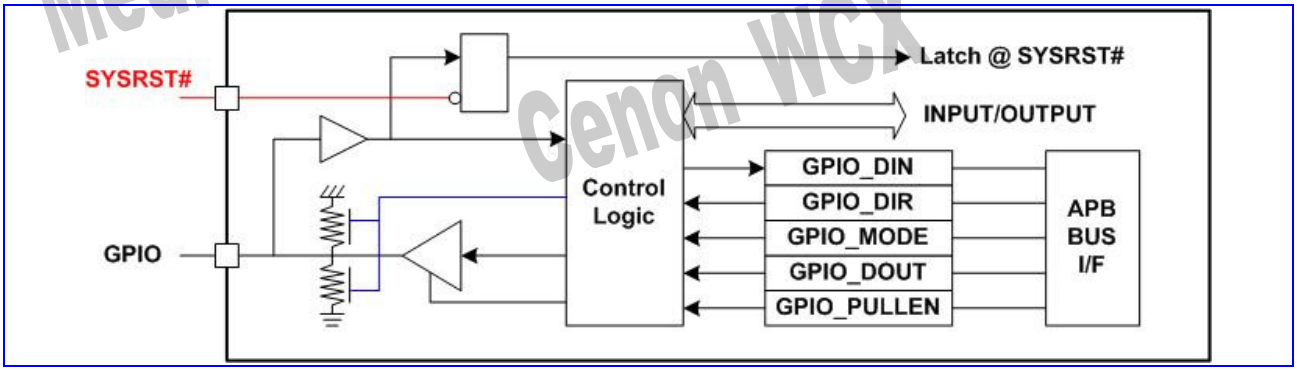



**Figure 34** GPIO Block Diagram

**GPIOs at RESET**

Upon hardware reset (SYSRST#), the digital IOs which are configurated in the mode of aux. function 0 (GPIOs) are all inputs and the following alternative usages of GPIO pins are enabled:

These GPIOs are used to latch the inputs upon reset to memorize the desired configuration to make sure that the system restarts or boots in the right mode.

### 4.4.1 Register Definitions

**0x8002_0000 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 | GPIO 15 | GPIO 14 | GPIO 13 | GPIO 12 | GPIO 11 | GPIO 10 | GPIO 9 | GPIO 8 | GPIO 7 | GPIO 6 | GPIO 5 | GPIO 4 | GPIO 3 | GPIO 2 | GPIO 1 | GPIO 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 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 0x8002_0010 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 | GPIO 31 | GPIO 30 | GPIO 29 | GPIO 28 | GPIO 27 | GPIO 26 | GPIO 25 | GPIO 24 | GPIO 23 | GPIO 22 | GPIO 21 | PGIO 20 | GPIO 19 | GPIO 18 | GPIO 17 | GPIO 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 |

### 0x8002_0020 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 | GPIO 47 | GPIO 46 | GPIO 45 | GPIO 44 | GPIO 43 | GPIO 42 | GPIO 41 | GPIO 40 | GPIO 39 | GPIO 38 | GPIO 37 | GPIO 36 | GPIO 35 | GPIO 34 | GPIO 33 | GPIO 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 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### 0x8002_0030 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 | GPIO 63 | GPIO 62 | GPIO 61 | GPIO 60 | GPIO 59 | GPIO 58 | GPIO 57 | GPIO 56 | GPIO 55 | GPIO 54 | GPIO 53 | GPIO 52 | GPIO 51 | GPIO 50 | GPIO 49 | GPIO 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 |

### 0x8002_0040 GPIO direction control register 5 GPIO_DIR5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESERVED | | | | | | | | | GPIO 70 | GPIO 69 | GPIO 68 | GPIO 67 | GPIO 66 | GPIO 65 | GPIO 64 |
| Type | Reserved | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | Reserved | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**GPIO*n*** GPIO direction control

- **0** GPIOs are configured as input
- **1** GPIOs are configured as output

### 0x8002_0050 GPIO pull-up/pull-down enable register 1 GPIO_PULLE1

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

### 0x8002_0060 GPIO pull-up/pull-down enable register 2 GPIO_PULLEN 2

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

### 0x8002_0070 GPIO pull-up/pull-down enable register 3 GPIO_PULLEN 3

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | GPIO 47 | GPIO 46 | GPIO 45 | GPIO 44 | GPIO 43 | GPIO 42 | GPIO 41 | GPIO 40 | GPIO 39 | GPIO 38 | GPIO 37 | GPIO 36 | GPIO 35 | GPIO 34 | GPIO 33 | GPIO 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 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### 0x8002_0080 GPIO pull-up/pull-down enable register 4 GPIO_PULLEN4

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | GPIO 63 | GPIO 62 | GPIO 61 | GPIO 60 | GPIO 59 | GPIO 58 | GPIO 57 | GPIO 56 | GPIO 55 | GPIO 54 | GPIO 53 | GPIO 52 | GPIO 51 | GPIO 50 | GPIO 49 | GPIO 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 |

### 0x8002_0090 GPIO pull-up/pull-down enable register 5 GPIO_PULLEN5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESERVED | | | | | | | | | GPIO 70 | GPIO 69 | GPIO 68 | GPIO 67 | GPIO 66 | GPIO 65 | GPIO 64 |
| Type | Reserved | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | Reserved | | | | | | | | | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**GPIO*n*** GPIO pull up/down enable. **GPIO29, GPIO30, GPIO31, does not have the pull up/down options.**

**0** GPIOs pull up/down is not enabled

**1** GPIOs pull up/down is enabled

### 0x8002_00A0 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 | 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 |

## 0x8002_00B0 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 | 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 |

## 0x8002_00C0 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 | 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 |

## 0x8002_00D0 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 | 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 |

## 0x8002_00E0 GPIO data inversion control register 5 GPIO_DINV5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESERVED | | | | | | | | | INV 70 | INV 69 | INV 68 | INV 67 | INV 66 | INV 65 | INV 64 |
| Type | Reserved | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | Reserved | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**INVn** GPIO inversion control

**0** GPIOs data inversion disable

**1** GPIOs data inversion enable

## 0x8002_00F0 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 | GPIO 15 | GPIO 14 | GPIO 13 | GPIO 12 | GPIO 11 | GPIO 10 | GPIO 9 | GPIO 8 | GPIO 7 | GPIO 6 | GPIO 5 | GPIO 4 | GPIO 3 | GPIO 2 | GPIO 1 | GPIO 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

### 0x8002_0100 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 | GPIO 31 | GPIO 30 | GPIO 29 | GPIO 28 | GPIO 27 | GPIO 26 | GPIO 25 | GPIO 24 | GPIO 23 | GPIO 22 | GPIO 21 | PGIO 20 | GPIO 19 | GPIO 18 | GPIO 17 | GPIO 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 |

### 0x8002_0110 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 | GPIO 47 | GPIO 46 | GPIO 45 | GPIO 44 | GPIO 43 | GPIO 42 | GPIO 41 | GPIO 40 | GPIO 39 | GPIO 38 | GPIO 37 | GPIO 36 | GPIO 35 | GPIO 34 | GPIO 33 | GPIO 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 |

### 0x8002_0120 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 | GPIO 63 | GPIO 62 | GPIO 61 | GPIO 60 | GPIO 59 | GPIO 58 | GPIO 57 | GPIO 56 | GPIO 55 | GPIO 54 | GPIO 53 | GPIO 52 | GPIO 51 | GPIO 50 | GPIO 49 | GPIO 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 |

### 0x8002_0130 GPIO data output register 5 GPIO_DOUT5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESERVED | | | | | | | | | GPIO 70 | GPIO 69 | GPIO 68 | GPIO 67 | GPIO 66 | GPIO 65 | GPIO 64 |
| Type | Reserved | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | Reserved | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**GPIOn** GPIO data output control

**0** GPIOs data output 0

**1** GPIOs data output 1

### 0x8002_0140 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 | GPIO 15 | GPIO 14 | GPIO 13 | GPIO 12 | GPIO 11 | GPIO 10 | GPIO 9 | GPIO 8 | GPIO 7 | GPIO 6 | GPIO5 | GPIO 4 | GPIO 3 | GPIO 2 | GPIO 1 | GPIO 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

## 0x8002_0150 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 | GPIO 31 | GPIO 30 | GPIO 29 | GPIO 28 | GPIO 27 | GPIO 26 | GPIO 25 | GPIO 24 | GPIO 23 | GPIO 22 | GPIO2 1 | PGIO 20 | GPIO 19 | GPIO 18 | GPIO 17 | GPIO 16 |
| 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 |

## 0x8002_0160 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 | GPIO 47 | GPIO 46 | GPIO 45 | GPIO 44 | GPIO 43 | GPIO 42 | GPIO 41 | GPIO 40 | GPIO 39 | GPIO 38 | GPIO3 7 | GPIO 36 | GPIO 35 | GPIO 34 | GPIO 33 | GPIO 32 |
| 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 |

## 0x8002_0170 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 | GPIO 63 | GPIO 62 | GPIO 61 | GPIO 60 | GPIO 59 | GPIO 58 | GPIO 57 | GPIO 56 | GPIO 55 | GPIO 54 | GPIO5 3 | GPIO 52 | GPIO 51 | GPIO 50 | GPIO 49 | GPIO 48 |
| 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 |

## 0x8002_0180 GPIO data input register 5 GPIO_DIN5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESERVED | | | | | | | | | | GPIO 69 | GPIO 68 | GPIO 67 | GPIO 66 | GPIO 65 | GPIO 64 |
| Type | Reserved | | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | Reserved | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 |

**GPIOn** GPIOs data input

## 0x8002_0210 GPIO Specific Mode 0 GPIO_SPMODE0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | EA15_SPMODE | | EA14_SPMODE | | EA13_SPMODE | | EA12_SPMODE | | EA11_SPMODE | | EA10_SPMODE | | EA9_SPMODE | | EA8_SPMODE | |
| Type | 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 | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | EA7_SPMODE | | EA6_SPMODE | | EA5_SPMODE | | EA4_SPMODE | | EA3_SPMODE | | EA2_SPMODE | | EA1_SPMODE | | EA0_SPMODE | |
| Type | 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 | |

**EA0_SPMODE**

- **00** EMI EA0
- **01** LPA0
- **10** Reserved
- **11** SEN2LCM_A0

**EA1_SPMODE**

- **00** EMI EA1
- **01** LRD_B
- **10** Reserved
- **11** Reserved

**EA2_SPMODE**

- **00** EMI EA2
- **01** LWR_B
- **10** Reserved
- **11** SEN2LCM_WR_B

**EA3_SPMODE**

- **00** EMI EA3
- **01** NLD0
- **10** LSCK
- **11** CMDAT0

**EA4_SPMODE**

- **00** EMI EA4
- **01** NLD1
- **10** LSA0
- **11** CMDAT1

**EA5_SPMODE**

**00** EMI EA5
**01** NLD2
**10** LSDA
**11** CMDAT2

**EA6_SPMODE**
**00** EMI EA6
**01** NLD3
**10** LSDI
**11** CMDAT3

**EA7_SPMODE**
**00** EMI EA7
**01** NLD4
**10** Reserved
**11** CMDAT4

**EA8_SPMODE**
**00** EMI EA8
**01** NLD5
**10** Reserved
**11** CMDAT5

**EA9_SPMODE**
**00** EMI EA9
**01** NLD6
**10** Reserved
**11** CMDAT6

**EA10_SPMODE**
**00** EMI EA10
**01** NLD7
**10** Reserved
**11** CMDAT7

**EA11_SPMODE**
**00** EMI EA11
**01** NLD8
**10** Reserved
**11** Reserved

**EA12_SPMODE**
**00** EMI EA12

**01** NLD9
**10** CMMCLK
**11** CMMCLK

**EA13_SPMODE**
**00** EMI EA13
**01** NLD10
**10** CAM_CSK
**11** CMPCLK

**EA14_SPMODE**
**00** EMI EA14
**01** NLD11
**10** CAM_CSD
**11** HSYNC

**EA15_SPMODE**
**00** EMI EA15
**01** NLD12
**10** CMPDN
**11** VSYNC

### 0x8002_0220 GPIO Specific Mode 1 GPIO_SPMODE1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | | | |
| Type | 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 | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | UTXD1_SPMODE | | URXD1_SPMODE | | RESERVED | ED7_SPMODE | | | RESERVED | ED6_SPMODE | | | EA24_SPMODE | | EA23_SPMODE | |
| Type | 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 | |

**EA23_SPMODE**
**00** EMI EA23
**01** Reserved
**10** Reserved
**11** LSCE0_B

**EA24_SPMODE**
**00** EMI EA24

**01** Reserved
**10** Reserved
**11** LPCE0_B

**ED6_SPMODE**
**000** EMI ED6
**001** JRTCK
**010** CMRST
**011** LSCE1_B
**100** SEN2LCM_CS_B
**101** CAM_SDA
**110** LPCE1_B
**111** Reserved

**ED7_SPMODE**
**000** EMI ED7
**001** JRTCK
**010** CMPDN
**011** LPCE1_B
**100** SEN2LCM_CS_B
**101** CAM_SCL
**110** LSCE1_B
**111** Reserved

**URXD1_SPMODE**
**00** URXD1
**01** EINT2
**10** LSCK
**11** Reserved

**UTXD1_SPMODE**
**00** UTXD1
**01** EINT3
**10** LSDA**11** Reserved

**GPIO+0300h CLK_OUT0 setting** **CLKO_MODE1**

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

**CLKOUT** select the clock output source

- **0** JRTCK
- **1** f26m_ck
- **2** f13m_ck
- **3** f65m_ck, 6.5MHz
- **4** f32k_ck
- **5** dsp1_ck
- **6** dsp2_ck
- **7** mcu_hclk_ck
- **8** ahb_hclk_ck
- **9** slow_ck
- **A** fmcu_ck, PLL output, 104MHz
- **B** fdsp_ck, PLL output, 104MHz
- **C** fusb_ck, USB PHY clock output, 30MHz
- **D** fgsm_ck, GPLL output, 104MHz
- **E** dsp1_gated_ck
- **F** dsp2_gated_ck

### GPIO+0310h CLK_OUT1 setting CLKO_MODE1

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

### GPIO+0320h CLK_OUT2 setting CLKO_MODE2

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

### GPIO+0330h CLK_OUT3 setting CLKO_MODE3

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

| Type | | | | | | | | | | | | | R/W | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset | | | | | | | | | | | | | 0 | | | |

### GPIO+0340h CLK_OUT4 setting CLKO_MODE4

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

### GPIO+0350h CLK_OUT5 setting CLKO_MODE5

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

### GPIO+0360h CLK_OUT6 setting CLKO_MODE6

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

### GPIO+0360h CLK_OUT7 setting CLKO_MODE7

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

### 0x8002_0190 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 | | 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 | | 2 |

**GPIO0_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** PWM1 output
- **010** CLKSQ_SEL
- **011** Reserved
- **100** EMI ED bus 0 monitor
- **101** Serial flash debug output 0
- **110** USB debug output 0
- **111** USB debug output 8

**GPIO1_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** Key-pad column 6, KCOL6
- **010** External interrupt 4, EINT4
- **011** Clock monitor 0
- **100** EMI ED bus 1 monitor
- **101** Serial flash debug output 3
- **110** USB debug output 1
- **111** USB debug output 9

**GPIO2_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** Key-pad column 5, KCOL5
- **010** TDMA event validate
- **011** CAM_SDA
- **100** EMI ED bus 2 monitor
- **101** Serial flash debug output 1
- **110** MIXEDSYS_MON_DATA0
- **111** Reserved

**GPIO3_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** Key-pad column 4, KCOL4

010 TDMA_BTXEN
011 CAM_SCL
100 EMI ED bus 3 monitor
101 Clock monitor 1
110 Reserved
111 Reserved

GPIO4_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad column 3, KCOL3
010 UART1 CTS
011 BSI clock
100 EMI ED bus 4 monitor
101 Serial flash debug output 2
110 MIXEDSYS_MON_DATA0
111 RF_AUXOUT

GPIO5_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad column 2, KCOL2
010 CTIRQ2
011 host DSP ICE data
100 EMI ED bus 5 monitor
101 URXD3
110 Reserved
111 Reserved

GPIO6_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad column 1, KCOL1
010 CTIRQ1
011 Host DSP ICE clock
100 EMI ED bus 6 monitor
101 EDICK
110 USB debug output 2
111 USB debug output 10

**GPIO7_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Key-pad column 0, KCOL0
- **010** DTIRQ
- **011** Host DSP ICE mode-select
- **100** EMI ED bus 7 monitor
- **101** Serial flash debug output 4
- **110** USB debug output 3
- **111** USB debug output 11



MediaTek Confidential Release for Cenon WCX

## 0x8002_01A0 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 | | GPIO15_M | | | | GPIO14_M | | | | GPIO13_M | | | | GPIO12_M | | |
| Type | | R/W | | | | R/W | | | | R/W | | | | R/W | | |
| Reset | | 0 | | | | 1 | | | | 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 | | |

**GPIO8_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Key-pad row 5, KROW5
- **010** External interrupt 5, EINT5
- **011** Clock monitor 3
- **100** EMI ED bus 8 monitor
- **101** EDI DAT
- **110** MIXEDSYS_MON_DATA1
- **111** Reserved

**GPIO9_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Key-pad row 4, KROW4
- **010** TDMA serial data output bit 0
- **011** SRCLKENA | SRECLKENAI
- **100** EMI ED bus 9 monitor
- **101** EDI WS
- **110** Reserved
- **111** Reserved

**GPIO10_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Key-pad row 3, KROW3
- **010** UART1 RTS_B
- **011** Reserved
- **100** LSA0
- **101** Serial flash debug output 5

110 MIXEDSYS_MON_DATA1
111 RF_TEST_EN_DCXO_O

GPIO11_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad row 2, KROW2
010 BTXFS
011 DSP2 ICE data
100 LSCK
101 UTXD3
110 Reserved
111 Reserved

GPIO12_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad row 1, KROW1
010 BRXEN
011 DSP2 ICE clock
100 LSDA
101 Reserved
110 Reserved
111 Reserved

GPIO13_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad row 0, KROW0
010 BRXFS
011 DSP2 IMS
100 LSDI
101 Serial flash debug output 6
110 USB debug output 4
111 USB debug output 12

GPIO14_M GPIO mode selection
000 Configured as GPIO function
001 external memory chip-select 2, ECS2_B

**010** LCD CS0
**011** LSCE0_B
**100** EMI ECS1_B monitor
**101** Reserved
**110** SEN2LCM_CS_B
**111** Reserved

**GPIO15_M** GPIO mode selection
**000** Configured as GPIO function
**001** DAI clock output
**010** TDMA serial data output bit 1
**011** EDI clock
**100** EMI ED[10] monitor
**101** Reserved
**110** Reserved
**111** Reserved

## 0x8002_01B0 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 | | GPIO23_M | | | | GPIO22_M | | | | GPIO21_M | | | | GPIO20_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 | | GPIO19_M | | | | GPIO18_M | | | | GPIO17_M | | | | GPIO16_M | | |
| Type | | R/W | | | | R/W | | | | R/W | | | | R/W | | |
| Reset | | 0 | | | | 0 | | | | 0 | | | | 0 | | |

**GPIO16_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** DAI PCM output
- **010** TDMA_FS
- **011** EDI data
- **100** EMI ED[11] monitor
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO17_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** DAI PCM input
- **010** Reserved
- **011** Reserved
- **100** EMI ED[12] monitor
- **101** Serial flash debug output 7
- **110** MIXEDSYS_MON_DATA 2
- **111** RF_TEST_DCXODELAY

**GPIO18_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** DAI reset
- **010** Reserved
- **011** Reserved
- **100** EMI ED[13] monitor
- **101** Serial flash debug output 8

110 MIXEDSYS_MON_DATA 3
111 RF_TEST_EN_O

GPIO19_M GPIO mode selection
000 Configured as GPIO function
001 DAI sync.
010 TDMA_CK
011 EDI_WS
100 EMI ED[14] monitor
101 Reserved
110 Reserved
111 Reserved

GPIO20_M GPIO mode selection
000 Configured as GPIO function
001 URXD3
010 UART2 flow control, UCTS2_B
011 Reserved
100 EMI EA[18] monitor
101 Serial flash debug output 10
110 USB debug output 5
111 USB debug output 13

GPIO21_M GPIO mode selection
000 Configured as GPIO function
001 UTXD3
010 UART2 flow control, URTS2_B
011 Reserved
100 EMI EA[19] monitor
101 Serial flash debug output 11
110 USB debug output 6
111 USB debug output 14

GPIO22_M GPIO mode selection
000 Configured as GPIO function
001 URXD2

010 UART1 CTS_B
011 CAM_SCL
100 EMI EA[20] monitor
101 Serial flash debug output 12
110 MIXEDSYS_MON_DATA 4
111 DBG_RF_TEST_SCLK_O

GPIO23_M GPIO mode selection
000 Configured as GPIO function
001 UTXD2
010 UART1 RTS_B
011 CAM_SDA
100 EMI EA[21] monitor
101 Clock monitor 2
110 MIXEDSYS_MON_DATA 5
111 DBG_RF_TEST_SDATAI_O

## 0x8002_01C0 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 | | 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 | | 1 | | | | 1 | | | | 0 | | | | 0 | | |

**GPIO24_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** UART1 flow control, UCTS1_B
- **010** CAM_SCL
- **011** LCD data 13
- **100** EMI EUB_B monitor
- **101** EINT5
- **110** Reserved
- **111** Reserved

**GPIO25_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** UART1 flow control, URTS1_B
- **010** CAM_SDA
- **011** LCD data 14
- **100** EMI ELB_B monitor
- **101** EINT6
- **110** Reserved
- **111** Reserved

**GPIO26_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** EINT0
- **010** Reserved
- **011** Clock monitor 4
- **100** Reserved
- **101** Serial flash debug 14

110 MIXEDSYS_MON_DATA 6
111 DBG_RF_TEST_SDATAO_O

GPIO27_M GPIO mode selection
000 Configured as GPIO function
001 EINT1
010 Reserved
011 Clock monitor 3
100 Reserved
101 Serial flash debug 15
110 MIXEDSYS_MON_DATA 7
111 Reserved

GPIO28_M GPIO mode selection
000 Configured as GPIO function
001 BPI_BUS 4
010 Reserved
011 Reserved
100 EMI EADV_B monitor
101 Reserved
110 Reserved
111 Reserved

GPIO29_M GPIO mode selection
000 Configured as GPIO function
001 SIM2 IO
010 Reserved
011 Reserved
100 Reserved
101 Reserved
110 Reserved
111 Reserved

GPIO30_M GPIO mode selection
000 Configured as GPIO function
001 SIM2 CLK

010 Reserved
011 Reserved
100 Reserved
101 Reserved
110 Reserved
111 Reserved

GPIO31_M GPIO mode selection
000 Configured as GPIO function
001 SIM2 RST
010 Reserved
011 Reserved
100 Reserved
101 Reserved
110 Reserved
111 Reserved

## 0x8002_01D0 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 | | 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 | | 1 | | | | 0 | | | | 0 | | | | 0 | | |

**GPIO32_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Reserved
- **010** Reserved
- **011** Reserved
- **100** EMI ED15 monitor
- **101** Serial flash debug output 9
- **110** USB debug output 7
- **111** USB debug output 15

**GPIO33_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Clock monitor 2
- **010** Reserved
- **011** LSCE1_B
- **100** EMI EA16 monitor
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO34_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Clock monitor 6
- **010** SRCLKENA|SRCLKENAI
- **011** Reserved
- **100** EMI EA17 monitor
- **101** Serial flash debug output 13

110 Reserved
111 Reserved

GPIO35_M GPIO mode selection
000 Configured as GPIO function
001 SRECLKENAI
010 Reserved
011 Reserved
100 Reserved
101 Reserved
110 Reserved
111 Reserved

GPIO36_M GPIO mode selection
000 Configured as GPIO function
001 MCINS
010 Reserved
011 LCD data 15
100 EMI ERD_B monitor
101 EINT6
110 Reserved
111 Reserved

GPIO37_M GPIO mode selection
000 Configured as GPIO function
001 MSDC clock output
010 Reserved
011 Reserved
100 EMI ECLK monitor
101 Reserved
110 Reserved
111 DBG_RF_TEST_SCLK_I

GPIO38_M GPIO mode selection
000 Configured as GPIO function
001 MSDC data

010 Reserved
011 Reserved
100 EMI EWAIT monitor
101 Reserved
110 Reserved
111 DBG_RF_TEST_EN_I

GPIO39_M GPIO mode selection
000 Configured as GPIO function
001 MSDC command
010 Reserved
011 Reserved
100 EMI EWR_B monitor
101 Reserved
110 Reserved
111 DBG_RF_TEST_SDATA_I

## 0x8002_01E0 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 | | GPIO47_M | | | | GPIO46_M | | | | GPIO45_M | | | | GPIO44_M | | |
| Type | | R/W | | | | R/W | | | | R/W | | | | R/W | | |
| Reset | | 0 | | | | 4 | | | | 0 | | | | 0 | | |
| 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 | | | | 0 | | | | 0 | | | | 0 | | |

**GPIO40_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** LCD TE
- **010** Reserved
- **011** Reserved
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO41_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** LCD reset
- **010** Reserved
- **011** Reserved
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO42_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Reserved
- **010** LCD_CS1_B
- **011** LSCE1_B
- **100** Reserved
- **101** Reserved

110 SEN2LCM_CS_B
111 EINT2

GPIO43_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad column 7, KCOL7
010 EINT2
011 Clock monitor 2
100 LSCK
101 JRTCK
110 Reserved
111 Reserved

GPIO44_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad row 6, KROW6
010 Reserved
011 Reserved
100 LSDA
101 Reserved
110 Reserved
111 Reserved

GPIO45_M GPIO mode selection
000 Configured as GPIO function
001 Key-pad row 7, KROW7
010 EINT3
011 Clock monitor 5
100 CAM_SDA
101 Reserved
110 Reserved
111 Reserved

GPIO46_M GPIO mode selection
000 Configured as GPIO function
001 Reserved
010 Reserved

011 Reserved
100 JRTCK
101 Reserved
110 Reserved
111 Reserved

GPIO47_M GPIO mode selection
000 Configured as GPIO function
001 Camera data bit 0, CMDAT0
010 CAM_CSD
011 Slave DSP task ID 0
100 Reserved
101 Reserved
110 Reserved
111 Reserved



MediaTek Confidential Release for Cenon WCX

## 0x8002_01F0 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 | | 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 | | |

**GPIO48_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Camera data bit 1, CMDAT1
- **010** LSDA
- **011** Slave DSP task ID 1
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO49_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Camera data bit 2, CMDAT2
- **010** Reserved
- **011** Slave DSP task ID 2
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO50_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** Camera data bit 3, CMDAT3
- **010** Reserved
- **011** Slave DSP task ID 3
- **100** Reserved
- **101** Reserved

**110** Reserved
**111** Reserved

**GPIO51_M** GPIO mode selection
**000** Configured as GPIO function
**001** Camera data bit 4, CMDAT4
**010** Reserved
**011** Slave DSP task ID 4
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

**GPIO52_M** GPIO mode selection
**000** Configured as GPIO function
**001** Camera data bit 5, CMDAT5
**010** Reserved
**011** Slave DSP task ID 5
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

**GPIO53_M** GPIO mode selection
**000** Configured as GPIO function
**001** Camera data bit 6, CMDAT6
**010** Reserved
**011** Slave DSP task ID 6
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

**GPIO54_M** GPIO mode selection
**000** Configured as GPIO function
**001** Camera data bit 7, CMDAT7

010 Reserved
011 Host DSP task ID 0
100 Reserved
101 Reserved
110 Reserved
111 Reserved

GPIO55_M GPIO mode selection
000 Configured as GPIO function
001 CMHREF
010 Reserved
010 Host DSP task ID 1
100 Reserved
101 Reserved
110 Reserved
111 Reserved

## 0x8002_0200 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 | | GPIO63_M | | | | GPIO62_M | | | | GPIO61_M | | | | GPIO60_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 | | GPIO59_M | | | | GPIO58_M | | | | GPIO57_M | | | | GPIO56_M | | |
| Type | | R/W | | | | R/W | | | | R/W | | | | R/W | | |
| Reset | | 0 | | | | 0 | | | | 0 | | | | 0 | | |

**GPIO56_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** CMVREF
- **010** Reserved
- **011** Reserved
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO57_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** CMPDN
- **010** LSCK
- **011** Reserved
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO58_M** GPIO mode selection

- **000** Configured as GPIO function
- **001** CMMCLK
- **010** Reserved
- **011** Reserved
- **100** Reserved

**101** Reserved
**110** Reserved
**111** Reserved

**GPIO59_M** GPIO mode selection
**000** Configured as GPIO function
**001** CMPCLK
**010** CAM_CSK
**011** Reserved
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

**GPIO60_M** GPIO mode selection
**000** Configured as GPIO function
**001** CMRST
**010** Reserved
**011** Reserved
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

**GPIO61_M** GPIO mode selection
**000** Configured as GPIO function
**001** EDI_CK
**010** Reserved
**011** BPI_BUS_5
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

**GPIO62_M** GPIO mode selection
**000** Configured as GPIO function

**001** EDI DAT
**010** Reserved
**011** BPI_BUS_6
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

**GPIO63_M** GPIO mode selection
**000** Configured as GPIO function
**001** EDI WS
**010** Reserved
**011** BPI_BUS_7
**100** Reserved
**101** Reserved
**110** Reserved
**111** Reserved

## 0x8002_0230 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 | | | | | | GPIO70_M | | | | GPIO69_M | | | | GPIO68_M | | |
| Type | | | | | | R/W | | | | R/W | | | | R/W | | |
| Reset | | | | | | 0 | | | | 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 | | | | 1 | | | | 1 | | | | 1 | | |

**GPIO64_M** GPIO mode selection
**000** Configured as GPIO function

001 SCK
010 LSCK
011 Clock monitor 2
100 Reserved
101 Reserved
110 Reserved
111 Reserved

GPIO65_M GPIO mode selection
000 Configured as GPIO function
001 SWP
010 LSA0
011 Reserved
100 Reserved
101 Reserved
110 Reserved
111 Reserved

GPIO66_M GPIO mode selection
000 Configured as GPIO function
001 SHOLD
010 LSCE0_B
011 Reserved
100 Reserved
101 Reserved
110 Reserved
111 Reserved

GPIO67_M GPIO mode selection
000 Configured as GPIO function
001 SCS
010 LSCE1_B
011 Reserved
100 Reserved
101 Reserved
110 Reserved

**111** Reserved

**GPIO68_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** SIN
- **010** LSDI
- **011** EINT3
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO69_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** SOUT
- **010** LSDA
- **011** Reserved
- **100** Reserved
- **101** Reserved
- **110** Reserved
- **111** Reserved

**GPIO70_M** GPIO mode selection
- **000** Configured as GPIO function
- **001** Reserved
- **010** EINT4
- **011** Clock monitor 2
- **100** CAM_SCL
- **101** Reserved
- **110** Reserved
- **111** Reserved



## 0x8002_0240 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 | GPIO 15 | GPIO 14 | GPIO 13 | GPIO 12 | GPIO 11 | GPIO 10 | GPIO 9 | GPIO 8 | GPIO 7 | GPIO 6 | GPIO 5 | GPIO 4 | GPIO 3 | GPIO 2 | GPIO 1 | GPIO 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 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

## 0x8002_0250 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 | GPIO 31 | GPIO 30 | GPIO 29 | GPIO 28 | GPIO 27 | GPIO 26 | GPIO 25 | GPIO 24 | GPIO 23 | GPIO 22 | GPIO 21 | PGIO 20 | GPIO 19 | GPIO 18 | GPIO 17 | GPIO 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 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

## 0x8002_0260 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 | GPIO 47 | GPIO 46 | GPIO 45 | GPIO 44 | GPIO 43 | GPIO 42 | GPIO 41 | GPIO 40 | GPIO 39 | GPIO 38 | GPIO 37 | GPIO 36 | GPIO 35 | GPIO 34 | GPIO 33 | GPIO 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 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

## 0x8002_0270 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 | GPIO 63 | GPIO 62 | GPIO 61 | GPIO 60 | GPIO 59 | GPIO 58 | GPIO 57 | GPIO 56 | GPIO 55 | GPIO 54 | GPIO 53 | GPIO 52 | GPIO 51 | GPIO 50 | GPIO 49 | GPIO 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 |

**GPIO*n*** GPIO pull up/down selection, only valid if corresponding GPIO_PULLEN is high

- **0** GPIOs pull down is selected
- **1** GPIOs pull up is selected

## 0x8002_0280 GPIO Pull Up/Down Select5 GPIO_PULLSEL5

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESERVED | | | | | | | | | GPIO 70 | GPIO 69 | GPIO 68 | GPIO 67 | GPIO 66 | GPIO 65 | GPIO 64 |
| Type | Reserved | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Reset | Reserved | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**GPIO*n*** GPIO pull up/down selection, only valid if corresponding GPIO_PULLEN is high

**0** GPIOs pull down is selected

**1** GPIOs pull up is selected

### GPIO +0370h OE read-back selection GPIO_TM

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

**TM_DIR** Select to read GPIO_DIRx as the real pad output-enable or the MCU-configured GPIO_DIR

**0** MCU-configured

**1** Real pad OE

### GPIO+xxx4h GPIO xxx register SET GPIO_XXX_SET

For all registers addresses listed above, writing to the +4h addresse offset will perform a bit-wise **OR** function between the 16bit written value and the 16bit register value already existing in the corresponding GPIO_xxx registers.

Eg.

If GPIO_DIR1 (GPIO+0000h) = 16'h0F0F,

writing GPIO_DIR1_SET (GPIO+0004h) = 16'F0F0 will result in GPIO_DIR1 = 16'hFFFF.

### GPIO+xxx8h GPIO xxx register CLR GPIO_XXX_CLR

For all registers addresses listed above, writing to the +8h addresse offset will perform a bit-wise **AND-NOT** function between the 16bit written value and the 16bit register value already existing in the corresponding GPIO_xxx registers.

Eg.

If GPIO_DIR1 (GPIO+0000h) = 16'h0F0F,

writing GPIO_DIR1_CLR (GPIO+0008h) = 16'0F0F will result in GPIO_DIR1 = 16'h0000.

## CONFG +0700h Misc usage ACIF_CON0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | LCD_EA_SR | LCD_EA_E8 | LCD_EA_E4 | LCD_EA_E2 | SF_RDSEL | | SF_TDSEL | | SF_IES | SF_SMT | RESERVED | RESERVED | SFE2 | SFSR | SFE8 | SFE4 |
| 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 | | 1 | 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 | BPIBUSSR | LCDSR | LCDE4 | LCDE8 | CMPCLKSMT | CMMCLKSR | CMMCLKE8 | CMMCLKE4 | RESERVED | | BT_POWENSR | BT_POWENE4 | BT_32KSR | BT_32KE4 | D2ICK_SMT | D1ICK_SMT |
| 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 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | | | 0 | 0 | 1 | 0 | 0 | 0 |

**D1ICK_SMT**

- **0** Host DSP ice clock without Schmitt trigger
- **1** Host DSP ice clock with Schmitt trigger.

**D2ICK_SMT**

- **0** Slave DSP ice clock without Schmitt trigger
- **1** Slave DSP ice clock with Schmitt trigger.

**BT_32K E4**

- **0** 2 mA
- **1** 6 mA.

**BT_32K SR**

- **0** Fast slew rate
- **1** Slow slew rate

**BT_POWEN E4**

- **0** 2 mA
- **1** 6 mA.

**BT_ POWEN SR**

- **0** Fast slew rate
- **1** Slow slew rate

**CMMCLK E8**

- **0** + 0 mA
- **1** + 8 mA. CMMCLK default is 4 mA.

**CMMCLK E4**

- **0** + 0 mA

1 + 4 mA. CMMCLK default is 4 mA.

**CMMCLK SR**

0 Fast slew rate

1 Slow slew rate

**CMPCLK SMT**

0 Disable SMT

1 Enable SMT

**LCD E4**

0 + 0 mA

1 + 4 mA. LCD related control pins default is 4 mA.

**LCD E8**

0 + 0 mA

1 + 8 mA. LCD related control pins default is 4 mA.

**LCD SR**

0 Fast slew rate

1 Slow slew rate

**BPIBUS SR** Note that BPIBUS driving current settings are located at BPI module

0 Fast slew rate

1 Slow slew rate

**SF E4** Serial Flash driving current setting

0 + 0 mA

1 + 4 mA. Default is 2 mA.

**SF E8** Serial Flash driving current setting

0 + 0 mA

1 + 8 mA. Default is 2 mA.

**SF SR** Serial Flash output slew rate

0 Fast slew rate

1 Slow slew rate

**SF E2** Serial Flash driving current setting

0 + 0 mA

1 + 2 mA. Default is 2 mA.

**SF_SMT** SF SMT

0 Disable

1 Enable

**SF_IES** SF IES

0 Disable

**1** Enable

**SF_TDSEL** IO_CUP RDSEL option for SF

**00** Default mode

**Others** Experiment mode

**SF_RDSEL** IO_CUP TDSEL option for SF

**00** Default mode

**Others** Experiment mode

**LCD_EA E2** EA[15:12] driving current setting

**0** + 0 mA

**1** + 4 mA. Default is 2 mA.

**LCD_EA E4** EA[15:12] driving current setting

**0** + 0 mA

**1** + 4 mA. Default is 2 mA.

**LCD_EA E8** EA[15:12] driving current setting

**0** + 0 mA

**1** + 8 mA. Default is 2 mA.

**LCD_EA SR** EA[15:12] output slew rate

**0** Fast slew rate

**1** Slow slew rate



**CONFG +0704h** **Misc usage** **ACIF_CON1**

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | LCD_ED_SR | LCD_ED_E8 | LCD_ED_E4 | LCD_ED_E2 | LCD_EA_O_DEL | | | | LCD_EA_RD SEL | | LCD_EA_TD SEL | | LCD_EA_IES | LCD_EA_SMT | LCD_EA_PD | LCD_EA_PU |
| Type | 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 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | RESERVED | KP_COL SR | KP_COL E2 | KP_COL E4 | RESERVED | CM SR | CM E8 | CM E4 | RESERVED | UART SR | UART E4 | UART E8 | RESERVED | EDI SR | EDI E4 | EDI E8 |
| 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 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

**EDI E4** EDI driving current setting
- **0** + 0 mA
- **1** + 4 mA. EDI related control and data pins default is 4 mA.

**EDI E8** EDI driving current setting
- **0** + 0 mA
- **1** + 8 mA. EDI related control and data pins default is 4 mA.

**EDI SR**
- **0** Fast slew rate
- **1** Slow slew rate

**UART E4** UART data and control driving current setting(does not include UCTS1_B/URTS1_B)
- **0** + 0 mA
- **1** + 4 mA. UART related control and data pins default is 4 mA.

**UART E8** UART data and control driving current setting(does not include UCTS1_B/URTS1_B)
- **0** + 0 mA
- **1** + 8 mA. UART related control and data pins default is 4 mA.

**UART SR** UART Output slew rate control (does not include UCTS1_B/URTS1_B)
- **0** Fast slew rate
- **1** Slow slew rate

**CM E8** Camera output signal driving current setting
- **0** + 0 mA
- **1** + 8 mA. Camera related control and data pins default is 4 mA.

**CM E4** Camera output signal driving current setting
- **0** + 0 mA
- **1** + 4 mA. Camera related control and data pins default is 4 mA.

**CM SR**
- **0** Fast slew rate
- **1** Slow slew rate

**KP_COL E2** Key pad column driving current setting
- **0** + 0 mA
- **1** + 2 mA. Default is 2 mA.

**KP_COL E4**
- **0** + 0 mA
- **1** + 4 mA. Default is 2 mA.

**KP_COL SR**
- **0** Fast slew rate
- **1** Slow slew rate

**LCD_EA_PU** EA[15:12] PU
- **0** Disable
- **1** Enable

**LCD_EA_PD** EA[15:12] PD
- **0** Disable
- **1** Enable

**LCD_EA_SMT** EA[15:12] SMT
- **0** Disable
- **1** Enable

**LCD_EA_IES** EA[15:12] IES
- **0** Disable
- **1** Enable

**LCD_EA_TDSEL** IO_CUP RDSEL option for EA[15:12]
- **00** Default mode
- **Others** Experiment mode

**LCD_EA_RDSEL** IO_CUP TDSEL option for EA[15:12]
- **00** Default mode
- **Others** Experiment mode

**LCD_EA_O_DEL** IO_CUP Output delay for EA[15:12]

**LCD_ED E2** ED[7:6] and EA[24:23] driving current setting
- **0** + 0 mA
- **1** + 4 mA. Default is 2 mA.

**LCD_ED E4** ED[7:6] and EA[24:23] driving current setting
- **0** + 0 mA
- **1** + 4 mA. Default is 2 mA.

**LCD_ED E8** ED[7:6] and EA[24:23] driving current setting
- **0** + 0 mA
- **1** + 8 mA. Default is 2 mA.

**LCD_ED SR** ED[7:6] and EA[24:23] output slew rate
- **0** Fast slew rate
- **1** Slow slew rate



**CONFG +0708h** **Misc usage** **ACIF_CON2**

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

| Name | RESERVED | 2V8_I2C SR | 2V8_I2C E4 | 2V8_I2C E2 | RESERVED | KROW_SLCM SR | KROW_SLCM E2 | KROW_SLCM E4 | RESERVED | MCINS SR | MCINS E4 | MCINS E8 | LCD_CSSR | LCD_CSE8 | LCD_CSE4 | LCD_CSE2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | RESERVED | | PWMSR | PWME8 | RESERVED | WATCHDOGSR | WATCHDOGE2 | WATCHDOGE4 | SRCLKENAISR | SRCLKENAIE4 | SD_PWRENSR | SD_PWRENE4 | KP_ROWSR | KP_ROWE4 | CLK_RDSEL | CMD_RDSEL |
| 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 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**CMD_RDSEL** MDDR IO_CUP RDSEL option for CMD_MACRO

- **0** Default mode
- **1** Experiment mode

**CLK_RDSEL** MDDR IO_CUP RDSEL option for CLK_MACRO

- **0** Default mode
- **1** Experiment mode

**KP_ROW E2** Key pad row driving current setting

- **0** + 0 mA
- **1** + 2 mA. Default is 2 mA.

**KP_ROW E4** Key pad row driving current setting

- **0** + 0 mA
- **1** + 4 mA. Default is 2 mA.

**KP_ROW SR**

- **0** Fast slew rate
- **1** Slow slew rate

**SD_PWREN E4** MSDC power enable driving current setting

- **0** + 0 mA
- **1** + 4 mA. Default is 2 mA.

**SD_PWREN SR**

- **0** Fast slew rate
- **1** Slow slew rate

**SRCLKENAI E4** Driving current setting

- **0** + 0 mA
- **1** + 4 mA. Default is 4 mA.

**SRCLKENAI SR**

**0** Fast slew rate
**1** Slow slew rate

**WATCHDOG E2** Driving current setting
**0** + 0 mA
**1** + 2 mA. Default is 2 mA.

**WATCHDOG E4** Driving current setting
**0** + 0 mA
**1** + 4 mA. Default is 2 mA.

**WATCHDOG SR**
**0** Fast slew rate
**1** Slow slew rate

**PWM E8** Driving current setting
**0** + 0 mA
**1** + 8 mA. Default is 4 mA.

**PWM SR**
**0** Fast slew rate
**1** Slow slew rate

**LCD_CS E2** ECS[3:2] driving current setting
**0** + 0 mA
**1** + 4 mA. Default is 2 mA.

**LCD_CS E4** ECS[3:2] driving current setting
**0** + 0 mA
**1** + 4 mA. Default is 2 mA.

**LCD_CS E8** ECS[3:2] driving current setting
**0** + 0 mA
**1** + 8 mA. Default is 2 mA.

**LCD_CS SR** ECS[3:2] output slew rate
**0** Fast slew rate
**1** Slow slew rate

**MCINS E4** MCINS/UCTS1_B/URTS1_B output signal driving current setting
**0** + 0 mA
**1** + 4 mA. Default is 4 mA.

**MCINS E8** MCINS/UCTS1_B/URTS1_B output signal driving current setting
**0** + 0 mA
**1** + 8 mA. Default is 4 mA.

**MCINS SR** MCINS/UCTS1_B/URTS1_B output slew rate control

**0** Fast slew rate

**1** Slow slew rate

**KROW_SLCM E4** KROW[3:0] output signal driving current setting

**0** + 0 mA

**1** + 4 mA. Default is 4 mA.

**KROW_SCLM E8** KROW[3:0] output signal driving current setting

**0** + 0 mA

**1** + 8 mA. Default is 4 mA.

**KROW_SCLM SR** KROW[3:0] output signal slew rate

**0** Fast slew rate

**1** Slow slew rate

**2V8_I2C E2** KROW7 and GPIO70 output signal driving current setting

**0** + 0 mA

**1** + 2 mA. Default is 4 mA.

**2V8_I2C E4** KROW7 and GPIO70 output signal driving current setting

**0** + 0 mA

**1** + 4 mA. Default is 4 mA.

**2V8_I2C SR** KROW7 and GPIO70 output signal slew rate

**0** Fast slew rate

**1** Slow slew rate

## CONFG +070Ch Misc usage ACIF_CON3

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | JRTCK SR | JRTCK E2 | JRTCK E4 | RESERVED | KP_SLCM E8 | KP_SLCM E4 | KP_SLCM E2 |
| Type | 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 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | EMI_IES | MCDPUPD | MCCPUPD | MCCKPUPD | TDSEL | RDSEL | ESERVED | BSI_EXT_SEL | JTDOSR | JTDOE2 | RESERVED | | | DAISR | DAIE2 | DAIE4 |
| 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 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 1 | 1 | | | | 1 | 1 | 0 |

**DAI E2** Driving current setting

**0** + 0 mA

1 + 2 mA. Default is 2 mA.

**DAI E4** Driving current setting

0 + 0 mA

1 + 4 mA. Default is 2 mA.

**DAI SR**

0 Fast slew rate

1 Slow slew rate

**JTDO E2** Driving current setting

0 + 0 mA

1 + 2 mA. Default is 2 mA.

**JTDO SR**

0 Fast slew rate

1 Slow slew rate

**BSI_EXT_SEL** BSI selection

0 Internal BSI

1 External BSI

**RDSEL** IO_CUP RDSEL option for GPIO

0 Default mode

1 Experiment mode

**TDSEL** IO_CUP TDSEL option for GPIO

0 Default mode

1 Experiment mode

**MCCK_PUPD** MCCK pad PUPD port connection

0 Pullup

1 Pulldown

**MCC_PUPD** MSDC command pad PUPD port connection

0 Pullup

1 Pulldown

**MCD_PUPD** MSDC data pads PUPD port connection

0 Pullup

1 Pulldown

**EMI_IES** EMI IES

0 Disable

1 Enable

**KP_SCLM E4** KROW6 and KCOL7 driving current setting

**0** + 0 mA
**1** + 4 mA. Default is 4 mA.

**KP_SCLM E8** KROW6 and KCOL7 driving current setting
**0** + 0 mA
**1** + 8 mA. Default is 4 mA.

**KP_SCLM SR** KROW6 and KCOL7 output signal slew rate
**0** Fast slew rate
**1** Slow slew rate

**JRTCK E4** JRTCK driving current setting
**0** + 0 mA
**1** + 4 mA. Default is 4 mA.

**JRTCK E2** JRTCK driving current setting
**0** + 0 mA
**1** + 2 mA. Default is 4 mA.

**JRTCK SR** JRTCK output signal slew rate
**0** Fast slew rate
**1** Slow slew rate

The pull-up pull-down resistance value for MCCK, MSDC command pad (MCCM0) and MSDC data pads (MCDA0) is determined by the following tables

| PUPD | R0 | R1 | Result |
|---|---|---|---|
| 0 | 0 | 0 | high-Z |
| 0 | 0 | 1 | pull-up with 47K |
| 0 | 1 | 0 | pull-up with 47K |
| 0 | 1 | 1 | pull-up with 23.5K |
| 1 | 0 | 0 | high-Z |
| 1 | 0 | 1 | pull-down with 47K |
| 1 | 1 | 0 | pull-down with 47K |
| 1 | 1 | 1 | pull-down with 23.5K |

**Table 43** MSDC Input Ports Definition

MediaTek Confidential Release for Cenon WCX

| PAD | PUPD | Aux. Func. 0 | Aux. Func. 1 |
|---|---|---|---|
| MCDA0 (GPIO38) | PU 47K | 1. 0x8001070c[14] = 0<br>2. 0x80020070[6] = 1 | 1. 0x8001070c[14] = 0<br>2. 0x81110000[27:26]=0x1 or 0x2 |
| | PU 23.5K | NA | 1. 0x8001070c[14] = 0<br>2. 0x81110000[27:26]=0x3 |
| | High-Z | 1. 0x80020070[6] = 0 | 1. 0x81110000[27:26]=0x0 |
| | PD 47K | 1. 0x8001070c[14] = 1<br>2. 0x80020070[6] = 1 | 1. 0x8001070c[14] = 1<br>2. 0x81110000[27:26]=0x1 or 0x2 |
| | PD 23.5K | NA | 1. 0x8001070c[14] = 1<br>2. 0x81110000[27:26]=0x3 |
| MCCK (GPIO37) | PU 47K | 1. 0x8001070c[12] = 0<br>2. 0x80020070[5] = 1 | 1. 0x8001070c[12] = 0<br>2. 0x80020070[5] = 1 |
| | PU 23.5K | NA | NA |
| | High-Z | 1. 0x80020070[5] = 0 | 1. 0x80020070[5] = 0 |
| | PD 47K | 1. 0x8001070c[12] = 1<br>2. 0x80020070[5] = 1 | 1. 0x8001070c[12] = 1<br>2. 0x80020070[5] = 1 |
| | PD 23.5K | NA | NA |
| MCCM0 (GPIO39) | PU 47K | 1. 0x8001070c[13] = 0<br>2. 0x80020070[7] = 1 | 1. 0x8001070c[13] = 0<br>2. 0x81110000[25:24]=0x1 or 0x2 |
| | PU 23.5K | NA | 1. 0x8001070c[13] = 0<br>2. 0x81110000[25:24]=0x3 |
| | High-Z | 1. 0x80020070[7] = 0 | 1. 0x81110000[25:24]=0x0 |
| | PD 47K | 1. 0x8001070c[13] = 1<br>2. 0x80020070[7] = 1 | 1. 0x8001070c[13] = 1<br>2. 0x81110000[25:24]=0x1 or 0x2 |
| | PD 23.5K | NA | 1. 0x8001070c[13] = 1<br>2. 0x81110000[25:24]=0x3 |

| | | | |
|---|---|---|---|
| MCINS (GPIO36) | PU | 1. 0x80020070[4] = 1<br>2. 0x80020260[4] = 1 | 1. 0x81110014[9:8]=0x2 |
| | High-Z | 1. 0x80020070[4] = 0 | 1. 0x81110014[9:8]=0x0 |
| | PD | 1. 0x80020070[4] = 1<br>2. 0x80020260[4] = 0 | 1. 0x81110014[9:8]=0x1 |

**Table 44** MT6253EL MSDC PADs Configurations



## 4.5 General Purpose Timer

### 4.5.1 General Description

Four general-purpose timers are provided. Three timers are 16 bits long and one timer is 32 bits long and run independently of each other. GPT1~3 use 32k clock source to count, whereas GPT4 uses 26M clock source. 26M clock source could be gated when system enters sleep mode and this will cause GPT4 stop counting, whereas 32k clock source is always toggling. GPT1 and GPT2 can operate in one of two modes: one-shot mode and auto-repeat mode; GPT3 and GPT4 are free running timer. In one-shot mode, when the timer counts down and reaches zero, it is halted. In auto-repeat mode, when the timer reaches zero, it simply resets to countdown initial value and repeats the countdown to zero; this loop repeats until the disable 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 does not take effect until the next time the timer is restarted. In auto-repeat mode, the new countdown start value is used on the next countdown iteration. Therefore, before enabling the gptimer, the desired values for GPTIMER_DAT and the GPTIMER_PRESCALER registers must first be set.

### 4.5.2 Register Definitions

| REGISTER ADDRESS | REGISTER NAME | SYNONYM |
|---|---|---|
| 0x8106_0000 | General Purpose Timer 1 Control Register | GPTIMER1_CON |
| 0x8106_0004 | General Purpose Timer 1 Time-Out Interval Register | GPTIMER1_DAT |
| 0x8106_0008 | General Purpose Timer 2 Control Register | GPTIMER2_CON |
| 0x8106_000C | General Purpose Timer 2 Time-Out Interval Register | GPTIMER2_DAT |
| 0x8106_0010 | General Purpose Timer Status Register | GPTIMER_STA |

| | | |
|---|---|---|
| 0x8106_0014 | General Purpose Timer 1 Prescalar Register | GPTIMER1_PRESCALAR |
| 0x8106_0018 | General Purpose Timer 2 Prescalar Register | GPTIMER2_PRESCALAR |
| 0x8106_001C | General Purpose Timer 3 Control Register | GPTIMER3_CON |
| 0x8106_0020 | General Purpose Timer 3 Time-Out Interval Register | GPTIMER3_DAT |
| 0x8106_0024 | General Purpose Timer 3 Prescalar Register | GPTIMER3_PRESCALAR |
| 0x8106_0028 | General Purpose Timer 4 Control Register | GPTIMER4_CON |
| 0x8106_002C | General Purpose Timer 4 Data Register | GPTIMER4_DAT |

**Table 45** [General Purpose Timer Register Map]

## 0x8106_0000h GPT1 Control register — GPTIMER1_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 | EN | MODE | | | | | | | | | | | | | | |
| Type | R/W | R/W | | | | | | | | | | | | | | |
| Reset | 0 | 0 | | | | | | | | | | | | | | |

**MODE** This register controls GPT1 to count repeatedly (in a loop) or just one-shot.

**0** One-shot mode is selected.

**1** Auto-repeat mode is selected.

**EN** This register controls GPT1 to start counting or to stop.

**0** GPT1 is disabled.

**1** GPT1 is enabled.

## 0x8106_0004h GPT1 Time-Out Interval register — GPTIMER1_DAT

| 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 [15:0] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | FFFFh | | | | | | | | | | | | | | | |

**CNT [15:0]** Initial counting value. GPT1 counts down from GPTIMER1_DAT. When GPT1 counts down to zero, a GPT1 interrupt is generated.

## 0x8106_0008h GPT2 Control register — GPTIMER2_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 | EN | MODE | | | | | | | | | | | | | | |
| Type | R/W | R/W | | | | | | | | | | | | | | |
| Reset | 0 | 0 | | | | | | | | | | | | | | |

**MODE** This register controls GPT2 to count repeatedly (in a loop) or just one-shot.

- **0** One-shot mode is selected
- **1** Auto-repeat mode is selected

**EN** This register controls GPT2 to start counting or to stop.

- **0** GPT2 is disabled.
- **1** GPT2 is enabled.

## 0x8106_000Ch GPT2 Time-Out Interval register — GPTIMER2_DAT

| 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 [15:0] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | FFFFh | | | | | | | | | | | | | | | |

**CNT [15:0]** Initial counting value. GPT2 counts down from GPTIMER2_DAT. When GPT2 counts down to zero, a GPT2 interrupt is generated.

## 0x8106_0010h GPT Status register — GPTIMER_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 | | | | | | | | | | | | | | GPT2 | GPT1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | | | | | | RC | RC |
| Reset | | | | | | | | | | | | | | 0 | 0 |

This register illustrates the gptimer timeout status. Each flag is set when the corresponding timer countdown completes, and can be cleared when the CPU reads the status register.

## 0x8106_0014h GPT1 Prescaler register

**GPTIMER1_PRESCALER**

| 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 [2:0] | | |
| Type | | | | | | | | | | | | | | R/W | | |
| Reset | | | | | | | | | | | | | | 100b | | |

**PRESCALER** This register controls the counting clock for gptimer1.

- **000** 16384 Hz
- **001** 8192 Hz
- **010** 4096 Hz
- **011** 2048 Hz
- **100** 1024 Hz
- **101** 512 Hz
- **110** 256 Hz
- **111** 128 Hz

## 0x8106_0018h GPT2 Prescaler register

**GPTIMER2_PRESCALER**

| 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 [2:0] | | |
| Type | | | | | | | | | | | | | | R/W | | |
| Reset | | | | | | | | | | | | | | 100b | | |

**PRESCALER** This register controls the counting clock for gptimer2.

**000** 16384 Hz
**001** 8192 Hz
**010** 4096 Hz
**011** 2048 Hz
**100** 1024 Hz
**101** 512 Hz
**110** 256 Hz
**111** 128 Hz

## 0x8106_001Ch GPT3 Control register

**GPTIMER3_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 | | | | | | | | | | | | | | | | EN |
| Type | | | | | | | | | | | | | | | | R/W |
| Reset | | | | | | | | | | | | | | | | 0 |

**EN** This register controls GPT3 to start counting or to stop.

**0** GPT3 is disabled.
**1** GPT3 is enabled.

## 0x8106_0020h GPT3 Time-Out Interval register

**GPTIMER3_DAT**

| 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 [15:0] | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |
| Reset | 0 | | | | | | | | | | | | | | | |

**CNT [15:0]** If EN=1, GPT3 is a free running timer. This register records GPT3 value. If EN=0, this register is cleared to 0.

## 0x8106_0024h GPT3 Prescaler register

**GPTIMER3_PRESCALER**

| 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 [2:0] | | |
| Type | | | | | | | | | | | | | | R/W | | |
| Reset | | | | | | | | | | | | | | 100b | | |

**PRESCALER** This register controls the counting clock for gptimer3.

- **000** 16384 Hz
- **001** 8192 Hz
- **010** 4096 Hz
- **011** 2048 Hz
- **100** 1024 Hz
- **101** 512 Hz
- **110** 256 Hz
- **111** 128 Hz

## 0x8106_0028h GPT4 Control register

**GPTIMER4_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 | | | | | | | | | | | | | | | LOCK | EN |
| Type | | | | | | | | | | | | | | | R/W | R/W |
| Reset | | | | | | | | | | | | | | | 0 | 0 |

**EN** This register controls GPT4 to start counting or to stop.

- **0** GPT4 is disabled.
- **1** GPT4 is enabled.

**LOCK** This register controls GPT4 EN bit can be modified or not

- **0** No lock. Software can configure EN bit
- **1** Lock. Software cannot program EN bit until reset

### 0x8106_002Ch GPT4 Data register

**GPTIMER4_DAT**

| 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 | | | | | | | | | | | | | | | |
| 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 | | | | | | | | | | | | | | | |

**CNT [31:0]** If EN=1, GPT4 is a free running timer. This register records GPT4 value. If EN=0, this register is cleared to 0.

This register is not allowed continuous read. Need at least 1 26M clock cycle between 2 APB reads.

# 4.6 UART

## 4.6.1 General Description

The baseband chipset houses three UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external devices.

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.

**Error! Reference source not found.**Fig 7 shows the block diagram of the UART device.

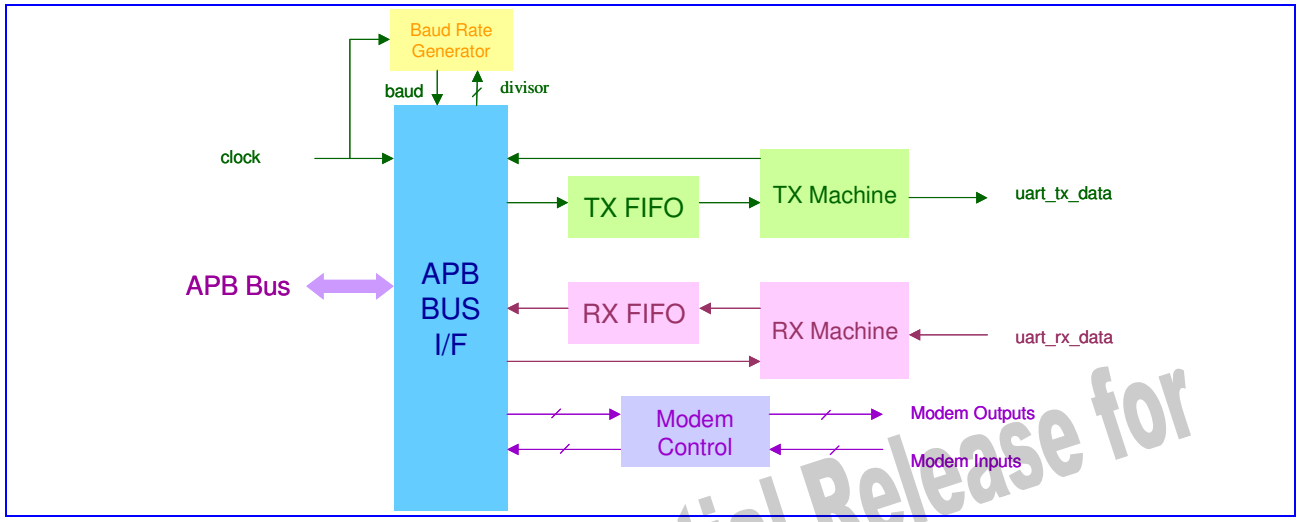



**Figure 35** Block Diagram of UART

## 4.6.2 Register Definitions

UART1 Register Mapping Table:

| MCU Register Addr. | Register Function | Acronym |
|---|---|---|
| 81030000h | RX Buffer Register | UARTn_RBR |

| | | |
|---|---|---|
| 81030000h | TX Holding Register | UARTn_THR |
| 81030004h | Interrupt Enable Register | UARTn_IER |
| 81030008h | Interrupt Identification Register | UARTn_IIR |
| 81030008h | FIFO Control Register | UARTn_FCR |
| 8103000Ch | Line Control Register | UARTn_LCR |
| 81030010h | Modem Control Register | UARTn_MCR |
| 81030014h | Line Status Register | UARTn_LSR |
| 81030018h | Modem Status Register | UARTn_MSR |
| 8103001Ch | Scratch Register | UARTn_SCR |
| 81030000h | Divisor Latch (LS) | uartn_dll |
| 81030004h | Divisor Latch (MS) | uartn_dlM |
| 81030008h | Enhanced Feature Register | UARTn_EFR |
| 81030010h | XON1 | UARTn_XON1 |
| 81030014h | XON2 | UARTn_XON2 |
| 81030018h | XOFF1 | UARTn_XOFF1 |
| 8103001Ch | XOFF2 | UARTn_XOFF2 |
| 81030020h | AUTOBAUD_EN | UARTn_AUTOBAUD_EN |
| 81030024h | HIGH SPEED UART | UARTn_HIGHSPEED |
| 81030028h | SAMPLE_COUNT | UARTn_SAMPLE_COUNT |
| 8103002Ch | SAMPLE_POINT | UARTn_SAMPLE_POINT |

| | | |
|---|---|---|
| 81030030h | AUTOBAUD_REG | UARTn_AUTOBAUD_REG |
| 81030034h | Rate Fix Address | UARTn_RateFix_ad |
| 81030038h | AUTOBAUDSAMPLE | UARTn_AUTOBAUDSAMPLE |
| 8103003Ch | Guard time added register | UARTn_GUARD |
| 81030040h | Escape character register | UARTn_ESCAPE_DAT |
| 81030044h | Escape enable register | UARTn_ESCAPE_EN |
| 81030048h | Sleep enable register | UARTn_SLEEP_EN |
| 8103004Ch | Virtual FIFO enable register | UARTn_VFIFO_EN |
| 81030050h | Rx Trigger Address | UARTn_RXTRI_AD |
| 81030054h | Fractional Divider LSB Address | UARTn_Fracdiv_l |
| 81030058h | Fractional Divider MSB Address | UARTn_FRACDIV_M |

UART2 Register Mapping Table:

| MCU Register Addr. | Register Function | Acronym |
|---|---|---|
| 81040000h | RX Buffer Register | UARTn_RBR |
| 81040000h | TX Holding Register | UARTn_THR |
| 81040004h | Interrupt Enable Register | UARTn_IER |
| 81040008h | Interrupt Identification Register | UARTn_IIR |

| | | |
|---|---|---|
| 81040008h | FIFO Control Register | UARTn_FCR |
| 8104000Ch | Line Control Register | UARTn_LCR |
| 81040010h | Modem Control Register | UARTn_MCR |
| 81040014h | Line Status Register | UARTn_LSR |
| 81040018h | Modem Status Register | UARTn_MSR |
| 8104001Ch | Scratch Register | UARTn_SCR |
| 81040000h | Divisor Latch (LS) | uartn_dll |
| 81040004h | Divisor Latch (MS) | uartn_dlM |
| 81040008h | Enhanced Feature Register | UARTn_EFR |
| 81040010h | XON1 | UARTn_XON1 |
| 81040014h | XON2 | UARTn_XON2 |
| 81040018h | XOFF1 | UARTn_XOFF1 |
| 8104001Ch | XOFF2 | UARTn_XOFF2 |
| 81040020h | AUTOBAUD_EN | UARTn_AUTOBAUD_EN |
| 81040024h | HIGH SPEED UART | UARTn_HIGHSPEED |
| 81040028h | SAMPLE_COUNT | UARTn_SAMPLE_COUNT |
| 8104002Ch | SAMPLE_POINT | UARTn_SAMPLE_POINT |
| 81040030h | AUTOBAUD_REG | UARTn_AUTOBAUD_REG |
| 81040034h | Rate Fix Address | UARTn_RateFix_ad |
| 81040038h | AUTOBAUDSAMPLE | UARTn_AUTOBAUDSAMPLE |
| 8104003Ch | Guard time added | UARTn_GUARD |

| | | |
|---|---|---|
| | register | |
| 81040040h | Escape character register | UARTn_ESCAPE_DAT |
| 81040044h | Escape enable register | UARTn_ESCAPE_EN |
| 81040048h | Sleep enable register | UARTn_SLEEP_EN |
| 8104004Ch | Virtual FIFO enable register | UARTn_VFIFO_EN |
| 81040050h | Rx Trigger Address | UARTn_RXTRI_AD |
| 81040054h | Fractional Divider LSB Address | UARTn_Fracdiv_l |
| 81040058h | Fractional Divider MSB Address | UARTn_FRACDIV_M |

UART3 Register Mapping Table:

| MCU Register Addr. | Register Function | Acronym |
|---|---|---|
| 81050000h | RX Buffer Register | UARTn_RBR |
| 81050000h | TX Holding Register | UARTn_THR |
| 81050004h | Interrupt Enable Register | UARTn_IER |
| 81050008h | Interrupt Identification Register | UARTn_IIR |
| 81050008h | FIFO Control Register | UARTn_FCR |
| 8105000Ch | Line Control Register | UARTn_LCR |
| 81050010h | Modem Control Register | UARTn_MCR |

| | | |
|---|---|---|
| 81050014h | Line Status Register | UARTn_LSR |
| 81050018h | Modem Status Register | UARTn_MSR |
| 8105001Ch | Scratch Register | UARTn_SCR |
| 81050000h | Divisor Latch (LS) | uartn_dll |
| 81050004h | Divisor Latch (MS) | uartn_dlM |
| 81050008h | Enhanced Feature Register | UARTn_EFR |
| 81050010h | XON1 | UARTn_XON1 |
| 81050014h | XON2 | UARTn_XON2 |
| 81050018h | XOFF1 | UARTn_XOFF1 |
| 8105001Ch | XOFF2 | UARTn_XOFF2 |
| 81050020h | AUTOBAUD_EN | UARTn_AUTOBAUD_EN |
| 81050024h | HIGH SPEED UART | UARTn_HIGHSPEED |
| 81050028h | SAMPLE_COUNT | UARTn_SAMPLE_COUNT |
| 8105002Ch | SAMPLE_POINT | UARTn_SAMPLE_POINT |
| 81050030h | AUTOBAUD_REG | UARTn_AUTOBAUD_REG |
| 81050034h | Rate Fix Address | UARTn_RateFix_ad |
| 81050038h | AUTOBAUDSAMPLE | UARTn_AUTOBAUDSAMPLE |
| 8105003Ch | Guard time added register | UARTn_GUARD |
| 81050040h | Escape character register | UARTn_ESCAPE_DAT |
| 81050044h | Escape enable register | UARTn_ESCAPE_EN |

| | | |
|---|---|---|
| 81050048h | Sleep enable register | UARTn_SLEEP_EN |
| 8105004Ch | Virtual FIFO enable register | UARTn_VFIFO_EN |
| 81050050h | Rx Trigger Address | UARTn_RXTRI_AD |
| 81050054h | Fractional Divider LSB Address | UARTn_Fracdiv_l |
| 81050058h | Fractional Divider MSB Address | UARTn_FRACDIV_M |

n = 1, 2, 3; for uart1, uart2 and uart3 respectively.

### UARTn+0000h RX Buffer Register — UARTn_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. Read-only register. The received data can be read by accessing this register. Modified when LCR[7] = 0.

### UARTn+0000h TX Holding Register — UARTn_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. Write-only register. The data to be transmitted is written to this register, and then sent to the PC via serial communication.
Modified when LCR[7] = 0.

### UARTn+0004h Interrupt Enable Register — UARTn_IER

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | CTSI | RTSI | XOFFI | VFF_FC_EN | EDSSI | ELSI | ETBEI | ERBFI |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

**IER** By storing a '1' to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt is disabled.

IER[3:0] are modified when LCR[7] = 0.

IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.

**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.

**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.

**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.

**VFF_FC_EN** Enable flow control triggered by RX FIFO full when VFIFO_EN is set.

**EDSSI** When set ("1"), an interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.

**0** No interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.

**1** An interrupt is generated if DDCD, TERI, DDSR or DCTS (MSR[4:1]) becomes set.

**ELSI** When set ("1"), an interrupt is 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.

**ETBEI** When set ("1"), an interrupt is 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 is 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 is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level

**ERBFI** When set ("1"), an interrupt is generated if the RX Buffer contains data.

**0** No interrupt is generated if the RX Buffer contains data.

**1** An interrupt is generated if the RX Buffer contains data.

## UARTn+0008h Interrupt Identification Register — UARTn_IIR

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | FIFOE | | ID4 | ID3 | ID2 | ID1 | ID0 | NINT |
| Type | | | | | | | | | RO | | | | | | | |
| Reset | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**IIR** Identify if there are pending interrupts; ID4 and ID3 are presented only when EFR[4] = 1.
The following table gives the IIR[5:0] codes associated with the possible interrupts:

| IIR[5:0] | Priority Level | Interrupt | Source |
|---|---|---|---|
| 000001 | - | No interrupt pending | |
| 000110 | 1 | Line Status Interrupt | BI, FE, PE or OE set in LSR<br>(Under IER[2]=1) |
| 000100 | 2 | RX Data Received | RX Data received or RX Trigger Level reached.<br>(Under IER[0]=1) |
| 001100 | 2 | RX Data Timeout | Timeout on character in RX FIFO.<br>(Under IER[0]=1) |
| 000010 | 3 | TX Holding Register Empty | TX Holding Register empty or TX FIFO Trigger Level reached.<br>(Under IER[1]=1) |
| 000000 | 4 | Modem Status change | DDCD, TERI, DDSR or DCTS set in MSR<br>(Under IER[3]=1) |
| 010000 | 5 | Software Flow Control | XOFF Character received<br>(Under IER[5]=1) |
| 100000 | 6 | Hardware Flow Control | CTS or RTS Rising Edge<br>(Under IER[6]=1, EDR[6]=1) |

**Table 46** The IIR[5:0] codes associated with the possible interrupts

Line Status Interrupt: A RX Line Status Interrupt (IIR[5:0`] == 000110b) is 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) is generated if EFRBI (IER[0]) is set and either RX Data is 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 Timeout Interrupt:

When virtual FIFO mode is disabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO contains at least one character;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register, or on a CPU read from the RX FIFO.

The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1, and is cleared by reading RX FIFO.

When virtual FIFO mode is enabled, RX Data Timeout Interrupt is generated if all of the following apply:

1. FIFO is empty;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.

The timeout timer is restarted on receipt of a new byte from the RX Shift Register.

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, the TX FIFO becomes empty. The interrupt is cleared by writing to the TX Holding Register or TX FIFO if FIFO enabled.

Modem Status Change Interrupt: A Modem Status Change Interrupt (IIR[5:0] = 000000b) is generated if EDSSI (IER[3]) is set and either DDCD, TERI, DDSR or DCTS (MSR[3:0]) 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) is generated if 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) is generated if 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.

## UARTn+0008h FIFO Control Register — UARTn_FCR

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | RFTL1 | RFTL0 | TFTL1 | TFTL0 | DMA1 | CLRT | CLRR | FIFOE |
| Type | | | | | | | | | WO | | | | | | | |

**FCR** 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

**FCR[7:6]** RX FIFO trigger threshold. (RX FIFO contains total 24 bytes.)

**0** 1

**1** 6
**2** 12
**3** **RXTRIG**

**FCR[5:4]** TX FIFO trigger threshold (TX FIFO contains total 16 bytes.)
**0** 1
**1** 4
**2** 8
**3** 14 (FIFOSIZE - 2)

**DMA1** This bit determines the DMA mode, which the TXRDY and RXRDY pins support. TXRDY and RXRDY act to support single-byte transfers between the UART and memory (DMA mode 0) or multiple byte transfers (DMA mode1). Note that this bit has no effect unless the FIFOE bit is set as well
**0** The device operates in DMA Mode 0.
**1** The device operates in DMA Mode 1.
TXRDY – mode0: Goes active (low) when the TX FIFO or the TX Holding Register is empty. And it becomes inactive when a byte is written to the Transmit channel.
TXRDY – mode1: Goes active (low) when there are no characters in the TX FIFO. And it becomes inactive when the TX FIFO is full.
RXRDY – mode0: Becomes active (low) when at least one character is in the RX FIFO or the RX Buffer Register is full. And it becomes inactive when there are no more characters in the RX FIFO or RX Buffer register.
RXRDY – mode1: Becomes active (low) when the RX FIFO Trigger Level is reached or an RX FIFO Character Timeout occurs. And it goes inactive when the RX FIFO is empty.

**CLRT** Clear Transmit FIFO. This bit is self-clearing.
**0** Leave TX FIFO intact.
**1** Clear all the bytes in the TX FIFO.

**CLRR** Clear Receive FIFO. This bit is self-clearing.
**0** Leave RX FIFO intact.
**1** Clear all the bytes in the RX FIFO.

**FIFOE** FIFO Enabled. This bit must be set to 1 for any of the 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.

## UARTn+000Ch Line Control Register — UARTn_LCR

| 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 | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**LCR** Line Control Register. Determines characteristics of serial communication signals. Modified when LCR[7] = 0.

**DLAB** Divisor Latch Access Bit.

- **0** The RX and TX Registers are read/written at Address 0 and the IER register is read/written at Address 4.
- **1** The Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.

**SB** Set Break

- **0** No effect
- **1** SOUT signal is forced into the "0" state.

**SP** Stick Parity

- **0** No effect.
- **1** The Parity bit is forced into a defined state, depending on the states of EPS and PEN:
  If EPS=1 & PEN=1, the Parity bit is set and checked = 0.
  If EPS=0 & PEN=1, the Parity bit is set and checked = 1.

**EPS** Even Parity Select

- **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.

**PEN** Parity Enable

- **0** The Parity is neither transmitted nor checked.
- **1** The Parity is transmitted and checked.

**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.

**WLS1, 0** Word Length Select.

- **0** 5 bits
- **1** 6 bits
- **2** 7 bits
- **3** 8 bits

## UARTn+0010h Modem Control Register — UARTn_MCR

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | XOFF STATUS | | DCM_EN | LOOP | OUT2 | OUT1 | RTS | DTR |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**MCR** Modem Control Register. Control 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.

**XOFF Status** This is a read-only bit.
- **0** When an XON character is received.
- **1** When an XOFF character is received.

**DCM_EN** DCM Function enable
- **0** DCM function is forbidden.
- **1** DCM function is permit.

**LOOP** Loop-back control bit.
- **0** No loop-back is enabled.
- **1** Loop-back mode is enabled.

**OUT2** Controls the state of the output NOUT2, even in loop mode.
- **0** NOUT2=1.
- **1** NOUT2=0.

**OUT1** Controls the state of the output NOUT1, even in loop mode.
- **0** NOUT1=1.
- **1** NOUT1=0.

**RTS** Controls the state of the output NRTS, even in loop mode.
- **0** NRTS=1.
- **1** NRTS=0.

**DTR** Control the state of the output NDTR, even in loop mode.
- **0** NDTR=1.
- **1** NDTR=0.

## UARTn+0014h Line Status Register UARTn_LSR

| 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 | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

**LSR** Line Status Register.
Modified when LCR[7] = 0.

**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.

**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 is set whenever the TX FIFO and the TX Shift Register are empty. If FIFOs are disabled, the bit is set whenever TX Holding Register and TX Shift Register are empty.

**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).

**BI** Break Interrupt.

0 Reset by the CPU reading this register

1 If the FIFOs are disabled, this bit is 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 is associated with a corresponding character in the FIFO and is 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.

**FE** Framing Error.

0 Reset by the CPU reading this register

1 If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit. If the FIFOs are enabled, the state of this bit is revealed when the byte it refers to is the next to be read.

**PE** Parity Error

0 Reset by the CPU reading this register

1 If the FIFOs are disabled, this bit is set if the received data did not have a valid parity bit. If the FIFOs are enabled, the state of this bit is revealed when the referred byte is the next to be read.

**OE** Overrun Error.

0 Reset by the CPU reading this register.

1 If the FIFOs are disabled, this bit is set if the RX Buffer was not read by the CPU before new data from the RX Shift Register overwrote the previous contents.
If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX Shift Register becomes full. OE is set as soon as this happens. The character in the Shift Register is then overwritten, but not transferred to the FIFO.

**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.

## UARTn+0018h Modem Status Register — UARTn_MSR

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

| Name | | | | | | | | | DCD | RI | DSR | CTS | DDCD | TERI | DDSR | DCTS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | | | | | | Input | Input | Input | Input | 0 | 0 | 0 | 0 |

Note: After a reset, D4-D7 are inputs. A modem status interrupt can be cleared by writing '0' or set by writing '1' to this register. D0-D3 can be written to.

Modified when LCR[7] = 0.

**MSR** Modem Status Register

**DCD** Data Carry Detect.

When Loop = "0", this value is the complement of the NDCD input signal.

When Loop = "1", this value is equal to the OUT2 bit in the Modem Control Register.

**RI** Ring Indicator.

When Loop = "0", this value is the complement of the NRI input signal.

When Loop = "1", this value is equal to the OUT1 bit in the Modem Control Register.

**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.

**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.

**DDCD** Delta Data Carry Detect.

**0** The state of DCD has not changed since the Modem Status Register was last read

**1** Set if the state of DCD has changed since the Modem Status Register was last read.

**TERI** Trailing Edge Ring Indicator

**0** The NRI input does not change since this register was last read.

**1** Set if the NRI input changes from "0" to "1" since this register was last read.

**DDSR** Delta Data Set Ready

**0** Cleared if the state of DSR has not changed since this register was last read.

**1** Set if the state of DSR has changed since this register was last read.

**DCTS** Delta Clear To Send

**0** Cleared if the state of CTS has not changed since this register was last read.

**1** Set if the state of CTS has changed since this register was last read.

## UARTn+001Ch Scratch Register — UARTn_SCR

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

A general purpose read/write register. After reset, its value is un-defined.

Modified when LCR[7] = 0.

## UARTn+0000h Divisor Latch (LS) — UARTn_DLL

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

## UARTn+0004h Divisor Latch (MS) — UARTn_DLM

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

Note: DLL & DLM can only be updated if DLAB is set (“1”).. Note too that division by 1 generates a BAUD signal that is constantly high.

Modified when LCR[7] = 1.

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 13, 26 MHz and 52 MHz. The effective clock enable generated is 16 x the required baud rate.

| BAUD | 6.5MHz | 13MHz | 26MHz | 52MHz |
|---|---|---|---|---|
| 110 | 3693 | 7386 | 14773 | 29545 |
| 300 | 1354 | 2708 | 5417 | 10833 |
| 1200 | 338 | 677 | 1354 | 2708 |
| 2400 | 169 | 338 | 677 | 1354 |
| 4800 | 85 | 169 | 339 | 677 |
| 9600 | 42 | 85 | 169 | 339 |
| 19200 | 21 | 42 | 85 | 169 |
| 38400 | 11 | 21 | 42 | 85 |
| 57600 | 7 | 14 | 28 | 56 |

| | | | | |
|---|---|---|---|---|
| 115200 | * | 6 | 14 | 28 |

**Table 47** Divisor needed to generate a given baud rate

## UARTn+0008h Enhanced Feature Register — UARTn_EFR

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | AUTO CTS | AUTO RTS | | ENABLE-E | SW FLOW CONT[3:0] | | | |
| Type | | | | | | | | | R/W | R/W | | R/W | R/W | | | |
| Reset | | | | | | | | | 0 | 0 | | 0 | 0 | | | |

*NOTE: Only when LCR=BF'h

**Auto CTS** Enables hardware transmission flow control
- **0** Disabled.
- **1** Enabled.

**Auto RTS** Enables hardware reception flow control
- **0** Disabled.
- **1** Enabled.

**Enable-E** Enable enhancement features.
- **0** Disabled.
- **1** Enabled.

**CONT[3:0]** 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

## UARTn+0010h XON1 — UARTn_XON1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | XON1[7:0] | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

### UARTn+0014h XON2 UARTn_XON2

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | XON2[7:0] | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

### UARTn+0018h XOFF1 UARTn_XOFF1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | XOFF1[7:0] | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

### UARTn+001Ch XOFF2 UARTn_XOFF2

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | XOFF2[7:0] | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

*Note: XON1, XON2, XOFF1, XOFF2 are valid only when LCR=BFh.

### UARTn+0020h AUTOBAUD_EN UARTn_AUTOBAUD_EN

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

**AUTOBAUD_EN** Auto-baud enable signal

- **0** Auto-baud function disable
- **1** Auto-baud function enable (UARTn+0024h SPEED should be set 0)

### UARTn+0024h HIGH SPEED UART UARTn_HIGHSPEED

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

**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 / {DLM, DLL}

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 13 MHz based on different HIGHSPEED value.

| BAUD | HIGHSPEED = 0 | HIGHSPEED = 1 | HIGHSPEED = 2 |
|---|---|---|---|
| 110 | 7386 | 14773 | 29545 |
| 300 | 2708 | 7386 | 14773 |
| 1200 | 677 | 2708 | 7386 |
| 2400 | 338 | 677 | 2708 |
| 4800 | 169 | 338 | 677 |
| 9600 | 85 | 169 | 338 |
| 19200 | 42 | 85 | 169 |
| 38400 | 21 | 42 | 85 |
| 57600 | 14 | 21 | 42 |
| 115200 | 7 | 14 | 21 |
| 230400 | * | 7 | 14 |
| 460800 | * | * | 7 |
| 921600 | * | * | * |

**Table 48** Divisor needed to generate a given baud rate from 13MHz based on different HIGHSPEED value

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 26 MHz based on different HIGHSPEED value.

| BAUD | HIGHSPEED = 0 | HIGHSPEED = 1 | HIGHSPEED = 2 |
|---|---|---|---|
| 110 | 14773 | 29545 | 59091 |
| 300 | 5417 | 14773 | 29545 |
| 1200 | 1354 | 5417 | 14773 |
| 2400 | 677 | 1354 | 5417 |
| 4800 | 339 | 677 | 1354 |
| 9600 | 169 | 339 | 667 |
| 19200 | 85 | 169 | 339 |

| | | | |
|---|---|---|---|
| 38400 | 42 | 85 | 169 |
| 57600 | 28 | 42 | 85 |
| 115200 | 14 | 28 | 42 |
| 230400 | 7 | 14 | 28 |
| 460800 | * | 7 | 14 |
| 921600 | * | * | 7 |

**Table 49** Divisor needed to generate a given baud rate from 26 MHz based on different HIGHSPEED value

The table below shows the divisor needed to generate a given baud rate from CLK inputs of 52MHz based on different HIGHSPEED value.

| BAUD | HIGHSPEED = 0 | HIGHSPEED = 1 | HIGHSPEED = 2 |
|---|---|---|---|
| 110 | 29545 | 59091 | 118182 |
| 300 | 10833 | 29545 | 59091 |
| 1200 | 2708 | 10833 | 29545 |
| 2400 | 1354 | 2708 | 10833 |
| 4800 | 677 | 1354 | 2708 |
| 9600 | 339 | 677 | 1354 |
| 19200 | 169 | 339 | 677 |
| 38400 | 85 | 169 | 339 |
| 57600 | 56 | 85 | 169 |
| 115200 | 28 | 56 | 85 |
| 230400 | 14 | 28 | 56 |
| 460800 | 7 | 14 | 28 |
| 921600 | * | 7 | 14 |

**Table 50**
Divisor needed to generate a given baud rate from 52 MHz based on different HIGHSPEED value

### UARTn+0028h SAMPLE_COUNT UARTn_SAMPLE_COUNT

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

When HIGHSPEED=3, the sample_count is the threshold value for UART sample counter (sample_num).

**Count from 0 to sample_count.**

### UARTn+002Ch SAMPLE_POINT UARTn_SAMPLE_POINT

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

When HIGHSPEED=3, UART gets the input data when sample_count=sample_num.

e.g. system clock = 13MHz, 921600 = 13000000 / 14

sample_count = 14 and sample point = 6 (sample the central point to decrease the inaccuracy)

**The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 and remove the decimal.**

### UARTn+0030h AUTOBAUD_REG UARTn_AUTOBAUD_REG

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | BAUD_STAT[3:0] | | | | BAUDRATE[3:0] | | | |
| Type | | | | | | | | | RO | | | | RO | | | |
| Reset | | | | | | | | | 0 | | | | 0 | | | |

**BAUD_RATE** Autobaud baud rate

- **0** 115200
- **1** 57600
- **2** 38400
- **3** 19200
- **4** 9600
- **5** 4800
- **6** 2400
- **7** 1200
- **8** 300
- **9** 110

**BAUDSTAT** 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

## UARTn+0034h Rate Fix Address UARTn_RATEFIX_AD

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | RESTRICT | FREQ_SEL | AUTOBAUD_RATE_FIX | RXTE_FIX |
| Type | | | | | | | | | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | | | | | | | | | 0 | 0 | 0 | 0 |

**RATE_FIX** When you set "rate_fix"(34H[0]), you can transmit and receive data only if
1) the f13m_en is enable and the freq_sel (34H[2]) is set to 1, or
2) the f26m_en is enable and the freq_sel (34H[2]) is set to 0.

**AUTOBAUD_RATE_FIX** When you set "autobaud_rate_fix"(34H[1]), you can tx/rx the autobaud packet only if
1) the f13m_en is enable and the freq_sel (34H[2]) is set to 1, or
2) the f26m_en is enable and the freq_sel (34H[2]) is set to 0.

**FREQ_SEL**
**0** Select f26m_en for rate_fix and autobaud_rate_fix
**1** Select f13m_en for rate_fix and autobaud_rate_fix

**RESTRICT** The "restrict" (34H[3]) is used to set a more condition for the autobaud fsm starting point

## UARTn+0038h AUTOBAUDSAMPLE

UARTn_AUTOBAUDSAMPLE

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

Since the system clock may change, autobaud sample duration should change as system clock changes.
When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13.

## UARTn+003Ch Guard time added register

UARTn_GUARD

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

**GUARD_CNT** Guard interval count value. Guard interval = (1/(system clock / **div_step** / div )) * GUARD_CNT.

**GUARD_EN** Guard interval add enable signal.

- **0** No guard interval added.
- **1** Add guard interval after stop bit.

## UARTn+0040h Escape character register

UARTn_ESCAPE_DAT

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

**ESCAPE_DAT** Escape character added before software flow control data and escape character, i.e. if tx data is xon (31h), with esc_en =1, uart transmits data as esc + CEh (~xon).

## UARTn+0044h Escape enable register

UARTn_ESCAPE_EN

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

**ESC_EN** Add escape character in transmitter and remove escape character in receiver by UART.

- **0** Do not deal with the escape character.

1 Add escape character in transmitter and remove escape character in receiver.

## UARTn+0048h Sleep enable register — UARTn_SLEEP_EN

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

**SLEEP_EN** For sleep mode issue

0 Do not deal with sleep mode indicate signal

1 To activate hardware flow control or software control according to software initial setting when chip enters sleep mode. Releasing hardware flow when chip wakes up; but for software control, uart sends xon when awaken and when FIFO does not reach threshold level.

## UARTn+004Ch RX Virtual FIFO enable register — UARTn_RXVFF_EN

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

**RXVFF_EN** UART RX Virtual FIFO mechanism enable signal.

0 Disable RX VFIFO mode.

1 Enable RX VFIFO mode. When UART RX virtual mode is enabled, the flow control is based on the DMA threshold, and generates a timeout interrupt for DMA.

## UARTn+0050h Rx Trigger Address — UARTn_RXTRI_AD

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

**RXTRIG** When {rtm,rtl}=2'b11, The Rx FIFO threshold will be Rxtrig.
The value is suggested to be less than half of RX FIFO size, which is 24 Bytes.

## UARTn+0054h Fractional Divider LSB Address — UARTn_FRACDIV_L

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

| Name | | | | | | | | | FRACDIV_L | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**FRACDIV_L** Add sampling count (+1) from state data7 to data0, in order to contribute fractional divisor.

### UARTn+0058h Fractional Divider MSB Address — UARTn_FRACDIV_M

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

**FRACDIV_M** Add sampling count when in state stop to parity, in order to contribute fractional divisor.

### UARTn+005Ch — FIFO Control Register — UARTn_FCR_RD

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | RFTL1 | RFTL0 | TFTL1 | TFTL0 | DMA1 | CLRT | CLRR | FIFOE |
| Type | | | | | | | | | | | | | | | RO | |

Please refer to UARTn_FCR register.

## 4.7 Real Time Clock

### 4.7.1 General Description

The Real Time Clock (RTC) module provides time and data information. The clock is based on a 32.768KHz oscillator with 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 via the BBWAKEUP pin. 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.

### 4.7.2 Register Definitions

| REGISTER ADDRESS | REGISTER NAME | SYNONYM |
|---|---|---|

| | | |
|---|---|---|
| 0x810c_0000 | Baseband power up | RTC_BBPU |
| 0x810c_0004 | RTC IRQ status | RTC_IRQ_STA |
| 0x810c_0008 | RTC IRQ enable | RTC_IRQ_EN |
| 0x810c_000C | Counter increment IRQ enable | RTC_CII_EN |
| 0x810c_0010 | RTC alarm mask | RTC_AL_MASK |
| 0x810c_0014 | RTC seconds time counter register | RTC_TC_SEC |
| 0x810c_0018 | RTC minutes time counter register | RTC_TC_MIN |
| 0x810c_001C | RTC hours time counter register | RTC_TC_HOU |
| 0x810c_0020 | RTC day-of-month time counter register | RTC_TC_DOM |
| 0x810c_0024 | RTC day-of-week time counter register | RTC_TC_DOW |
| 0x810c_0028 | RTC month time counter register | RTC_TC_MTH |
| 0x810c_002C | RTC year time counter register | RTC_TC_YEA |
| 0x810c_0030 | RTC second alarm setting register | RTC_AL_SEC |
| 0x810c_0034 | RTC minute alarm setting register | RTC_AL_MIN |
| 0x810c_0038 | RTC hour alarm setting register | RTC_AL_HOU |
| 0x810c_003C | RTC day-of-month alarm setting register | RTC_AL_DOM |
| 0x810c_0040 | RTC day-of-week alarm setting register | RTC_AL_DOW |
| 0x810c_0044 | RTC month alarm setting register | RTC_AL_MTH |
| 0x810c_0048 | RTC year alarm setting register | RTC_AL_YEA |
| 0x810c_004C | XOSC bias current control register | RTC_XOSCCALI |
| 0x810c_0050 | RTC_POWERKEY1 register | RTC_POWERKEY1 |
| 0x810c_0054 | RTC_POWERKEY2 register | RTC_POWERKEY2 |
| 0x810c_0058 | PDN1 | RTC_PDN1 |
| 0x810c_005C | PDN2 | RTC_PDN2 |
| 0x810c_0064 | Spare register for specific purpose | RTC_SPAR1 |
| 0x810c_0068 | Lock / unlock scheme to prevent RTC miswriting | RTC_PROT |
| 0x810c_006c | One-time calibration offset | RTC_DIFF |
| 0x810c_0070 | Repeat calibration offset | RTC_CALI |
| 0x810c_0074 | Enable the transfers from core to RTC in the queue | RTC_WRTGR |

**Table 51** RTC Register Map

## 0x810c_0000 Baseband power up RTC_BBPU

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | KEY_BBPU | | | | | | | | DBING | CBUSY | RELOAD | CLRPKY | AUTO | BBPU | WRITE_EN | PWREN |
| Type | WO | | | | | | | | RO | RO | WO | WO | R/W | R/W | R/W | R/W |

**KEY_BBPU** A bus write is acceptable only when KEY_BBPU=0x43.

**DBING** This bit indicates RTC is still de-bouncing.

**CBUSY** The read/write channels between RTC / Core is busy. This bit indicates high after software program sequence to anyone of RTC data registers and enable the transfer by RTC_WRTGR=1. By the way, it is high after the reset from low to high because RTC reload process.

**RELOAD** Reload the values from RTC domain to Core domain. Generally speaking, RTC will reload synchronize the data from RTC to core when reset from 0 to 1. This bit can be treated as debug bit.

**CLRPKY** Clear powerkey1 and powerkey2 at the same time. In some cases, software may clear powerkey1 & powerkey2. The BBWAKEUP depends on the matching specific patterns of powerkey1 and powerkey2. If any one of powerkey1 or powerkey2 or BBPU is cleared, BBWAKEUP goes low immediately. Software can't program the other control bits without power. By program RTC_BBPU with CLRPKY=1 and BBPU=0 condition, RTC can clear powerkey1, powerkey2 and BBPU at the same moment.

**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 from high to low.
- **1** BBWAKEUP is automatically in the low state when SYSRST# transitions from high to low.

**BBPU** Controls the power of PMIC. If powerkey1=A357h and powerkey2=67D2h, PMIC takes on the value programmed by software; otherwise PMIC is low.

- **0** Power down
- **1** Power on

**WRITE_EN** When WRITE_EN is write 0 by the MCU, the RTC programing interface is disabled immediately (MCU can't program RTC). After the debounce counter is time-out, the interface enabled again (MCU can program RTC). The debounce counter time-out period is decided by RTC_PDN1. Note that the WRITE_EN value read out is meaningless. The hardware only care about the "write-0 action" to WRITE_EN control bit.

When WRITE_EN==0, avoid to "read out RTC_BBPU, AND/OR something and write back", like this -> *RTC_BBPU=*RTC_BBPU|RTC_BBPU_KEY|0x1. This would disable RTC write interface for a while and hard to debug.

**PWREN**

- **0** RTC alarm has no action on power switch.
- **1** When an RTC alarm occurs, BBPU is set to 1 and the system powers on by RTC alarm wakeup.

### 0x810c_0004 RTC IRQ status RTC_IRQ_STA

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | | TCSTA | ALSTA |
| Type | | | | | | | | | | | | | | | R/C | R/C |

**ALSTA** This register 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.

**TCSTA** This register 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.

### 0x810c_0008 RTC IRQ enable RTC_IRQ_EN

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | ONESHOT | TC_EN | AL_EN |
| Type | | | | | | | | | | | | | | R/W | R/W | R/W |

The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

**ONESHOT** Controls automatic reset of AL_EN and TC_EN.

**AL_EN** This register 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.

**TC_EN** This register 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 when ONESHOT is high upon generation of the corresponding IRQ.

### 0x810c_000C Counter increment IRQ enable RTC_CII_EN

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | 1/8SECCII | 1/4SECCII | 1/2SECCII | YEACII | MTHCII | DOWCII | DOMCII | HOUCII | MINCII | SECCII |
| Type | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.

**SECCII** Set this bit to 1 to activate the IRQ at each second update.

**MINCII** Set the bit to 1 to activate the IRQ at each minute update.
**HOUCII** Set the bit to 1 to activate the IRQ at each hour update.
**DOMCII** Set the bit to 1 to activate the IRQ at each day-of-month update.
**DOWCII** Set the bit to 1 to activate the IRQ at each day-of-week update.
**MTHCII** Set the bit to 1 to activate the IRQ at each month update.
**YEACII** Set the bit to 1 to activate the IRQ at each year update.
**1/2SECCII** Set the bit to 1 to activate the IRQ at each one-half of a second update.
**1/4SECCII** Set the bit to 1 to activate the IRQ at each one-fourth of a second update.
**1/8SECCII** Set the bit to 1 to activate the IRQ at each one-eighth of a second update.

## 0x810c_0010 RTC alarm mask RTC_AL_MASK

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | YEA_MSK | MTH_MSK | DOW_MSK | DOM_MSK | HOU_MSK | MIN_MSK | SEC_MSK |
| Type | | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

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 1 in RTC_AL_MASK (i.e. RTC_AL_MASK=0x7f) and PWREN=1 in RTC_BBPU, it means alarm comes EVERY SECOND, not disabled.

**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.

**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.

**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.

**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.

**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.

**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.

**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.



## 0x810c_0014 RTC seconds time counter register RTC_TC_SEC

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

**TC_SECOND** The second initial value for the time counter. The range of its value is: 0-59.

## 0x810c_0018 RTC minutes time counter register RTC_TC_MIN

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

**TC_MINUTE** The minute initial value for the time counter. The range of its value is: 0-59.

## 0x810c_001C RTC hours time counter register RTC_TC_HOU

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

**TC_HOUR** The hour initial value for the time counter. The range of its value is: 0-23.

## 0x810c_0020 RTC day-of-month time counter register RTC_TC_DOM

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

**TC_DOM** The 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.

### 0x810c_0024 RTC day-of-week time counter register — RTC_TC_DOW

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

**TC_DOW** The day-of-week initial value for the time counter. The range of its value is: 1-7.

### 0x810c_0028 RTC month time counter register — RTC_TC_MTH

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

**TC_MONTH** The month initial value for the time counter. The range of its value is: 1-12.

### 0x810c_002C RTC year time counter register — RTC_TC_YEA

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

**TC_YEAR** The year initial value for the time counter. The range of its value is: 0-127. (2000-2127)

### 0x810c_0030 RTC second alarm setting register — RTC_AL_SEC

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

**AL_SECOND** The second value of the alarm counter setting. The range of its value is: 0-59.

### 0x810c_0034 RTC minute alarm setting register — RTC_AL_MIN

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

**AL_MINUTE** The minute value of the alarm counter setting. The range of its value is: 0-59.

## 0x810c_0038 RTC hour alarm setting register RTC_AL_HOU

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

**AL_HOUR** The hour value of the alarm counter setting. The range of its value is: 0-23.

## 0x810c_003C RTC day-of-month alarm setting register RTC_AL_DOM

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

**AL_DOM** The 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.

## 0x810c_0040 RTC day-of-week alarm setting register RTC_AL_DOW

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

**AL_DOW** The day-of-week value of the alarm counter setting. The range of its value is: 1-7.

## 0x810c_0044 RTC month alarm setting register RTC_AL_MTH

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

**AL_MONTH** The month value of the alarm counter setting. The range of its value is: 1-12.

## 0x810c_0048 RTC year alarm setting register RTC_AL_YEA

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

**AL_YEAR** The year value of the alarm counter setting. The range of its value is: 0-127. (2000-2127)

## 0x810c_004C XOSC bias current control register — RTC_XOSCCALI

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | XOSCCALI | | | | |
| Type | | | | | | | | | | | | WO | | | | |

The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

**XOSCCALI** This register controls the XOSC32 bias current.

## 0x810c_0050 RTC_POWERKEY1 register — RTC_POWERKEY1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RTC_POWERKEY1 | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |

## 0x810c_0054 RTC_POWERKEY2 register — RTC_POWERKEY2

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RTC_POWERKEY2 | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |

These register sets are used to determine if the real time clock has been programmed by 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, therefore the time values shown are incorrect. Software needs to know if the real time clock has been programmed. Hence, these two registers are defined to solve this power-on issue. After software programs the correct value, these two register sets do not need to be updated. In addition to programming the correct time value, when the contents of these register sets are wrong, the interrupt is not generated. Therefore, the real time clock does not generate the interrupts before the software programs the registers; unwanted interrupt due to wrong time value do not occur. The correct values of these two register sets are:

**RTC_POWERKEY1** A357h
**RTC_POWERKEY2** 67D2h

## 0x810c_0058 PDN1 — RTC_PDN1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | RTC_PDN1[7:0] | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |

**RTC_PDN1[3:1]** is for reset de-bounce mechanism. When RTC_POWERKEY1 & RTC_POWERKEY2 do not match the correct values, RTC_PDN1[3:1] is set to 3 (011 in binary).

- **0** 2ms
- **1** 8ms
- **2** 32ms
- **3** 128ms
- **4** 256ms
- **5** 512ms
- **6** 1024ms
- **7** 2048ms

**RTC_PDN1[7:4] & RTC_PDN1[0]** is the spare register for software to keep power on and power off state information.

## 0x810c_005C PDN2 RTC_PDN2

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | RTC_PDN2[7:0] | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |

**RTC_PDN2** The spare register for software to keep power on and power off state information.

## 0x810c_0064 Spare register for specific purpose RTC_SPAR1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RTC_SPAR1 | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |

**RTC_SPAR1** This register is reserved for specific purpose.

## 0x810c_0068 Lock / unlock scheme to prevent RTC miswriting RTC_PROT

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

**RTC_PROT** The RTC write interface is protected by RTC_PROT. Whether the RTC writing interface is enabled or not is decided by RTC_PROT contents. When RTC_POWERKEY1 & RTC_POWERKEY2 are not equal to the correct values, the RTC writing interface is always enabled. But when they match, users have to perform Unlock flow to enable the writing interface.

Unlock flow: Step1: *RTC_PROT=0x586a;
Step2: *RTC_WRTRG=1;
Step3: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec
Step4: *RTC_PROT=0x9136;
Step5: *RTC_WRTRG=1;
Step6: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec

Lock flow: Step1: *RTC_PROT=0x0;
Step2: *RTC_WRTRG=1;
Step3: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec

Once the normal RTC content writing is complete, it is suggested to perform Lock flow to turn off the interface to avoid accident writing.
The RTC_PROT contents will be corrupt when reset.

## 0x810c_006c One-time calibration offset RTC_DIFF

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

The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

**RTC_DIFF** These registers are used to adjust the internal counter of RTC. It effects once and returns to zero in done.

In some cases, you observe the RTC is faster or slower than the standard. To change RTC_TC_SEC is coarse and may cause alarm problem. RTC_DIFF provides a finer time unit. An internal 15-bit counter accumulates in each 32768-HZ clock. Entering a non-zero value into the RTC_DIFF causes the internal RTC counter increases or decreases RTC_DIFF when RTC_DIFF changes to zero again. RTC_DIFF represents as 2's completement form.

For example, if you fill in 0xfff into RTC_DIFF, the internal counter decreases 1 when RTC_DIFF returns to zero. In other words, you can only use RTC_DIFF continuously if RTC_DIFF is equal to zero now.

Note: RTC_DIFF ranges from 0x800 (-2048) to 0x7fd (2045). 0x7ff & 0x7fe are forbid to use.

## 0x810c_0070 Repeat calibration offset RTC_CALI

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

| Type | | | | | | | | | | R/W |
|---|---|---|---|---|---|---|---|---|---|---|

The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

**RTC_CALI** These registers provide a repeat calibration scheme. RTC_CALI provides 7-bit calibration capability in 8-second duration; in other words, 5-bit calibration capability in each second. RTC_CALI represents in 2's complement form, such that you can adjust RTC increasing or decreasing.

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)

### 0x810c_0074 Enable the transfers from core to RTC in the queue RTC_WRTGR

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | | | WRTGR |
| Type | | | | | | | | | | | | | | | | WO |

**WRTGR** This register enables the transfers from core to RTC. After you modify all the RTC registers you'd like to change, you must write RTC_WRTGR to 1 to trigger the transfer. The prior writing operations are queued at core power domain. The pending data will not be transferred to RTC domain until WRTGR=1.
After WRTGR=1, the pending data is transferred to RTC domain sequentially in order of register address, from low to high. For example: RTC_BBPU -> RTC_IRQ_EN -> RTC_CII_EN -> RTC_AL_MASK -> RTC_TC_SEC -> etc. The CBUSY in RTC_BBPU is equal to 1 in writing process. You can observe CBUSY to determine when the transmission completes.

## 4.8 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. Seven input channels allow diverse applications in this unit.

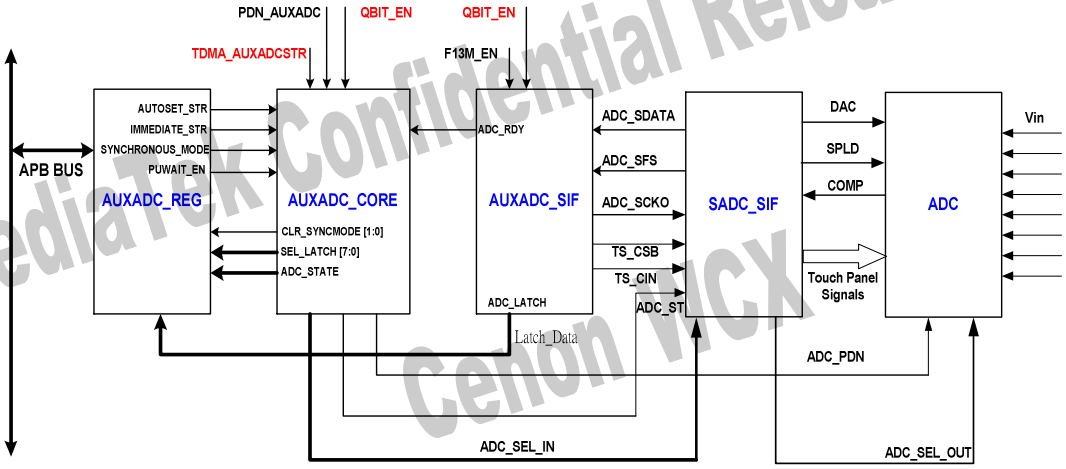



Figure 36 Auxadc Architecture

Each channel can operate in one of two modes: immediate mode and timer-triggered mode. The mode of each channel can be individually selected through register AUXADC_CON0. For example, if the flag SYN0 in the register AUXADC_CON0 is set, the channel 0 is set in timer-triggered mode. Otherwise, the channel operates in immediate mode.

In immediate mode, the A/D converter samples the value once only when the flag in the AUXADC_CON1 register has been set. For example, if the flag IMM0 in AUXADC_CON1 is set, the A/D converter samples 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, the value for channel 1 is stored in register AUXADC_DAT1, etc.

If the AUTOSET flag in the register AUXADC_CON3 is set, the auto-sample function is enabled. The A/D converter samples the data for the channel in which the corresponding data register has been read. For example, in the case where the SYN1 flag is not set, the AUTOSET flag is set, when the data register AUXADC_DAT0 has been read, the A/D converter samples the next value for channel 1 immediately.

If multiple channels are selected at the same time, the task is performed sequentially on every selected channel. For example, if AUXADC_CON1 is set to 0x7f, that is, all 7 channels are selected, the state machine in the unit starts sampling from channel 6 to channel 0, and saves the values of each input channel in the respective registers. The same process also applies in timer-triggered mode.

In 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 the register TDMA_AUXEV1, which is placed in the TDMA timer. For example, if AUXADC_CON0 is set to 0x7f, all 7 channels are selected to be in timer-triggered mode. The state machine samples all 7 channels sequentially and save the values in registers from AUXADC_DAT0 to AUXADC_DAT6, as it does in immediate mode.

There is a dedicated timer-triggered scheme for channel 0. This scheme is enabled by setting the SYN7 flag in the register AUXADC_CON2. The timing offset for this event is stored in the register TDMA_AUXEV0 in the TDMA timer. The sampled data triggered by this specific event is stored in the register AUXADC_DAT7. It is used to separate the results of two individual software routines that perform actions on the auxiliary ADC unit.

The AUTOCLR*n* in the register AUXADC_CON3 is set when it is intended to sample only once after setting timer-triggered mode. If AUTOCLR1 flag has been set, after the data for the channels in timer-triggered mode has been stored, the SYN*n* flags in the register AUXADC_CON0 are cleared. If AUTOCLR0 flag has been set, after the data for the channel 0 has been stored in the register AUXADC_DAT7, the SYN7 flag in the register AUXADC_CON2 is cleared.

The usage of the immediate mode and timer-triggered mode are mutually exclusive in terms of individual channels.

The PUWAIT_EN bit in the registers AUXADC_CON3 is used to power up the analog port in advance. This ensures that the power has ramped up to the stable state before A/D converter starts the conversion. The analog part is automatically powered down after the conversion is completed.

There are only two external pins (channel 4~5) for voltage detection. The other channels (0~3) are for battery voltage, battery current, and charger, respectively.

Touch Panel:

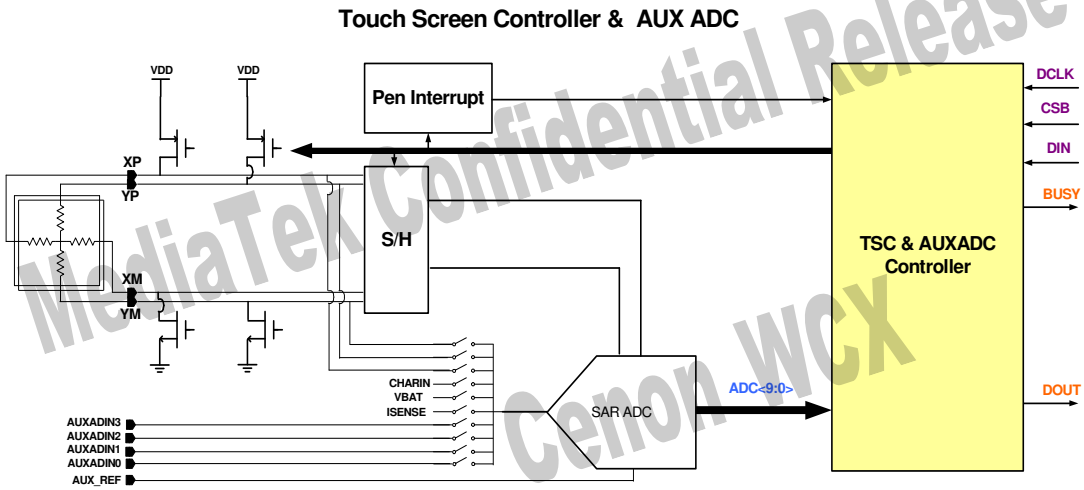



Figure 37 Touch Panel Circuit Structure

Besides the normal sampling of external input voltage, auxadc includes the sampling of the touch panel function. For the specified axis, SW should program **AUX_TS_CMD** first, and then trigger touch panel's sample in the register **AUX_TS_CON**. The touch panel sampling waveform is shown as follows. After SW polls status bit in the register **AUXADC_CON3** to know that the touch panel sample is finished. SW can read back the specified axis value from the register **AUX_TS_DAT0**.

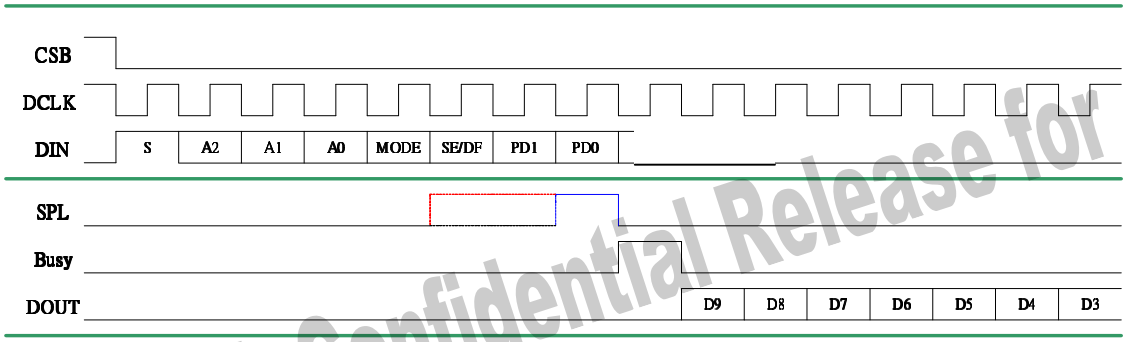



Figure 38 Touch Panel Sampling Waveform

S: Start bit

A2~A0: Addressing bits

Mode: 10bit or 8bit

SE/DF: Single End or Differential mode

PD1~0: Power Down Command

These values are defined in the register **AUX_TS_CMD**. In the following table, it shows the relationship between AUX_TS_CMD and touch panel control signals.

| SE/DFB | A2 | A1 | A0 | X/Y Driver | X/Y Pass | SEL< 3:0 > | NOTE |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | ALL OFF | ALL OFF | | TBD |
| 0 | 0 | 0 | 1 | X+ / X- off<br>Y+ / Y- on | X+ / X- off<br>Y+ / Y- on | 1000 | Y Position |
| 0 | 0 | 1 | 0 | ALL OFF | ALL OFF | 0011 | IN3 |
| 0 | 0 | 1 | 1 | X+ / Y- off<br>Y+ / X- on | X+ / Y- off<br>Y+ / X- on | 1000 | $Z_1$ Position |
| 0 | 1 | 0 | 0 | X+ / Y- off<br>Y+ / X- on | X+ / Y- off<br>Y+ / X- on | 1010 | $Z_2$ Position |
| 0 | 1 | 0 | 1 | X+ / X- on<br>Y+ / Y- off | X+ / X- on<br>Y+ / Y- off | 1001 | X Position |
| 0 | 1 | 1 | 0 | ALL OFF | ALL OFF | 0100 | IN4 |
| 0 | 1 | 1 | 1 | ALL OFF | ALL OFF | | TBD |
| 1 | 0 | 0 | 0 | ALL OFF | ALL OFF | | TBD |
| 1 | 0 | 0 | 1 | X+ / X- off<br>Y+ / Y- on | ALL OFF | 1000 | Y Position |
| 1 | 0 | 1 | 0 | ALL OFF | ALL OFF | 0011 | IN3 |
| 1 | 0 | 1 | 1 | X+ / Y- off<br>Y+ / X- on | ALL OFF | 1000 | $Z_1$ Position |
| 1 | 1 | 0 | 0 | X+ / Y- off<br>Y+ / X- on | ALL OFF | 1010 | $Z_2$ Position |
| 1 | 1 | 0 | 1 | X+ / X- on<br>Y+ / Y- off | ALL OFF | 1001 | X Position |
| 1 | 1 | 1 | 0 | ALL OFF | ALL OFF | 0100 | IN4 |
| 1 | 1 | 1 | 1 | ALL OFF | ALL OFF | | TBD |

Table 52 Relationship between commands and touch panel control signals

## 4.8.1 Register Definitions

| Register Address | Register Function | Acronym |
|---|---|---|
| 0x82050000 | Auxiliary ADC control register 0 | AUXADC_CON0 |
| 0x82050004 | Auxiliary ADC control register 1 | AUXADC_CON1 |
| 0x82050008 | Auxiliary ADC control register 2 | AUXADC_CON2 |
| 0x82050010 | Auxiliary ADC channel 0 data register | AUXADC_DAT0 |
| 0x82050014 | Auxiliary ADC channel 1 data register | AUXADC_DAT1 |
| 0x82050018 | Auxiliary ADC channel 2 data register | AUXADC_DAT2 |
| 0x8205001C | Auxiliary ADC channel 3 data register | AUXADC_DAT3 |

| | | |
|---|---|---|
| 0x82050020 | Auxiliary ADC channel 4 data register | AUXADC_DAT4 |
| 0x82050024 | Auxiliary ADC channel 5 data register | AUXADC_DAT5 |
| 0x82050028 | Auxiliary ADC channel 6 data register | AUXADC_DAT6 |
| 0x8205002C | Auxiliary ADC channel 0 data register for TDMA event 0 | AUXADC_DAT7 |
| 0x82050030 | Touch Screen Debounce Time | AUX_TS_DEBT |
| 0x82050034 | Touch Screen Sample Command | AUX_TS_CMD |
| 0x82050038 | Touch Screen Control | AUX_TS_CON |

**Table 53** Auxadc Registers



## 0x82050000 Auxiliary ADC control register 0 AUXADC_CON0

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | SYN6 | 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 |

**SYN*n*** These 7 bits define whether the corresponding channel is sampled or not in timer-triggered mode. It is associated with timing offset register TDMA_AUXEV1. It supports multiple flags. The flags can be automatically cleared after those channel have been sampled if AUTOCLR1 in the register AUXADC_CON3 is set. To monitor ISENSE and BATSNS, ISENSE_OUT_EN & VBAT_OUT_EN in the register PMIC_CONG (Analog Front End functional spec) must be set to 1 in advanced.

**0** The channel is not selected.

**1** The channel is selected.

## 0x82050004 Auxiliary ADC control register 1 AUXADC_CON1

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | IMM6 | 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 |

**IMM*n*** These 7 bits are set individually to sample the data for the corresponding channel. It supports multiple flags.

**0** The channel is not selected.

**1** The channel is selected.

## 0x82050008 Auxiliary ADC control register 2 AUXADC_CON2

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

| Name | | | | | | | | | | | | | | | SYN7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | | | | | | | R/W |
| Reset | | | | | | | | | | | | | | | 0 |

**SYN7** This bit is used only for channel 0 and is to be associated with timing offset register TDMA_AUXEV0 in the TDMA timer in timer-triggered mode. The flag can be automatically cleared after channel 0 has been sampled if AUTOCLR0 in the register AUXADC_CON3 is set.

**0** The channel is not selected.

**1** The channel is selected.

## 0x8205000C Auxiliary ADC control register 3 AUXADC_CON3

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | AUTOSET | | | | PUWAIT_EN | | AUTOCLR1 | AUTOCLR0 | | | | | | | | STA |
| Type | R/W | | | | R/W | | R/W | R/W | | | | | | | | RO |
| Reset | 0 | | | | 0 | | 0 | 0 | | | | | | | | 0 |

**AUTOSET** This field 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.

**PUWAIT_EN** Thus field enables the power warm-up period to ensure power stability before the SAR process takes place. It is recommended to activate this field.

**0** The mode is not enabled.

**1** The mode is enabled.

**AUTOCLR1** The field defines the auto-clear mode of the module for event 1. In auto-clear mode, each timer-triggered channel gets samples of the specified channels once the SYN*n* bit in the register AUXADC_CON0 has been set. The SYN*n* bits are automatically cleared and the channel is not enabled again by the timer event except when the SYN*n* flags are set again.

**0** The automatic clear mode is not enabled.

**1** The automatic clear mode is enabled.

**AUTOCLR0** The field defines the auto-clear mode of the module for event 0. In auto-clear mode, the timer-triggered channel 0 gets the sample once the SYN7 bit in the register AUXADC_CON2 has been set. The SYN7 bit is automatically cleared and the channel is not enabled again by the timer event 0 except when the SYN7 flag is set again.

**0** The automatic clear mode is not enabled.

**1** The automatic clear mode is enabled.

**STA** The field defines the state of the module.

**0** This module is idle.

1 This module is busy.

## 0x82050010 Auxiliary ADC channel 0 register AUXADC_DAT0

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

The register stores the sampled data for the channel 0. There are 8 registers of the same type for the corresponding channel . The overall register definition is listed in **Table 54**.

| Register Address | Register Function | Acronym |
|---|---|---|
| 0x82050010 | Auxiliary ADC channel 0 data register | AUXADC_DAT0 |
| 0x82050014 | Auxiliary ADC channel 1 data register | AUXADC_DAT1 |
| 0x82050018 | Auxiliary ADC channel 2 data register | AUXADC_DAT2 |
| 0x8205001C | Auxiliary ADC channel 3 data register | AUXADC_DAT3 |
| 0x82050020 | Auxiliary ADC channel 4 data register | AUXADC_DAT4 |
| 0x82050024 | Auxiliary ADC channel 5 data register | AUXADC_DAT5 |
| 0x82050028 | Auxiliary ADC channel 6 data register | AUXADC_DAT6 |
| 0x8205002C | Auxiliary ADC channel 0 data register for TDMA event 0 | AUXADC_DAT7 |

**Table 54** Auxiliary ADC data register list

## 0x82050030 Touch Screen Debounce Time AUX_TS_DEBT

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

**DEBOUNCE TIME** While the analog touch screen irq signal is from high to low level, auxadc will issue an interrupt after the debounce time.

## 0x82050034 Touch Screen Sample Command AUX_TS_CMD

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | ADDRESS | | | MODE | SE/DF | PD | |
| Type | | | | | | | | | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**ADDRESS** Define which x or y or z data will be sampled.

- **001** Y Position
- **011** Z1 Position
- **100** Z2 Position
- **101** X Position
- **Others** Reserved

**MODE** Select the sample resolution

- **0** 10-bit resolution
- **1** 8-bit resolution

**SE/DF** Mode selection

- **0** Differential mode
- **1** Single-end mode

**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

## 0x82050038 Touch Screen Control AUX_TS_CON

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

**SPL** Touch Screen Sample Trigger

- **0** No Action
- **1** While SW writes 1'b1, auxadc will trigger the touch screen process. After the sample process of touch screen finishes, this bit will be disserted.

**ST** Touch Screen Status

- **0** Touch Screen is idle.
- **1** Touch Screen is touched.

## 0x8205003C Touch Screen Sample DATA AUX_TS_DAT0

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

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Type | | | | | | | RO |
| Reset | | | | | | | 0 |

This register stores the touch screen sample data.

# 4.9 I2C / SCCB Controller

## 4.9.1 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.

### 4.9.1.1 Feature Support

I2C compliant master mode operation

Adjustable clock speed for LS/FS mode operation.

7bit/10 bit addressing support.

High Speed mode support.

Slave Clock Extension support.

START/STOP/REPEATED START condition

Manual/DMA Transfer Mode

Multi write per transfer (up to 8 data bytes for non dma mode and 255 data bytes for dma mode)

Multi read per transfer (up to 8 data bytes for non dma mode and 255 data bytes for dma mode)

Multi transfer per transaction (up to 256 write transfers or 256 read transfers with dma mode)

DMA mode with Fifo Flow Control and bus signal holding

Combined format transfer with length change capability.

Active drive / wired-and I/O configuration

### 4.9.1.2 Manual/DMA Transfer Mode

The controller offers 2 types of transfer mode, Manual and DMA.

When 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.

When DMA mode is enabled, the data to and from the FIFO is controlled via DMA transfer and can therefore support up to 255 bytes of consecutive read or write, with the data read from or write to another memory space. When DMA mode is enabled, flow control mechanism is also implemented to hold the bus clk when FIFO underflow or overflow condition is encountered.

### 4.9.1.3 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 = the number of bytes within the transfer.**

**transaction = this is the top unit. Everything combined equals 1 transaction.**

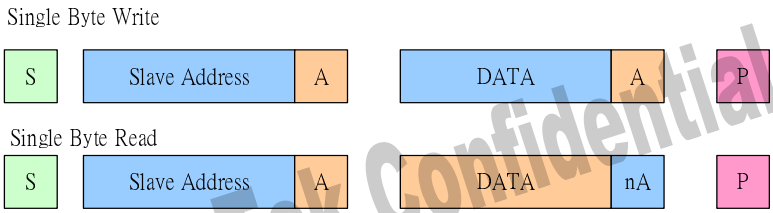
**Transaction length = the number of transfers to be conducted.**

**)**

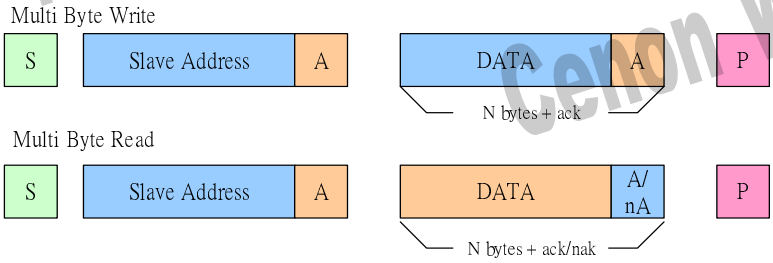
Master to slave dir

Slave to master dir

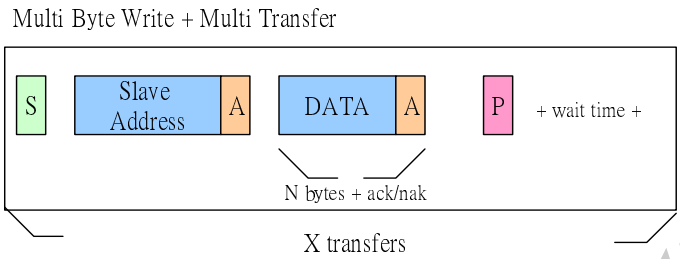
**Single Byte Access**

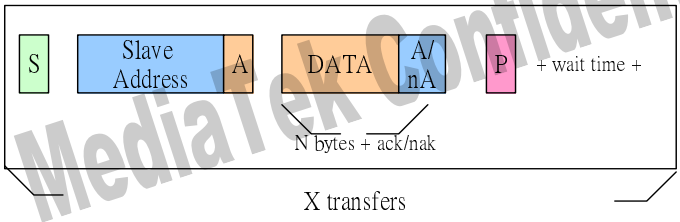



**Multi Byte Access**

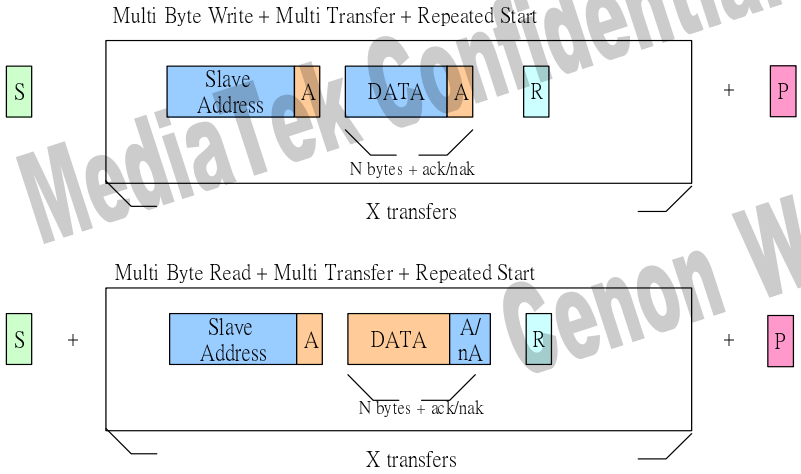



**Multi Byte Transfer + Multi Transfer (same direction)**



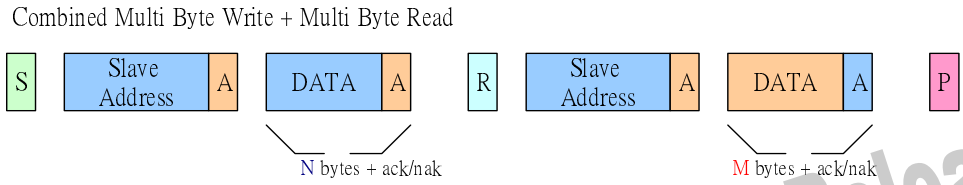



**Multi Byte Transfer + Multi Transfer w RS (same direction)**



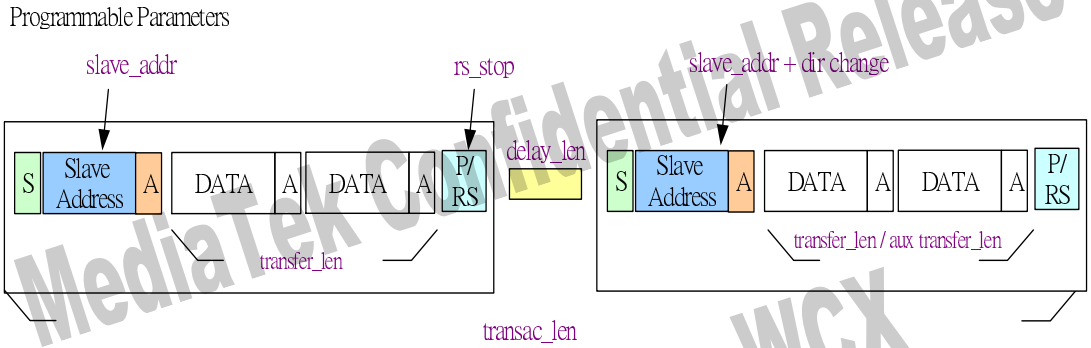

**Combined Write/Read with Repeated Start (direction change)**

***(Note: Only supports Write and then Read sequence. Read and then Write is not supported)***

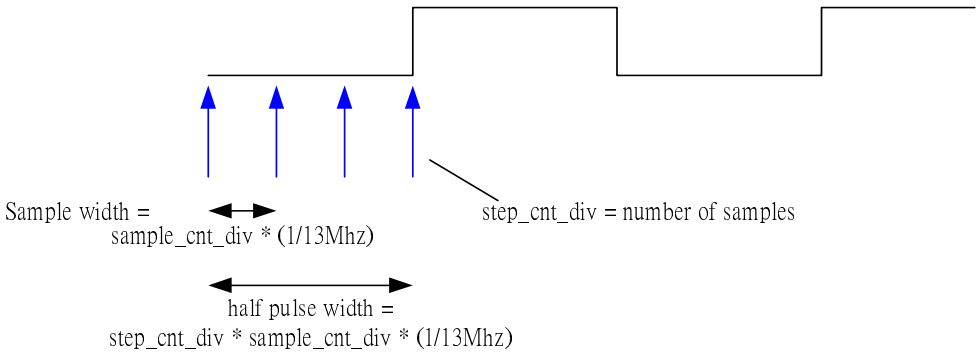



## 4.9.2 Programming Examples

**Common Transfer Programmable Parameters**




**Output Waveform Timing Programmable Parameters**




## 4.9.3 Register Definitions

### I2CREG+0000h Data Port Register DATA_PORT

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

**DATA_PORT[7:0]** This is the 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 the fifo.

(DEBUG ONLY) If the fifo_apb_debug bit is set, then the FIFO can be read and write by the APB

### I2CREG+0004h Slave Address Register SLAVE_ADDR

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

**SLAVE_ADDR [7:0]** This 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. 1 = master read, 0 = master write.

### I2CREG+0008h Interrupt Mask Register INTR_MASK

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | DEBUG | HS_NACKERR | ACKERR | TRANSAC_COMP |
| Type | | | | | | | | | | | | | R.W | R/W | R/W | R/W |
| Reset | | | | | | | | | | | | | 1 | 1 | 1 | 1 |

This register provides masks for the corresponding interrupt sources as indicated in intr_stat register.

1 = allow interrupt

0 = disable interrupt

Note: while disabled, the corresponding interrupt will not be asserted, however the intr_stat will still be updated with the status. Ie. mask does not affect intr_stat register values.

### I2CREG+000Ch Interrupt Status Register INTR_STAT

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

| Name | | | | | | | | | | | | | HS_NACKERR | ACKERR | TRANSAC_COMP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | | | | | W1C | W1C | W1C |
| Reset | | | | | | | | | | | | | 0 | 0 | 0 |

When an interrupt is issued by 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 write 1 cleared.

**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.

**ACKERR** This status is asserted if ACK error detection is enabled. If enabled, ackerr will cause transaction to end and stop will be issued.

**TRANSAC_COMP** This status is asserted when a transaction has completed successfully.

## I2CREG+0010h Control Register CONTROL

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | TRANSFER_LEN_CHANGE | ACKERR_DET_EN | DIR_CHANGE | CLK_EXT_EN | DMA_EN | RS_STOP | |
| Type | | | | | | | | | | R/W | R/W | RW | RW | RW | RW | R/W |
| Reset | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**TRANSFER_LEN_CHANGE** This options specifies whether or not to change the transfer length after the fist transfer completes. If enabled, the transfers after the first transfer will use the transfer_len_aux parameter.

**ACKERR_DET_EN** This option 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 asserts ackerr interrupt. Mcu shall handle this case appropriately and then resets the fifo address before reissuing transaction again. If this option is disabled, the controller will ignore slave ack error and keep on scheduled transaction.

0 disable
1 enable

**DIR_CHANGE** This option is used for 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

**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, master controller will enter a high wait state until the slave releases the SCL line.

**DMA_EN** By default, this is disabled, and fifo data shall be manually prepared by mcu. This default setting should be used for transfer sizes of less than 8 data bytes and no multiple transfer is configured. When enabled, dma requests are turned on, and the fifo data should be prepared in memory.

**RS_STOP** In LS/FS mode, this bit affects multi-transfer transaction only. It controls whether or not 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

## I2CREG+0014h Transfer Length Register (Number of Bytesper Transfer) TRANSFER_LEN

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

| Name | | | | TRANSFER_LEN_AUX | | | | | | | TRANSFER_LEN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | R/W | | | | | | | R/W |
| Reset | | | | 'h1 | | | | | | | 'h1 |

**TRANSFER_LEN_AUX[4:0]** This field is valid only when dir_change is set to 1. This indicates the number of DATA BYTES to be transferred in 1 transfer unit (excluding slave address byte) for the transfers following the direction change. I.e., if dir_change =1, then the first write transfer length depends on transfer_len, while the second read transfer length depend on transfer_len_aux. Dir change is always after the first transfer.

(NOTE) The value must be set greater than 1, otherwise no transfer will take place.

**TRANSFER_LEN[7:0]** This indicates the number of DATA BYTES to be transferred in 1 transfer unit (excluding slave address byte)

(NOTE) The value must be set greater than 1, otherwise no transfer will take place.

## I2CREG+0018h Transaction Length Register (Number of Transfers per Transaction) TRANSAC_LEN

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | TRANSAC_LEN | | | | | | | |
| Type | | | | | | | | | | | | | | | | R/W |
| Reset | | | | | | | | | | | | | | | | 'h1 |

**TRANSAC_LEN[7:0]** This indicates the number of TRANSFERS to be transferred in 1 transaction

(NOTE) The value must be set greater than 1, otherwise no transfer will take place.

## I2CREG+001Ch Inter Delay Length Register DELAY_LEN

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | DELAY_LEN | | | | | | | |
| Type | | | | | | | | | R/W | | | | | | | |
| Reset | | | | | | | | | 'h2 | | | | | | | |

**DELAY_LEN[3:0]** This sets the wait delay between consecutive transfers when RS_STOP bit is set to 0. (the unit is same as the half pulse width)

## I2CREG+0020h Timing Control Register TIMING

| 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 | R/W | R/W | | | | R/W | | | | | R/W | | | | | |
| Reset | 'h0 | 'h1 | | | | 'h3 | | | | | 'h3 | | | | | |

LS/FS only. This register is used to control the output waveform timing. Each half pulse width (ie. each high or low pulse) is equal to = step_cnt_div * (sample_cnt_div * 1/13Mhz)

**SAMPLE_CNT_DIV[2:0]** Used for LS/FS only. This adjusts the width of each sample. (sample width = sample_cnt_div * 1/13Mhz)

**STEP_CNT_DIV[5:0]** This specifies the number of samples per half pulse width (ie. each high or low pulse)

**DATA_READ_ADJ** When set to 1, data latch in sampling time during master reads are adjusted according to DATA_READ_TIME value. Otherwise, by default, data is latched in at half of the high pulse width point. This value must be set to less or equal to half the high pulse width.

**DATA_READ_TIME[2:0]** This value is valid only when DATA_READ_ADJ is set to 1. This can be used to adjust so that data is latched in at earlier sampling points (assuming data is settled by then)

## I2CREG+0024h Start Register START

| 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** This register starts the transaction on the bus. It is auto deasserted at the end of the transaction.

## I2CREG+0030h Fifo Status Register FIFO_STAT

| 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 | | | | RO | | | | RO | | | | | | RO | RO |
| Reset | 0 | | | | 0 | | | | 0 | | | | 0 | 0 | 0 | 0 |

**RD_ADDR[3:0]** The current rd address pointer. (only bit [2:0] has physical meaning)

**WR_ADDR[3:0]** The current wr address pointer. (only bit [2:0] has physical meaning)

**FIFO_OFFSET[3:0]** wr_addr[3:0] – rd_addr[3:0]

**WR_FULL** This indicates that the fifo is full.

**RD_EMPTY** This indicates that the fifo is empty.

## I2CREG+0034h Fifo Thresh Register FIFO_THRESH

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | TX_TRIG_THRESH | | | | | | | | RX_TRIG_THRESH | | |
| Type | | | | | | RW | | | | | | | | R/W | | |
| Reset | | | | | | 'h7 | | | | | | | | 'h0 | | |

**DEBUG ONLY.** By default, these values do not need to be adjusted. Note! for RX, no timeout mechanism is implemented. Therefore, RX_trig_thresh must be left at 0, or there would be data left in the fifo that is not fetched by DMA controller.

**TX_TRIG_THRESH[2:0]** When tx fifo level is below this value, tx dma request is asserted.

**RX_TRIG_THRESH[2:0]** When rx fifo level is above this value, rx dma request is asserted.

## I2CREG+0038h Fifo Address Clear Register FIFO_ADDR_CLR

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | | | FIFO_ADDR_CR |
| Type | | | | | | | | | | | | | | | | WO |
| Reset | | | | | | | | | | | | | | | | 0 |

**FIFO_ADDR_CLR** When written with a 1'b1, a 1 pulse fifo_addr_clr is generated to clear the fifo address to back to 0.

## I2CREG+0040h IO Config Register IO_CONFIG

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | IDLE_OE_EN | IO SYNC EN | SDA_IO_CONFIG | SCL_IO_CONFIG |
| Type | | | | | | | | | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | | | | | | | | | 0 | 0 | 0 | 0 |

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.

**IDLE_OE_EN**
0 don't drive bus in idle state
1 drive bus in idle state

**IO_SYNC_EN** DEBUG ONLY: When set to 1, scl and sda inputs will be first dual synced by bclk_ck. This should not be needed. Only reserved for debugging.

**SDA_IO_CONFIG**
0 normal tristate io mode
1 open-drain mode

**SCL_IO_CONFIG**
0 normal tristate io mode
1 open-drain mode

## I2CREG+0044h RESERVED DEBUG Register DEBUG

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

NOTE: This register is for DEBUG ONLY. The bits are R/W, do not change the values from the default value.

## I2CREG+0048h High Speed Mode Register HS

| 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 | | R/W | | | | R/W | | | | R/W | | | | | R/W | R/W |
| Reset | | 0 | | | | 1 | | | | 0 | | | | | 1 | 0 |

This register contains options for supporting high speed operation features

Each HS half pulse width (ie. each high or low pulse) is equal to = step_cnt_div * (sample_cnt_div * 1/13Mhz)

**HS_SAMPLE_CNT_DIV[2:0]** When high speed mode is entered after the master code transfer has been completed, the sample width becomes dependent on this parameter.

**HS_STEP_CNT_DIV[2:0]** When high speed mode is entered after the master code transfer has been completed, the number of samples per half pulse width becomes dependent on this value.

**MASTER_CODE[2:0]** This is the 3 bit programmable value for the master code to be transmitted.

**HS_NACKERR_DET_EN** This enables NACKERR detection during the master code transmission. When enabled, if NACK is not received after master code has been transmitted, the transaction will terminated with a STOP condition.

**HS_EN** This enables the high speed transaction. (note: rs_stop must be set to 1 as well)

## I2CREG+0050h Soft Reset Register SOFTRESET

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | | | SOFT_RESET |
| Type | | | | | | | | | | | | | | | | WO |
| Reset | | | | | | | | | | | | | | | | 0 |

**SOFT_RESET** When written with a 1'b1, a 1 pulse soft reset is used as synchronous reset to reset the I2C internal hardware circuits.

## I2CREG+0064h Debug Status Register DEBUGSTAT

| 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 | 1 | 0 | 0 | | | |

**BUS_BUSY** DEBUG ONLY: valid when bus_detect_en is 1. bus_busy = 1 indicates a start transaction has been detected and no stop condition has been detected yet.

**MASTER_WRITE** DEBUG ONLY: 1 = current transfer is in the master write dir

**MASTER_READ** DEBUG ONLY: 1 = current transfer is in the master read dir

**MASTER_STATE[3:0]** DEBUG ONLY: reads back the current master_state.

0: idle state;

1: i2c master is preparing sending 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 transmitting 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 transmitting ack bit, SCL=0, SDA=ack (ack bit can be changed when SCL=0);

6: i2c master/slave is transmitting ack bit, SCL=1, SDA=0 (ack bit is stable when SCL=1);

7: i2c master is preparing sending out stop bit or repeated-start bit, SCL=0, SDA=0/1 (0: means stop bit; 1: means repeated-start bit);

8: i2c master is sending out stop bit or repeated-start bit, SCL=1, SDA=1/0 (1: means stop bit; 0: means repeated-start bit);

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 writing data into fifo; For reading transaction, it

means fifo is full and i2c master is waiting for dma controller reading data from fifo, SCL=0, SDA=don't care;

12: i2c master is preparing sending 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 transmitting nack bit, SCL=0, SDA=nack bit (nack bit can be changed when SCL=0); This state is used only in high-speed transaction;

15: i2c master/slave is transmitting nack bit, SCL=1, SDA=1; This state is used only in high-speed transaction;

## I2CREG+0068h Debug Control Register DEBUGCTRL

| 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 | R/W |
| Reset | | | | | | | | | | | | | | | 0 | 0 |

**APB_DEBUG_RD** This bit is 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.

**FIFO_APB_DEBUG** This is used for trace32 debug purposes. When using trace32, and the memory map is shown, turning this bit on will block the normal apb read access. Apb read access to the fifo is then enabled by writing to apb_debug_rd.

0 disable
1 enable

# 4.10 USB Device Controller

## 4.10.1 General Description

This chip provides a USB function interface that is in compliance with Universal Serial Bus Specification Rev 1.1. The USB device controller supports only full-speed (12Mbps) operation. The Bluetooth chip can make use of this widely available USB interfaces to transmit/receive data with USB hosts, typically PC/laptop.

There provides 6 endpoints in the USB device controller besides the mandatory control endpoint, where among them, 4 endpoints are for IN transactions and 2 endpoints are for OUT transactions. Word, half-word, and byte access are all allowed for loading and unloading the FIFO. The controller features 4 DMA channels to accelerate the data transfer for ACL and SCO data streams. The features of the endpoints are as follows:

1. Endpoint 0: The control endpoint feature 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. DMA transfer is not supported.
2. 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.
3. 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.
4. IN endpoint 3: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.
5. IN endpoint 4: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.
6. OUT endpoint 1: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Channel 5 Read Transfer is supported.
7. OUT endpoint 2: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Channel 7 Read Transfer is supported.

For each endpoint except 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 the control endpoint can be configured to be a bulk, interrupt or isochronous endpoints. Composite device is also supported. The IN endpoint 1 and the OUT endpoint 1 shares the same endpoint number but they can be use separately. So is the situation as the endpoint 2

The USB device uses cable-powered feature for the transceiver but only drains little current. An internal pull-up resistor has been integrated across Vbus and D+ signal. The switch on/off of the pull-up resistor can 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.

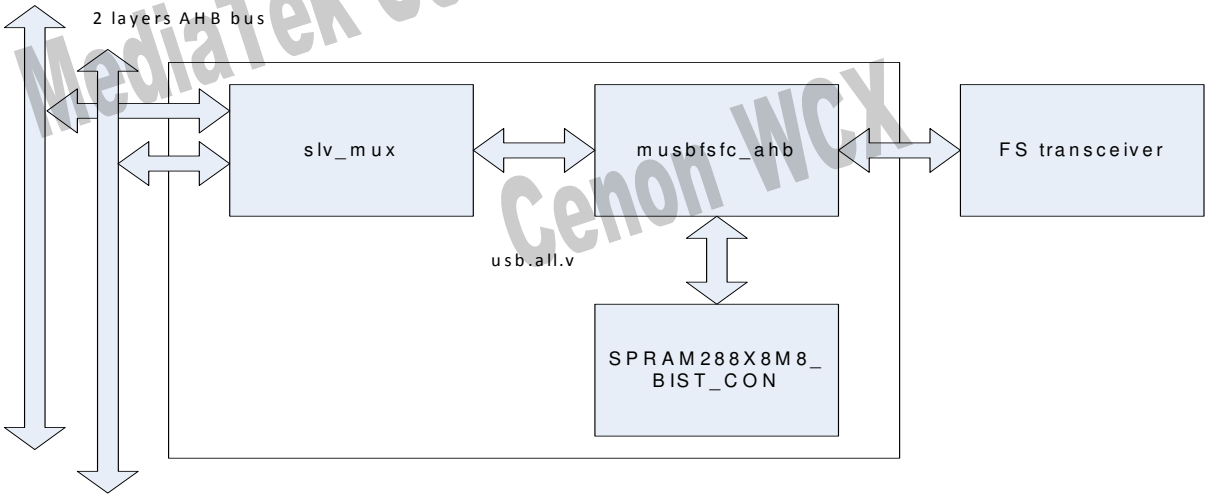



Figure1. USB11 controller system diagram 1

## 4.10.2 Register Definitions

### USB+0000h USB function address register USB_FADDR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | UPD | FADDR | | | | | | |
| Type | RO | R/W | | | | | | |
| Reset | 0 | 0 | | | | | | |

This is an 8-bit register that should be written with the function's 7-bit address (received through a SET_ADDRESS description). It is then used for decoding the function address in subsequent token packets.

**UPD** Set when FADDR is written. It's cleared when the new address takes effect (at the end of the current transfer).
**FADDR** The function address of the device.

## USB+0001h USB power control register USB_POWER

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | ISO_UP | | | SWRSTENAB | RESET | RESUME | SUSPMODE | SUSPENAB |
| Type | R/W | | | R/W | RO | R/W | RO | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**ISO_UP** When set by the MCU, the core will wait for an SOF token from the time INPKTRDY is set before sending the packet.

**SWRSTENAB** Set by the MCU to enable the mode in which the device can only be reset by the software after detecting reset signals on the bus. In case the software is delayed by other high-priority process and can't make it to read the command from the buffer before the hardware reset the device after detecting the reset signal on the bus, the command will be lost. That's why the software-reset mode is effective. When the flag is enabled, the hardware state machine can't reset by itself, but rather can be reset by the software. In that sense, the software and the hardware can keep synchronous on detecting the reset signal.

**RESET** The read-only bit is set when **Reset** signaling is present on the bus.

**RESUME** Set by the MCU to generate **Resume** signaling when the function is in suspend mode. The MCU should clear this bit after 10 ms (a maximum of 15 ms) to end Resume signaling.

**SUSPMODE** Set by the USB core when **Suspend** mode is entered. It is cleared when sets the Resume bit of this register.

**SUSPENAB** Set by the MCU to enable device into **Suspend** mode when Suspend signaling is received on the bus.

## USB+0002h USB IN endpoints interrupt register USB_INTRIN

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | EP4 | EP3 | EP2 | EP1 | EP0 |
| Type | | | | RC | RC | RC | RC | RC |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

This is a read-only register that indicates which of the interrupts for IN endpoints 0 to 4 are currently active. It must set be setting after USB_INTRINE has been set .All active interrupts will be cleared when this register is read.

**EP4** IN endpoint #4 interrupt.

**EP3** IN endpoint #3 interrupt.

**EP2** IN endpoint #2 interrupt.

**EP1** IN endpoint #1 interrupt.

**EP0** IN endpoint #0 interrupt.

## USB+0004h USB OUT endpoints interrupt register USB_INTROUT

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | | EP2 | EP1 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Type | | | | | | RC | RC | |
| Reset | | | | | | 0 | 0 | |

This is a read-only register that indicates which of the interrupts for OUT endpoints 1, and 2 are currently active. It must set be setting after USB_INTROUTE has been set. All active interrupts will be cleared when this register is read.

**EP2** OUT endpoint #2 interrupt.

**EP1** OUT endpoint #1 interrupt.

## USB+0006h USB general interrupt register USB_INTRUSB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | SOF | RESET | RESUME | SUSP |
| Type | | | | | RC | RC | RC | RC |
| Reset | | | | | 0 | 0 | 0 | 0 |

This is a read-only register that indicates which USB interrupts are currently active. It must set be setting after USB_INTRUSBE has been set. All active interrupts will be cleared when this register is read.

**SOF** Set at the start of each frame.

**RESET** Set when **Reset** signaling is detected on the bus.

**RESUME** Set when Resume signaling is detected on the bus while the USB core is in suspend mode.

**SUSP** Set when Suspend signaling is detected on the bus.

## USB+0007h USB IN endpoints interrupt enable register USB_INTRINE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | EP4 | EP3 | EP2 | EP1 | EP0 |
| Type | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 1 | 1 | 1 | 1 | 1 |

This register provides interrupt enable bits for the interrupts in USB_INTRIN. On reset, the bits corresponding to endpoint 0 and all IN endpoints are set to 1.

**EP4** IN endpoint 4 interrupt enable.

**EP3** IN endpoint 3 interrupt enable.

**EP2** IN endpoint 2 interrupt enable.

**EP1** IN endpoint 1 interrupt enable.

**EP0** IN endpoint 0 interrupt enable.

## USB+0009h USB OUT endpoints interrupt enable register USB_INTROUTE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | | EP2 | EP1 | |
| Type | | | | | | R/W | R/W | |

| Reset | | | | | | 1 | 1 | |
|---|---|---|---|---|---|---|---|---|

This register provides interrupt enable bits for the interrupts in USB_INTROUT. On reset, the bits corresponding to all OUT endpoints are set to 1.

**EP2** OUT endpoint 2 interrupt enable.
**EP1** OUT endpoint 1 interrupt enable.

## USB+000Bh USB general interrupt enable register — USB_INTRUSBE

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | SOF | RESET | RESUME | SUSP |
| Type | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 1 | 1 | 0 |

This register provides interrupt enable bits for each of the interrupts for USB_INTRUSB.

**SOF** SOF interrupt enable
**RESET** Reset interrupt enable
**RESUME** Resume interrupt enable
**SUSP** Suspend interrupt enable

## USB+000Ch USB frame count #1 register — USB_FRAME1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | NUML | | | | | | | |
| Type | RO | | | | | | | |
| Reset | 0 | | | | | | | |

The register holds the lower 8 bits of the last received frame number.

**NUML** The lower 8 bits of the frame number.

## USB+000Dh USB frame count #2 register — USB_FRAME2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | | NUMH | | |
| Type | | | | | | RO | | |
| Reset | | | | | | 0 | | |

The register holds the upper 3 bits of the last received frame number.

**NUMH** The upper 3 bits of the frame number.

## USB+000Eh USB endpoint register index USB_INDEX

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | INDEX | | | |
| Type | | | | | R/W | | | |
| Reset | | | | | 0 | | | |

The register determines which endpoint control/status registers are to be accessed at addresses USB+10h to USB+17h. Each IN endpoint and each OUT endpoint have their 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 any one time. Before accessing an endpoint's control/status registers, the endpoint number should be written to the USB_INDEX register to ensure that the correct control/status registers appear in the memory map.

**INDEX** The index of the endpoint.

## USB+000Fh USB reset control USB_RSTCTRL

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SWRST | | | | RSTCNTR | | | |
| Type | R/W | | | | R/W | | | |
| Reset | 0 | | | | 0 | | | |

The register is used to control the reset process when the device detects the reset command issued from the host.

**SWRST** If the flag SWRSTENAB in the register USB_POWER is set to be 1, the software enable mode is enabled, and the device can be reset by writing this flag to be 1.

**RSTCNTR** The field signifies the duration for the reset operation to take place after detecting reset signal on the bus. It's only enabled when software reset is not enabled. If the value is equal to zero, the duration is 2.5us. Otherwise, the duration is equal to this value multiplied by 341 and then added by 2.5 in unit of us. The range consequently starts from 2.5us to 5122.5 us.

## USB+0011h USB control/status register for endpoint 0 USB_EP0_CSR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | SSETUPEND | SOUTPKTRDY | SENDSTALL | SETUPEND | DATAEND | SENTSTALL | INPKTRDY | OUTPKTRDY |
| Type | R/WS | R/WS | R/WS | RO | R/WS | R/WC | R/WS | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The register is used for all control/status of endpoint 0. The register is active when USB_INDEX register is set to 0.

**SSETUPEND** The MCU writes a 1 to this bit to clear the SETUPEND bit. It's cleared automatically. Only active when a transaction has been started.

**SOUTPKTRDY** The MCU writes a 1 to this bit to clear the OUTPKTRDY bit. It's cleared automatically. Only active when an OUT transaction has been started.

**SENDSTALL** The MCU writes a 1 to this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically.

**SETUPEND** This bit will be set when a control transaction ends before the DATAEND bit has been set. An interrupt will be generated and FIFO flushed at this time. The bit is cleared by the MCU writing a 1 to the SSETUPEND bit.

**DATAEND** The MCU sets 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's cleared automatically

**SENTSTALL** This bit is set when a STALL handshake is transmitted. The MCU should clear this bit by writing a 0.

**INPKTRDY** The MCU sets this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet has been transmitted. An interrupt is generated when this bit is set.

**OUTPKTRDY** This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The MCU clears this bit by setting the SOUTPKTRDY bit.

## USB+0016h USB byte count register — USB_EP0_COUNT

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | COUNT | | | | | | |
| Type | | RO | | | | | | |
| Reset | | 0 | | | | | | |

The register indicates the number of received data bytes in the endpoint 0. The value returned is valid while OUTPKTRDY bit of USB_EP0_CSR register is set. The register is active when USB_INDEX register is set to 0.

**COUNT** The number of received data bytes in the endpoint 0.

## USB+0010h USB maximum packet size register for IN endpoint 1~4 — USB_EP_INMAXP

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MAXP | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

The register holds the maximum packet size for transactions through the currently selected IN endpoint – in units of one byte. In setting the value, the programmer should note the constraints placed by the USB Specification on packet size for bulk interrupt, and isochronous transactions in full-speed operations. There is an INMAXP register for each IN endpoint except endpoint 0. The registers are active when 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 could cause unexpected results. If a value greater than the configured IN FIFO size for the endpoint is written to the register, the value will be automatically changed to the IN FIFO size. If the value written to the register is less 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 packets have been sent from the endpoint, the endpoint IN FIFO should be completely flushed after writing the new value to the register.

**MAXP** The maximum packet size in units of one byte.

## USB+0011h USB control/status register #1 for IN endpoint 1~4 USB_EP_INCSR1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | ABORTPKT_EN | CLRDATATOG | SENTSTALL | SENDSTALL | FLUSHFIFO | UNDERRUN | FIFONOTEMPTY | INPKTRDY |
| Type | R/W | WO | R/WC | R/W | WO | R/WC | RO | R/WS |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The register provides control and status bits for IN transactions through the currently selected endpoint. There is an INCSR1 register for each IN endpoint except endpoint 0. The registers are active when USB_INDEX register is set to 1, 2, and 3, respectively.

**ABORTPKT_EN** When MCU write ABORTPKT_EN as 1, FLUSHFIFO switch to abort 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 aborting packet, EP will issue an interrupt. Programmer should wait for this interrupt to make sure the packet is aborted.

**CLRDATATOG** The MCU writes a 1 to this bit to reset the endpoint IN data toggle to 0.

**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 a 0 to this bit.

**SENDSTALL** The MCU writes a 1 to this bit to issue a STALL handshake to an IN token. The MCU clears this bit to terminate the stall condition.

**FLUSHFIFO** The MCU writes a 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 to be corrupted.
If ABORTPKT_EN is enabled and this bit is set , the function of this bit becomes ABORTPKT to abort the next packet to be transmitted from the endpoint IN FIFO and doesn't need to set INPKTRDY. The same with FLUSHFIFO function, this bit is only active when endpoint is idle.

**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.

**FIFONOTEMPTY** This bit is set when there is at least 1 packet in the IN FIFO.

**INPKTRDY** The MCU sets this bit after loading a data packet into the FIFO. Only active when an IN transaction has been started. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

## USB+0012h USB control/status register #2 for IN endpoint 1~4 USB_EP_INCSR2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | AUTOSET | ISO | MODE | DMAENAB | RFCDATATOG | | | |
| Type | R/W | R/W | R/W | R/W | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | | | |

The register provides further control bits for IN transactions through the currently selected endpoint. There is an INCSR2 register for each IN endpoint except endpoint 0. The registers are active when USB_INDEX register is set to 1, 2, and 3, respectively.

**AUTOSET** If the MCU sets the bit, INPKTRDY will be automatically set when data of the maximum packet size (value in INMAXP) is loaded into the IN FIFO. If a packet of less than the maximum packet size is loaded, then INPKTRDY will have to be set manually. When 2 packets are in the IN FIFO then INPKTRDY will also be automatically set when the first packet has been sent, if the second packet is the maximum packet size.

**ISO** The MCU sets this bit to enable the IN endpoint for isochronous transfer, and clears it to enable the IN endpoint for bulk/interrupt transfers.

**MODE** The MCU sets this bit to enable the endpoint direction as IN, and clears it to enable the endpoint direction as OUT. It's valid only where the same endpoint FIFO is used for both IN and OUT transaction.

**DMAENAB** The MCU sets this bit to enable the DMA request for the IN endpoint.

**FRCDATATOG** The MCU sets this bit to force the endpoint's IN data toggle to switch after each data packet is sent regardless of whether an ACK was received. This can be used by interrupt IN endpoints which are used to communicate rate feedback for isochronous endpoints.

## USB+0013h USB maximum packet size register for OUT endpoint 1~2 USB_EP_OUTMAXP

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | MAXP | | | | | | | |
| Type | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

This register holds the maximum packet size for transactions through the currently selected OUT endpoint – in units of one byte. In setting 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 endpoint 0. The registers are active when 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 could 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 greater 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 less 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.

**MAXP** The maximum packet size in units of one byte.

## USB+0014h USB control/status register #1 for OUT endpoint 1~2 USB_EP_OUTCSR1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | CLRDATATOG | SENTSTALL | SENDSTALL | FLUSHFIFO | DATAERROR | OVERRUN | FIFOFULL | OUTPKTRDY |
| Type | WO | R/WC | R/W | WO | RO | R/WC | RO | R/WC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The register provides control status bits for OUT transactions through the currently selected endpoint. The registers are active when USB_INDEX register is set to 1 and 2 respectively.

**CLRDATATOG** The MCU writes a 1 to this bit to reset the endpoint data toggle to 0.

**SENTSTALL** The bit is set when a STALL handshake is transmitted. The MCU should clear this bit by writing a 0.

**SENDSTALL** The MCU writes a 1 to this bit to issue a STALL handshake. The MCU clears this bit to terminate the stall condition. This bit has no effect if the OUT endpoint is in isochronous mode.

**FLUSHFIFO** The MCU writes a 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 to be corrupted.

**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 isochronous mode.

**OVERRUN** The bit is set if an OUT packet cannot be loaded into the OUT FIFO. The MCU should clear the bit by writing a zero. This bit is only valid in isochronous mode.

**FIFOFULL** This bit is set when no more packets can be loaded into the OUT FIFO.

**OUTPKTRDY** The bit is set when a data packet has been received. The MCU should clear (write a 0 to) the bit when the packet has been unloaded from the OUT FIFO. An interrupt is generated when the bit is set. When receiving null packet , OUTPKTRDY has been set after USB_INTROUT[1] is high.

## USB+0015h USB control/status register #2 for OUT endpoint 1~2 USB_EP_OUTCSR2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | AUTOCLEAR | ISO | DMAENAB | DMAMODE | | | | |
| Type | R/W | R/W | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

The register provides further control bits for OUT transactions through the currently selected endpoint. The registers are active when USB_INDEX register is set to 1 and 2 respectively.

**AUTOCLEAR** If the MCU sets this bit then the OUTPKTRDY bit will be automatically cleared when a packet of OUTMAXP bytes has been unloaded from the OUT FIFO. When packets of less then the maximum packet size are unloaded, OUTPKTRDY will have to be cleared manually.

**ISO** The MCU sets this bit to enable the OUT endpoint for isochronous transfers, and clears it to enable the OUT endpoint for bulk/interrupt transfers.

**DMAENAB** The MCU sets this bit to enable the DMA request for the OUT endpoint.

**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); and 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 the bit to select DMA mode 1 and clears this bit to select DMA mode 0.

## USB+0016h USB OUT endpoint byte counter register LSB part for endpoint 1~2 USB_EP_COUNT1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | NUML | | | | | | | |
| Type | RO | | | | | | | |
| Reset | 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 the register USB_OUTCSR1 is set. The registers are active when USB_INDEX register is set to 1 and 2 respectively.

**NUML** The lower 8 bits of the number of received data bytes for the OUT endpoint.

## USB+0017h USB OUT endpoint byte counter register MSB part for endpoint 1~2 USB_EP_COUNT2

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | | | | | | NUMH | | |
| Type | | | | | | RO | | |
| Reset | | | | | | 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 the register USB_EP_OUTCSR1 is set. The registers are active when USB_INDEX register is set to 1 and 2 respectively.

**NUMH** The upper 8 bits of the number of received data bytes for the OUT endpoint.

## USB+0020h USB endpoint 0 FIFO access register USB_EP0_FIFO

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DB3 | | | | | | | | DB2 | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | DB1 | | | | | | | | DB0 | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |

The register provides MCU access to the FIFO for the endpoint 0. Writing to this register loads data into the FIFO for the endpoint 0. Reading from this register unloads data from the FIFO for the endpoint 0.

The register provides word, half-word, and byte mode access. If word or half-word accesses are performed, the less significant byte corresponds to the prior byte to load in or unload from the FIFO.

**DB0** The first byte to be loaded into or unloaded from the FIFO.
**DB1** The second byte to be loaded into or unloaded from the FIFO.

**DB2** The third byte to be loaded into or unloaded from the FIFO.
**DB3** The forth byte to be loaded into or unloaded from the FIFO.

## USB+0024h USB endpoint 1 FIFO access register USB_EP1_FIFO

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DB3 | | | | | | | | DB2 | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | DB1 | | | | | | | | DB0 | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |

The register provides MCU access to the IN FIFO and the OUT FIFO for the endpoint 1. Writing to the register loads data into the IN FIFO for the endpoint 1. Reading from the register unloads data from the OUT FIFO for the endpoint 1.

The register provides word, half-word, and byte mode access. If word or half-word accesses are performed, the less significant byte corresponds to the prior byte to load in the IN FIFO or unload from the OUT FIFO.

**DB0** The first byte to be loaded in the IN FIFO or unloaded from the OUT FIFO.
**DB1** The second byte to be loaded in the IN FIFO or unloaded from the OUT FIFO.
**DB2** The third byte to be loaded in the IN FIFO or unloaded from the OUT FIFO.
**DB3** The forth byte to be loaded in the IN FIFO or unloaded from the OUT FIFO.

## USB+0028h USB endpoint 2 FIFO access register USB_EP2_FIFO

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DB3 | | | | | | | | DB2 | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | DB1 | | | | | | | | DB0 | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |

The register provides MCU access to the IN FIFO and the OUT FIFO for the endpoint 2. Writing to the register loads data into the IN FIFO for the endpoint 2. Reading from the register unloads data from the OUT FIFO for the endpoint 2.

The register provides word, half-word, and byte mode access. If word or half-word accesses are performed, the less significant byte corresponds to the prior byte to load in the IN FIFO or unload from the OUT FIFO.

**DB0** The first byte to be loaded into the IN FIFO or unloaded from the OUT FIFO.
**DB1** The second byte to be loaded into the IN FIFO or unloaded from the OUT FIFO.
**DB2** The third byte to be loaded into the IN FIFO or unloaded from the OUT FIFO.
**DB3** The forth byte to be loaded into the IN FIFO or unloaded from the OUT FIFO.

## USB+002Ch USB endpoint 3 FIFO access register USB_EP3_FIFO

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

| Name | DB3 | | | | | | | | DB2 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | R/W | | | | | | | | R/W | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | DB1 | | | | | | | | DB0 | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |

The register provides MCU access to the IN FIFO and the OUT FIFO for the endpoint 3. Writing to the register loads data into the IN FIFO for the endpoint 3. Reading from the register unloads data from the OUT FIFO for the endpoint 3.

The register provides word, half-word, and byte mode access. If word or half-word accesses are performed, the less significant byte corresponds to the prior byte to load in the IN FIFO or unload from the OUT FIFO.

**DB0** The first byte to be loaded into the IN FIFO or unloaded from the OUT FIFO.

**DB1** The second byte to be loaded into the IN FIFO or unloaded from the OUT FIFO.

**DB2** The third byte to be loaded into the IN FIFO or unload from the OUT FIFO.

**DB3** The forth byte to be loaded into the IN FIFO or unload from the OUT FIFO.

### USB+0240h USB PHY control USB_CON

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | NULL PKT_ FIX | | | | DMPU LLUP | DPPU LLUP |
| Type | | | | | | | | | | | R/W | | | | R/W | R/W |
| Reset | | | | | | | | | | | 1 | 0 | 0 | 0 | 0 | 0 |

**DPPULLUP** Pull-up enable pin. Enable the pull up 1.5KOhm pull up on D+ pin as a full speed device by setting it to high

**DMPULLUP** Pull-up enable pin. Enable the pull up 1.5KOhm pull up on D- pin as a full speed device by setting it to high

**NULLPKT_FIX** If NULLPKT_FIX is setting as “1”, USB controller will not issue a DMAreq when receive a null packet.

.

## 4.10.3 System integration guide

### 4.10.3.1 USB device configuration

The target audience of this section is the software engineer.

The USB device controller features one control endpoint and 6 other endpoints. The configuration of interfaces and endpoints can be accommodated by software for specific functions, basically supporting Bluetooth HCI, and device firmware upgrade.

Bluetooth HCI transport layer defines the configuration of endpoints and interfaces.

✓ One voice channel with 16-bit encoding.

| Endpoint | Endpoint type | Maximum packet size (Bytes) | Maximum bandwidth (Bytes/ms) | Minimum bandwidth (Bytes/ms) | Double buffer in controller | Generic DMA |
|---|---|---|---|---|---|---|
| Endpoint 0 (command) | Control | 64 | | | No | No |
| Endpoint 1 IN | Bulk (IN) | 64 | | 1024 | No | Yes |
| Endpoint 2 OUT | Bulk (OUT) | 64 | | 1024 | No | Yes |
| Endpoint 2 IN | Bulk (IN) | 64 | 1024 | | No | Yes |
| Endpoint 2 OUT | Bulk (OUT) | 64 | 1024 | | No | Yes |
| Endpoint 3 IN | Interrupt (IN) | 16 | | 16 | No | No |
| Endpoint 4 IN | Interrupt (IN) | 16 | | 16 | No | No |

*When the maximum packet size is less than one half of the device FIFO size (64 Bytes), the double buffer will be automatically enabled by hardware

The pull-up resister on USB transceiver is initially disconnected when boot-up. No external resistor is required. Software should enable it after performing the configuration of USB device controller.

### 4.10.3.2 System infrastructure configuration

The clock, interrupt, and DMA are defined as the system infrastructure. It requires several steps to bring up USB. Those steps should be done in sequence to prevent from malfunction.

Power on:

1. Enable USB PLL.
2. Enable USB clock after USB PLL is settled.
3. Unmask the USB interrupt in the interrupt controller.
4. Enable the pull-up resistor.

USB device controller can generate the interrupt when conditions are met as defined in USB_INTRINE, USB_INTROUTE, and USB_INTRUSBE.

The generic DMA controller is used to move data from or to the USB device controller. The USB device controller will use at most 4 DMA channels for ACL and SCO. The user should use half channel DMA since only the half channel DMA has the hardware flow control.

The USB FIFO provides byte and word accesses to the read/write port of USB_EP0_FIFO, USB_EP1_FIFO, USB_EP2_FIFO, USB_EP3_FIFO, and USB_EP4_FIFO. If the data buffer allocated in memory is word aligned, the user can enable word transfer in DMA controller. If the data buffer allocated in memory is not word aligned, the user should set to byte aligned and set B2W in DMAx_CON to 1 to enable fast byte-to-word transfer. Please refer to DMA section foe more detail.

#### 4.10.3.3 Power on/off USB PHY and Controller Sequence

1 Power on sequence after plug-in.

1.1 Turn on Vusb(PHY 3.3v power) – the control register is in PMIC document.

1.2 Turn on USB AHB clock(78MHz) – the control register is in config document.

1.3 Turn on internal 48MHz PLL – the control register is in clock document.

1.4 Wait 50 usec. (PHY 3.3v power stable time)

1.5 Turn on USB PHY BIAS Current control → reg[USB+08C1h] bit3 = 1. (RG_USB11_FSLS_ENBGRI).

1.6 Wait 10 usec.

1.7 Setting D+ pull up register for connecting Host → reg[USB+0240h] bit 0 =1(PUB)

2 Power off sequence after plug-out.

2.1 Release D+ pull up register for disconnecting Host → Setting reg[USB+0240h] bit 0 =0 (PUB)

2.2 Turn off USB PHY BIAS Current control → reg[USB+08C1h] bit3 = 0. (RG_USB11_FSLS_ENBGRI).

2.3 Turn off Vusb(PHY 3.3v power) – the control register is in PMIC document

## 4.11 SD/MMC Memory Card Controller

### 4.11.1 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 as well as the MultiMediaCard (MMC) bus protocol as defined in MMC system specification version 4.1. Since SD Memory Card bus protocol is backward compatible to MMC bus protocol, the

controller is capable of working well as the host on MMC bus under control of proper firmware. Hereafter, the controller is also abbreviated as SD/MMC 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
- Shared pins for Memory Stick and SD/MMC Memory Card
- 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
- Automatic command execution capability when an interrupt from Memory Stick
- Data rate up to 26 Mbps in serial mode, 26x4 Mbps (26x8 Mbps if 8-bit data line for SD/MMC card is configured) in parallel model, the module is targeted at 26 MHz operating clock
- Serial clock rate on SD/MMC bus is programmable
- Card detection capabilities during sleep mode
- Controllability of power for memory card
- Not support SPI mode for SD/MMC Memory Card
- Not support multiple SD Memory Cards

## 4.11.2 Overview

### 4.11.2.1 Pin Assignment

Since the controller can only be configured as either the host of Memory Stick or the host of SD/MMC Memory Card at one time, pins for Memory Stick and SD/MMC Memory Card are shared in order to save pin counts. The following lists pins required for Memory Stick and SD/MMC Memory Card. **Table 55** shows how they are shared. In **Table 55**, all I/O pads have embedded both pull up and pull down resistor because they are shared by both the Memory Stick and SD/MMC Memory Card. Pins 2,4,5,8 are only useful for SD/MMC Memory Card. Pull down resistor for these pins can

be used for power saving. All embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers if optimal pull-up or pull-down resistors are required on the system board. The pin VDDPD is used for power saving. Power for Memory Stick or SD/MMC Memory Card can be shut down by programming the corresponding control register. The pin WP (Write Protection) is only valid when the controller is configured for SD/MMC Memory Card. It is used to detect the status of Write Protection Switch on SD/MMC Memory Card.

| No. | Name | Type | MMC | SD | MS | MSPRO | Description |
|---|---|---|---|---|---|---|---|
| 1 | SD_CLK | O | CLK | CLK | SCLK | SCLK | Clock |
| 2 | SD_DAT3 | I/O/PP | | CD/DAT3 | | DAT3 | Data Line [Bit 3] |
| 3 | SD_DAT0 | I/O/PP | DAT0 | DAT0 | SDIO | DAT0 | Data Line [Bit 0] |
| 4 | SD_DAT1 | I/O/PP | | DAT1 | | DAT1 | Data Line [Bit 1] |
| 5 | SD_DAT2 | I/O/PP | | DAT2 | | DAT2 | Data Line [Bit 2] |
| 6 | SD_CMD | I/O/PP | CMD | CMD | BS | BS | Command Or Bus State |
| 7 | SD_PWRON | O | | | | | VDD ON/OFF |
| 8 | SD_WP | I | | | | | Write Protection Switch in SD |
| 9 | SD_INS | I | VSS2 | VSS2 | INS | INS | Card Detection |

**Table 55** Sharing of pins for Memory Stick and SD/MMC Memory Card Controller

### 4.11.2.2 Card Detection

For Memory Stick, the host or connector should provide a pull up resistor on the signal INS. Therefore, the signal INS will be logic high if no Memory Stick is on line. The scenario of card detection for Memory Stick is shown in **Figure 39**. Before Memory Stick is inserted or powered on, on host side SW1 shall be closed and SW2 shall be opened for card detection. It is the default setting when the controller is powered on. Upon insertion of Memory Stick, the signal INS will have a transition from high to low. Hereafter, if Memory Stick is removed then the signal INS will return to logic high. If card insertion is intended to not be supported, SW1 shall be opened and SW2 closed always.

For SD/MMC Memory Card, detection of card insertion/removal by hardware is also 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 scenario of card detection for SD/MMC Memory Card is shown in **Figure 40**. Before SD/MMC Memory Card is inserted or powered on, SW1 and SW2 shall be opened for card detection on the host side. Meanwhile, pull down resistor $R_{CD}$ on system board shall attach onto the signal CD/DAT3 by the output signal RCDEN. In addition, SW3 on the card is default to be closed. Upon insertion of SD/MMC Memory Card, the signal CD/DAT3 will have a transition from low to high. If SD/MMC Memory Card is removed then the signal CD/DAT3 will return to logic low. After the card identification process, pull down resistor $R_{CD}$ on system board shall disconnect with the signal CD/DAT3 and SW3 on the card shall be opened for normal operation.

Since the scheme above needs a mechanical switch such as a relay on system board, it is not ideal enough. Thus, a dedicated pin “INS” is used to perform card insertion and removal for SD/MMC. The pin “INS” will connect to the pin “VSS2” of a SD/MMC connector. It is shown in **Figure 39**.

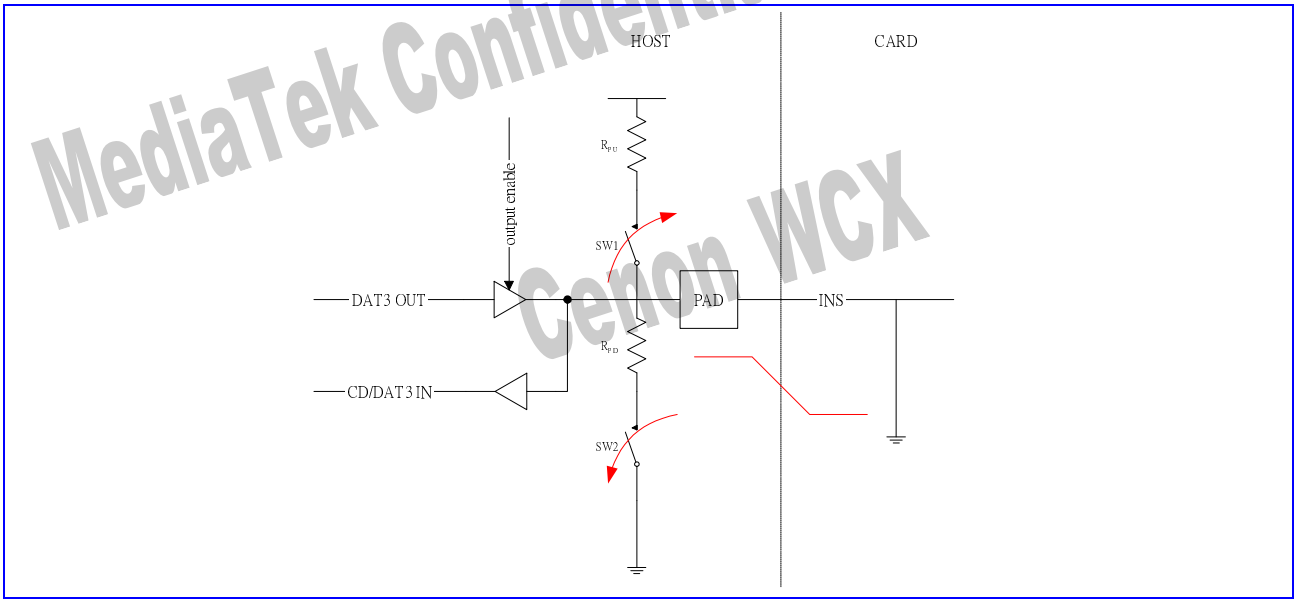



**Figure 39** Card detection for Memory Stick

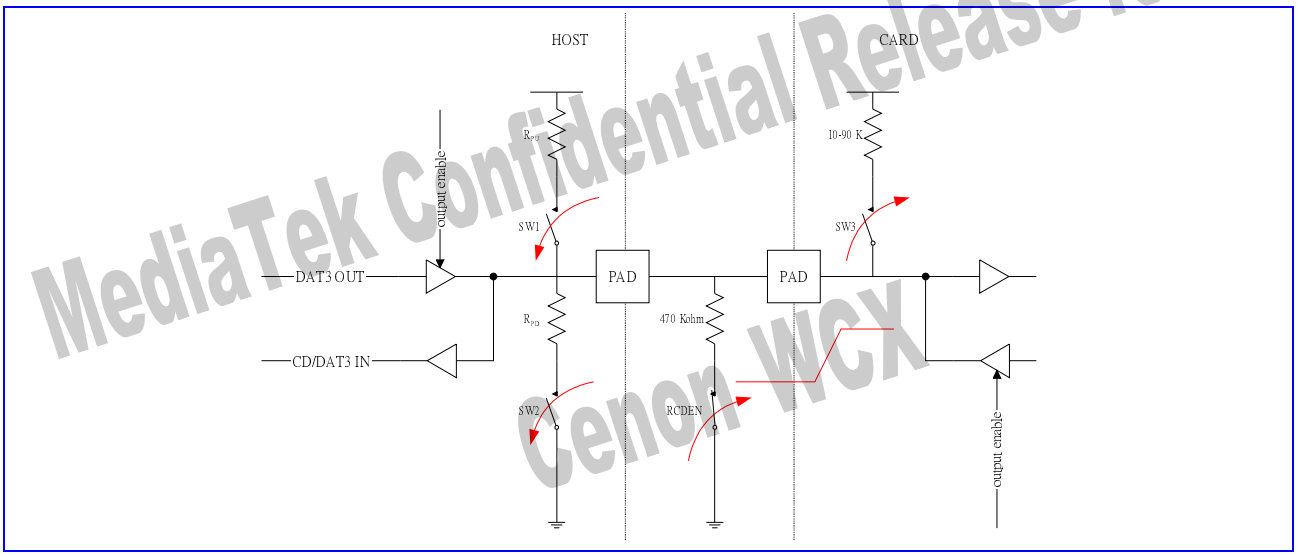



**Figure 40** Card detection for SD/MMC Memory Card

### 4.11.3 Register Definitions

For MT6252, MSDC base address is 0x81110000.

| REGISTER ADDRESS | REGISTER NAME | SYNONYM |
|---|---|---|
| MSDC + 0000h | MS/SD Memory Card Controller Configuration Register | MSDC_CFG |
| MSDC + 0004h | MS/SD Memory Card Controller Status Register | MSDC_STA |
| MSDC + 0008h | MS/SD Memory Card Controller Interrupt Register | MSDC_INT |
| MSDC + 000Ch | MS/SD Memory Card Controller Data Register | MSDC_DAT |
| MSDC + 00010h | MS/SD Memory Card Pin Status Register | MSDC_PS |
| MSDC + 00014h | MS/SD Memory Card Controller IO Control Register | MSDC_IOCON |
| MSDC + 0020h | SD Memory Card Controller Configuration Register | SDC_CFG |
| MSDC + 0024h | SD Memory Card Controller Command Register | SDC_CMD |
| MSDC + 0028h | SD Memory Card Controller Argument Register | SDC_ARG |
| MSDC + 002Ch | SD Memory Card Controller Status Register | SDC_STA |
| MSDC + 0030h | SD Memory Card Controller Response Register 0 | SDC_RESP0 |
| MSDC + 0034h | SD Memory Card Controller Response Register 1 | SDC_RESP1 |
| MSDC + 0038h | SD Memory Card Controller Response Register 2 | SDC_RESP2 |
| MSDC + 003Ch | SD Memory Card Controller Response Register 3 | SDC_RESP3 |
| MSDC + 0040h | SD Memory Card Controller Command Status Register | SDC_CMDSTA |
| MSDC + 0044h | SD Memory Card Controller Data Status Register | SDC_DATSTA |
| MSDC + 0048h | SD Memory Card Status Register | SDC_CSTA |
| MSDC + 004Ch | SD Memory Card IRQ Mask Register 0 | SDC_IRQMASK0 |
| MSDC + 0050h | SD Memory Card IRQ Mask Register 1 | SDC_IRQMASK1 |
| MSDC + 0054h | SDIO Configuration Register | SDIO_CFG |
| MSDC + 0058h | SDIO Status Register | SDIO_STA |

**Table 56** MS/SD Controller Register Map

#### 4.11.3.1 Global Register Definitions

**MSDC+0000h MS/SD Memory Card Controller Configuration Register** **MSDC_CFG**

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

| Name | FIFOTHD | | | | PRCFG2 | | PRCFG1 | | PRCFG0 | | VDDPD | RCDEN | DIRQEN | PINEN | DMAEN | INTEN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | R/W | | | | R/W | | R/W | | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0001 | | | | 01 | | 01 | | 10 | | 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 | RST | NOCRC | | MSDC |
| Type | R/W | | | | | | | | R/W | R/W | R/W | R/W | W | R/W | | R/W |
| Reset | 00000000 | | | | | | | | 0 | 0 | 1 | 0 | 0 | 0 | | 0 |

The register is for general configuration of the MS/SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

**MSDC** The register bit is used to configure the controller as SD/MMC Memory card mode. CLK/CMD/DAT line is pull low when SD/MMC memory card mode is disable..

**0** SD/MMC Memory card disable.

**1** SD/MMC Memory card enable.

**NOCRC** CRC Disable. A '1' indicates that data transfer without CRC is desired. For write data block, data will be transmitted without CRC. For read data block, CRC will not be checked. It is for testing purpose.

**0** Data transfer with CRC is desired.

**1** Data transfer without CRC is desired.

**RST** Software Reset. Writing a '1' to the register bit will cause internal synchronous reset of MS/SD controller, but does not reset register settings.

**0** Otherwise

**1** Reset MS/SD controller

**CLKSRC** The register bit specifies which clock is used as source clock of memory card. If MUC clock is used, the fastest clock rate for memory card is 104/4=26MHz. If MCPLL clock is used, the fastest clock rate for memory card is 48/2 =24MHz.

**0** Use MCU clock as source clock of memory card.

**1** Use MCPLL clock as source clock of memory card.

**STDBY** Standby Mode. If the module is powered down, operating clock to the module will be stopped. At the same time, clock to card detection circuitry will also be stopped. If detection of memory card insertion and removal is desired, write '1' to the register bit. If interrupt for detection of memory card insertion and removal is enabled, interrupt will take place whenever memory is inserted or removed.

**0** Standby mode is disabled.

**1** Standby mode is enabled.

**RED** Rise Edge Data. The register bit is used to determine that serial data input is latched at the falling edge or the rising edge of serial clock. The default setting is at the rising edge. If serial data has worse timing, set the register bit to '1'. **When memory card has worse timing on return 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.

**SCLKON** Serial Clock Always On. It is for debugging purpose.

**0** Not to have serial clock always on.

**1** To have serial clock always on.

**SCLKF** The register field controls clock frequency of serial clock on MS/SD bus. Denote clock frequency of MS/SD bus serial clock as $f_{slave}$ and clock frequency of the MS/SD controller as $f_{host}$ which is 104 or 52 MHz. Then the value of the register field is as follows. **Note that the allowable maximum frequency of $f_{slave}$ is 26MHz. While changing clock rate, it needs " 1T clock period before change + 1T clock period after change" for HW signal to re-synchronize.**

**00000000b** $f_{slave} = (1/2) * f_{host}$

**00000001b** $f_{slave} = (1/(4*1)) * f_{host}$

**00000010b** $f_{slave} = (1/(4*2)) * f_{host}$

**00000011b** $f_{slave} = (1/(4*3))* f_{host}$

…

**00010000b** $f_{slave} = (1/(4*16))* f_{host}$

…

**11111111b** $f_{slave} = (1/(4*255)) * f_{host}$

**INTEN** Interrupt Enable. Note that if interrupt capability is disabled then application software must poll the status of the register MSDC_STA to check for any interrupt request.

**0** Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD/MMC 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/MMC Memory Card or Memory Stick.

**DMAEN** DMA Enable. Note that if DMA capability is disabled then application software must poll the status of the register MSDC_STA for checking any data transfer request. If DMA is desired, the register bit must be set 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/MMC 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/MMC Memory Card or Memory Stick.

**PINEN** Pin Interrupt Enable. 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.

**DIRQEN** Data Request Interrupt Enable. 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.

**RCDEN** The register bit controls the output pin RCDEN that is used for card identification process when the controller is for SD/MMC Memory Card. Its output will control the pull down resistor on the system board to connect or disconnect with the signal CD/DAT3.

**0** The output pin RCDEN will output logic low.

**1** The output pin RCDEN will output logic high.

**VDDPD** The register bit controls the output pin VDDPD that is used for power saving. The output pin VDDPD will control power for memory card.

**0** The output pin VDDPD will output logic low. The power for memory card will be turned off.

**1** The output pin VDDPD will output logic high. The power for memory card will be turned on.

**PRCFG0** Pull Up/Down Register Configuration for the pin WP. The default value is 10.

**00** Pull up resistor and pull down resistor in the I/O pad of the pin WP are all disabled.

**01** Pull down resistor in the I/O pad of the pin WP is enabled.

**10** Pull up resistor in the I/O pad of the pin WP is enabled.

**11** Use keeper of IO pad.

**PRCFG1** Pull Up/Down Register Value for the pins CMD/BS. The default value is 0b01. **Note that pull up configuration for the pins CMD/BS with the register bit MCCPUPD in ACIF_CON3 setting to 1. Pull down configuration for the pins CMD/BS with the register bit MCCPUPD in ACIF_CON3(0x8001070c) setting to 0.**

**00** Pull up resistor and pull down resistor in the I/O pad of the pin CMD/BS are all disabled.

**01** Pull up/down resistor in the I/O pad of the pin CMD/BS value is 47k.

**10** Pull up/down resistor in the I/O pad of the pin CMD/BS value is 47k.

**11** Pull up/down resistor in the I/O pad of the pin CMD/BS value is 23.5k.

**PRCFG2** Pull Up/Down Register Value for the pins DAT0, DAT1, DAT2, DAT3. The default value is 0b01. **Note that pull up configuration for the pins DAT0, DAT1, DAT2, DAT3 with the register bit MCDPUPD in ACIF_CON3 setting to 1. Pull down configuration for the pins DAT0, DAT1, DAT2, DAT3 with the register bit MCDPUPD in ACIF_CON3(0x8001070c) setting to 0. And DAT pin enable configuration from DAT0 to DAT3 is the register bit from GPIO71 to GPIO68 in GPIO_PULLSEL5(0x80020090).**

**00** Pull up resistor and pull down resistor in the I/O pad of the pin DAT are all disabled.

**01** Pull up/down resistor in the I/O pad of the pin DAT value is 47k.

**10** Pull up/down resistor in the I/O pad of the pin DAT value is 47k.

**11** Pull up/down resistor in the I/O pad of the pin DAT value is 23.5k.

**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 larger 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 larger 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.
**…**
**1000** Threshold value is 8.
**others** Invalid

## MSDC+0004h MS/SD Memory Card Controller Status Register MSDC_STA

| 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 | W | | | | | | | RO | | | | RO | RO | RO | RO |
| Reset | 0 | - | | | | | | | 0000 | | | | 0 | 0 | 0 | 0 |

The register contains the status of FIFO, interrupts and data requests.

**BF** The register bit indicates if FIFO in MS/SD controller is full.
**0** FIFO in MS/SD controller is not full.
**1** FIFO in MS/SD controller is full.

**BE** The register bit indicates if FIFO in MS/SD controller is empty.
**0** FIFO in MS/SD controller is not empty.
**1** FIFO in MS/SD controller is empty.

**DRQ** The register bit indicates if any data transfer is required. While any data transfer is required, the register bit still will be active even if the register bit DIRQEN in the 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. While the register bit DIRQEN in the register MSDC_CFG is disabled, the second method is used.
**0** No DMA request exists.
**1** DMA request exists.

**INT** The register bit indicates if any interrupt exists. While any interrupt exists, the register bit still will be active even if the register bit INTEN in the register MSDC_CFG is disabled. MS/SD controller can interrupt MCU by issuing interrupt request to Interrupt Controller, or software/application polls the register endlessly to check if any interrupt request exists in MS/SD controller. While the register bit INTEN in the register MSDC_CFG is

disabled, the second method is used. For read commands, it is possible that timeout error takes place. Software can read the status register to check if timeout error takes place without OS time tick support or data request is asserted. Note that the register bit will be cleared when reading the register MSDC_INT.

**0** No interrupt request exists.

**1** Interrupt request exists.

**FIFOCNT** FIFO Count. The register field shows how many valid entries are in FIFO.

**0000** There is 0 valid entry in FIFO.

**0001** There is 1 valid entry in FIFO.

**0010** There are 2 valid entries in FIFO.

**…**

**1000** There are 8 valid entries in FIFO.

**others** Invalid

**FIFOCLR** Clear FIFO. Writing '1' to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.

**0** No effect on FIFO.

**1** Clear the content of FIFO clear and reset the status of FIFO controller.

**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.

## MSDC+0008h MS/SD Memory Card Controller Interrupt Register MSDC_INT

| 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 |

The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, the register bit INTEN of the register MSDC_CFG is set to '0. It implies that software interrupt can be implemented by polling the register bit INT of the register MSDC_STA and this register. **However, if hardware interrupt is desired, remember to clear the register before setting the register bit INTEN of the register MSDC_CFG to '1'. Or undesired hardware interrupt arisen from previous interrupt status may take place.**

**DIRQ** Data Request Interrupt. The register bit indicates if any interrupt for data request exists. Whenever data request exists and data request as an interrupt source is enabled, i.e., the register bit DIRQEN in the register MSDC_CFG is set to '1', the register bit will be active. It will be reset when reading it. For software, data requests can be recognized by polling the 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.

**PINIRQ** Pin Change Interrupt. The register bit indicates if any interrupt for memory card insertion/removal exists. Whenever memory card is inserted or removed and card detection interrupt is enabled, i.e., the register bit PINEN in the 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.

**SDCMDIRQ** SD Bus CMD Interrupt. The register bit indicates if any interrupt for SD CMD line exists. 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.

**SDDATIRQ** SD Bus DAT Interrupt. The register bit indicates if any interrupt for SD DAT line exists. Whenever interrupt for SD DAT line exists, i.e., any bit in the 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.

**SDMCIRQ** SD Memory Card Interrupt. The register bit indicates if any interrupt for SD Memory Card exists. Whenever interrupt for SD Memory Card exists, i.e., any bit in the 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.

0 No SD Memory Card interrupt.
1 SD Memory Card interrupt exists.

**SDR1BIRQ** SD/MMC R1b Response Interrupt. The register bit will be active when a SD/MMC command with R1b response finishes and the DAT0 line has transition from busy to idle state. **Single block write commands with R1b response will cause the interrupt when the command completes no matter successfully or with CRC error. However, multi-block write commands with R1b response do not cause the 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 the interrupt. Single block read command with R1b response will cause the interrupt when the command completes but multi-block read commands do not. Note that STOP_TRANS commands (with R1b response) behind multi-block read commands will cause the interrupt.**

0 No interrupt for SD/MMC R1b response.
1 Interrupt for SD/MMC R1b response exists.

## MSDC+000Ch MS/SD Memory Card Controller Data Register MSDC_DAT

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DATA[31:16] | | | | | | | | | | | | | | | |
| Type | 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 | | | | | | | | | | | | | | | |

The register is used to read/write data from/to FIFO inside MS/SD controller. Data access is in unit of 32 bits.

## MSDC+0010h MS/SD Memory Card Pin Status Register MSDC_PS

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | CMD | DAT | | | | | | | |
| Type | | | | | | | | RO | RO | | | | | | | |
| Reset | | | | | | | | - | - | | | | | | | |
| 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 | R/W | R/W | R/W |
| Reset | 0000 | | | | | | | | | | | 0 | 1 | 0 | 0 | 0 |

The register is used for card detection. When the memory card controller is powered on, and the system is powered on, the power for the memory card is still off unless power has been supplied by the PMIC. Meanwhile, 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/MMC.

For detecting card insertion, first pull up INS pin, and then enable card detection and input pin at the same time. After 32 cycles of controller clock, status of pin changes will emerge. For detecting card removal, just keep enabling card detection and input pin.

**CDEN** Card Detection Enable. The register bit is used to enable or disable card detection.
- **0** Card detection is disabled.
- **1** Card detection is enabled.

**PIEN0** The register bit is used to control input pin for card detection.
- **0** Input pin for card detection is disabled.
- **1** Input pin for card detection is enabled.

**POEN0** The register bit is used to control 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.

**PIN0** The register 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.

**PINCHG** Pin Change. The register bit indicates the status of card insertion/removal. If 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.

**CDDEBOUNCE** The register field specifies the time interval for card detection de-bounce. Its default value is 0. It means that de-bounce interval is 32 cycle time of 32KHz. The interval will extend one cycle time of 32KHz by increasing the counter by 1.

**DAT** Memory Card Data Lines.

**CMD** Memory Card Command Lines.

## MSDC+0014h MS/SD Memory Card Controller IO Control Register MSDC_IOCON

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DLT | | | | | | | | | | | CRCDIS | CMDSEL | INTLH | | DSW |
| Type | R/W | | | | | | | | | | | R/W | R/W | R/W | | R/W |
| Reset | 00000010 | | | | | | | | | | | 0 | 0 | 0 | | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | CMDRE | | | | | | PRCFG3 | | SRCFG1 | SRCFG0 | ODCCFG1 | | | ODCCFG0 | | |
| Type | R/W | | | | | | R/W | | R/W | R/W | R/W | | | R/W | | |
| Reset | 0 | | | | | | 10 | | 1 | 1 | 000 | | | 011 | | |

The register specifies **Output Driving Capability** and **Slew Rate** of IO pads for MSDC. The reset value is suggestion setting. If output driving capability of the pins DAT0, DAT1, DAT2 and DAT3 is too large, it's 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.

**ODCCFG0** Output driving capability the pins CMD/BS and SCLK

**000** 4mA

**010** 12mA

**100** 8mA

**110** 16mA

**ODCCFG1** Output driving capability the pins DAT0, DAT1, DAT2 and DAT3

**000** 4mA

**010** 12mA

**100** 8mA

| | |
|---|---|
| 110 | 16mA |
| **SRCFG0** | Output driving capability the pins CMD/BS and SCLK |
| 0 | Fast Slew Rate |
| 1 | Slow Slew Rate |
| **SRCFG1** | Output driving capability the pins DAT0, DAT1, DAT2 and DAT3 |
| 0 | Fast Slew Rate |
| 1 | Slow Slew Rate |
| **PRCFG3** | Pull Up/Down Register Configuration for the pin INS. The default value is 10. |
| 00 | Pull up resistor and pull down resistor in the I/O pad of the pin INS are all disabled. |
| 01 | Pull down resistor in the I/O pad of the pin INS is enabled. |
| 10 | Pull up resistor in the I/O pad of the pin INS is enabled. |
| 11 | Use keeper of IO pad. |
| **CMDRE** | The register bit is used to determine whether the host should latch response token (which is sent from card on CMD line ) at rising edge or falling edge of serial clock. |
| 0 | Host latches response at rising edge of serial clock |
| 1 | Host latches response at falling edge of serial clock |
| **DSW** | The register bit is used to determine whether the host should latch data with 1-T delay or not. For SD/MMC card, this bit is suggest to be 0. |
| 0 | Host latches the data with 1-T delay |
| 1 | Host latches the data without 1-T delay |
| **INTLH** | This field is used to select the latch timing for SDIO multi-block read interrupt. Note that, SDIO is not support in MT6252. |
| 00 | Host latches INT at the second backend clock after the end bit of current data block from card is received. (This is the default setting) |
| 01 | Host latches INT at the first backend clock after the end bit of current data block from card is received. |
| 10 | Host latches INT at the second backend clock after the end bit of current data block from card is received. |
| 11 | Host latches INT at the third backend clock after the end bit of current data block from card is received. |
| **CMDSEL** | The register bit is used to determine 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. |
| **CRCDIS** | The register bit is used to switch-off the data CRC check for SD/MMC read data. |
| 0 | CRC Check is on. |
| 1 | CRC Check is off. |

**DLT** Data Latch Timing. The register is used for SW to select the latch timing on data line.

**Figure 41** illustrates the data line latch timing. sclk_out is the serial clock output to card. div_clk is the internal clock used for generating divided clock. The number "1 2 1 2" means the current sclk_out is divided from div_clk by a ratio of 2. data_in is the output data from card, and latched_data(r)/(f) is the rising/falling edge latched data inside the host (configured by RED in MSDC_CFG). In this example, SCLKF(in MSDC_CFG) is set to 8'b0 which means the division ratio is 2, and DLT is set to 1. Note that the value of DLT CANNOT be set as 0 and its value should not exceed the division ratio ( in the example, the division ratio is 2). Also note that, the latching time will be one div_clk later than the indicated DLT value and the falling edge is always half div_clk ahead from rising edge. The default value of DLT is set to 8'b2.

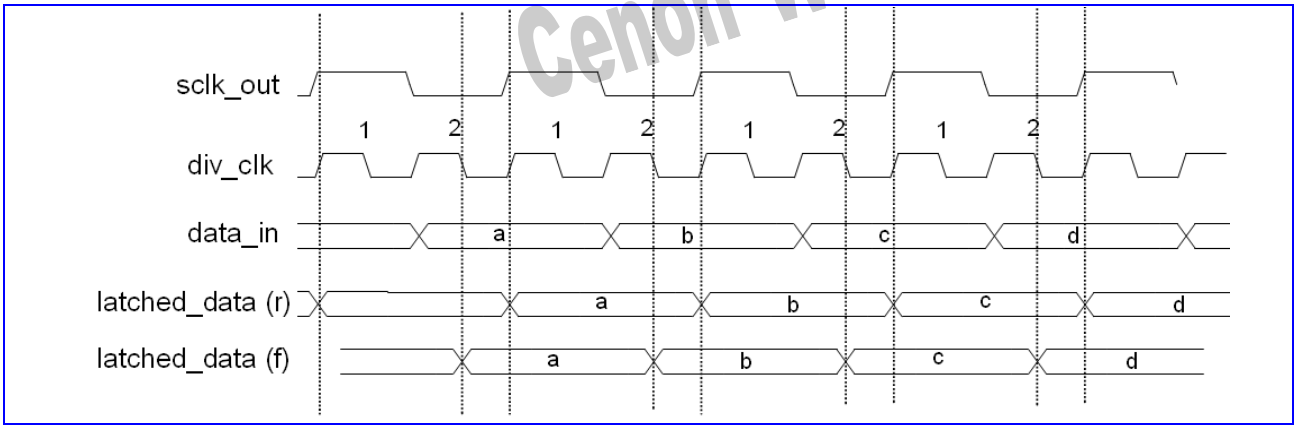



**Figure 41** Illustration of data line latch timing

### 4.11.3.2 SD Memory Card Controller Register Definitions

#### MSDC+0020h SD Memory Card Controller Configuration Register SDC_CFG

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DTOC | | | | | | | | WDOD | | | | SDIO | | MDLEN | SIEN |
| Type | R/W | | | | | | | | R/W | | | | R/W | | R/W | R/W |
| Reset | 00000000 | | | | | | | | 0000 | | | | 0 | | 0 | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | BSYDLY | | | | BLKLEN | | | | | | | | | | | |
| Type | R/W | | | | R/W | | | | | | | | | | | |
| Reset | 1000 | | | | 00000000000 | | | | | | | | | | | |

The register is used for configuring the MS/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 have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

**BLKLEN** It refers to Block Length. The register field is used to define the length of one block in unit of byte in a data transaction. The maximal value of block length is 2048 bytes.

- **000000000000** Reserved.
- **000000000001** Block length is 1 byte.
- **000000000010** Block length is 2 bytes.
- …
- **011111111111** Block length is 2047 bytes.
- **100000000000** Block length is 2048 bytes.

**BSYDLY** The register field is only valid for the commands with R1b response. If the command has a response of R1b type, MS/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 operations in SD/MMC 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, then the controller will abandon the detection.

- **0000** No extend.
- **0001** Extend one more serial clock cycle.
- **0010** Extend two more serial clock cycles.
- …
- **1111** Extend fifteen more serial clock cycle.

**SIEN** Serial Interface Enable. It should be enabled as soon as possible before any command.

- **0** Serial interface for SD/MMC is disabled.
- **1** Serial interface for SD/MMC is enabled.

**SDIO** SDIO Enable. Note that, SDIO is not support in MT6252.

- **0** SDIO mode is disabled
- **1** SDIO mode is enabled

**MDLEN** Multiple Data Line Enable. 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 MultiMediaCard is applied. If an MultiMediaCard is applied and 4-bit data line is enabled, then 4 bits will be output every serial clock. Therefore, data integrity will fail. Note that, Only 1-bit mode could be supported in MT6252.

- **0** 4-bit Data line is disabled.
- **1** 4-bit Data line is enabled.

**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 extend.
**0001** Extend one more serial clock cycle.
**0010** Extend two more serial clock cycles.
…
**1111** Extend fifteen more serial clock cycle.

**DTOC** Data Timeout 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 description of the register bit RDINT for reference.

**00000000** Extend 65,536 more serial clock cycle.
**00000001** Extend 65,536x2 more serial clock cycle.
**00000010** Extend 65,536x3 more serial clock cycle.
…
**11111111** Extend 65,536x 256 more serial clock cycle.

## MSDC+0024h SD Memory Card Controller Command Register SDC_CMD

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CTOC | | | | | | | | | | | | | | | CMDFAIL |
| Type | R/W | | | | | | | | | | | | | | | R/W |
| Reset | 01000000 | | | | | | | | | | | | | | | 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 | R/W | R/W | R/W | R/W | | R/W | R/W | | | R/W | R/W | | | | | |
| Reset | 0 | 0 | 0 | 00 | | 0 | 000 | | | 0 | 000000 | | | | | |

The register defines a SD Memory Card command and its attribute. Before MS/SD controller issues a transaction onto SD bus, application shall specify other relative setting such as argument for command. After application writes the register, MS/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/MMC bus run 128 cycles before issuing the command.

**CMD** SD Memory Card command. It is totally 6 bits.

**BREAK** Abort a 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.

**RSPTYP** The register field defines response type for the command. For commands with R1 and R1b response, the register SDC_CSTA (not SDC_STA) will update after response token is received. This register SDC_CSTA contains the status of the SD/MMC and it will be used as response interrupt sources. Note that if CMD7 is used with all 0's RCA then RSPTYP must be "000". And the command "GO_TO_IDLE" also have RSPTYP='000'.

**000** There is no response for the command. For instance, 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, but 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, MS/SD controller must monitor the data line 0 for card busy status from the bit time that is two or four serial clock cycles after the command end bit to check if operations in SD/MMC Memory Card have 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 four serial clock cycles. The second case is that the card is in idle state or under a scenario of receiving a stop transmission command between data blocks when multiple block write command is in progress. The register bit is valid only when the command has a response token.

**IDRT** Identification Response Time. The register bit indicates if the command has a response with $N_{ID}$ (that is, 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 $N_{ID}$ response time.

**DTYPE** The register field defines data token type for the command.

**00** No data token for the command

**01** Single block transaction

**10** Multiple block transaction. That is, the command is a multiple block read or write command.

**11** Stream operation. It only shall be used when an MultiMediaCard is applied.

**RW** The register bit defines the command is a read command or write command. The register bit is valid only when the command will cause a transaction with data token.

**0** The command is a read command.

1 The command is a write command.

**STOP** The register bit 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.

**INTC** The register bit 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.

## MSDC+0028h SD Memory Card Controller Argument Register SDC_ARG

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

The register contains the argument of the SD/MMC Memory Card command.

## MSDC+002Ch SD Memory Card Controller Status Register SDC_STA

| 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 | R | | | | | | | | | | | RO | RO | RO | RO | RO |
| Reset | - | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 |

The register contains various status of MS/SD controller as the controller is configured as the host of SD Memory Card.

**BESDCBUSY** The register field indicates if MS/SD controller is busy, that is, any transmission is going on CMD or DAT line on SD bus. This bit shows backend controller's SDC busy state. The busy state is sync from card clock domain to bus clock domain.

0 Backend MS/SD controller is idle.

1 Backend MS/SD controller is busy.

**BECMDBUSY** The register field indicates if any transmission is going on CMD line on SD bus. This bit shows backend controller's CMD busy state. The busy state is sync from card clock domain to bus clock domain.

0 Backend MS/SDC Controller gets the info that no transmission is going on CMD line on SD bus.

1 Backend MS/SDC Controller gets the info that there exists transmission going on CMD line on SD bus.

**BEDATBUSY** The register field indicates if any transmission is going on DAT line on SD bus.

0 Backend MS/SDC Controller gets the info that no transmission is going on DAT line on SD bus.

1 Backend MS/SDC Controller gets the info that there exists transmission going on DAT line on SD bus.

**FECCMDBUSY** The register field indicates if any transmission is 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 exists transmission going on CMD line on SD bus.

**FEDATBUSY** The register field indicates if any transmission is 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, then checking if the register bit is '0' before issuing next command with data would not guarantee that the controller is idle. In this situation, use the register bit BESDCBUSY.**

0 No transmission is going on DAT line on SD bus.

1 There exists transmission going on DAT line on SD bus.

**WP** It is used to detect 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. The pin WP (Write Protection) is also only useful while the controller is configured for SD Memory Card. Note that, write protection feature is not support in MT6252.

1 Write Protection Switch ON. It means that memory card is desired to be write-protected.

0 Write Protection Switch OFF. It means that memory card is writable.

**CTOC** Command Timeout Counter. The period between an end bit of command and a start bit of response except CMD2 and ACMD41. The unit of the counter is one serial clock cycle.

## MSDC+0030h SD Memory Card Controller Response Register 0 SDC_RESP0

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

The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.

## MSDC+0034h SD Memory Card Controller Response Register 1 SDC_RESP1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESP [63:48] | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Name | RESP [47:32] |
|---|---|
| Type | RO |

The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.

## MSDC+0038h SD Memory Card Controller Response Register 2 SDC_RESP2

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESP [95:80] | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | RESP [79:64] | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |

The register contains parts of the last SD/MMC Memory Card bus response. See description for the register field SDC_RESP3.

## MSDC+003Ch SD Memory Card Controller Response Register 3 SDC_RESP3

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESP [127:112] | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | RESP [111:96] | | | | | | | | | | | | | | | |
| Type | RO | | | | | | | | | | | | | | | |

The register contains parts of the last SD/MMC Memory Card bus response. The register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD/MMC 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 response token is stored in the register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of response token is stored in the register field SDC_RESP0.

## MSDC+0040h SD Memory Card Controller Command Status Register SDC_CMDSTA

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | MMCIRQ | RSPCRCERR | CMDTO | CMDRDY |
| Type | | | | | | | | | | | | | RC | RC | RC | RC |
| Reset | | | | | | | | | | | | | 0 | 0 | 0 | 0 |

The register contains the status of MS/SD controller during command execution and that of MS/SD bus protocol after command execution when MS/SD controller is configured as the host of SD/MMC Memory Card. The register will also be used as interrupt sources. The register will be cleared when reading the register. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will deassert the interrupt.

**CMDRDY** For command without response, the register bit will be '1' once the command completes on SD/MMC bus. For command with response, the register bit will be '1' whenever the command is issued onto SD/MMC bus and its corresponding response is received **without CRC error**.

**0** Otherwise.

**1** Command with/without response finish successfully without CRC error.

**CMDTO** Timeout on CMD detected. A '1' indicates that MS/SD controller detected a timeout condition while waiting for a response on the CMD line.

**0** Otherwise.

**1** MS/SD controller detected a timeout condition while waiting for a response on the CMD line.

**RSPCRCERR** CRC error on CMD detected. A '1' indicates that MS/SD controller detected a CRC error **after reading a response from the CMD line.**

**0** Otherwise.

**1** MS/SD controller detected a CRC error after reading a response from the CMD line.

**MMCIRQ** MMC requests an interrupt. A '1' indicates that a MMC supporting command class 9 issued an interrupt request.

**0** Otherwise.

**1** A '1' indicates that a MMC supporting command class 9 issued an interrupt request.

## MSDC+0044h SD Memory Card Controller Data Status Register SDC_DATSTA

| 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 |

The register contains the status of MS/SD controller during data transfer on DAT line(s) when MS/SD controller is configured as the host of SD/MMC Memory Card. The register also will be used as interrupt sources. The register will be cleared when reading the register. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will deassert the interrupt.

**BLKDONE** The register bit indicates the status of data block transfer.

**0** Otherwise.

**1** A data block was successfully transferred.

**DATTO** Timeout on DAT detected. A '1' indicates that MS/SD controller detected a timeout condition while waiting for data token on the DAT line.

**0** Otherwise.

**1** MS/SD controller detected a timeout condition while waiting for data token on the DAT line.

**DATCRCERR** CRC error on DAT detected. A '1' indicates that MS/SD controller detected a CRC error after reading a block of data from the DAT line or SD/MMC signaled a CRC error after writing a block of data to the DAT line.

**0** Otherwise.

**1** MS/SD controller detected a CRC error after reading a block of data from the DAT line or SD/MMC signaled a CRC error after writing a block of data to the DAT line.

## MSDC+0048h SD Memory Card Status Register SDC_CSTA

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CSTA [31:16] | | | | | | | | | | | | | | | |
| Type | RC | | | | | | | | | | | | | | | |
| Reset | 0000000000000000 | | | | | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | CSTA [15:0] | | | | | | | | | | | | | | | |
| Type | RC | | | | | | | | | | | | | | | |
| Reset | 0000000000000000 | | | | | | | | | | | | | | | |

After commands with R1 and R1b response this register contains the status of the SD/MMC card and it will be used as response interrupt sources. In all register fields, logic high indicates error and logic low indicates no error. The register will be cleared when reading the register. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will deassert the interrupt.

**CSTA31** **OUT_OF_RANGE**. The command's argument was out of the allowed range for this card.

**CSTA30** **ADDRESS_ERROR**. A misaligned address that did not match the block length was 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 occurred.

**CSTA27** **ERASE_PARAM**. An invalid selection of write-blocks for erase occurred.

**CSTA26** **WP_VIOLATION**. Attempt to program a write-protected block.

**CSTA25** Reserved. Return zero.

**CSTA24** **LOCK_UNLOCK_FAILED**. Set when a sequence or password error has been detected in lock/unlock card command or if there was an attempt to access a locked card.

**CSTA23** **COM_CRC_ERROR**. The CRC check of the previous command failed.

**CSTA22** **ILLEGAL_COMMAND**. Command not legal for the card state.

**CSTA21** **CARD_ECC_FAILED**. Card internal ECC was applied but failed to correct the data.

**CSTA20** **CC_ERROR**. Internal card controller error.
**CSTA19** **ERROR**. A general or an unknown error occurred during the operation.
**CSTA18** **UNDERRUN**. The card could not sustain data transfer in stream read mode.
**CSTA17** **OVERRUN**. The card could not 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 original) or permanent WP (unprotected) bits was made.
**CSTA[15:4]** Reserved. Return zero.
**CSTA3** **AKE_SEQ_ERROR**. Error in the sequence of authentication process
**CSTA[2:0]** Reserved. Return zero.

## MSDC+004Ch SD Memory Card IRQ Mask Register 0

**SDC_IRQMASK 0**

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQMASK [31:16] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 0000000000000000 | | | | | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQMASK [15:0] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 0000000000000000 | | | | | | | | | | | | | | | |

The register contains parts of SD Memory Card Interrupt Mask Register. See the register description of the register SDC_IRQMASK1 for reference. The register will mask interrupt sources from the register SDC_CMDSTA and SDC_DATSTA. IRQMASK[15:0] is for SDC_CMDSTA and IRQMASK[31:16] for SDC_DATSTA. A '1' in some bit of the register will mask the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is '1' then interrupt source from the register field CMDRDY of the register SDC_ CMDSTA will be masked. A '0' in some bit will not cause interrupt mask on the corresponding interrupt source from the register SDC_CMDSTA and SDC_DATSTA.

## MSDC+0050h SD Memory Card IRQ Mask Register 1

**SDC_IRQMASK 1**

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | IRQMASK [63:48] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Reset | 0000000000000000 | | | | | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | IRQMASK [47:32] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |

| Reset | 0000000000000000 |
|---|---|

The register contains parts of SD Memory Card Interrupt Mask Register. The registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD Memory Card Interrupt Mask Register. The register will mask interrupt sources from the register SDC_CSTA. A '1' in some bit of the register will mask the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is '1' then interrupt source from the register field OUT_OF_RANGE of the register SDC_ CSTA will be masked. A '0' in some bit will not cause interrupt mask on the corresponding interrupt source from the register SDC_ CSTA.

## MSDC+0054h SDIO Configuration Register SDIO_CFG

| 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 | | | | | | | | | | | | | | DSBSEL | INTSEL | INTEN |
| Type | | | | | | | | | | | | | | R/W | R/W | R/W |
| Reset | | | | | | | | | | | | | | 0 | 0 | 0 |

The register is used to configure functionality for SDIO. Note that, SDIO is not support in MT6252.

**INTEN** Interrupt enable for SDIO.

- **0** Disable
- **1** Enable

**INTSEL** Interrupt Signal Selection

- **0** Use data line 1 as interrupt signal
- **1** Use data line 5 as interrupt signal

**DSBSEL** Data Block Start Bit Selection.

- **0** Use data line 0 as start bit of data block and other data lines are ignored.
- **1** Start bit of a data block is received only when data line 0-3 all become low.

## MSDC+0058h SDIO Status Register SDIO_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 | | | | | | | | | | | | | | | | IRQ |
| Type | | | | | | | | | | | | | | | | RO |
| Reset | | | | | | | | | | | | | | | | 0 |

## 4.11.4 Application Notes

### 4.11.4.1 Initialization Procedures After Power On

```
Disable power down control for MSDC module
```

Remember to power on MSDC module before starting any operation to it.

### 4.11.4.2 Card Detection Procedures

The pseudo code is as follows:

```
MSDC_CFG.PRCFG0 = 2'b10
MSDC_PS = 2'b11
MSDC_CFG.VDDPD = 1
if(MSDC_PS.PINCHG) { // card is inserted
     . . .
}
```

The pseudo code segment perform the following tasks:

1. First pull up CD/DAT3 (INS) pin.
2. Enable card detection and input pin at the same time.
3. Turn on power for memory card.
4. Detect insertion of memory card.

### 4.11.4.3 Notes on Commands

For MS, check if MSC_STA.RDY is '1' before issuing any command.

For SD/MMC, if the command desired to be issued involves data line, for example, commands with data transfer or R1b response, check if SDC_STA.SDCBUSY is '0' before issuing. If the command desired to be issued does not involve data line, only check if SDC_STA.CMDBUSY is '0' before issuing.

### 4.11.4.4 Notes on Data Transfer

- For SD/MMC, if multiple-block-write command is issued then only issue STOP_TRANS command inter-blocks instead of intra-blocks.
- Once SW decides to issue STOP_TRANS commands, no more data transfer from or to the controller.

### 4.11.4.5 Notes on Frequency Change

Before changing the frequency of serial clock on MS/SD/MMC bus, it is necessary to disable serial interface of the controller. That is, set the register bit SIEN of the register SDC_CFG to '0' for SD/MMC controller, and set the register bit SIEN of the register MSC_CFG to '0' for Memory Stick controller. Serial interface of the controller needs to be enabled again before starting any operation to the memory card.

### 4.11.4.6 Notes on Response Timeout

If a read command doest not receive response, that is, it terminates with a timeout, then register SDC_DATSTA needs to be cleared by reading it. The register bit "DATTO" should be active. However, it may take a while before the register bit becomes active. The alternative is to send the STOP_TRANS command. However, this method will receive response with illegal-command information. Also, remember to check if the register bit SDC_STA.CMDBUSY is active before issuing the STOP_TRANS command. The procedure is as follows:

1. Read command => response time out
2. Issue STOP_TRANS command => Get Response
3. Read register SDC_DATSTA to clear it

### 4.11.4.7 Source or Destination Address is not word-aligned

It is possible that the source address is not word-aligned when data move from memory to MSDC. Similarly, destination address may be not word-aligned when data move from MSDC to memory. This can be solved by setting DMA byte-to-word functionality.

1. DMAn_CON.SIZE=0
2. DMAn_CON.BTW=1
3. DMAn_CON.BURST=2 (or 4)
4. DMAn_COUNT=byte number instead of word number
5. fifo threshold setting must be 1 (or 2), depending on DMAn_CON.BURST

Note n=4 ~ 11

### 4.11.4.8 Miscellaneous notes

- Siemens MMC card: When a write command is issued and followed by a STOP_TRANS command, Siemens MMC card will de-assert busy status even though flash programming has not yet finished. Software must use "Get Status" command to make sure that flash programming finishes.

MediaTek Confidential Release for Cenon WCX

# 5 Multi-Media Subsystem

MT6252 is specially designed to support multi-media terminals. It integrates several hardware based accelerators, like advanced LCD display controller, camera interface, hardware image Resizer and Rotator. This chapter describes those functional blocks in detail.

## 5.1 LCD Interface

### 5.1.1 General Description

MT6252H contains a versatile LCD controller which is optimized for multimedia applications. This controller supports many types of LCD modules and contains a rich feature set to enhance the functionality. These features are:

- Up to 240 x 320 resolution
- The internal frame buffer supports 8bpp indexed color, RGB 565, RGB 888, ARGB 8888, PARGB 8888 and YUYV422 format.
- Supports 8-bpp (RGB332), 12-bpp (RGB444), 16-bpp (RGB565), 18-bpp (RGB666) and 24-bpp (RGB888) LCD modules.
- 4 Layers Overlay with individual color depth, window size, vertical and horizontal offset, source key, alpha value and display rotation control(90°,180°, 270°, mirror and mirror then 90°, 180° and 270°)

For parallel LCD modules, the LCD controller can reuse external memory interface or use dedicated 8/9-bit parallel interface to access them and 8080 type interface is supported. It can transfer the display data from the internal SRAM or external SRAM/Flash Memory to the off-chip LCD modules. Additionally, when used with page-mode pSRAM, the LCM may be connected to the EMI to share pins. The LCD controller can be set to update pixel data to this interface.

For serial LCD modules, this interface performs parallel to serial conversion and supports 8, 9, 16, 18, 24 and 32 bit interfaces. The serial interface may use four pins – LSCE#, LSDA, LSCK and LSA0 or three pins – LSCE#, LSDA, LSCK to enter commands and data. In 3 wire mode, an extra bit representing the LSA0 pin is transferred before the MSB of each transaction

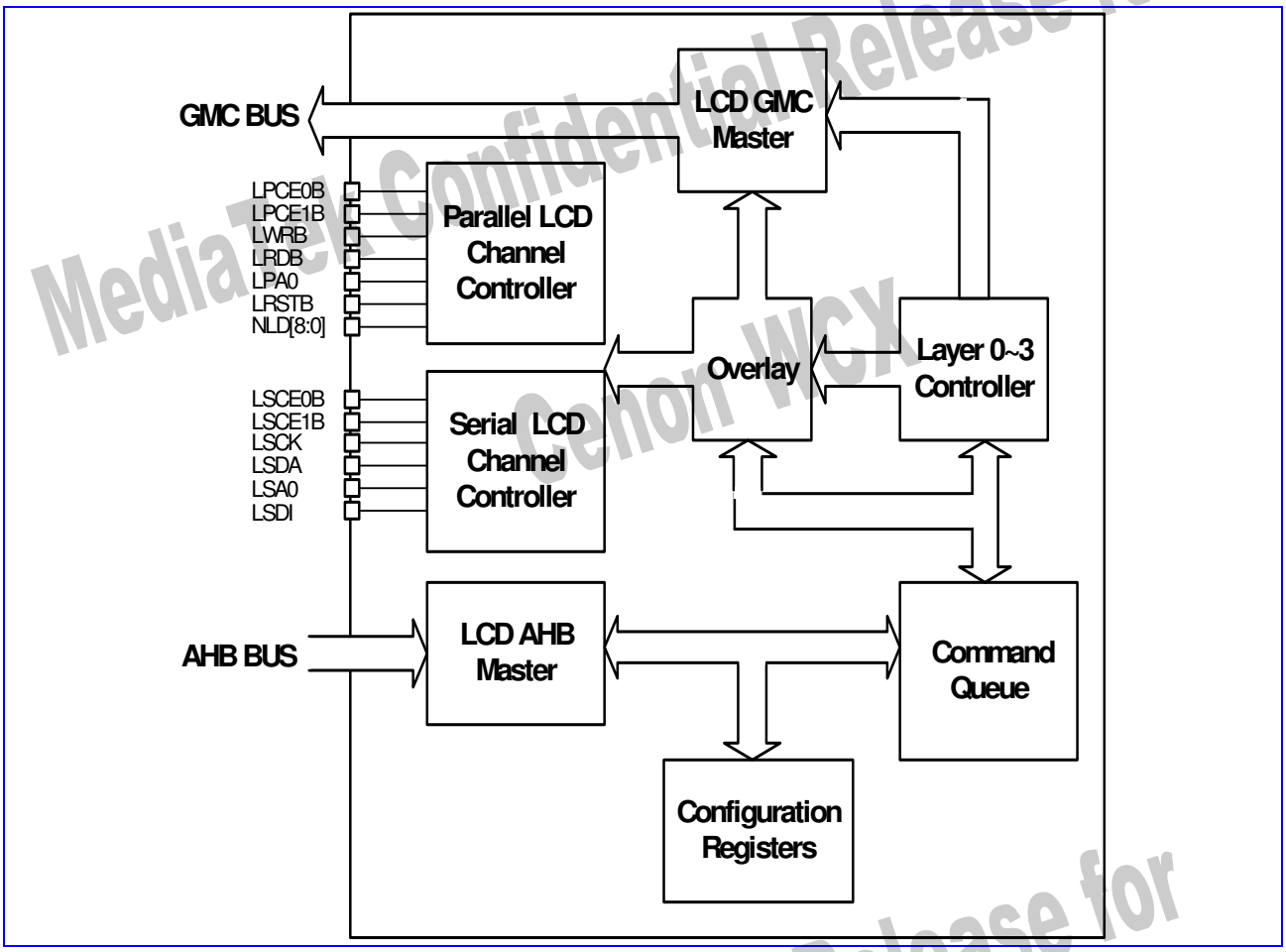



**Figure 42** LCD Interface Block Diagram

LCD = 0x9000_0000

| Address | Register Function | Width | Acronym |
|---|---|---|---|
| 9000_0000h | LCD Interface Status Register | 16 | **LCD_STA** |
| 9000_0004h | LCD Interface Interrupt Enable Register | 16 | **LCD_INTEN** |
| 9000_0008h | LCD Interface Interrupt Status Register | 16 | **LCD_INTSTA** |
| 9000_000Ch | LCD Interface Frame Transfer Register | 16 | **LCD_START** |
| 9000_0010h | LCD Parallel/Serial LCM Reset Register | 16 | **LCD_RSTB** |

| | | | |
|---|---|---|---|
| 9000_0020h | LCD Serial Interface Write Timing Register | 32 | **LCD_SCNF_WR** |
| 9000_0024h | LCD Serial Interface Read Timing Register | 32 | **LCD_SCNF_RD** |
| 9000_0028h | LCD Serial Interface Configuration Register | 32 | **LCD_SCNF** |
| 9000_002Ch | LCD Serial Interface Chip Select Register | 32 | **LCD_SCNF_CS** |
| 9000_0030h | LCD Parallel Interface 0 Configuration Register | 32 | **LCD_PCNF0** |
| 9000_0034h | LCD Parallel Interface 1 Configuration Register | 32 | **LCD_PCNF1** |
| 9000_003Ch | LCD Parallel Interface Size Configuration Register | 16 | **LCD_PDW** |
| 9000_0050h | LCD Tearing Control Register | 32 | **LCD_TECON** |
| 9000_0054h | LCD GMC Port Throttle Register | 32 | **LCD_GMCCON** |
| 9000_0060h | LCD ROI Window Write to Memory Address Register | 32 | **LCD_WROI_W2MADD** |
| 9000_0064h | LCD Calculation HTT Counter Register | 32 | **LCD_CALC_HTT** |
| 9000_0068h | LCD Sync LCM Size Register | 32 | **LCD_SYNC_LCM_SIZE** |
| 9000_006ch | LCD Sync Counter Register | 32 | **LCD_SYNC_CNT** |
| 9000_0070h | LCD ROI Window Write to Memory Pitch Register | 16 | **LCD_W2M_PITCH** |
| 9000_0074h | LCD ROI Window Write to Memory Offset Register | 32 | **LCD_WROI_W2MOFS** |
| 9000_0078h | LCD ROI Window Write to Memory Control Register | 16 | **LCD_WROI_W2MCON** |
| 9000_007Ch | LCD Palette Address Register | 32 | **LCD_PAL_ADD** |
| 9000_0080h | LCD ROI Window Control Register | 32 | **LCD_WROICON** |
| 9000_0084h | LCD ROI Window Offset Register | 32 | **LCD_WROIOFS** |
| 9000_0088h | LCD ROI Window Command Address Register | 16 | **LCD_WROICADD** |
| 9000_008ch | LCD ROI Window Data Address Register | 16 | **LCD_WROIDADD** |
| 9000_0090h | LCD ROI Window Size Register | 32 | **LCD_WROISIZE** |
| 9000_009Ch | LCD ROI Window Background Color Register | 32 | **LCD_WROI_BGCLR** |

| | | | |
|---|---|---|---|
| 9000_00A0h | LCD Write to LCM Continuous Count | 32 | **LCD_WROI_CONTI** |
| 9000_00B0h | LCD Layer 0 Window Control Register | 32 | **LCD_L0WINCON** |
| 9000_00B4h | LCD Layer 0 Color Key Register | 32 | **LCD_L0WINKEY** |
| 9000_00B8h | LCD Layer 0 Window Display Offset Register | 32 | **LCD_L0WINOFS** |
| 9000_00BCh | LCD Layer 0 Window Display Start Address Register | 32 | **LCD_L0WINADD** |
| 9000_00C0h | LCD Layer 0 Window Size | 32 | **LCD_L0WINSIZE** |
| 9000_00C4h | LCD Layer 0 Scroll Start Offset | 32 | **LCD_L0WINSCRL** |
| 9000_00C8h | LCD Layer 0 Memory Offset | 32 | **LCD_L0WINMOFS** |
| 9000_00CCh | LCD Layer 0 Memory Pitch | 16 | **LCD_L0WINPITCH** |
| 9000_00E0h | LCD Layer 1 Window Control Register | 32 | **LCD_L1WINCON** |
| 9000_00E4h | LCD Layer 1 Color Key Register | 32 | **LCD_L1WINKEY** |
| 9000_00E8h | LCD Layer 1 Window Display Offset Register | 32 | **LCD_L1WINOFS** |
| 9000_00ECh | LCD Layer 1 Window Display Start Address Register | 32 | **LCD_L1WINADD** |
| 9000_00F0h | LCD Layer 1 Window Size | 32 | **LCD_L1WINSIZE** |
| 9000_00F4h | LCD Layer 1 Scroll Start Offset | 32 | **LCD_L1WINSCRL** |
| 9000_00F8h | LCD Layer 1 Memory Offset | 32 | **LCD_L1WINMOFS** |
| 9000_00FCh | LCD Layer 1 Memory Pitch | 16 | **LCD_L1WINPITCH** |
| 9000_0110h | LCD Layer 2 Window Control Register | 32 | **LCD_L2WINCON** |
| 9000_0114h | LCD Layer 2 Color Key Register | 32 | **LCD_L2WINKEY** |
| 9000_0118h | LCD Layer 2 Window Display Offset Register | 32 | **LCD_L2WINOFS** |
| 9000_011Ch | LCD Layer 2 Window Display Start Address Register | 32 | **LCD_L2WINADD** |
| 9000_0120h | LCD Layer 2 Window Size | 32 | **LCD_L2WINSIZE** |
| 9000_0124h | LCD Layer 2 Scroll Start Offset | 32 | **LCD_L2WINSCRL** |
| 9000_0128h | LCD Layer 2 Memory Offset | 32 | **LCD_L2WINMOFS** |
| 9000_012Ch | LCD Layer 2 Memory Pitch | 16 | **LCD_L2WINPITCH** |
| 9000_0140h | LCD Layer 3 Window Control Register | 32 | **LCD_L3WINCON** |
| 9000_0144h | LCD Layer 3 Color Key Register | 32 | **LCD_L3WINKEY** |
| 9000_0148h | LCD Layer 3 Window Display Offset | 32 | **LCD_L3WINOFS** |

| | Register | | |
|---|---|---|---|
| 9000_014Ch | LCD Layer 3 Window Display Start Address Register | 32 | **LCD_L3WINADD** |
| 9000_0150h | LCD Layer 3 Window Size | 32 | **LCD_L3WINSIZE** |
| 9000_0154h | LCD Layer 3 Scroll Start Offset | 32 | **LCD_L3WINSCRL** |
| 9000_0158h | LCD Layer 3 Memory Offset | 32 | **LCD_L3WINMOFS** |
| 9000_015Ch | LCD Layer 3 Memory Pitch | 16 | **LCD_L3WINPITCH** |
| 9000_0170h | LCD Dither Control Register | 32 | **LCD_DITHER_CON** |
| 9000_01F0h | LCD Addcon Debug Register | 32 | **LCD_DB_ADDCON** |
| 9000_01F4h | LCD Maincon Debug Register | 32 | **LCD_DB_MCON** |
| 9000_01F8h | LCD W2mcon Debug Register | 32 | **LCD_DB_W2MCON** |
| 9000_01FCh | LCD Frame Counter Debug Register | 32 | **LCD_DB_COUNT** |
| 9000_4000h | LCD Parallel Interface 0 Command | 32 | **LCD_PCMD0** |
| 9000_4010h | LCD Parallel IF 0 Command Sync 0 | 32 | **LCD_ PCMD0_SYNC0** |
| 9000_4020h | LCD Parallel IF 0 Command Sync 1 | 32 | **LCD_ PCMD0_SYNC1** |
| 9000_4030h | LCD Parallel IF 0 Command HTT | 32 | **LCD_ PCMD0_HTT** |
| 9000_4100h | LCD Parallel Interface 0 Data | 32 | **LCD_PDAT0** |
| 9000_4110h | LCD Parallel IF 0 Data Sync 0 | 32 | **LCD_ PDAT0_SYNC0** |
| 9000_4120h | LCD Parallel IF 0 Data Sync 1 | 32 | **LCD_ PDAT0_SYNC1** |
| 9000_4130h | LCD Parallel IF 0 Data HTT | 32 | **LCD_ PDAT0_HTT** |
| 9000_5000h | LCD Parallel Interface 1 Command | 32 | **LCD_PCMD1** |
| 9000_5010h | LCD Parallel IF 1 Command Sync 0 | 32 | **LCD_ PCMD1_SYNC0** |
| 9000_5020h | LCD Parallel IF 1 Command Sync 1 | 32 | **LCD_ PCMD1_SYNC1** |
| 9000_5030h | LCD Parallel IF 1 Command HTT | 32 | **LCD_ PCMD1_HTT** |
| 9000_5100h | LCD Parallel Interface 1 Data | 32 | **LCD_PDAT1** |
| 9000_5110h | LCD Parallel IF 1 Data Sync 0 | 32 | **LCD_ PDAT1_SYNC0** |
| 9000_5120h | LCD Parallel IF 1 Data Sync 1 | 32 | **LCD_ PDAT1_SYNC1** |
| 9000_5130h | LCD Parallel IF 1 Data HTT | 32 | **LCD_ PDAT1_HTT** |
| 9000_8000h | LCD Serial Interface Command | 32 | **LCD_SCMD** |
| 9000_8010h | LCD Serial IF Command Sync 0 | 32 | **LCD_ SCMD _SYNC0** |
| 9000_8020h | LCD Serial IF Command Sync 1 | 32 | **LCD_ SCMD _SYNC1** |
| 9000_8030h | LCD Serial IF Command HTT | 32 | **LCD_ SCMD _HTT** |

| | | | |
|---|---|---|---|
| 9000_8100h | LCD Serial Interface Data | 32 | **LCD_SDAT** |
| 9000_8110h | LCD Serial IF Data Sync 0 | 32 | **LCD_ SDAT _SYNC0** |
| 9000_8120h | LCD Serial IF Data Sync 1 | 32 | **LCD_ SDAT _SYNC1** |
| 9000_8130h | LCD Serial IF Data HTT | 32 | **LCD_ SDAT _HTT** |
| 9000_C000h ~ 9000_C07Fh | LCD Interface Command/Parameter Register | 32 | **LCD_COMD** |

**Table 57** Memory map of LCD Interface



## 5.1.2 Register Definitions

The base address of LCD is 0x9000_0000. Note: All reserved fields in LCD registers should be set to 0.

### 9000_0000h LCD Interface Status Register LCD_STA

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | BUSY | WAIT_SYNC | WAIT_HTT | GMC | WAIT_CMDQ | RUN |
| Type | | | | | | | | | | | R | R | R | R | R | R |
| Reset | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 |

**RUN** LCD Interface start transfer commands and/or pixels.
**WAIT_CMDQ** LCD is waiting for command queue transfer to complete.
**GMC** LCD is processing a GMC request
**WAIT_HTT** LCD is waiting LCD_CALC_HTT.COUNT to reach LCD_CALC_HTT.TIMEOUT.
**WAIT_SYNC** LCD is waiting for LCM tearing-free sync signal or is counting the set TE delay.
**BUSY** LCD Interface is busy. If read as 1, this means the LCD may be in the process of waiting for a hardware trigger signal, waiting for tearing signal, sending commands to command queue, or writing pixels to memory/LCM.

### 9000_0004h LCD Interface Interrupt Enable Register LCD_INTEN

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

| Name | | | | | | | | | | | SYNC | HTT | | CMDQ_CPL | REG_CPL | CPL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | | | | | | | | | 0 | 0 | | 0 | 0 | 0 |

LCD interrupt is at interrupt number 16.

**CPL** Enable the interrupt when LCD frame transfer completes.

**REG_CPL** Enable the interrupt when LCD has entered transfer command/pixel state.

**CMDQ_CPL** Enable the interrupt when LCD command transfer completes.

**HTT** Enable the interrupt when LCD_CALC_HTT.COUNT reaches LCD_CALC_HTT.TIMEOUT.

**SYNC** Enable the interrupt when LCM tearing-free sync signal arrives and LCD has finished the set TE delay.

## 9000_0008h LCD Interface Interrupt Status Register LCD_INTSTA

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | SYNC | HTT | | CMDQ_CPL | REG_CPL | CPL |
| Type | | | | | | | | | | | R | R | | R | R | R |
| Reset | | | | | | | | | | | 0 | 0 | | 0 | 0 | 0 |

**CPL** Interrupt of LCD frame transfer completion occurs.

**REG_CPL** Interrupt when LCD has entered transfer command/pixel state.

**CMDQ_CPL** Interrupt when LCD command transfer completes.

**HTT** Interrupt when LCD_CALC_HTT.COUNT reaches LCD_CALC_HTT.TIMEOUT.

**SYNC** Interrupt of the arrival of LCM tearing-free sync signal occurs and LCD has finished the set TE delay

**LCD Operating States**

Below is a state diagram detailing the various states of the LCD controller. State transitions depend on the current hardware trigger and tearing settings. Please consult the table below for detailed explanations.

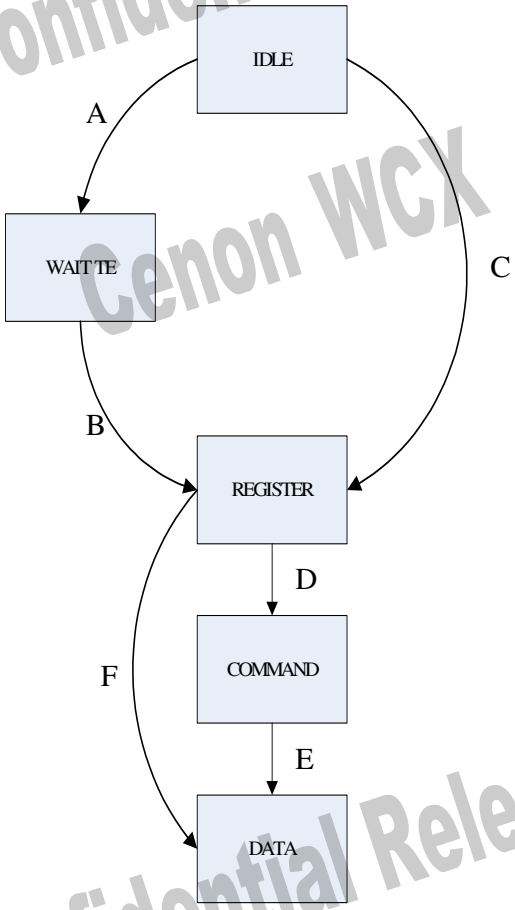



Figure 43 **LCD State Transitions**

| State | Action | Exit State | Exit Condition | IRQ | LCD_STA |
|---|---|---|---|---|---|
| IDLE | LCD is idle | If te_en = 1, then A;<br>Else C; | LCD_START.START has been changed to 0 from 1. | No IRQ | 0x00 |

| | | | | | |
|---|---|---|---|---|---|
| WAIT TE | If sync_mode = 0, then LCD is waiting for a tearing edge;<br><br>If sync_mode = 1, then LCD is waiting for SW to read the scanline port. | Always B | If sync_mode = 0, then LCD must receive a tearing edge and must wait for a period of time;<br><br>If sync _mode = 1, then SW must read the X_SYNC1 scanline port | SYNC | 0x30 |
| REGISTER | LCD is setting up register values for use | If command queue is enabled then D;<br>Else F; | LCD will transition in 1T | REG_CPL | 0x20 |
| COMMAND | LCD is transferring command queue data to the LCM | Always E | After LCD has finished transferring all command queue data. | CMDQ_CPL | 0x27 |
| DATA | LCD is transferring ROI data to the LCM | If te_repeat = 1, then WAIT_TE;<br>Else IDLE; | After LCD has finished transferring all ROI data to the LCM. | CPL | 0x25 |

## 9000_000Ch LCD Interface Frame Transfer Register LCD_START

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

**START** Start Control of LCD Frame Transfer. After LCD is started, please to not write to any other register because LCD does not have double registers. Only the SW_TE bit may be changed

**INT_RESET** Internal reset

**0** Not reset.

**1** Reset.

## 9000_0010h LCD Parallel/Serial Interface Reset Register LCD_RSTB

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

**RSTB** Parallel/Serial LCD Module Reset Control. It directly control the LRSTB pin.

## 9000_0020h LCD Serial Interface Write Timing Configuration Register LCD_SCNF_WR

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

**SCL_HIGH_COUNT** Number of clocks plus one to hold the SCL high for every bit transferred during a write cycle.
**SCL_LOW_COUNT** Number of clocks plus one to hold the SCL low for every bit transferred during a write cycle.

## 9000_0024h LCD Serial Interface Read Timing Configuration Register LCD_SCNF_RD

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

**SCL_HIGH_COUNT** Number of clocks plus one to hold the SCL high for every bit transferred during a read cycle
**SCL_LOW_COUNT** Number of clocks plus one to hold the SCL low for every bit transferred during a read cycle

## 9000_0028h LCD Serial Interface Configuration Register LCD_SCNF

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

**3-WIRE** Enable 3 wire mode. Serial interface will transfer an A0 bit before transferring the MSB of each transaction.

**SDI** Set to 1 to read data from LSDI pin, otherwise LCD will use the bi-directional LSDA pin.

**IF_SIZE** Interface size of Serial Interface. Each transaction will transmit this many bits

**000** 8 bits
**001** 9 bits
**010** 16 bits
**011** 18 bits
**100** 24 bits
**101** 32 bits

## 9000_002Ch LCD Serial Interface Chip Select Configuration Register LCD_SCNF_CS

| 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 | | | | | | | | | | | | | | | CS1 | CS0 |
| Type | | | | | | | | | | | | | | | R/W | R/W |
| Reset | | | | | | | | | | | | | | | 1 | 1 |

**CS0** Set Chip Select 0 level. This bit directly controls the LSCE0 pin
**CS1** Set Chip Select 1 level. This bit directly controls the LSCE1 pin

The serial interface can support 4 wire mode or 3 wire mode as shown below. In 3 wire mode, the A0 bit is transferred as part of the data before the MSB. The write and read timing registers can be used to set the timing of the interface. In the diagram below, X = SCL_LOW_COUNT+1 clock cycles and Y = SCL_HIGH_COUNT+1 clock cycles. When a read needs to be done, a write should be sent to the serial interface. Then (without changing CS) a read is performed by reading the serial interface. Notice the write and read may use different timing. The CS pin is controlled by software by writing to the LCD_SCNF_CS register. The LCD 104Mhz clock MUST BE ENABLED to use the Serial Interface.

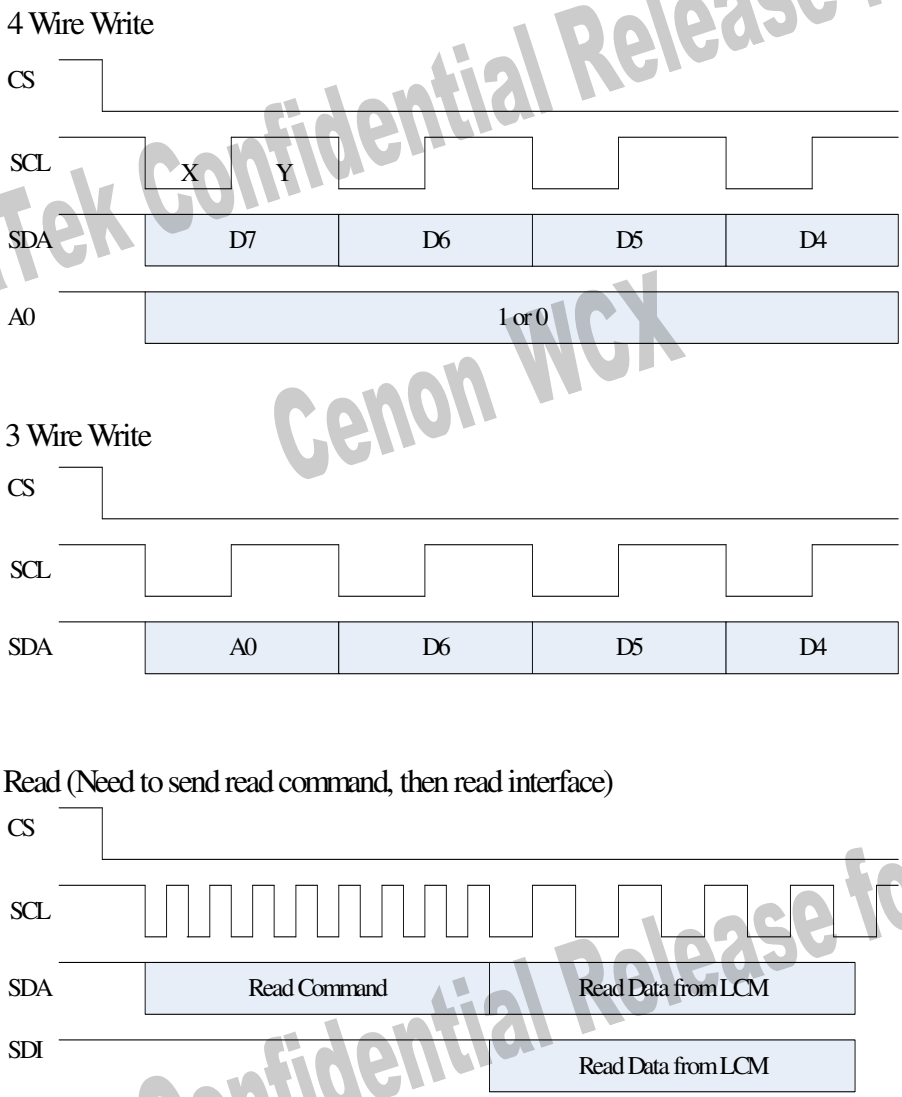



**Figure 44** Serial interface timing diagram

MediaTek Confidential Release for Cenon WCX

## 9000_0030h LCD Parallel Interface Configuration Register 0 LCD_PCNF0

| 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 | R/W | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | | | 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 | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**WST** Write Wait State Time, see Figure 45.
**C2WS** Chip Select (LPCEB0) to Write Strobe (LWRB) Setup Time, see Figure 45. **C2WS must <= WST.**
**C2WH** Chip Select (LPCEB0) to Write Strobe (LWRB) Hold Time, see Figure 45.
**RLT** Read Latency Time. See Figure 46.
**C2RS** Chip Select (LPCEB0) to Read Strobe (LRDB) Setup Time, see Figure 46. **C2RS must <= RLT.**
**C2RH** Chip Select (LPCEB0) to Read Strobe (LRDB) Hold Time, see Figure 46.

## 9000_0034h LCD Parallel Interface Configuration Register 1 LCD_PCNF1

| 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 | R/W | R/W |
| 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 | | | 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 | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**WST** Write Wait State Time, see Figure 45.
**C2WS** Chip Select (LPCEB1) to Write Strobe (LWRB) Setup Time, see Figure 45. **C2WS must <= WST.**
**C2WH** Chip Select (LPCEB1) to Write Strobe (LWRB) Hold Time, see Figure 45.
**RLT** Read Latency Time. See Figure 46.
**C2RS** Chip Select (LPCEB1) to Read Strobe (LRDB) Setup Time, see Figure 46. **C2RS must <= RLT.**
**C2RH** Chip Select (LPCEB1) to Read Strobe (LRDB) Hold Time, see Figure 46.

Parallel Interface Write timing
C2WS=2, WST=3, C2WH=0, LCD_WROICON.PERIOD=0, **C2WS must <= WST**

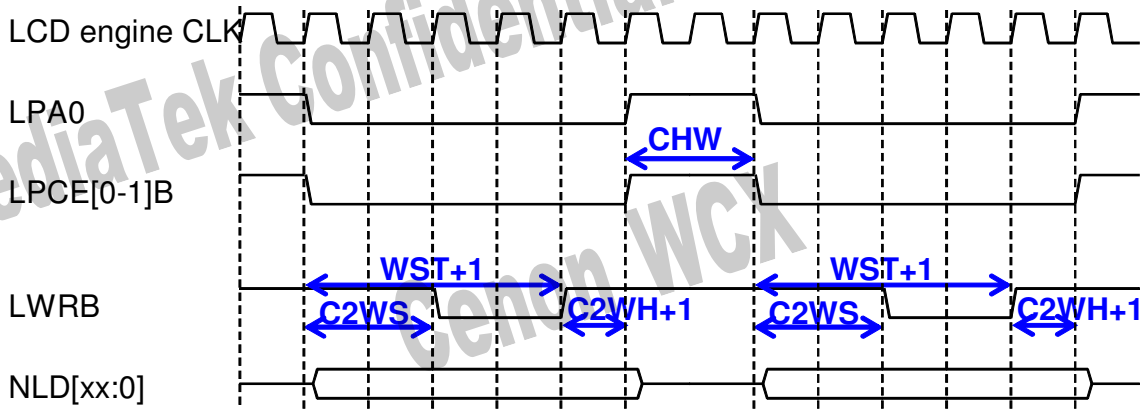
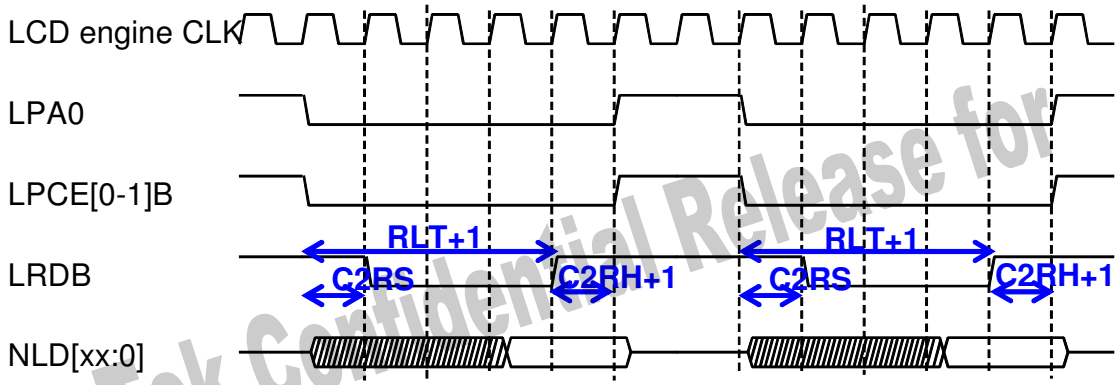



**Figure 45** Parallel interface write timing diagram

The time between transactions is CHW = (PERIOD>C2WH) ? PERIOD+1-C2WH : 1.

Parallel Inteface Read timing
C2RS=1, RLT=3, C2RH=0, LCD_WROICON.PERIOD=0, **C2RS must <= RLT**




NLD is sampled at the rising edge of LRDB.

**Figure 46** Parallel interface read timing diagram

## 9000_003Ch LCD Parallel Interface Data Width Configuration Register — LCD_PDW

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | P1_DW | | | | P0_DW | | |
| Type | | | | | | | | | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | | | | | | | | 0 | 0 | 0 | | 0 | 0 | 0 |

**P0_DW** Data width of parallel interface 0
- **000** 8 bit
- **001** 9 bit
- **010** 16 bit
- **others** Reserved

**P1_DW** Data width of parallel interface 1
- **000** 8 bit
- **001** 9 bit
- **010** 16 bit
- **others** Reserved

### LCD GPIO Table

| IO Name | Aux Function | LCD Function |
|---|---|---|
| KCOL7 | 4 | LSCK |
| KROW6 | 4 | LSDA |
| KROW3 | 4 | LSA0 |
| KROW2 | 4 | LSCK |
| KROW1 | 4 | LSDA |
| KROW0 | 4 | LSDI |
| SD_PWREN | 3 | LSCE1B |
| URXD1 | 2 | LSCK |
| UTXD1 | 2 | LSDA |

| | | |
|---|---|---|
| UCTS1_B | 3 | NLD13 |
| URTS1_B | 3 | NLD14 |
| MCINS | 3 | NLD15 |
| EA24 | 3 | LPCE0B |
| EA23 | 3 | LSCE0B |
| EA15 | 1 | NLD12 |
| EA14 | 1 | NLD12 |
| EA13 | 1 | NLD10 |
| EA12 | 1 | NLD9 |
| EA11 | 1 | NLD8 |
| EA10 | 1 | NLD7 |
| EA9 | 1 | NLD6 |
| EA8 | 1 | NLD5 |
| EA7 | 1 | NLD4 |
| EA6 | 1 | NLD3 |
| EA5 | 1 | NLD2 |
| EA4 | 1 | NLD1 |
| EA3 | 1 | NLD0 |
| EA2 | 1 | LWR_B |
| EA1 | 1 | LRD_B |
| EA0 | 1 | LPA0 |
| EA6 | 2 | LSDI |

| | | |
|---|---|---|
| EA5 | 2 | LSDA |
| EA4 | 2 | LSA0 |
| EA3 | 2 | LSCK |
| ECS3_B | 2 | LPCE1_B |
| ECS3_B | 3 | LSCE1_B |
| ECS2_B | 1 | CORE_ECS2_B |
| ECS2_B | 2 | LPCE0_B |
| ECS2_B | 3 | LSCE0_B |
| ED7 | 3 | LPCE1_B |
| ED6 | 3 | LSCE1_B |
| LPTE | 1 | LPTE |
| LRSTB | 1 | LRSTB |
| CMDAT1 | 2 | LSDA |
| CMPDN | 2 | LSCK |
| SCK | 2 | LSCK |
| SWP | 2 | LSA0 |
| SHOLD | 2 | LSCE0_B |
| SCS | 2 | LSCE1_B |
| SIN | 2 | LSDI |
| SOUT | 2 | LSDA |



## TE Signal Polarity

TE_EDGE_SEL can be used to select TE polarity for TE signal detection.

| TE_EDGE_SEL value | TE signal detection |
|---|---|
| 0 | Detect a TE signal at its rising edge. This setting is for active high TE signal. |
| 1 | Detect a TE signal at its falling edge. This setting if for active low TE signal. |

**Table 58** TE Signal Polarity

## Sync Mode 0

In sync mode = 0, LCD will update TE after counting a set number of horizontal sync lines. Each horizontal sync line is set by LCD_SYNC_LCM_SIZE.HTT. After receive a TE edge, LCD will count LCD_SYNC_CNT.WAITLINE number of lines and then begin updating the new frame to the LCM. To use this mode, please follow these steps:

1. Set LCD_TECON.SYNC_MODE = 0
2. Set LCD_SYNC_LCM_SIZE.HTT to the correct size. See below for more information on this.
3. Set LCD_SYNC_CNT.WAITLINE to the number of lines you wish to wait before updating the LCM.
4. Set other registers (ROI, Layer, etc.) and start the LCD controller by setting LCD_START.START = 1 (from 0).

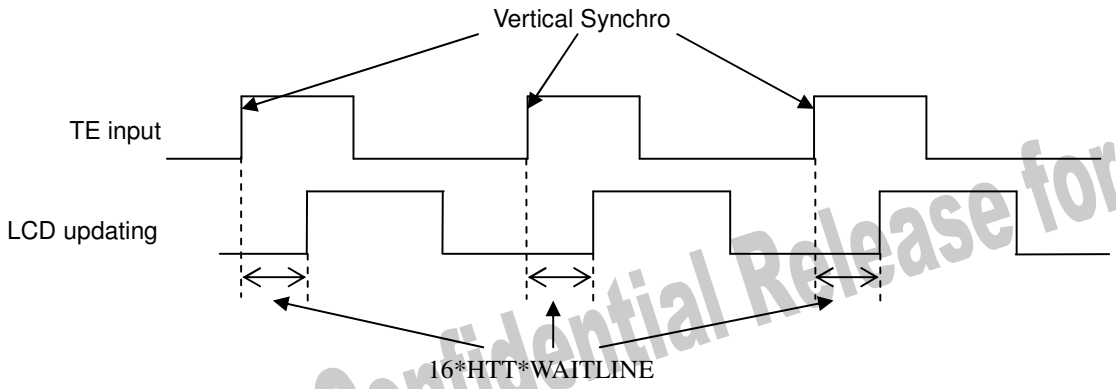



**Figure 47** SYNC_MODE = 0

## Sync Mode 1

In sync mode 1, LCD will not use the TE pin to detect the LCM scan line position. Instead, software must read the LCM scan line from the LCM register. When software reads a specified port, LCD will interpret this and automatically

begin its internal TE counter at the read LCM scan line position. Scan line 0 indicates the beginning of VSYNC as shown in the figure below. The LCD ROI begins during the V active region (vact).

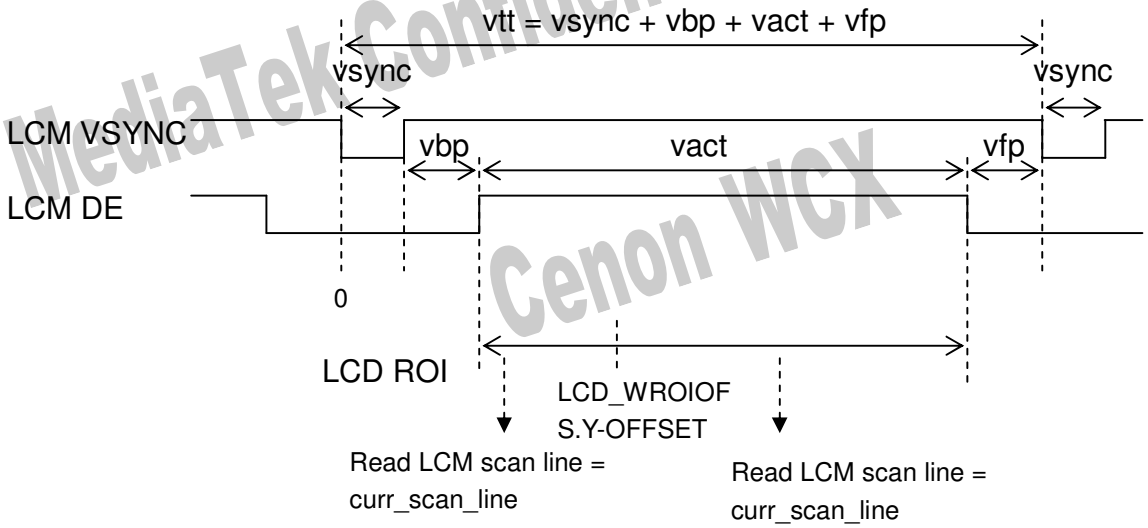



Figure 48 **LCM Scan Line Timing**

Typically, the scan line register is divided into 2 parameters and 1 dummy read. To read the current LCM scan line, software uses the following steps: (assume the LCM is on Parallel CS0)

1. Set LCD_SYNC_LCM.VTT size to the number of LCM scan lines including blanking.
2. Set LCD_SYNC_LCM.HTT to the correct timing parameter. See below for more information on this.
3. Set LCD_SYNC_CNT.WAITLINE to the LCM scan line you wish to start updating the frame.
4. Start the LCD by setting LCD_START.START = 1 (from 0).
5. Write Scan line command to port 0x4000 or 0x4100.
6. If the LCM needs a dummy read, then read port 0x4000 or 0x4100. This step can be skipped if no dummy read is required.
7. Read port 0x4010 or 0x4110 to latch the first parameter into the LCM internal counter.

8. Read port 0x4020 or 0x4120 to latch the second parameter into the LCM internal counter and begin the TE counter. SW must use an 8 bit read for this parameter or else the top byte will be covered. If the IF size is greater then 8/9 bits and there is only 1 parameter to read, the SW may skip step 6 and only use step 7. In this case, SW may use a 16 or 32 bit read to this port.
9. If you need to read another port then substitute the port number for the "4" in 0x4XXX. All ports can support this feature except the LCM on EMI port (0x6XXX). The "1" in 0x4100 indicates the A0 bit will be 1 when the read/write transaction is issued. Please confirm with the LCM datasheet to determine which port is appropriate.

In the figure below, the LCM has 240 total horizontal lines including blanking. Assume we want LCD to begin updating at Point A because the partial update begins at this point. In this case, we should set VTT = 240 and WAITLINE = 3. When SW takes steps 6 and 7 above, assume the returned value is Point B. This means the TE internal counter will count up to Line 239 and loop back to 0. The counter will count until Point A is reached and then begin updating the LCM. Note that Line 0 is typically not within the active LCM region.

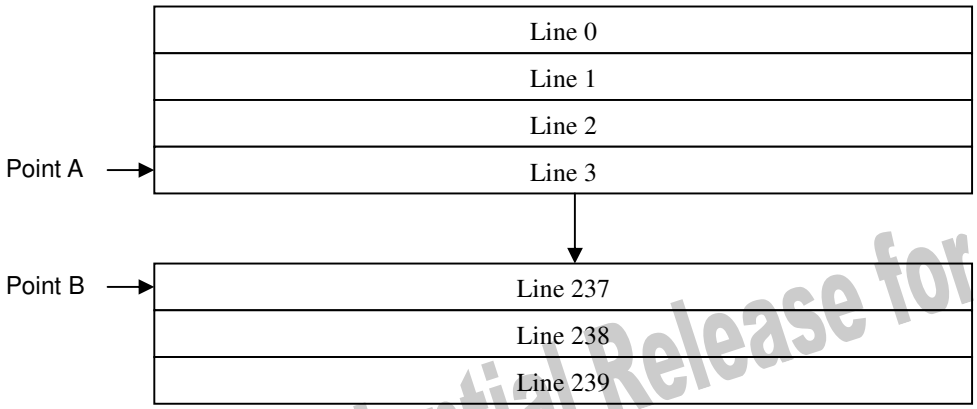



Figure 49 **TE Scan Line Example**

| Name | Function |
|---|---|
| X_SYNC0 | Latches the first parameter of the LCM scan line into the TE counter. |
| X_SYNC1 | Latches the second parameter of the LCM scan line into the TE counter and begin the |

| | |
|---|---|
| | counter. |
| X_HTT | Read once to begin HTT calculation. Read again to stop the calculation. This can only be used when LCD is idle. |

Table 59 **LCD TE Ports**

## HTT Calibration

The HTT parameter can be calculated from the LCM datasheet. However, if SW wants a more automatic method to calculate HTT, then SW can use the LCD TE counters. The steps are as follows:

1. Make sure LCD is in the IDLE state (LCD_START.START = 0 and LCD_STA = 0).
2. Set HTT = 256.
3. Set LCD_CALC_HTT.TIMEOUT to 128.
4. Enable the HTT calculation interrupt LCD_INTEN.HTT
5. To start the calculation, read the port 0x4030 or 0x4130 (depending on desired A0, assuming Parallel CS0). Note this will immediately start the calculation, so if there are many parameters to read, this port must be read last.
6. LCD will begin counting cycles. When LCD_CALC_HTT.COUNT reaches TIMEOUT, LCD will issue an interrupt. Software should read port 0x4030 or 0x4130 to stop the TE counters. Again, this port should be read last if there are many parameters to read.
7. Assume the first scan line read is SE0 and the second is SE1. Then the correct HTT value is:

   HTT = (COUNT*256) / (SE1 – SE0)

## 9000_0050h LCD Tearing Control Register LCD_TECON

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | | | |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | SW_TE | | | | | | | | | | | | TE_REPEAT | SYNC_MODE | TE_EDGE_SEL | SYNC_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 | 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 |

**SYNC_EN** Enable sync control. LCD controller will synchronize to LCM refresh timing.

**TE_EDGE_SEL** Select TE edge.

- **0** Rising edge
- **1** Falling edge

**SYNC_MODE** Sync control mode.

- **0** TE signaling mode. LCD starts to update LCM after receiving TE edge and plus a delay specified by LCD_working_clock_cycle_time*16*LCD_SYNC_CNT.HTT*LCD_SYNC_CNT.LINES in unit of ns. LCD_working_clock_cycle_time is 19.2ns.
- **1** Read LCM current scan line mode. LCD starts to update LCM after LCD_START.START is set to 1 from 0 and plus a delay specified by LCD_working_clock_cycle_time*16*LCD_SYNC_CNT.HTT*LCD_SYNC_CNT.LINES in unit of ns. LCD_working_clock_cycle_time is 19.2ns.

**TE_REPEAT** Repeat mode.

- **0** update LCM once every TE signal coming.
- **1** repeat updating LCM after TE signal coming.

**SW_TE** Software emulated TE signal. Write this bit from 0 to 1 will let LCD act like a TE signal received.

## 9000_0054h LCD GMC Port Control Register LCD_GMCCON

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

**MAX_BURST** Specify the maximum burst length of GMC request.

- **000** Single 4 bytes access
- **010** Burst 4 beats access and one beat is 4 bytes. Total data amount is 16 bytes per access.
- **011** Burst 8 beats access and one beat is 4 bytes. Total data amount is 32 bytes per access.
- **Others** Burst 4 beats access and one beat is 4 bytes. Total data amount is 16 bytes per access.

**THROTTLE_EN** Enable GMC port throttling.

**THROTTLE_PERIOD** Throttle down LCD GMC port access speed. It specifies how many AHB bus cycles between two GMC requests. There is a down counter, throttle_cnt, to count down for the interval between two

GMC requests. It counts from THROTTLE_PERIOD when lcd_greq rising to zero, and the next GMC request is allowed to issue when throttle_cnt=0. Please see Figure 50. The maximum GMC bandwidth is limited to

$$\frac{bytes\_per\_GMC\_access \times 1000}{throttle\_period \times AHB\_cycle\_time(ns)} = \frac{bytes\_per\_GMC\_access \times 1000}{throttle\_period \times 19.2(ns)} MBytes/sec$$

Bytes_per_GMC_access is specified by MAX_BURST.

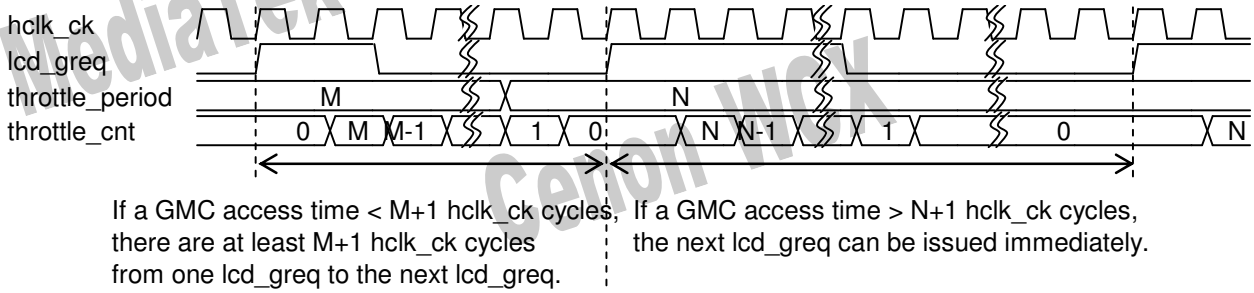



Figure 50 **GMC throttle mechanism**

## 9000_0060h Region of Interest Window Write to Memory Address FB Register LCD_WROI_W2MADD

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | W2M_ADDR | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | W2M_ADDR | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |

**W2M_ADDR** Write to memory address (byte address) for Frame Buffer, please see Figure 52. The address must be aligned according to the table below. Note: If the buffer is Page-size byte aligned, LCD will benefit from a ~2-3% bandwidth increase. For example, if the page size of the memory is 16bytes, then a 16byte alignment address will increase bandwidth.

| LCD_WROI_W2MCON.W2MFORMAT | Color format | ADDR alignment |
|---|---|---|
| 00 | RGB565 | 2 bytes alignment |
| 01 | RGB888 | no alignment constraint |

| | | |
|---|---|---|
| 10 | Constant A + RGB | 4 bytes alignment |
| 11 | Pixel A + RGB | 4 bytes alignment |

## 9000_0064h LCD Calculation HTT Counter Register

LCD_CALC_HTT

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | COUNT | | | | | | | | | | | |
| Type | R/W | R/W | R/W | R/W | 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 | | | | | TIMEOUT | | | | | | | | | | | |
| 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 | 1 |

**COUNT** COUNT will begin counting when the port X_HTT is read. Reading this port indicates software want to calibrate for HTT. Reading this port again will stop COUNT.

**TIMEOUT** Specify the time interval from timeout counter starting to timeout interrupt issuing. When COUNT reaches this number, LCD will issue an interrupt. This is only used for HTT calculation. TIMEOUT must be >0.

## 900A_0068h LCD Sync LCM Size Register

LCD_SYNC_LCM_SIZE

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

**VTT** The total number of horizontal lines on the LCM including blanking lines. VTT must be >0

**HTT** Indicates how long a LCM horizontal sync is in units of 16*T, where T is the cycle time. Cycle time is 19.2 ns. HTT must be >0.

## 9000_006Ch LCD Sync Counter Register LCD_SYNC_CNT

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | SCANLINE | | | | | | | | | | | |
| Type | R/W | R/W | R/W | R/W | 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 | | | | | WAITLINE | | | | | | | | | | | |
| 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 | 1 |

**SCANLINE** Indicates the current TE counter value.

**WAITLINE** SCANLINE will count until it reaches this value, a TE interrupt will be issued (if enabled) and LCD will begin updating the new frame. In TE SYNC MODE 0, this value is the number of lines to delay LCD update. In TE SYNC MODE1, this value is the scan line SW wishes LCD to update the frame. WAITLINE must be >0.

## 9000_0070h Region of Interest Window Write to Memory Pitch Register LCD_W2M_PITCH

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

**PITCH** Write to memory pitch in unit of byte, please see Figure 52. The pitch divided by the write to memory bpp must be equal or greater then the ROI width. If the write to memory bpp is 4, then the pitch must be divisible by 4. If the write to memory bpp is 2, then the pitch must be divisible by 2. If the write to memory bpp is 3 (RGB888), then the pitch may be any number greater then the ROI width * 3.

## 9000_0074h Region of Interest Window Write to Memory Offset Register LCD_WROI_W2MOFS

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

| Type | | | | | | R/W |
|---|---|---|---|---|---|---|

This control register is used to specify the offset of the ROI window from the LCD_WROI_W2MADDR when writing the ROI window's content to memory.

**X-OFFSET** the x offset of ROI window in the destination memory, please see Figure 52.

**Y-OFFSET** the y offset of ROI window in the destination memory, please see Figure 52.

## 9000_0078h Region of Interest Window Write to Memory Control Register LCD_WROI_W2MCON

| 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 | OUTPUT_ALPHA | | | | | | | | EMI2LCM_BANK | | WR_ULTR | ADDINC_DISABLE | RD_ULTRA | W2M_FORMAT | | W2LCM |
| Type | R/W | | | | | | | | R/W | | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0xff | | | | | | | | 0 | | 0 | 0 | 0 | 0 | | 0 |

This control register is effective only when the W2M bit is set in LCD_WROICON register.

**W2LCM** Write to LCM simultaneously. It is effective only when LCD_WROICON.W2M=1.

- **0** Not output to LCM, only write to memory.
- **1** Output to LCM and write to memory.

LCD_WROICON.W2M, LCD_WROI_W2MCON.W2LCM possible combinations

0x only output LCM
10 only output memory
11 output LCM and memory

**W2M_FORMAT** Write to memory format.

- **00** RGB565
- **01** RGB888
- **10** ARGB8888, alpha value is a constant specified by OUTPUT_ALPHA

**11** ARGB8888, alpha value is calculated by blending equation. When using this mode, all layers enabled must have color depth ARGB8888/PARGB8888

**ADDINC_DISABLE** Disable address increasing when writing to memory. Always write to the same address.
**RD_ULTRA** Enable LCD issue ultra GMC request for layer frame buffer read.
**WR_ULTRA** Enable LCD issue ultra GMC request for LCM on EMI write only. LCD does not issue ultra GMC request for W2M requests
**EMI2LCM_BANK** Choose the bank number that the LCM is connected to on EMI. Only used for LCM on EMI interface. Value 2'b00 represents CS0, 2'b01 is CS1 and so on. The LCM is typically placed on CS2 (2'b10). When using this interface, the interface size is set at LCD_WROICON.

.

When the LCM is placed on the EMI bus, software must access EMI to direct access LCM (not 0x9000_6000). The access location is as follows:

| Address | EMI Bank | Action |
|---|---|---|
| 0x0000_0000 | Bank 0 | Access Bank 0 with A0 = 0 |
| 0x0000_0002 | Bank 0 | Access Bank 0 with A0 = 1 |
| 0x0800_0000 | Bank 1 | Access Bank 1 with A0 = 0 |
| 0x0800_0002 | Bank 1 | Access Bank 1 with A0 = 1 |
| 0x1000_0000 | Bank 2 | Access Bank 2 with A0 = 0 |
| 0x1000_0002 | Bank 2 | Access Bank 2 with A0 = 1 |
| 0x1800_0000 | Bank 3 | Access Bank 3 with A0 = 0 |
| 0x1800_0002 | Bank 3 | Access Bank 3 with A0 = 1 |

## 9000_007Ch LCD Palette Address Register LCD_PAL_ADD

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

**PAL_ADDR** Address of palette placed in memory used for 8bpp indexed color . The palette should have 256 RGB565 entries and the address should be 2 byte aligned.

## 9000_0080h Region of Interest Window Control Register — LCD_WROICON

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | EN0 | EN1 | EN2 | EN3 | | FCNT | | SEND_RES_MOD | PERIOD | | | | | | | |
| 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 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | ENC | W2M | | COMMAND | | | | | FORMAT | | | | | | | |
| Type | R/W | R/W | | R/W | | | | | R/W | | | | | | | |

**FORMAT** LCD Module Data Format, see Table 60.

Bit 0 : Sequence

**0** BGR sequence.

**1** RGB sequence.

Bit 1 : Significance.

Bit 2 : Padding.

**0** Padding bits on LSBs.

**1** Padding bits on MSBs.

Bit 5-3 : Color format

000 for RGB332, 001 for RGB444, 010 for RGB565, 011 for RGB666, 100 for RGB888.

Bit 7-6 : Interface width

00 for 8-bit interface, 01 for 16-bit interface, 10 for 9-bit interface, 11 for 18-bit interface.

**COMMAND** Number of Commands to be sent to LCD module. Maximum is 31.

**W2M** Enable Write to Memory. ROI column and/or row size cannot be 0 if write to memory is enabled.

**ENC** Command Transfer Enable Control.

**0** Only send pixel data to LCM, not send commands in command queue.

**1** Send commands in command queue first, and then send pixel data to LCM. The number of commands to be sent is specified by COMMAND, and the command queue to be sent is specified by COM_SEL.

**PERIOD** Waiting period between two consecutive transfers, effective for both data and command.

**SEND_RES_MOD** Send the residue odd pixel method. When the output throughput is 2pixels/3cycles, if ROI width is odd, the last pixel of a line is not in any two-pixel-pair; If ROI height is also odd, the last pixel of a frame is not in any two-pixel-pair. This field is used to select the method to transmit the residue odd pixel of a line or of a frame.

**0** Send the residue odd pixel per line. In this mode, the last pixel of a line is combined with an extra byte, and sent to LCM. LCD driver should not care this extra byte.

For example, if the ROI's width x height is 3 x 2. The output sequence is

$R_0G_0$ --- pixel 0 of line 0

$B_0R_1$

$G_1B_1$

$R_2G_2$

$B_2R_1$ --- LCD driver should not care this $R_1$.

$R_0G_0$ --- pixel 0 of line 1

$B_0R_1$

$G_1B_1$

$R_2G_2$

$B_2R_1$ --- LCD driver should not care this $R_1$.

**1** Send the residue odd pixel per frame. In this mode, the last pixel of a line is combined with the first pixel of the next line as a two-pixel-pair, and is sent to LCM.

For example, if the ROI's width x height is 3 x 3. The output sequence is

$R_0G_0$ --- pixel 0 of line 0

$B_0R_1$

$G_1B_1$

$R_2G_2$

$B_2R_0$ --- pixel 0 of line 1

$G_0B_0$

$R_1G_1$

$B_1R_2$

$G_2B_2$

$R_0G_0$ --- pixel 0 of line 2

$B_0R_1$

$G_1B_1$

$R_2G_2$

$B_2R_1$ --- LCD driver should not care this $R_1$.

**FCNT** Enable frame update counter. Counter starts when LCD leaves the IDLE state and stops when it returns to IDLE.

**ENn** Layer Window Enable Control.

Note : If there are two pixels in one row, Blue color indicates the prior pixel in scan direction, and black color means the later pixel. The scan direction is from left-top to right-bottom, horizontal scan first. The skin color means the dummy bit of 9 bit interface.

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| RGB332 | 8 | x | x | 0 | 1pixel/1cycle | $R_2R_1R_0G_2G_1G_0B_1B_0$ |
| | | x | x | 1 | 1pixel/1cycle | $B_1B_0G_2G_1G_0R_2R_1R_0$ |
| RGB332 | 9 | x | x | 0 | 1pixel/1cycle | $B_0R_2R_1R_0G_2G_1G_0B_1B_0$ |
| | | x | x | 1 | 1pixel/1cycle | $R_0B_1B_0G_2G_1G_0R_2R_1R_0$ |
| | 16 | x | 0 | 0 | 2pixel/1cycle | $R_2R_1R_0G_2G_1G_0B_1B_0R_2R_1R_0G_2G_1G_0B_1B_0$ |
| | | x | 0 | 1 | 2pixel/1cycle | $B_1B_0G_2G_1G_0R_2R_1R_0B_1B_0G_2G_1G_0R_2R_1R_0$ |
| | | x | 1 | 0 | 2pixel/1cycle | $R_2R_1R_0G_2G_1G_0B_1B_0R_2R_1R_0G_2G_1G_0B_1B_0$ |
| | | x | 1 | 1 | 2pixel/1cycle | $B_1B_0G_2G_1G_0R_2R_1R_0B_1B_0G_2G_1G_0R_2R_1R_0$ |
| | 18 | x | 0 | 0 | 2pixel/1cycle | $xxR_2R_1R_0G_2G_1G_0B_1B_0R_2R_1R_0G_2G_1G_0B_1B_0$ |
| | | x | 0 | 1 | 2pixel/1cycle | $xxB_1B_0G_2G_1G_0R_2R_1R_0B_1B_0G_2G_1G_0R_2R_1R_0$ |
| | | x | 1 | 0 | 2pixel/1cycle | $xxR_2R_1R_0G_2G_1G_0B_1B_0R_2R_1R_0G_2G_1G_0B_1B_0$ |
| | | x | 1 | 1 | 2pixel/1cycle | $xxB_1B_0G_2G_1G_0R_2R_1R_0B_1B_0G_2G_1G_0R_2R_1R_0$ |
| RGB444 | 8 | x | 0 | 0 | 2pixel/3cycle | $R_3R_2R_1R_0G_3G_2G_1G_0$<br>$B_3B_2B_1B_0R_3R_2R_1R_0$<br>$G_3G_2G_1G_0B_3B_2B_1B_0$ |

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| RGB444 | 9 | x | 0 | 1 | 2pixel/3cycle | $B_3B_2B_1B_0G_3G_2G_1G_0$<br>$R_3R_2R_1R_0B_3B_2B_1B_0$<br>$G_3G_2G_1G_0R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 2pixel/3cycle | $G_3G_2G_1G_0B_3B_2B_1B_0$<br>$B_3B_2B_1B_0R_3R_2R_1R_0$<br>$R_3R_2R_1R_0G_3G_2G_1G_0$ |
| | | x | 1 | 1 | 2pixel/3cycle | $G_3G_2G_1G_0R_3R_2R_1R_0$<br>$R_3R_2R_1R_0B_3B_2B_1B_0$<br>$B_3B_2B_1B_0G_3G_2G_1G_0$ |
| | | x | 0 | 0 | 2pixel/3cycle | $G_0R_3R_2R_1R_0G_3G_2G_1G_0$<br>$R_0B_3B_2B_1B_0R_3R_2R_1R_0$<br>$B_0G_3G_2G_1G_0B_3B_2B_1B_0$ |
| | | x | 0 | 1 | 2pixel/3cycle | $G_0B_3B_2B_1B_0G_3G_2G_1G_0$<br>$B_0R_3R_2R_1R_0B_3B_2B_1B_0$<br>$R_0G_3G_2G_1G_0R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 2pixel/3cycle | $B_0G_3G_2G_1G_0B_3B_2B_1B_0$<br>$R_0B_3B_2B_1B_0R_3R_2R_1R_0$<br>$G_0R_3R_2R_1R_0G_3G_2G_1G_0$ |
| | | x | 1 | 1 | 2pixel/3cycle | $R_0G_3G_2G_1G_0R_3R_2R_1R_0$<br>$B_0R_3R_2R_1R_0B_3B_2B_1B_0$<br>$G_0B_3B_2B_1B_0G_3G_2G_1G_0$ |
| | 16 | 0 | x | 0 | 1pixel/1cycle | $R_3R_2R_1R_0G_3G_2G_1G_0B_3B_2B_1B_0$xxxx |
| | | 0 | x | 1 | 1pixel/1cycle | $B_3B_2B_1B_0G_3G_2G_1G_0R_3R_2R_1R_0$xxxx |
| | | 1 | x | 0 | 1pixel/1cycle | xxxx$R_3R_2R_1R_0G_3G_2G_1G_0B_3B_2B_1B_0$ |

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| | | 1 | x | 1 | 1pixel/1cycle | xxxx$B_3B_2B_1B_0G_3G_2G_1G_0R_3R_2R_1R_0$ |
| | 18 | 0 | x | 0 | 1pixel/1cycle | xx$R_3R_2R_1R_0G_3G_2G_1G_0B_3B_2B_1B_0$xxxx |
| | | 0 | x | 1 | 1pixel/1cycle | xx$B_3B_2B_1B_0G_3G_2G_1G_0R_3R_2R_1R_0$xxxx |
| | | 1 | x | 0 | 1pixel/1cycle | xxxxxx$R_3R_2R_1R_0G_3G_2G_1G_0B_3B_2B_1B_0$ |
| | | 1 | x | 1 | 1pixel/1cycle | xxxxxx$B_3B_2B_1B_0G_3G_2G_1G_0R_3R_2R_1R_0$ |
| RGB565 | 8 | x | 0 | 0 | 1pixel/2cycle | $R_4R_3R_2R_1R_0G_5G_4G_3$<br>$G_2G_1G_0B_4B_3B_2B_1B_0$ |
| | | x | 0 | 1 | 1pixel/2cycle | $B_4B_3B_2B_1B_0G_5G_4G_3$<br>$G_2G_1G_0R_4R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 1pixel/2cycle | $G_2G_1G_0B_4B_3B_2B_1B_0$<br>$R_4R_3R_2R_1R_0G_5G_4G_3$ |
| | | x | 1 | 1 | 1pixel/2cycle | $G_2G_1G_0R_4R_3R_2R_1R_0$<br>$B_4B_3B_2B_1B_0G_5G_4G_3$ |
| | 9 | x | 0 | 0 | 1pixel/2cycle | $G_3R_4R_3R_2R_1R_0G_5G_4G_3$<br>$B_0G_2G_1G_0B_4B_3B_2B_1B_0$ |
| | | x | 0 | 1 | 1pixel/2cycle | $G_3B_4B_3B_2B_1B_0G_5G_4G_3$<br>$R_0G_2G_1G_0R_4R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 1pixel/2cycle | $B_0G_2G_1G_0B_4B_3B_2B_1B_0$<br>$G_3R_4R_3R_2R_1R_0G_5G_4G_3$ |
| | | x | 1 | 1 | 1pixel/2cycle | $R_0G_2G_1G_0R_4R_3R_2R_1R_0$<br>$G_3B_4B_3B_2B_1B_0G_5G_4G_3$ |
| | 16 | x | x | 0 | 1pixel/1cycle | $R_4R_3R_2R_1R_0G_5G_4G_3G_2G_1G_0B_4B_3B_2B_1B_0$ |
| | | x | x | 1 | 1pixel/1cycle | $B_4B_3B_2B_1B_0G_5G_4G_3G_2G_1G_0R_4R_3R_2R_1R_0$ |
| | 18 | x | x | 0 | 1pixel/1cycle | xx$R_4R_3R_2R_1R_0G_5G_4G_3G_2G_1G_0B_4B_3B_2B_1B_0$ |

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| | | x | x | 1 | 1pixel/1cycle | $xxB_4B_3B_2B_1B_0G_5G_4G_3G_2G_1G_0R_4R_3R_2R_1R_0$ |
| RGB666 | 8 | 0 | 0 | 0 | 1pixel/3cycle | $R_5R_4R_3R_2R_1R_0xx$<br>$G_5G_4G_3G_2G_1G_0xx$<br>$B_5B_4B_3B_2B_1B_0xx$ |
| | | 0 | 0 | 1 | 1pixel/3cycle | $B_5B_4B_3B_2B_1B_0xx$<br>$G_5G_4G_3G_2G_1G_0xx$<br>$R_5R_4R_3R_2R_1R_0xx$ |
| | | 0 | 1 | 0 | 1pixel/3cycle | $B_5B_4B_3B_2B_1B_0xx$<br>$G_5G_4G_3G_2G_1G_0xx$<br>$R_5R_4R_3R_2R_1R_0xx$ |
| | | 0 | 1 | 1 | 1pixel/3cycle | $R_5R_4R_3R_2R_1R_0xx$<br>$G_5G_4G_3G_2G_1G_0xx$<br>$B_5B_4B_3B_2B_1B_0xx$ |
| | | 1 | 0 | 0 | 1pixel/3cycle | $xxR_5R_4R_3R_2R_1R_0$<br>$xxG_5G_4G_3G_2G_1G_0$<br>$xxB_5B_4B_3B_2B_1B_0$ |
| | | 1 | 0 | 1 | 1pixel/3cycle | $xxB_5B_4B_3B_2B_1B_0$<br>$xxG_5G_4G_3G_2G_1G_0$<br>$xxR_5R_4R_3R_2R_1R_0$ |
| | | 1 | 1 | 0 | 1pixel/3cycle | $xxB_5B_4B_3B_2B_1B_0$<br>$xxG_5G_4G_3G_2G_1G_0$<br>$xxR_5R_4R_3R_2R_1R_0$ |

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| RGB666 | 9 | 1 | 1 | 1 | 1pixel/3cycle | $xxR_5R_4R_3R_2R_1R_0$<br>$xxG_5G_4G_3G_2G_1G_0$<br>$xxB_5B_4B_3B_2B_1B_0$ |
| | | x | 0 | 0 | 1pixel/2cycle | $R_5R_4R_3R_2R_1R_0G_5G_4G_3$<br>$G_2G_1G_0B_5B_4B_3B_2B_1B_0$ |
| | | x | 0 | 1 | 1pixel/2cycle | $B_5B_4B_3B_2B_1B_0G_5G_4G_3$<br>$G_2G_1G_0R_5R_4R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 1pixel/2cycle | $G_2G_1G_0B_5B_4B_3B_2B_1B_0$<br>$R_5R_4R_3R_2R_1R_0G_5G_4G_3$ |
| | | x | 1 | 1 | 1pixel/2cycle | $G_2G_1G_0R_5R_4R_3R_2R_1R_0$<br>$B_5B_4B_3B_2B_1B_0G_5G_4G_3$ |
| | 16 | 0 | 0 | 0 | 2pixel/3cycle | $R_5R_4R_3R_2R_1R_0G_5G_4G_3G_2G_1G_0xxxx$<br>$B_5B_4B_3B_2B_1B_0R_5R_4R_3R_2R_1R_0xxxx$<br>$G_5G_4G_3G_2G_1G_0B_5B_4B_3B_2B_1B_0xxxx$ |
| | | 0 | 0 | 1 | 2pixel/3cycle | $B_5B_4B_3B_2B_1B_0G_5G_4G_3G_2G_1G_0xxxx$<br>$R_5R_4R_3R_2R_1R_0B_5B_4B_3B_2B_1B_0xxxx$<br>$G_5G_4G_3G_2G_1G_0R_5R_4R_3R_2R_1R_0xxxx$ |
| | | 0 | 1 | 0 | 2pixel/3cycle | $G_5G_4G_3G_2G_1G_0B_5B_4B_3B_2B_1B_0xxxx$<br>$B_5B_4B_3B_2B_1B_0R_5R_4R_3R_2R_1R_0xxxx$<br>$R_5R_4R_3R_2R_1R_0G_5G_4G_3G_2G_1G_0xxxx$ |
| | | 0 | 1 | 1 | 2pixel/3cycle | $G_5G_4G_3G_2G_1G_0R_5R_4R_3R_2R_1R_0xxxx$<br>$R_5R_4R_3R_2R_1R_0B_5B_4B_3B_2B_1B_0xxxx$<br>$B_5B_4B_3B_2B_1B_0G_5G_4G_3G_2G_1G_0xxxx$ |

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| | | 1 | 0 | 0 | 2pixel/3cycle | xxxx$R_5R_4R_3R_2R_1R_0G_5G_4G_3G_2G_1G_0$<br>xxxx$B_5B_4B_3B_2B_1B_0R_5R_4R_3R_2R_1R_0$<br>xxxx$G_5G_4G_3G_2G_1G_0B_5B_4B_3B_2B_1B_0$ |
| | | 1 | 0 | 1 | 2pixel/3cycle | xxxx$B_5B_4B_3B_2B_1B_0G_5G_4G_3G_2G_1G_0$<br>xxxx$R_5R_4R_3R_2R_1R_0B_5B_4B_3B_2B_1B_0$<br>xxxx$G_5G_4G_3G_2G_1G_0R_5R_4R_3R_2R_1R_0$ |
| | | 1 | 1 | 0 | 2pixel/3cycle | xxxx$G_5G_4G_3G_2G_1G_0B_5B_4B_3B_2B_1B_0$<br>xxxx$B_5B_4B_3B_2B_1B_0R_5R_4R_3R_2R_1R_0$<br>xxxx$R_5R_4R_3R_2R_1R_0G_5G_4G_3G_2G_1G_0$ |
| | | 1 | 1 | 1 | 2pixel/3cycle | xxxx$G_5G_4G_3G_2G_1G_0R_5R_4R_3R_2R_1R_0$<br>xxxx$R_5R_4R_3R_2R_1R_0B_5B_4B_3B_2B_1B_0$<br>xxxx$B_5B_4B_3B_2B_1B_0G_5G_4G_3G_2G_1G_0$ |
| | 18 | x | x | 0 | 1pixel/1cycle | $R_5R_4R_3R_2R_1R_0G_5G_4G_3G_2G_1G_0B_5B_4B_3B_2B_1B_0$ |
| | | x | x | 1 | 1pixel/1cycle | $B_5B_4B_3B_2B_1B_0G_5G_4G_3G_2G_1G_0R_5R_4R_3R_2R_1R_0$ |
| RGB888 | 8 | x | 0 | 0 | 1pixel/3cycle | $R_7R_6R_5R_4R_3R_2R_1R_0$<br>$G_7G_6G_5G_4G_3G_2G_1G_0$<br>$B_7B_6B_5B_4B_3B_2B_1B_0$ |
| RGB888 | 8 | x | 0 | 1 | 1pixel/3cycle | $B_7B_6B_5B_4B_3B_2B_1B_0$<br>$G_7G_6G_5G_4G_3G_2G_1G_0$<br>$R_7R_6R_5R_4R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 1pixel/3cycle | $B_7B_6B_5B_4B_3B_2B_1B_0$<br>$G_7G_6G_5G_4G_3G_2G_1G_0$<br>$R_7R_6R_5R_4R_3R_2R_1R_0$ |

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| | 9 | x | 1 | 1 | 1pixel/3cycle | $R_7R_6R_5R_4R_3R_2R_1R_0$<br>$G_7G_6G_5G_4G_3G_2G_1G_0$<br>$B_7B_6B_5B_4B_3B_2B_1B_0$ |
| | | x | 0 | 0 | 1pixel/3cycle | $R_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$G_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$B_0B_7B_6B_5B_4B_3B_2B_1B_0$ |
| | | x | 0 | 1 | 1pixel/3cycle | $B_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$G_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$R_0R_7R_6R_5R_4R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 1pixel/3cycle | $B_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$G_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$R_0R_7R_6R_5R_4R_3R_2R_1R_0$ |
| | | x | 1 | 1 | 1pixel/3cycle | $R_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$G_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$B_0B_7B_6B_5B_4B_3B_2B_1B_0$ |
| | 16 | x | 0 | 0 | 2pixel/3cycle | $R_7R_6R_5R_4R_3R_2R_1R_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$B_7B_6B_5B_4B_3B_2B_1B_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$G_7G_6G_5G_4G_3G_2G_1G_0B_7B_6B_5B_4B_3B_2B_1B_0$ |
| | | x | 0 | 1 | 2pixel/3cycle | $B_7B_6B_5B_4B_3B_2B_1B_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$R_7R_6R_5R_4R_3R_2R_1R_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$G_7G_6G_5G_4G_3G_2G_1G_0R_7R_6R_5R_4R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 2pixel/3cycle | $G_7G_6G_5G_4G_3G_2G_1G_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$B_7B_6B_5B_4B_3B_2B_1B_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$R_7R_6R_5R_4R_3R_2R_1R_0G_7G_6G_5G_4G_3G_2G_1G_0$ |

| format | I/F width | padding | significance | sequence | throughput (pixel/cycle) | output sequence |
|---|---|---|---|---|---|---|
| RGB888 | 16 | x | 1 | 1 | 2pixel/3cycle | $G_7G_6G_5G_4G_3G_2G_1G_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$R_7R_6R_5R_4R_3R_2R_1R_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$B_7B_6B_5B_4B_3B_2B_1B_0G_7G_5G_5G_4G_3G_2G_1G_0$ |
| | 18 | x | 0 | 0 | 2pixel/3cycle | $xxR_7R_6R_5R_4R_3R_2R_1R_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$xxB_7B_6B_5B_4B_3B_2B_1B_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$xxG_7G_6G_5G_4G_3G_2G_1G_0B_7B_6B_5B_4B_3B_2B_1B_0$ |
| | | x | 0 | 1 | 2pixel/3cycle | $xxB_7B_6B_5B_4B_3B_2B_1B_0G_7G_6G_5G_4G_3G_2G_1G_0$<br>$xxR_7R_6R_5R_4R_3R_2R_1R_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$xxG_7G_6G_5G_4G_3G_2G_1G_0R_7R_6R_5R_4R_3R_2R_1R_0$ |
| | | x | 1 | 0 | 2pixel/3cycle | $xxG_7G_6G_5G_4G_3G_2G_1G_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$xxB_7B_6B_5B_4B_3B_2B_1B_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$xxR_7R_6R_5R_4R_3R_2R_1R_0G_7G_6G_5G_4G_3G_2G_1G_0$ |
| | | x | 1 | 1 | 2pixel/3cycle | $xxG_7G_6G_5G_4G_3G_2G_1G_0R_7R_6R_5R_4R_3R_2R_1R_0$<br>$xxR_7R_6R_5R_4R_3R_2R_1R_0B_7B_6B_5B_4B_3B_2B_1B_0$<br>$xxB_7B_6B_5B_4B_3B_2B_1B_0G_7G_5G_5G_4G_3G_2G_1G_0$ |

**Table 60** WROICON.FORMAT List

## 9000_0084h Region of Interest Window Offset Register LCD_WROIOFS

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

**X-OFFSET** ROI Window Column Offset, please see Figure 51.
**Y-OFFSET** ROI Window Row Offset, please see Figure 51.

## 9000_0088h Region of Interest Window Command Address Register LCD_WROICADD

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

**ADDR** ROI Window Command Address. Specify the address that the commands will be sent to. Only eight possible values are allowed.

- **4000h** Commands are sent to LCD parallel interface 0 and LPA0 will be set to 0.
- **5000h** Commands are sent to LCD parallel interface 1 and LPA0 will be set to 0.
- **6000h** Commands are sent to LCD on EMI interface and LPA0 will be set to 0.
- **8000h** Commands are sent to LCD serial interface and LPA0 will be set to 0.
- **4100h** Commands are sent to LCD parallel interface 0 and LPA0 will be set to 1.
- **5100h** Commands are sent to LCD parallel interface 1 and LPA0 will be set to 1.
- **6100h** Commands are sent to LCD on EMI interface and LPA0 will be set to 1.
- **8100h** Commands are sent to LCD serial interface and LPA0 will be set to 1.

Note: In order to access port 6000 directly, software should access the EMI memory location instead. For example, if the LCM is connected to CS2 on the EMI, then address bit [28:27] = 10. Address bit[1] represents the LCM A0 bit. So if we want to write to the LCM with A0=1, then we write to address 0x1000_0002.

## 9000_008Ch Region of Interest Window Data Address Register LCD_WROIDADD

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

**ADDR** ROI Window Data Address. Specify the address that the data will be sent to. Only eight possible values are allowed.

- **4000h** Data are sent to LCD parallel interface 0 and LPA0 will be set to 0.
- **5000h** Data are sent to LCD parallel interface 1 and LPA0 will be set to 0.
- **6000h** Data are sent to LCD on EMI interface and LPA0 will be set to 0.

**8000h** Data are sent to LCD serial interface and LPA0 will be set to 0.
**4100h** Data are sent to LCD parallel interface 0 and LPA0 will be set to 1.
**5100h** Data are sent to LCD parallel interface 1 and LPA0 will be set to 1.
**6100h** Data are sent to LCD on EMI interface and LPA0 will be set to 1.
**8100h** Data are sent to LCD serial interface and LPA0 will be set to 1.

## 9000_0090h Region of Interest Window Size Register LCD_WROISIZE

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

**COLUMN** ROI Window Column Size in unit of pixel, please see Figure 51.
**ROW** ROI Window Row Size in unit of pixel, please see Figure 51.

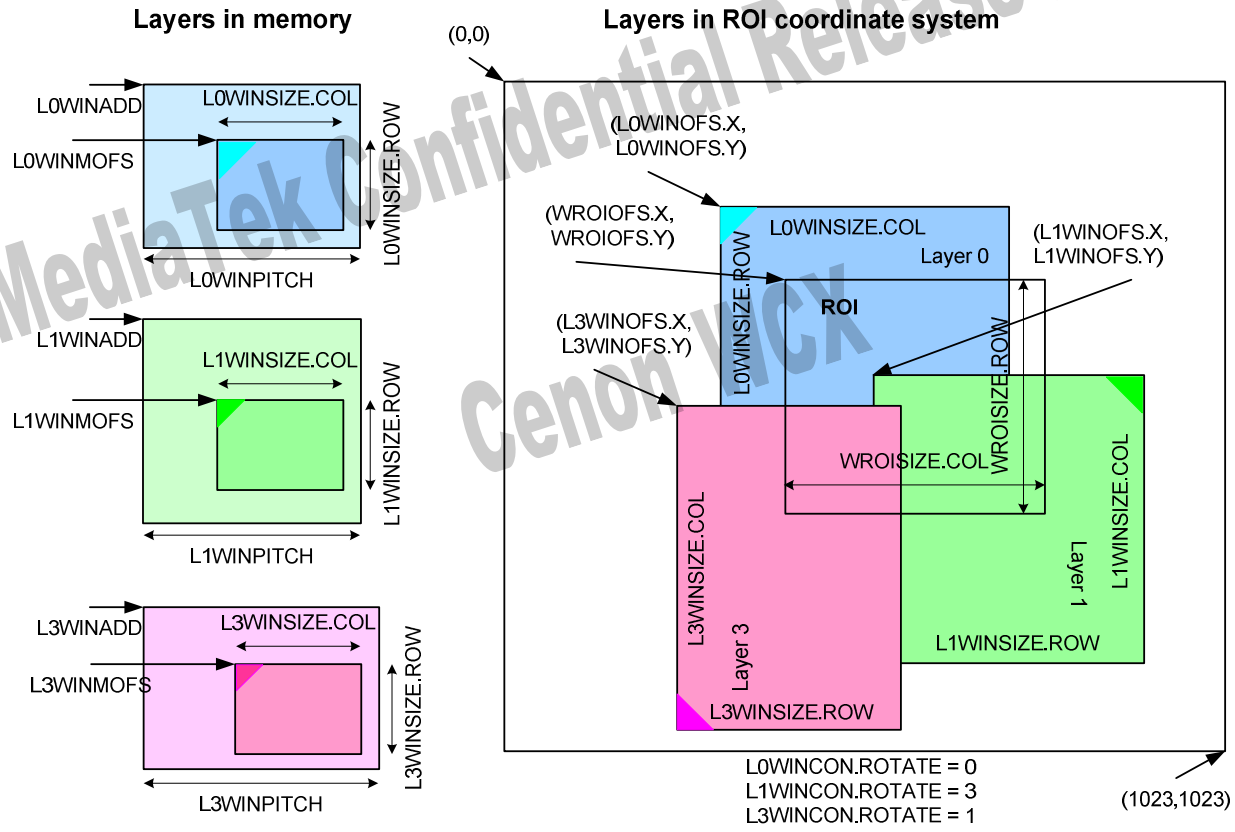



Figure 51 **Layers and ROI setting**

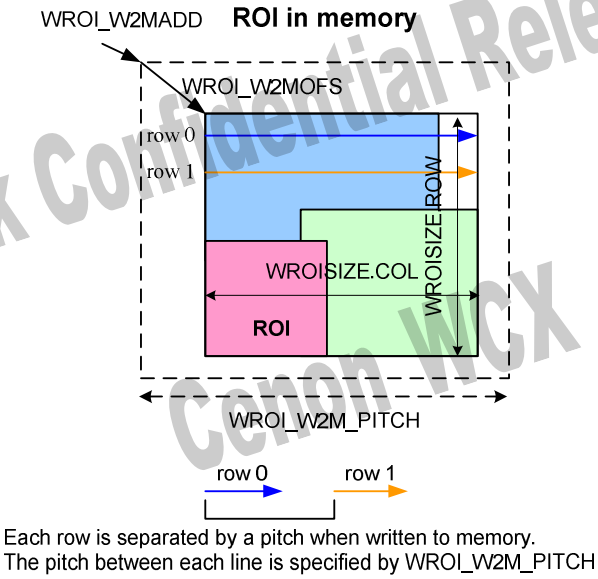



Figure 52 **ROI write to memory setting**

## 9000_009C Region of Interest Background Color Register — LCD_WROI_BGCLR

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ALPHA[7:0] | | | | | | | | RED[7:0] | | | | | | | |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | GREEN[7:0] | | | | | | | | BLUE[7:0] | | | | | | | |
| 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 |

**BLUE** Blue component of ROI window's background color
**GREEN** Green component of ROI window's background color
**RED** Red component of ROI window's background color
**ALPHA** Alpha component of ROI window's background color

## 9000_00A0h Region of Interest Window Write to LCM Continuous Count LCD_WROI_CONTI

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

**WROI_CONTI** Continuous write to LCM count without chip select de-asserted.

- **0** Continuously write 1 pixel to LCM without chip select de-asserted.
- **1** Continuously write 2 pixels to LCM without chip select de-asserted.
- **N** Continuously write N+1 pixels to LCM without chip select de-asserted.

**NOTE: IF LCD_WROICON.PERIOD != 0, this register won't take effect.**

## 9000_00B0h Layer 0 Window Control Register LCD_L0WINCON

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | RGB_SWP | | DST_KEYEN | | CLRFMT | | | | DITHER_EN | SCRL_EN | BYTE_SWP |
| Type | | | | | | | | 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 | SRC | SRC_KEYEN | ROTATE | | | | | ALPHA_EN | ALPHA | | | | | | | |
| Type | R/W | R/W | R/W | | | | | R/W | R/W | | | | | | | |

**ALPHA** Constant alpha value.

**ALPHA_EN** Enable Alpha blending

- **0** Disable Alpha blending
- **1** Enable Alpha blending, and the blending equations are specified by CLRFMT

**ROTATE** Rotation Configuration

- **000** no rotation
- **001** 90 degree rotation (counterclockwise, single request only)
- **010** 180 degree rotation (counterclockwise)
- **011** 270 degree rotation (counterclockwise, single request only)
- **100** Horizontal flip
- **101** Horizontal flip then 90 degree rotation (counterclockwise, single request only)

**110** Horizontal flip then 180 degree rotation (counterclockwise)
**111** Horizontal flip then 270 degree rotation (counterclockwise, single request only)

**SRC_KEYEN** Enable source color key. If the color format is YUYV422, this function is not supported.

**SRC** Disable auto-increment of the source pixel address. It makes the value of each pixel is the same as the first pixel of this frame. Just for debug.

**BYTE_SWP** Swap high byte and low byte of pixel data read from memory

**SCRL_EN** Enable scroll effect.

**DITHER_EN** Enable dithering. Please see LCD_DITERCON.

**CLRFMT** Color format
**000** 8bpp indexed color
**001** RGB 565
**010** YUYV422
**011** RGB 888
**100** ARGB 8888
**101** PARGB8888
**110** XRGB8888
**Others** Reserved

**DST_KEYEN** Enable destination color key. If the color format is YUYV422, this function is not supported.

**RGB_SWP** Swap RGB order of input pixel.

Note: SRC_KEYEN and DST_KEYEN are exclusive setting. They can't be enabled at the same time.

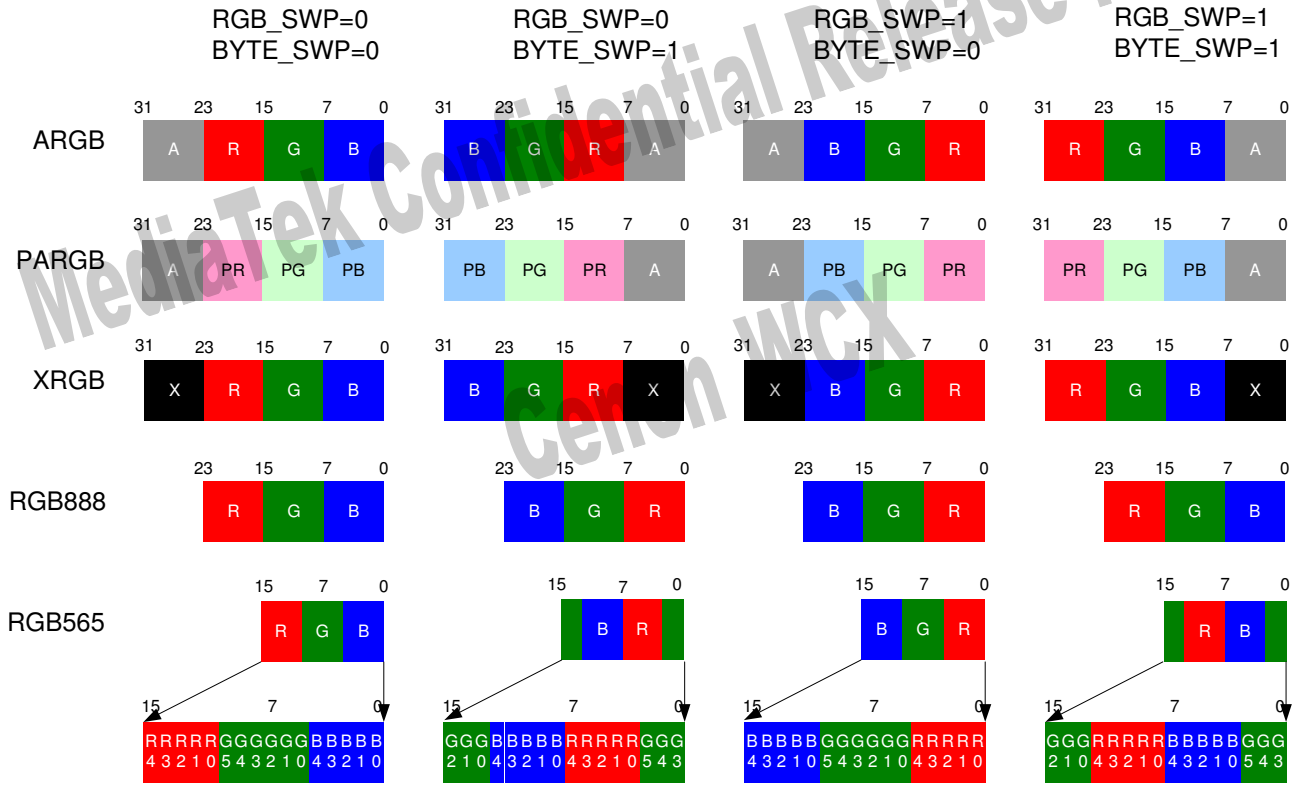



Figure 53 **Layer source RGB format**

The byte order in memory of YUYV422 is described in Figure 54 **YUYV422 byte order in memory**. $Y_0$ is the Y component of the first pixel, $P_0$. $Y_1$ is the Y component of the second pixel, $P_1$.

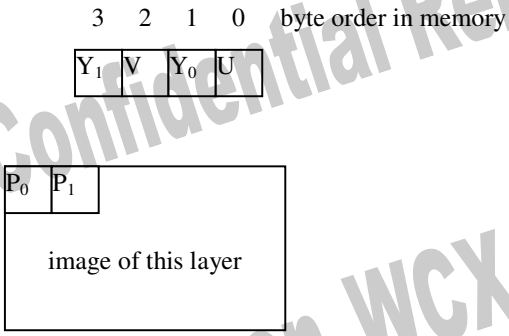



Figure 54 **YUYV422 byte order in memory**

**Note: When use YUYV422 mode, the pitch of this layer (LCD_LxWINPITCH) must be even, and the base address (LCD_LxWINADD) of this layer also must be 4-byte aligned. Source color key and destination color key are NOT supported in YUYV422 mode.**

Note: If color depth is YUYV422, the YUYV422 source will be translated to RGB domain and then overlaid. The YUV to RGB transformation is following the equations.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \frac{1}{32} \times \begin{bmatrix} 32 & 0 & 45 \\ 32 & -11 & -23 \\ 32 & 57 & 0 \end{bmatrix} \bullet \begin{pmatrix} Y \\ U-128 \\ V-128 \end{pmatrix}$$

The alpha blending formula is selected by source color format automatically.

If source color format is **RGB565, RGB888 or YUYV422** then the alpha blending formula is

dst.r = dst.r * (0xff - SCA ) / 0xff + src.r * SCA / 0xff;

dst.g = dst.g * (0xff - SCA ) / 0xff + src.g * SCA / 0xff;

dst.b = dst.b * (0xff - SCA ) / 0xff + src.b * SCA / 0xff;

dst.a = dst.a * (0xff - SCA ) / 0xff + SCA;

If source color format is **PARGB** then the alpha blending formula is

```
if ( SCA != 0xff) {
    dst.r = dst.r * (0xff - src.a * SCA / 0xff) / 0xff + src.r * SCA / 0xff;
    dst.g = dst.g * (0xff - src.a * SCA / 0xff) / 0xff + src.g * SCA / 0xff;
    dst.b = dst.b * (0xff - src.a * SCA / 0xff) / 0xff + src.b * SCA / 0xff;
    dst.a = dst.a * (0xff - src.a * SCA / 0xff) / 0xff + src.a * SCA / 0xff;
} else { // SCA == 0xff
    dst.r = dst.r * (0xff - src.a ) / 0xff + src.r;
    dst.g = dst.g * (0xff - src.a ) / 0xff + src.g;
    dst.b = dst.b * (0xff - src.a ) / 0xff + src.b;
    dst.a = dst.a * (0xff - src.a ) / 0xff + src.a
}
```

If source color format is **ARGB** then the alpha blending formula is

```
if ( SCA != 0xff ) {
    dst.r = dst.r * (0xff - src.a * SCA / 0xff) / 0xff + src.r * src.a / 0xff * SCA / 0xff;
    dst.g = dst.g * (0xff - src.a * SCA / 0xff) / 0xff + src.g * src.a / 0xff * SCA / 0xff;
    dst.b = dst.b * (0xff - src.a * SCA / 0xff) / 0xff + src.b * src.a / 0xff * SCA / 0xff;
    dst.a = dst.a * (0xff - src.a * SCA / 0xff) / 0xff + src.a * SCA / 0xff;
} else { // SCA == 0xff
    if SCA = 0xff
    dst.r = dst.r * (0xff - src.a ) / 0xff + src.r * src.a / 0xff;
    dst.g = dst.g * (0xff - src.a ) / 0xff + src.g * src.a / 0xff;
    dst.r = dst.b * (0xff - src.a ) / 0xff + src.b * src.a / 0xff;
    dst.a = dst.a * (0xff - src.a ) / 0xff + src.a;
}
```

src.r, src.g, src.b, and src.a are this layer's pixel value.
dst.r, dst.r, dst.b, and dst.a are the result of alpha blending of all lower layers.
Note: SCA is the source constant alpha specified by LCD_L0WINCON.ALPHA.

**Alpha blending hardware approximation**:

If source color format is **RGB565, RGB888 or YUYV422** then the hardware implements the following equation to approximate the above equation of 8-bit index color, RGB565, RGB888 or YUYV422. Only list red channel, other channels are the same.

```
tmp.r = SCA × (src.r - dst.r) + 255 * dst.r + 128;
dst'.r = (tmp.r + tmp.r >> 8) >> 8;
tmp_d.a = dst.a × (255 - SCA) + 128
tmp.a = (tmp_d.a + tmp_d.a >> 8) >> 8
dst'.a = src.a + tmp.a
```

If source color format is **PARGB** then the hardware implements the following equation to approximate the above equation of PARGB. Only list red channel, others are the same.

```
if ( SCA != 0xff ) {
    tmp_s.a = src.a × SCA + 128
    src'.a = (tmp_s.a + tmp_s.a >> 8) >> 8
    tmp_s.r = src.r × SCA + 128
    src'.r = (tmp_s.r + tmp_s.r >> 8) >> 8
    tmp_d.r = dst.r × (255 - src'.a) + 128
    tmp.r = (tmp_d.r + tmp_d.r >> 8) >> 8
    dst'.r = src'.r + tmp.r
} else { // SCA == 0xff
    tmp_d.r = dst.r × (255 - src.a) + 128
    tmp.r = (tmp_d.r + tmp_d.r >> 8) >> 8
    dst'.r = src.r + tmp.r
}
```

If source color format is **ARGB** then the hardware implements the following equation to approximate the above equation of ARGB. Only list red and alpha channels, others are the same.

```
if ( SCA != 0xff ) {
    tmp_s.a = src.a × SCA + 128;
    src'.a = (tmp_s.a + tmp_s.a >> 8) >> 8;
    tmp_d.a = dst.a × (255 - src'.a) + 128;
    tmp.a = (tmp_d.a + tmp_d.a >> 8) >> 8;
    dst'.a = src'.a + tmp.a;

    tmp.r = src'.a × (src.r - dst.r) + 255 * dst.r + 128;
    dst'.r = (tmp.r + tmp.r >> 8) >> 8;
} else { // SCA == 0xff
    tmp_d.a = dst.a × (255 - src.a) + 128;
    tmp.a = (tmp_d.a + tmp_d.a >> 8) >> 8;
    dst'.a = src.a + tmp.a;

    tmp.r = src.a × (src.r - dst.r) + 255 * dst.r + 128;
    dst'.r = (tmp.r + tmp.r >> 8) >> 8;
}
```

**Effect Ordering:** Each layer has many effects which can be turned on concurrently. The order the effects are applied are as follows:

1. Memory Offset and Pitch are first used to determine which part of the layer in memory to display.
2. If turned on, a scroll effect is then applied.
3. Rotation is applied to the layer.
4. Finally, swap and dither are applied in this order
5. The layer is alpha blended with previous layers and/or the ROI background.
6. If turned on, the gamma tables are applied to the final ROI output.
7. The ROI output is sent to the LCM and/or memory in the color format set by the corresponding register.

## 9000_00B4h Layer 0 Color Key Register — LCD_L0WINKEY

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

**CLRKEY** The source color key or destination color key, which depends on LCD_L0WINCON.SRC_KEYEN or LCD_L0WINCON.DST_KEYEN.

## 9000_00B8h Layer 0 Window Display Offset Register — LCD_L0WINOFS

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

**Y-OFFSET** Layer 0 Window Row Offset in unit of pixel, please see Figure 51.

**X-OFFSET** Layer 0 Window Column Offset in unit of pixel, please see Figure 51.

## 9000_00BCh Layer 0 Window Display Start Address Register — LCD_L0WINADD

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

**ADDR** Layer 0 source start address (byte address), please see Figure 51. The address must be aligned to layer color depth boundary as the following table. Note: If the buffer is Page-size byte aligned, LCD will

benefit from a ~2-3% bandwidth increase. For example, if the page size of the memory is 16bytes, then a 16byte alignment address will increase bandwidth.

| LCD_L0WINCON.CLRFMT | Color format | ADDR alignment |
|---|---|---|
| 000 | 8 bpp index | no alignment constraint |
| 001 | RGB565 | 2 bytes alignment |
| 010 | ARGB8888 or PARGB8888 | 4 bytes alignment |
| 011 | RGB888 | no alignment constraint |
| 100 | YUYV422 | 4 bytes alignment |

## 9000_00C0h Layer 0 Window Size — LCD_L0WINSIZE

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

**ROW** Layer 0 Window Row Size in unit of pixel, please see Figure 51.
**COLUMN** Layer 0 Window Column Size in unit of pixel, please see Figure 51.

## 9000_00C4h Layer 0 Scroll Start Offset — LCD_L0WINSCRL

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

**Y-OFFSET** Layer 0 Scroll Y Offset in unit of pixel, its value must satisfy Y-OFFSET < LCD_L0WINSIZE.ROW
**X-OFFSET** Layer 0 Scroll X Offset in unit of pixel, its value must satisfy X-OFFSET < LCD_L0WINSIZE.COLUMN

## 9000_00C8h Layer 0 Memory Offset — LCD_L0WINMOFS

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

**Y-OFFSET** Layer 0 Memory Y Offset in unit of pixel, please see Figure 51. Y-OFFSET < LCD_L0WINSIZE.ROW + PICTURE ROW

**X-OFFSET** Layer 0 Memory X Offset in unit of pixel, please see Figure 51. X-OFFSET < LCD_L0WINSIZE.COLUMN + PICTURE COLUMN

PICTURE ROW and PICTURE COLUMN refer to the total size of the image in memory. See Figure 51 for clarification.

## 9000_00CCh Layer 0 Memory Pitch — LCD_L0WINPITCH

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

**PITCH** Layer 0 Memory Pitch in unit of byte, please see Figure 51. This should be set to the total width of the image in memory (PICTURE COLUMN) times the number of bytes per pixel. For 4 bpp color depth settings, the pitch must be a multiple of 4. For 2 bpp color depth settings, the pitch must be a multiple of 2. For 3 bpp (RGB888) color depth settings, the pitch may be a multiple of any number.

**Note: Every field of every register of layer 1~3 is the same as that in layer 0 configuration register.**

## 9000_0170h LCD Dither Control Register — LCD_DITHER_CON

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

| Name | | | | | | | | | LFSR_R_SEED | | | | | | LFSR_G_SEED | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | 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 | LFSR_B_SEED | | | | | | DB_R | | | | DB_G | | | | DB_B | |
| Type | R/W | R/W | R/W | R/W | | | R/W | R/W | | | R/W | R/W | | | R/W | R/W |

**DB_B** Dither bit selection on blue channel

**00** dither to bit 1, bit [0] of output data is 0.

**01** dither to bit 2, bit [1:0] of output data are 0.

**10** dither to bit 3, bit [2:0] of output data are 0.

**11** dither to bit 4, bit [3:0] of output data are 0.

**DB_G** Dither bit selection on green channel

**00** dither to bit 1, bit [0] of output data is 0.

**01** dither to bit 2, bit [1:0] of output data are 0.

**10** dither to bit 3, bit [2:0] of output data are 0.

**11** dither to bit 4, bit [3:0] of output data are 0.

**DB_R** Dither bit selection on red channel

**00** dither to bit 1, bit [0] of output data is 0.

**01** dither to bit 2, bit [1:0] of output data are 0.

**10** dither to bit 3, bit [2:0] of output data are 0.

**11** dither to bit 4, bit [3:0] of output data are 0.

(e.g. RGB888 to RGB565, BD_R=2;BD_G=1;BD_B=2)

(e.g. RGB888 to RGB666, BD_R=1;BD_G=1;BD_B=1)

**LFSR_B_SEED** Seeds (LSB 4-bit) of blue channel Linear Feedback Shift Register

**LFSR_G_SEED** Seeds (LSB 4-bit) of green channel Linear Feedback Shift Register

**LFSR_R_SEED** Seeds (LSB 4-bit) of red channel Linear Feedback Shift Register

Note: LFSR_*_SEED shouldn't be zero, otherwise the linear feedback shift register will always be zero.

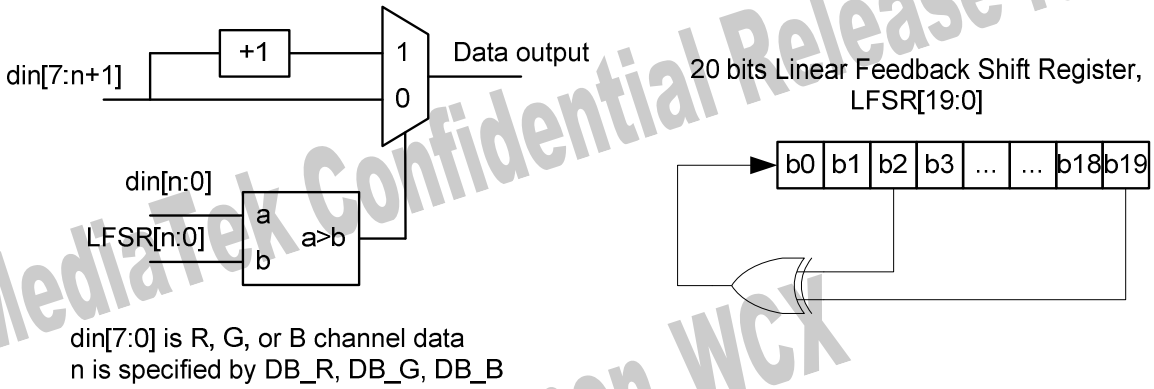



Figure 55 **Dither operation**

## 9000_01F0h Addcon Debug Register

**LCD_DB_ADDCON**

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

**Y-OFFSET** Current ROI pixel requesting from memory y offset.
**X-OFFSET** Current ROI pixel requesting from memory x offset.

## 9000_01F4h Maincon Debug Register

**LCD_DB_MCON**

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

**Y-OFFSET** Current ROI pixel writing to LCM y offset.
**X-OFFSET** Current ROI pixel writing to LCM x offset.

## 9000_01F8h W2mcon Debug Register — LCD_DB_W2MCON

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

**Y-OFFSET** Current ROI pixel writing to memory y offset.
**X-OFFSET** Current ROI pixel writing to memory x offset.

## 9000_01FCh Frame Count Debug Register — LCD_DB_COUNT

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

**FCOUNT** LCD Update Time in cycles. The real time is FCOUNT*T, where T is the LCD working clock cycle time.

**Below are registers which allow software to write/read directly to the DBI-B/DBI-C interface. The LCD allows software to issue consecutive writes/reads without delay. However on the last read/write, software should add a delay before writing to other memory location (non-LCD memory locations). Each port also has TE subports (ex. 0x4010, 0x5010, etc.) which adhere to the same rules as its master port. For example 0x4010 uses the same rules as 0x4000. However, these TE ports have special functions other then LCM write/read. Please refer to Table 59 for more information.**

## 9000_4000h LCD Parallel 0 Interface Command Port — LCD_PCMD0

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

**CMD** LCD parallel 0 command interface is mapped to this address. Read from/Write to this address will assert LCD parallel 0 chips select, and LPA0=0.

## 9000_4100h LCD Parallel 0 Interface Data Port LCD_PDAT0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DATA[31:16] | | | | | | | | | | | | | | | |
| Type | 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 | | | | | | | | | | | | | | | |

**DATA** LCD parallel 0 data interface is mapped to this address. Read from/Write to this address will assert LCD parallel 0 chips select, and LPA0=1.

## 9000_5000h LCD Parallel 1 Interface Command Port LCD_PCMD1

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

**CMD** LCD parallel 1 command interface is mapped to this address. Read from/Write to this address will assert LCD parallel 1 chips select, and LPA0=0.

## 9000_5100h LCD Parallel 1 Interface Data Port LCD_PDAT1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DATA[31:16] | | | | | | | | | | | | | | | |
| Type | 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 | | | | | | | | | | | | | | | |

**DATA** LCD parallel 1 data interface is mapped to this address. Read from/Write to this address will assert LCD parallel 1 chips select, and LPA0=1.

| Parallel I/F data width (bits) | AHB write transaction size (bits) | data sequence in parallel I/F |
|---|---|---|
| 8 | 8 | hwdata[7:0] |
| | 16 | hwdata[7:0]<br>hwdata[15:8] |
| | 32 | hwdata[7:0]<br>hwdata[15:8]<br>hwdata[23:16]<br>hwdata[31:24] |
| 9 | 8 | hwdata[8:0] |
| | 16 | hwdata[8:0] |
| | 32 | hwdata[8:0]<br>hwdata[24:16] |
| 16 | 8 | hwdata[15:0] |
| | 16 | hwdata[15:0] |
| | 32 | hwdata[15:0]<br>hwdata[31:16] |
| 18 | 8 | hwdata[17:0] |
| | 16 | hwdata[17:0] |
| | 32 | hwdata[17:0] |

Table 61 **Parallel interface transmission sequence when MCU writes to LCD_PDAT* or LCD_PCMD***

## 9000_8000h LCD Serial Interface Command Port LCD_SCMD

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

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

**CMD** LCD serial command interface is mapped to this address. Write to this address will assert LSA0=0 in 4-wire mode or A0 bit=0 in 3-wire mode.

## 9000_8100h LCD Serial Interface Data Port LCD_SDAT

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DATA[31:16] | | | | | | | | | | | | | | | |
| Type | 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 | | | | | | | | | | | | | | | |

**DATA** LCD serial data interface is mapped to this address. Write to this address will assert LSA0=1 in 4-wire mode or A0 bit=1 in 3-wire mode.

| LCD_SCNF.3_WIRE | LCD_SCNF.IF_SIZE | data sequence in serial I/F |
|---|---|---|
| 0 | 8 | hwdata[7:0] |
| | 9 | hwdata[8:0] |
| | 16 | hwdata[15:0] |
| | 18 | hwdata[17:0] |
| | 24 | hwdata[23:0] |
| | 32 | hwdata[31:0] |
| 1 | 8 | {cmd_data, hwdata[7:0]} |
| | 9 | {cmd_data, hwdata[8:0]} |
| | 16 | {cmd_data, hwdata[15:0]} |

| | | |
|---|---|---|
| | 18 | {cmd_data, hwdata[17:0]} |
| | 24 | {cmd_data, hwdata[23:0]} |
| | 32 | {cmd_data, hwdata[31:0]} |

Table 62 **Serial interface transmission sequence when MCU writes to LCD_SDAT, LCD_SCMD**

Serial transmission starts from the MSB of "data sequence in serial I/F". “cmd_data” is the command/data indicator, 0 for command and 1 for data. If MCU writes the command port, cmd_data=0. If MCU writes the data port, cmd_data=1. To read a serial LCM, software must first write a read command to this interface. Next software, can read this interface to retrieve data. CS should be kept low while performing this operation. Please note that the AHB transaction size cannot be less then the LCD_SCNF.IF_SIZE or unknown data may be transferred to the LCM.

### 9000_C000h~C07F LCD Interface Command/Parameter Registers LCD_COMD

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | C0 | | | | | | COMM[17:16] | |
| Type | | | | | | | | | R/W | | | | | | R/W | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | COMM[15:0] | | | | | | | | | | | | | | | |
| Type | R/W | | | | | | | | | | | | | | | |

**COMM** Command and Parameter for LCD Module.

**C0** Command/data indicator.

- **0** Write COMM[17:0] to ROI Data Address(LCD_WROIDADD).
- **1** Write COMM[17:0] to ROI Command Address (LCD_WROICADD).

There are 16 entries in command queue 1. When LCD is in the busy state, software must not read/write to the command queue.

## 5.2 Camera Interface

MT6252 CAM support VGA Sensor YUV422/RGB565 interface. No other color process include in CAM.

### 5.2.1 Register Table

**CAM** : **84020000**h

| REGISTER ADDRESS | REGISTER NAME | SYNONYM |
|---|---|---|
| CAM + 0000h | TG Phase Counter Register | CAM_PHSCNT |
| CAM + 0004h | Sensor Size Configuration Register | CAM_CAMWIN |
| CAM + 0008h | TG Grab Range Start/End Pixel Configuration Register | CAM_GRABCOL |
| CAM + 000Ch | TG Grab Range Start/End Line Configuration Register | CAM_GRABROW |
| CAM + 0010h | Sensor Mode Configuration Register | CAM_CSMODE |
| CAM + 0018h | View Finder Mode Control Register | CAM_VFCON |
| CAM + 001Ch | Camera Module Interrupt Enable Register | CAM_INTEN |
| CAM + 0020h | Camera Module Interrupt Status Register | CAM_INTSTA |
| CAM + 0024h | Camera Module Path Config Register | CAM_PATH |
| CAM + 0190h | Sensor Test Module Configuration Register 1 | CAM_MDLCFG1 |
| CAM + 0194h | Sensor Test Module Configuration Register 2 | CAM_MDLCFG2 |
| CAM + 01A0h | Camera to CRZ Control Register | CAMCRZ_CTRL |
| CAM + 01A4h | Camera to CRZ Status Register | CAMCRZ_STA |
| CAM + 01D4h | TG Status Register | TG_STATUS |
| CAM + 01D8h | Cam Reset Register | CAM_RESET |

**Table 63** Camera Interface Register Map

### 5.2.1.1 TG Register Definitions

**CAM+0000h TG Phase Counter Register** **CAM_PHSCNT**

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | PCEN | | CLKEN | CLKPOL | CLKCNT | | | | CLKRS | | | | CLKFL | | | |
| Type | R/W | | R/W | R/W | R/W | | | | R/W | | | | R/W | | | |
| Reset | 0 | | 0 | 0 | 1 | | | | 0 | | | | 1 | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | HVALID_EN | CAM_PCLK_INV | | PXCLK_IN | CLKFL_POL | | TGCLK_SEL | | PIXCNT | | | | DLATCH | | | |
| Type | R/W | R/W | | R/W | R/W | | R/W | | R/W | | | | R/W | | | |
| Reset | 0 | 0 | | 0 | 0 | | 0 | | 1 | | | | 1 | | | |

**PCEN** TG phase counter enable control

**CLKEN** Enable sensor master clock (mclk) output to sensor.

**CLKPOL** Sensor master clock polarity control

**CLKCNT** Sensor master clock frequency divider control.
Sensor master clock will be TGCLK_SEL/(CLKCNT+1)

**CLKRS** Sensor master clock rising edge control

**CLKFL** Sensor master clock falling edge control

**HVALID_EN** Sensor hvalid or href enable

**CAM_PCLK_INV** Pixel clock inverse in CAM.

**PXCLK_IN** Pixel clock sync enable. If sensor master based clock is from 48MHz, PXCLK_IN must be enabled.

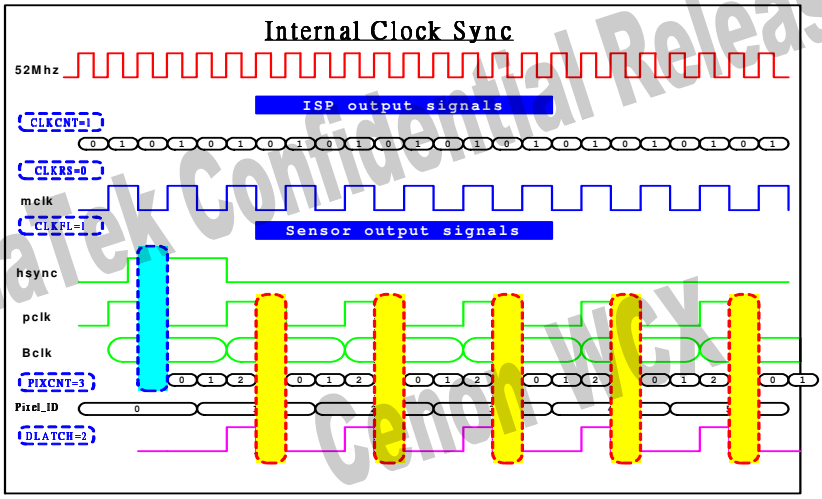
**CLKFL_POL** Sensor clock falling edge polarity

**TGCLK_SEL** Sensor master based clock selection (0: 52 Mhz, 1: 48MHz, 2: 104MHz)

**PIXCNT** Sensor data latch frequency control

**DLATCH** Sensor data latch position control

**Example waveform**(CLKCNT=1,CLKRS=0,CLKFL=1,PIXCNT=3,DLATCH=2)




## CAM+0004h Sensor Size Configuration Register CAM_CAMWIN

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

**PIXEL** Total input pixel number

**LINE** Total input line number

## CAM+0008h TG Grab Range Start/End Pixel Configuration Register CAM_GRABCOL

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

**START** Grab start pixel number (first pixel start from 0)

**END** Grab end pixel number (first pixel start from 0)

## CAM+000Ch TG Grab Range Start/End Line Configuration Register CAM_GRABROW

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

**START** Grab start line number (first line start from 1, line 0 is inside VSYNC). Note that this number should be set start from 1, set 0 to this parameter will cause memory output data uncompleted.

**END** Grab end line number (first line start from 1)

## CAM+0010h Sensor Mode Configuration Register CAM_CSMODE

| 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 | | | | | | | | | VSPOL | HSPOL | PWRON | RST | AUTO | | | EN |
| Type | | | | | | | | | R/W | R/W | R/W | R/W | R/W | | | R/W |
| Reset | | | | | | | | | 0 | 0 | 0 | 0 | 0 | | | 0 |

**VSPOL** Sensor Vsync input polarity

**HSPOL** Sensor Hsync input polarity

**AUTO** Auto lock sensor input horizontal pixel numbers enable

**EN** Sensor process counter enable

## CAM+0018h View Finder Mode Control Register CAM_VFCON

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | AV_SYNC_SEL | VD_INT_POL | | | AV_SYNC_LINENO[11:0] | | | | | | | | | | | |
| Type | R/W | R/W | | | R/W | | | | | | | | | | | |
| Reset | 0 | 0 | | | 0 | | | | | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | | | | SP_DELAY[2:0] | | | SP_MODE | TAKE_PIC | | | | FR_CON | | |
| Type | | | | | | R/W | | | R/W | R/W | | | | R/W | | |
| Reset | | | | | | 0 | | | 0 | 0 | | | | 0 | | |

**AV_SYNC_SEL** Av_sync start point selection
- **0** Start from AV_SYNC_LINENO
- **1** Start from vsync

**VD_INT_POL** Vsync interrupt polarity
- **0** Vsync rising edge
- **1** Vsync Falling edge

**AV_SYNC_LINENO** Av_sync desired line counts

**SP_DELAY[2:0]** Still Picture Mode Frame Delay. When SP_DELAY is nonzero, TAKE_PIC will start to trigger after SP_DELAY Vsync singal.

**SP_MODE** Still Picture Mode
- **0** Preview mode, ISP will process every frame sensor send
- **1** Capture mode, ISP will only process first frame sensor send after TAKE_PIC is set

**TAKE_PIC** Take Picture Request

**FR_CON** Frame Sampling Rate Control
- **000** Every frame is sampled
- **001** One frame is sampled every 2 frames
- **010** One frame is sampled every 3 frames
- **011** One frame is sampled every 4 frames
- **100** One frame is sampled every 5 frames
- **101** One frame is sampled every 6 frames
- **110** One frame is sampled every 7 frames
- **111** One frame is sampled every 8 frames

## CAM+001Ch Camera Module Interrupt Enable Register CAM_INTEN

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | VSYNC_INT_EN | | | | | | | | | | | | | | | INT_WCLR_EN |
| Type | R/W | | | | | | | | | | | | | | | R/W |
| Reset | 0 | | | | | | | | | | | | | | | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | | | | | | AV_SYNC_INT | FLASH_INT | | | ISPDONE | IDLE | | REZOVRUN | EXPDO |
| Type | | | | | | | | R/W | R/W | | | R/W | R/W | | R/W | R/W |
| Reset | | | | | | | | 0 | 0 | | | 0 | 0 | | 0 | 0 |

**VSYNC_INT_EN** Vsync interrupt and Flash interrupt switch
- **0** Flash interrupt
- **1** Vsync interrupt

**INT_WCLR_EN** interrupt write clear enable

**PCA_INT** PCA interrupt enable

**FLK_INT_EN** Flicker interrupt enable

**AV_SYNC_INT** AV sync interrupt enable

**FLASH_INT** Flash interrupt enable, note that **VSYNC_INT_EN** switch flash and vsync interrupt

**AEDONE** AE done interrupt enable

**ISPDONE** ISP done interrupt enable

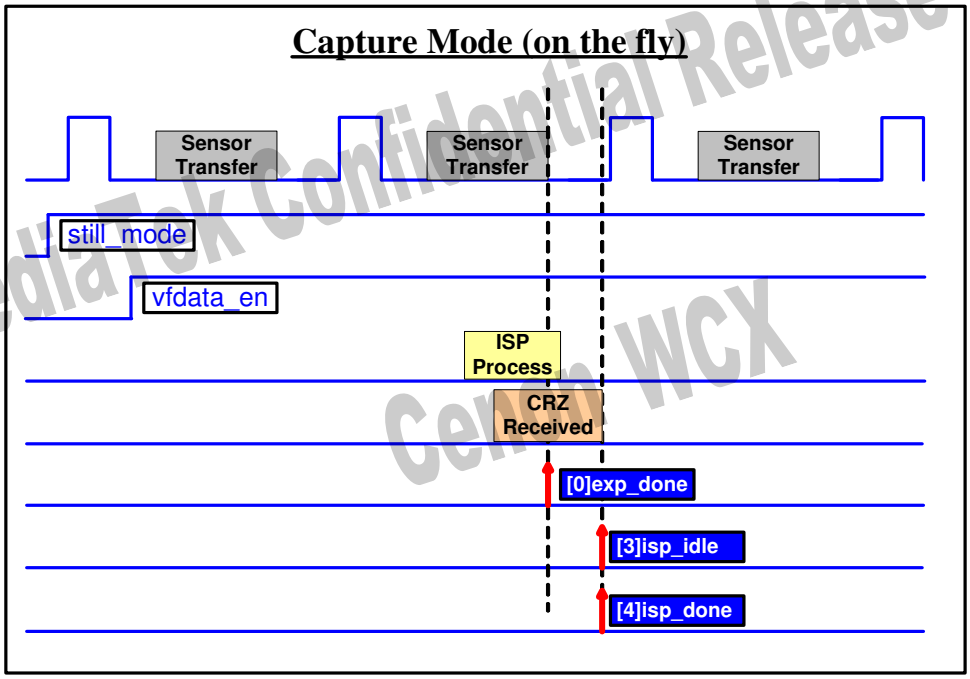
**IDLE** Returning idle state interrupt enable

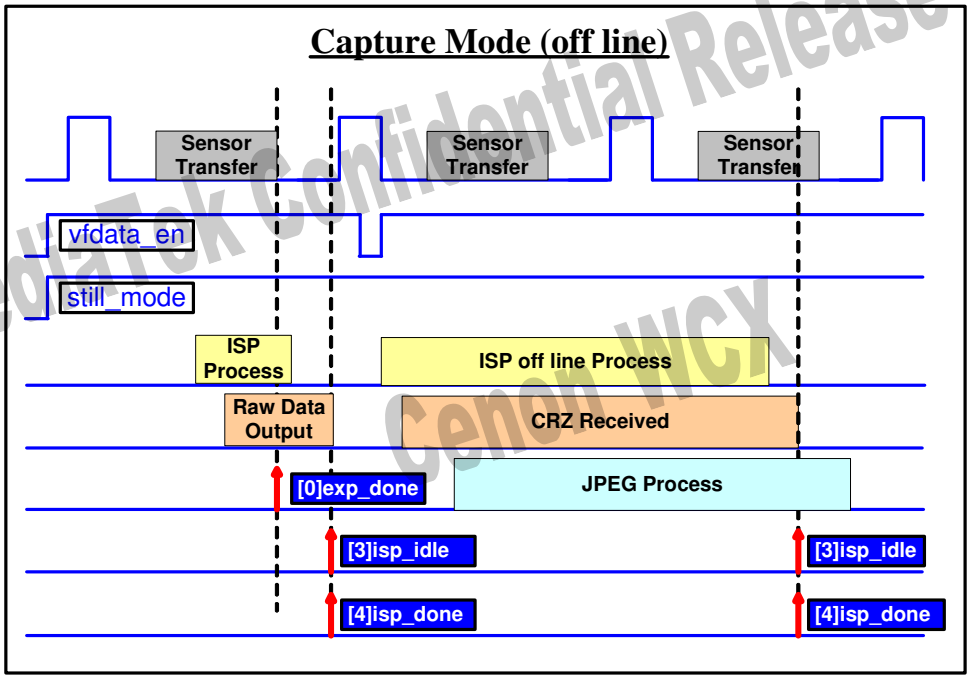
**REZOVRUN** CRZ overrun interrupt enable

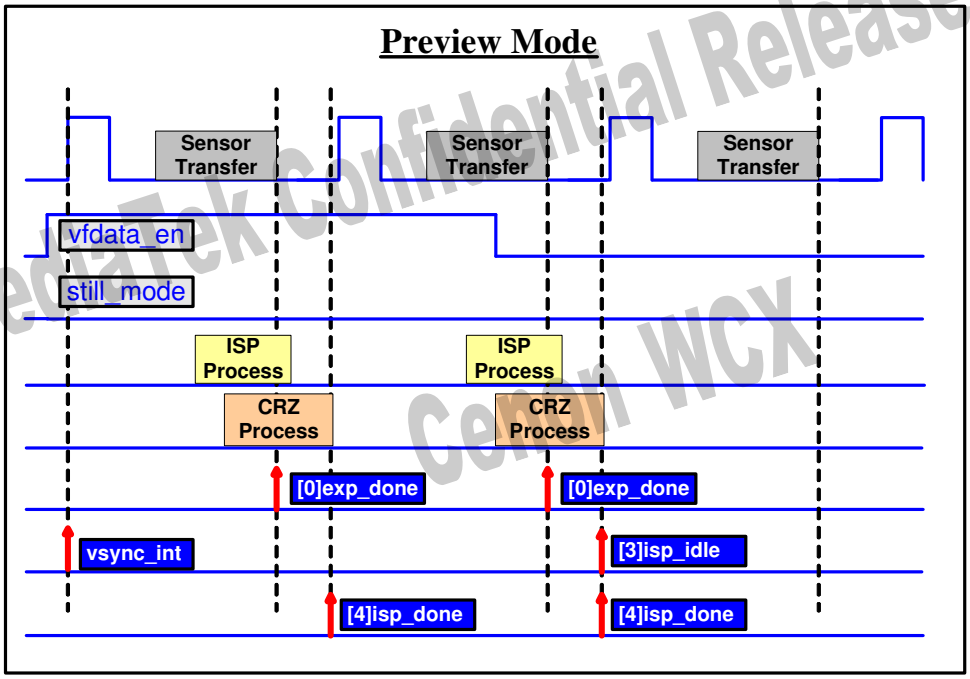
**EXPDO** Exposure done interrupt enable



## Interrupt Condition Chart








## CAM+0020h Camera Module Interrupt Status Register CAM_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 | | | | | | | | AV_SYNC_INT | FLASH_INT | | | ISPDONE | IDLE | | REZOVRUN | EXPDO |
| Type | | | | | | | | R | R | | | R | R | | R | R |
| Reset | | | | | | | | 0 | 0 | | | 0 | 0 | | 0 | 0 |

This register show the status of corresponding bit of CAM_INTEN. Note that default interrupt status is read clear. If CAM_INTEN[16] is set 1, then interrupt status is write clear.

**AV_SYNC_INT** AV sync interrupt status, occurred when desired line count equal **AV_SYNC_LINENO** in CAM_VFCON(CAM+0018h)

**FLASH_INT** TG interrupt status, occurred when flash light pulse is done

**ISPDONE** ISP done interrupt status, occurred when ISP finish full frame process

**IDLE** Returning idle state interrupt status, occurred when ISP is in IDLE state

**REZOVRUN** Resizer over run interrupt status, occurred when ISP to CRZ buffer is overrun

**EXPDO** Exposure done interrupt status, occurred when sensor send full frame

## CAM+0024h Camera Module Path Config Register CAM_PATH

| 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 | | | SWAP_Y | SWAP_CBCR | | INTYPE_SEL | | | | | | | | | | |
| Type | | | R/W | R/W | | R/W | | | | | | | | | | |
| Reset | | | 0 | 0 | | 0 | | | | | | | | | | |

**SWAP_Y** YCbCr in Swap Y, note that **INTYPE_SEL** should be set to 001 or 101

**SWAP_CBCR** YCbCr in Swap Cb Cr, note that **INTYPE_SEL** should be set to 001 or 101

**INTYPE_SEL** Input type selection

**000** Reserved

**001** YUV422 Format

**101** YCbCr422 Format

**010** RGB565 Format

**Others** Reserved

## CAM+0190h Sensor Test Model Configuration Register 1 CAM_MDLCFG1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | VSYNC | | | | | | | | IDLE_PIXEL_PER_LINE | | | | | | | |
| Type | R/W | | | | | | | | R/W | | | | | | | |
| Reset | 0 | | | | | | | | 0 | | | | | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | LINECHG_EN | FULL_RANGE | | | ON | RST | STILL | PATTERN | PIXEL_SEL | | CLK_DIV | | | |
| Type | | | 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 | | | |

**VSYNC** VSYNC high duration in line unit(IDLE_PIXEL_PER_LINE + PIXEL)

**IDLE_PIXEL_PER_LINE** HSYNC low duration in pixel unit

**LINECHG_EN** Pattern 0 2 lines change mode enable

**FULL_RANGE** Sensor Model Full Range Enable. When full range is enable, pattern data value will increase

progressively every line output.

**ON** Enable Sensor Model.

**RST** Reset Sensor Model

**STILL** Still picture Mode. Set 1 will generate fix pattern.

**PATTERN** Sensor Model Test Pattern Selection

**PIXEL_SEL** Sensor Model output pixel selection.

- **00** All pixels
- **01** 01 pixel
- **10** 10 pixel
- **11** 00 and 11 pixels

**CLK_DIV** Pixel_Clock/System_Clock Ratio

## CAM +0194h Sensor Test Model Configuration Register 2 CAM_MDLCFG2

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

**LINE** Sensor Model Line Number

**PIXEL** Sensor Model Pixel Number (HSYNC high duration in pixel unit)

## CAM +01A0h Camera to CRZ Control Register CAMCRZ_CTRL

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

| Name | | | | CAMCRZ_INIT_EN | CAMCRZ_INIT_PERIOD | | | | | | REZ_OVRUN_FLIMIT_EN | REZ_OVRUN_FLIMIT_NO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | R/W | R/W | | | | | | R/W | R/W | | | |
| Reset | | | | 0 | 10 | | | | | | 0 | 0 | | | |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Name | | | | | | | | | | | | | | | | |
| Type | | | | | | | | | | | | | | | | |
| Reset | | | | | | | | | | | | | | | | |

**CAMCRZ_INIT_EN** Camera to CRZ frame initialization scheme enable. In frame initialization, camera will block signal to CRZ and assert initial signal to CRZ when pixel dropped.

**CAMCRZ_INIT_PERIOD** Camera to CRZ initialization signal active period

**REZ_OVRUN_FLIMIT_EN** Camera to CRZ interface overrun frame count limit enable. When enabled, if REZ_OVRUN_FCOUNT (CAM+01A4[19:16]) equal REZ_OVRUN_FLIMIT_NO,REZOVRUN (CAM+0020[1]) will be asserted.

**REZ_OVRUN_FLIMIT_NO** Camera to CRZ interface overrun frame count limit number.

## CAM +01A4h Camera to CRZ Status Register — CAMCRZ_STA

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

**CAMCRZ_FIFOCNT** Camera to CRZ FIFO Count. There are 48 pixels buffer in Camera.

**REZ_OVRUN_FCOUNT** Camera to Resizer interface over run frame count. This frame count is for capture mode. REZ_OVRUN_FCOUNT will be reset every time TAKE_PIC (CAM _ 0018 [6]) is set to 0, will be add 1 when REZOVRUN occurred in a frame.

## CAM+01D4h TG Status Register — TG_STATUS

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | CAM_BUSY | CAPTURE_BUSY | SYN_VF_DATA_EN | LINE_COUNT[11:0] | | | | | | | | | | | |

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

**CAM_BUSY** CAM busy flag

**CAPTURE_BUSY** Capture busy flag

**SYN_VF_DATA_EN** View finder double buffer register status

**LINE_COUNT** Input frame line counter

**PIXEL_COUNT** Input frame pixel counter

## CAM +01D8h CAM RESET Register CAM_RESET

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

**CAM_CS** CAM status. There are 6 states

**1** cam_idle

**2** preview_idle

**4** preview_process

**8** preview_wait

**16** capture_process

**32** capture_done

**SW_RESET** Camera software reset. When software reset is enable, camera will automatically stop process including memory access and will assert hardware reset at suitable time. After reset is done, SW_RESET will be de-assert.

| | |
|---|---|
| **CAM_FRAME_COUNT** | CAM process frame count number |
| **HARD_RESET** | ISP reset. Note that reset should be assert longer than 100us to make sure GMC command done. |



MediaTek Confidential Release for Cenon WCX

# 6 Analog Front-end & Analog Blocks

## 6.1 General Description

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 APB bus write and read cycle for specific addresses related to analog front-end control. During writing or reading of any of these control registers, there is a latency associated with transferring of data to or from the analog front-end. Dedicated data interface of each analog block is implemented in the corresponding digital block. The Analog Blocks includes the following analog function for complete GSM/GPRS base-band signal processing:

1. *Base-band RX*: For I/Q channels base-band A/D conversion
2. *RF Control*: One DAC for automatic power control (APC) is included. Its output is provided to external RF power amplifier.
3. *Auxiliary ADC*: Providing an ADC for battery and other auxiliary analog function monitoring
4. *Audio mixed-signal blocks:* It provides complete analog voice signal processing including microphone amplification, A/D conversion, D/A conversion, earphone driver, and etc. Besides, dedicated stereo D/A conversion and amplification for audio signals are included).
5. *Clock Generation*: A clock squarer for shaping system clock, and three PLLs that provide clock signals to DSP, MCU, USB, and SFC units are included
6. *XOSC32*: It is a 32-KHz crystal oscillator circuit for RTC application Analog Block Descriptions

### 6.1.1 BBRX

#### 6.1.1.1 Block Descriptions

The receiver (RX) performs base-band I/Q channels downlink analog-to-digital conversion:

3. *Analog input multiplexer:* For each channel, a 4-input multiplexer that supports offset calibration is included.
4. *A/D converter:* Two 14-bit sigma-delta ADCs perform I/Q digitization for further digital signal processing.

#### 6.1.1.2 Functional Specifications

The functional specifications of the base-band downlink receiver are listed in the following table.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| N | Resolution | | 14 | | Bit |
| FC | Clock Rate | | 26 | | MHz |
| FS | Output Sampling Rate | | 13/12 | | MSPS |
| | Input Swing<br>When GAIN='00'<br>When GAIN='01'<br>When GAIN='10'<br>When GAIN='11' | | 0.8*AVDD<br>0.4*AVDD<br>0.57*AVDD<br>0.33*AVDD | | Vpk,diff |
| OE | Offset Error | | +/- 10 | | mV |
| FSE | Full Swing Error | | +/- 30 | | mV |
| | I/Q Gain Mismatch | | | 0.5 | dB |
| SINAD | Signal to Noise and Distortion Ratio<br>- 45kHz sine wave in [0:90] kHz bandwidth<br>- 145kHz sine wave in [10:190] kHz bandwidth | 65<br>65 | | | dB<br>dB |
| ICN | Idle channel noise<br>- [0:90] kHz bandwidth<br>- [10:190] kHz bandwidth | | | -74<br>-70 | dB<br>dB |
| DR | Dynamic Range<br>- [0:90] kHz bandwidth<br>- [10:190] kHz bandwidth | 74<br>70 | | | dB<br>dB |
| RIN | Input Resistance | 75 | | | kΩ |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| | Current Consumption<br>Power-up<br>Power-Down | | <br>5<br>5 | | <br>mA<br>µA |

**Table 64** Base-band Downlink Specifications

## 6.1.2 APC-DAC

### 6.1.2.1 Block Descriptions

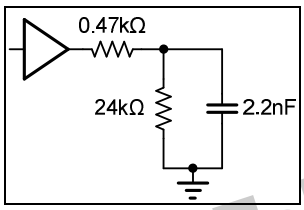
The APC-DAC is a 10-bits DAC with output buffer aimed for automatic power control. Here blow are its functional specification tables.

### 6.1.2.2 Function Specifications

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| N | Resolution | | 10 | | Bit |
| FS | Sampling Rate | | | 1.0833 | MSPS |
| | 99% Settling Time (Full Swing on Maximal Capacitance) | | | 5 | µS |
| | 99% Settling Time (100 Code) | | | 1/1.0833 | µS |
| | Output Swing | 0 | | AVDD | V |
| | Power-down Glitch* (After Serial Resistor) | | | 10 | mV |
| | Drive Capacitance | | 200 | | pF |
| | Drive Resistance | | 10 | | kΩ |
| DNL | Differential Nonlinearity | | +/- 0.5 | | LSB |
| INL | Integral Nonlinearity | | +/- 1.0 | | LSB |
| OE | Offset Error | | +/- 30 | | mV |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| | Current Consumption<br>Power-up<br>Power-Down | | <br>200<br>1 | | <br>µA<br>µA |

**Table 65** APC-DAC Specifications

* APCDAC power-down glitch reference load configuration:




## 6.1.3 Auxiliary ADC

### 6.1.3.1 Block Descriptions

The auxiliary ADC includes the following functional blocks:

1. *Analog Multiplexer:* The analog multiplexer selects signal from one of the several auxiliary input pins. Real word message to be monitored, like temperature, should be transferred to the voltage domain.
2. *10 bits A/D Converter:* The ADC converts the multiplexed input signal to 10-bit digital data.

### 6.1.3.2 Function Specifications

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

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| N | Resolution | | 10 | | Bit |
| FC | Clock Rate | | 1.0833 | | MHz |
| *FS | Sampling Rate @ N-Bit | | 1.0833/(N +1) | | MSPS |
| | Input Range | 0 | | AVDD | V |
| CIN | Input Capacitance<br>Unselected Channel<br>Selected Channel | | | <br>50<br>1.2 | <br>fF<br>pF |
| RIN | Input Resistance<br>Unselected Channel<br>Selected Channel | <br>10<br>1.8 | | | <br>MΩ<br>MΩ |
| | Clock Latency | | 11 | | 1/FC |
| DNL | Differential Nonlinearity | | +1.0/-1.0 | | LSB |

| | | | | | |
|---|---|---|---|---|---|
| INL | Integral Nonlinearity | | +1.0/-1.0 | | 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 | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| | Current Consumption<br>Power-up<br>Power-Down | | <br>150<br>1 | | <br>μA<br>μA |

**Table 66** The Functional specification of Auxiliary ADC

* FS: Sampling rate can reach 1.0833/(N+1) MSPS with special software configuration

** FSE: Full swing error compared with real time analog power supply of AUXADC (AVDD)

## 6.1.4 Audio mixed-signal blocks

### 6.1.4.1 Block Descriptions

Audio mixed-signal blocks (AMB) integrate complete voice uplink/downlink and audio playback functions. As shown in the following figure, it includes mainly three parts. The first consists of stereo audio DACs and headset amplifiers for audio playback and external radio playback. The second is the voice downlink path, including voice-band amplifiers, which produces voice signal to earphone or other auxiliary output device. The last is the voice uplink path, which is the interface between microphone (or other auxiliary input device) input and MT6252 DSP. A set of bias voltage is provided for external electric microphone.

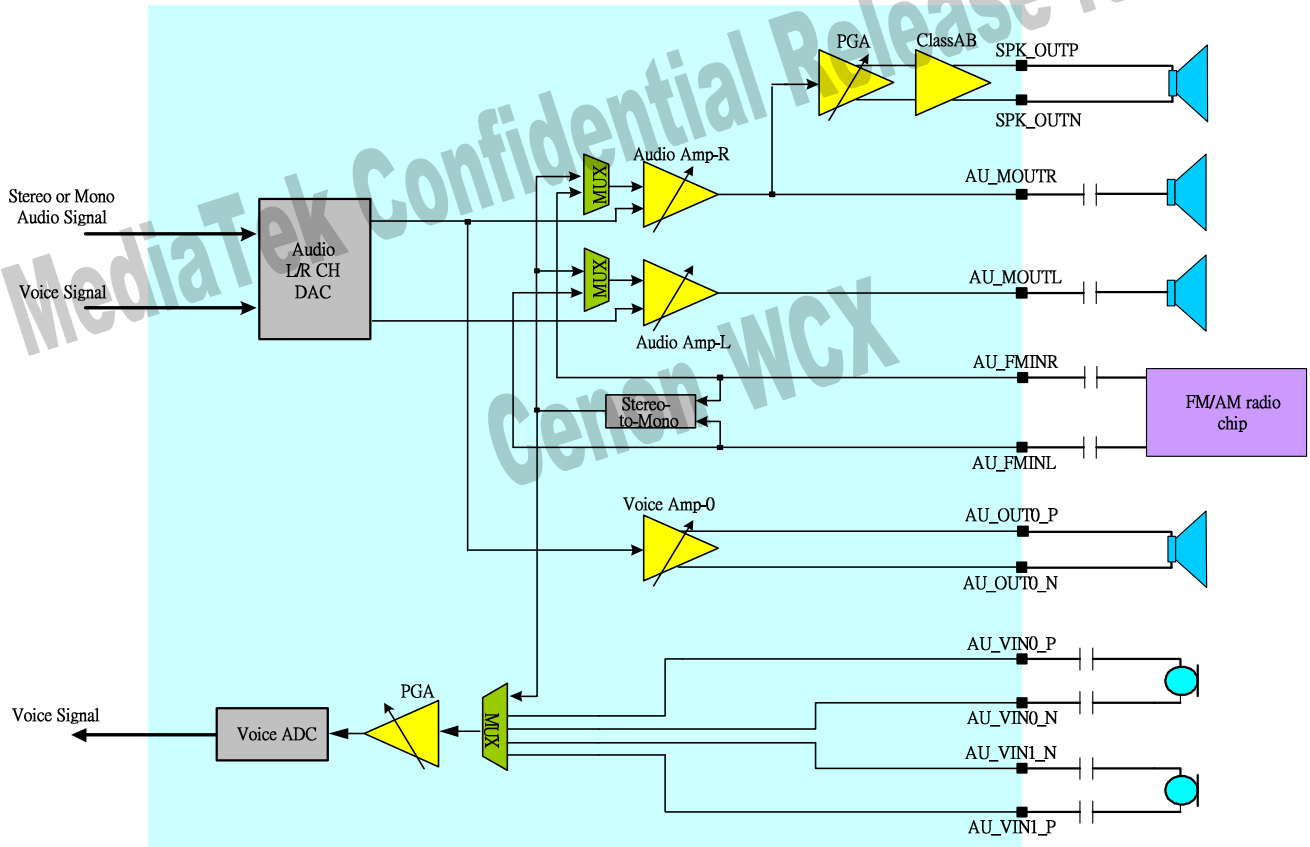



**Figure 56** Block diagram of audio mixed-signal blocks.

### 6.1.4.2 Functional Specifications

The following table gives functional specifications of voice-band uplink/downlink blocks.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| FS | Sampling Rate | | 6500 | | KHz |
| CREF | Decoupling Cap Between AU_VCM and AGND28_AFE | | 1 | | uF |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |

| | | | | | |
|---|---|---|---|---|---|
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| IDC | Current Consumption | | 5 | | mA |
| VMIC | Microphone Biasing Voltage | | 1.9 | 2.2 | V |
| IMIC | Current Draw From Microphone Bias Pins | | | 2 | mA |
| Uplink Path[1] | | | | | |
| IDC | Current Consumption | | 1.5 | | mA |
| SINAD | Signal to Noise and Distortion Ratio<br>Input Level: -40 dbm0<br>Input Level: 0 dbm0 | 29 | 69 | | dB<br>dB |
| RIN | Input Impedance (Differential) | 13 | 20 | 27 | KΩ |
| ICN | Idle Channel Noise | | | -67 | dBm0 |
| XT | Crosstalk Level | | | -66 | dBm0 |
| Downlink Path[2] | | | | | |
| SINAD | Signal to Noise and Distortion Ratio<br>Input Level: -40 dBm0<br>Input Level: 0 dBm0 | 29 | 69 | | dB<br>dB |
| RLOAD | Output Resistor Load (Differential) | 25.6 | | | Ω |
| CLOAD | Output Capacitor Load | | | 1000 | pF |
| ICN | Idle Channel Noise of Transmit Path | | | -67 | dBm0 |
| XT | Crosstalk Level on Transmit Path | | | -66 | dBm0 |

**Table 67** Functional specifications of analog voice blocks

Functional specifications of the audio blocks are described in the following.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| FCK | Clock Frequency | | 6.5 | | MHz |

[1] For uplink-path, not all gain setting of VUPG meets the specification listed on table, especially for the several lowest gains. The minimum gain that meets the specification is to be determined.

[2] For downlink-path, not all gain setting of VDPG meets the specification listed on table, especially for the several lowest gains. The minimum gain that meets the specification is to be determined.

| | | | | | |
|---|---|---|---|---|---|
| Fs | Sampling Rate | 32 | 44.1 | 48 | KHz |
| AVDD | Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| IDC | Current Consumption | | 5 | | mA |
| PSNR | Peak Signal to Noise Ratio | | 80 | | dB |
| DR | Dynamic Range | | 80 | | dB |
| VOUT | Output Swing for 0dBFS Input Level | | 0.7 | | Vrms |
| THD | Total Harmonic Distortion<br>22mW at 32 Ω Load | | | -60 | dB |
| RLOAD | Output Resistor Load (Single-Ended) | 32 | | | Ω |
| CLOAD | Output Capacitor Load | | | 200 | pF |
| XT | L-R Channel Cross Talk Suppression | 70 | | | dB |

**Table 68** Functional specifications of the analog audio blocks

## 6.1.5 Clock Squarer

### 6.1.5.1 Block Descriptions

In MT6252MT6252, RF is integrated and 26MHz is provided internally. Therefore, there is no dedicated input pin for 26MHz clock. For DCXO in RF, the output clock waveform is sinusoidal with small amplitude (about several hundred mV). To make MT6252MT6252 digital circuits function well, clock squarer is designed to convert such a small signal to a rail-to-rail clock signal with excellent duty-cycle.

### 6.1.5.2 Function Specifications

The functional specification of clock squarer is shown in **Table 69**.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| Fin | Input Clock Frequency | | 26 | | MHz |
| Fout | Output Clock Frequency | | 26 | | MHz |
| Iin | Input Signal Amplitude | | 75 | | uA |

| | | | | | |
|---|---|---|---|---|---|
| DcycIN | Input Signal Duty Cycle | | 50 | | % |
| DcycOUT | Output Signal Duty Cycle | DcycIN-5 | | DcycIN+5 | % |
| TR | Rise Time on Pin CLKSQOUT | | | 5 | ns/pF |
| TF | Fall Time on Pin CLKSQOUT | | | 5 | ns/pF |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| | Current Consumption | | 150 | | uA |

**Table 69** The Functional Specification of Clock Squarer

## 6.1.6 Phase Locked Loop

### 6.1.6.1 Block Descriptions

MT6252MT6252/E includes three PLLs: MCU PLL, SFC PLL, USB PLL. MCU PLL and SFC PLL are to provide 104MHz and 78MHz output clock while accepts 26MHz signal. USB PLL is designed to also accept 26MHz input clock signal and provides 48MHz, 78MHz, and 104MHz output clocks.

### 6.1.6.2 Function Specifications

The functional specification of MCU PLL is shown in the following table.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| Fin | Input Clock Frequency | | 26 | | MHz |
| Fout | Output Clock Frequency | 98 | | 104 | MHz |
| | Min. output frequency step | | 0.5 | | MHz |
| | Output Clock Cycle-to-Cycle Jitter (p-p) | | | 200 | ps |
| | Output Clock Duty Cycle | 45 | 50 | 55 | % |
| | Settling Time<br>for power on<br>For band switching | | 160<br>80 | 200<br>100 | us |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |

| | | | | | |
|---|---|---|---|---|---|
| | Current Consumption<br>For AVDD<br>For DVDD | | 0.8<br>0.2 | | mA |
| | Power Down Current Consumption<br>For AVDD<br>For DVDD | | | 1<br>2.5 | uA |

**Table 70** The Functional Specification of DSP/MCU PLL

The functional specification of SFC PLL is shown in the following table.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| Fin | Input Clock Frequency | | 26 | | MHz |
| Fout | Output Clock Frequency | 72 | | 78 | MHz |
| | Min. output frequency step | | 0.5 | | MHz |
| | Output Clock Cycle-to-Cycle Jitter (p-p) | | | 200 | ps |
| | Output Clock Duty Cycle | 45 | 50 | 55 | % |
| | Settling Time<br>for power on<br>For band switching | | 160<br>80 | 200<br>100 | us |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| | Current Consumption<br>For AVDD<br>For DVDD | | 0.8<br>0.2 | | mA |
| | Power Down Current Consumption<br>For AVDD<br>For DVDD | | | 1<br>2.5 | uA |

**Table 8** The Functional Specification of SFC PLL

The functional specification of USB PLL is shown below.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| Fin | Input Clock Frequency | | 26 | | MHz |
| Fout | Output Clock Frequency | | 48/78/104 | | MHz |
| | Output Clock Cycle-to-Cycle Jitter (p-p) | | | 200 | ps |
| | Output Clock Duty Cycle(78/104MHz) | 45 | 50 | 55 | % |
| | Settling Time | | 10 | 20 | us |
| DVDD | Digital Power Supply | 1.08 | 1.2 | 1.32 | V |
| AVDD | Analog Power Supply | 2.52 | 2.8 | 3.08 | V |
| T | Operating Temperature | -20 | | 85 | ℃ |
| | Current Consumption<br>For AVDD<br>For DVDD | | <br>1.2<br>0.25 | | mA |
| | Power Down Current Consumption<br>For AVDD<br>For DVDD | | | <br>1<br>2.5 | uA |

**Table 9** The Functional Specification of USB PLL

## 6.1.7 32-KHz Crystal Oscillator

### 6.1.7.1 Block Descriptions

The low-power 32-KHz crystal oscillator XOSC32 is designed to work with an external piezoelectric 32.768KHz crystal and a load composed of two functional capacitors, as shown in the following figure.

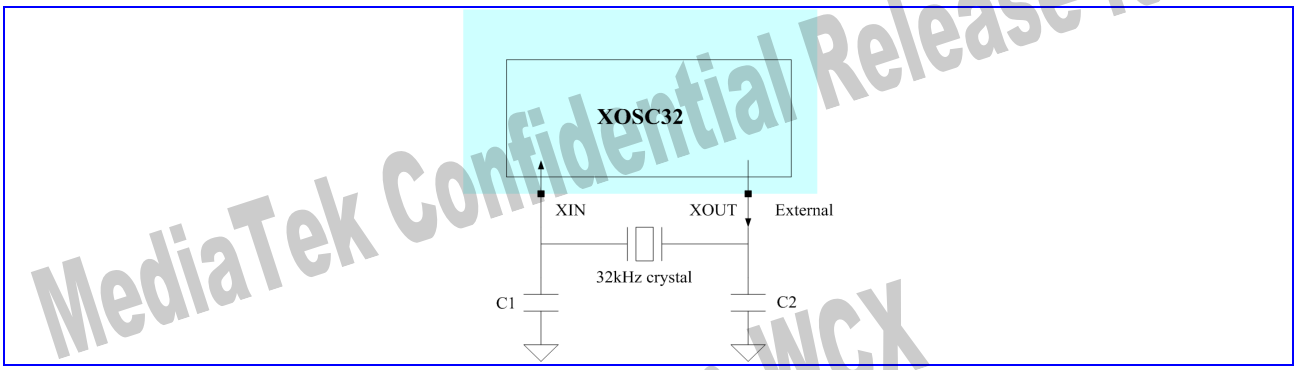



**Figure 57** Block diagram of XOSC32

### 6.1.7.2 Functional specifications

The functional specification of XOSC32 is shown in the following table.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| AVDDRTC | Analog power supply | 1 | 2.8 | 3.0 | V |
| Tosc | Start-up time | | | 1 | sec |
| Dcyc | Duty cycle | 20 | 50 | 80 | % |
| TR | Rise time on XOSCOUT | | TBD | | ns/pF |
| TF | Fall time on XOSCOUT | | TBD | | ns/pF |
| | Current consumption | | | 5 | µA |
| | Leakage current | | 1 | | µA |
| T | Operating temperature | -20 | | 85 | ℃ |

**Table 71** Functional Specification of XOSC32

Here below are a few recommendations for the crystal parameters for use with XOSC32.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| F | Frequency range | | 32768 | | Hz |
| GL | Drive level | | | 5 | uW |
| Δf/f | Frequency tolerance | | +/- 20 | | ppm |
| ESR | Series resistance | | | 50 | KΩ |

| | | | | | |
|---|---|---|---|---|---|
| C0 | Static capacitance | | | 1.6 | pF |
| CL[3] | Load capacitance | 6 | | 12.5 | pF |

**Table 72** Recommended Parameters of the 32KHz crystal

## 6.2 ABB Register Definitions

| ADDRESS | TITLE | DESCRIPTION |
|---|---|---|
| 8301_000C | WR_PATH | Switch configuration path control register |
| 8301_0100 | ACIF_VOICE_CON0 | VOICE control register 0 |
| 8301_0104 | ACIF_VOICE_CON1 | VOICE control register 1 |
| 8301_0108 | ACIF_VOICE_CON2 | VOICE control register 2 |
| 8301_010C | ACIF_VOICE_CON3 | VOICE control register 3 |
| 8301_010C | ACIF_VOICE_CON3 | VOICE control register 3 |
| 8301_0110 | ACIF_VOICE_CON4 | VOICE control register 4 |
| 8301_0200 | ACIF_AUDIO_CON0 | AUDIO control register 0 |
| 8301_0204 | ACIF_AUDIO_CON1 | AUDIO control register 1 |
| 8301_0208 | ACIF_AUDIO_CON2 | AUDIO control register 2 |
| 8301_0300 | ACIF_BBRX_CON | BBRX control register |
| 8301_0600 | ACIF_APC_CON0 | APC control register 0 |
| 8301_0604 | ACIF_APC_CON1 | APC control register 1 |
| 8301_0700 | ACIF_AUX_CON0 | AUX control register 0 |
| 8301_0704 | ACIF_AUX_CON1 | AUX control register 1 |

[3] CL is the parallel combination of C1 and C2 in the block diagram. It should be adjusted according to PCB design to get a preferred frequency accuracy.

## 6.2.1 Register setting path

**8301000Ch** **WR_PATH** **WR_PATH**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CCI_VSDM_PWDB | CCI_ASDM_PWDB | CCI_ESDM_PWDB | CCI_SIMLS_PWDB | CCI_APC_PWDB | CCI_AUX_PWDB | | ABB_PRST_MODE | PMU_PRST_MODE | SIM_PRST_MODE | CCI_VTX2AU_SEL | CCI_ACD_MODE | PMIC_WR_PATH | MODEM_WR_PATH | VBI_WR_PATH | ABI_WR_PATH |
| Type | RW | RW | RW | RW | RW | RW | | RW | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**WR_PATH { PMIC_WR_PATH, MODEM_WR_PATH, VBI_WR_PATH, ABI_WR_PATH }** The bit is to facilitate ACD members for verifying purpose; the hardware supports write path switching, without being disturbed by existing MCU load. However, when with manually control, all registers addresses are offset by 0x1000. For example, MCU configures **ACIF_AUDIO_CON0** through the address 0x8301_0200, while the manually control path take effect when configuring 0x8301_1200. Notice that before finishing manual control, the register must be reset to be 0. The modem part includes BBRX, BBTX, APC and AUX.

**0** switch the register setting to MCU side.

**1** switch the register setting to manually control by TRACE32 through JTAG.

**ACD_MODE** The register bit decides the input/output path of the mixed-mode module. For ABI and VBI, it can be configured to feed the pattern from AFE or from GPIO (shared with A_FUNC_MODE). For APC and AUX, the input selection interface is divided at either MIXED_DIG or GPIO (also shared with A_FUNC_MODE). As for the BBRX, the output pattern can be bypass to GPIO with this register bit being true. The bit is for convenient debug-usage in normal mode, such that the data pattern can be observed or be feed-in by external device, while control register setting still comes from the chip internally(By use of JTAG). It should be notice that this special debug mode should be accompanied by proper setting of GPIO, which decides the PAD OE when in normal function.

**0** data pattern comes from chip internally, and the output data cannot be bypassed to GPIO.

**1** analog debug mode in normal function.

**VTX2AU_SEL** The register bit select the input source to AUDIO DAC.

**0** normal path

**1** loop-back path (select VOICE TX-ADC data)

**ABIST_MODE** The register bit control the GPIO as mixedsys monitor pins, shared with ABIST_MODE.

**0** disbale

**1** enable

**AUX_PWDB** The register bit control the power-down of AUX at A_FUNC_MODE or ACD_MODE=1

**0** power-down

**1** power-up

**APC_PWDB** The register bit control the power-down of APC at A_FUNC_MODE or ACD_MODE=1

**0** power-down

**1** power-up

**ESDM_PWDB** The register bit control the power-down of BBRX at A_FUNC_MODE or ACD_MODE=1

**0** power-down

**1** power-up

**ASDM_PWDB** The register bit control the power-down of ASDM at A_FUNC_MODE or ACD_MODE=1

**0** power-down

**1** power-up

**VSDM_PWDB** The register bit control the power-down of VSDM at A_FUNC_MODE or ACD_MODE=1

**0** power-down

**1** power-up

| NAME | A_FUNC_TEST | NAME | A_FUNC_TEST | NAME | A_FUNC_TEST |
|---|---|---|---|---|---|
| PWM | A_FUNC_ABB_RESETB | EA14 | CCI_PMU_SDO | EA4 | CCI_SIM_SDO |
| KCOL4 | A_FUNC_ABB_DIN[9] | LPTE | CCI_PMU_SCLK | EA3 | CCI_SIM_SCLK |
| KCOL3 | A_FUNC_ABB_DIN[8] | SCK(L)/KROW6(E) | CCI_PMU_SFSI | EA2 | CCI_SIM_SFSI |
| KCOL2 | A_FUNC_ABB_DIN[7] | EINT1 | CCI_PMU_SDI | EA1 | CCI_SIM_SDI |
| KCOL1 | A_FUNC_ABB_DIN[6] | ED4 | A_FUNC_PMU_RESETB | EA0 | A_FUNC_SIM_RESETB |
| KCOL0 | A_FUNC_ABB_DIN[5] | EA15 | A_FUNC_PMU_CK | ED3(E)/SWP(L) | A_FUNC_SIM_DOUT[1] |
| KROW4 | A_FUNC_ABB_DIN[4] | EA13 | A_FUNC_PMU_DIN | ED2(E)/SHOLD(L) | A_FUNC_SIM_DOUT[0] |
| KROW3 | A_FUNC_ABB_DIN[3] | EA12 | A_FUNC_PMU_DOUT[7] | BPI_BUS4 | A_FUNC_SIM_DIN[6] |
| KROW2 | A_FUNC_ABB_DIN[2] | EA11 | A_FUNC_PMU_DOUT[6] | BPI_BUS3 | A_FUNC_SIM_DIN[5] |
| KROW1 | A_FUNC_ABB_DIN[1] | EA10 | A_FUNC_PMU_DOUT[5] | BPI_BUS2 | A_FUNC_SIM_DIN[4] |
| KROW0 | A_FUNC_ABB_DIN[0] | EA9 | A_FUNC_PMU_DOUT[4] | BPI_BUS1 | A_FUNC_SIM_DIN[3] |
| DAICLK | A_FUNC_ABB_CK | EA8 | A_FUNC_PMU_DOUT[3] | BPI_BUS0 | A_FUNC_SIM_DIN[2] |
| DAIPCMOUT | CCI_ABB_SDO | EA7 | A_FUNC_PMU_DOUT[2] | ED7 | A_FUNC_SIM_DIN[1] |
| DAIPCMIN | CCI_ABB_SCLK | EA6 | A_FUNC_PMU_DOUT[1] | ED6 | A_FUNC_SIM_DIN[0] |
| DAIRST | CCI_ABB_SDI | EA5 | A_FUNC_PMU_DOUT[0] | LRSTB | EXT_BBWAKEUP |

| | | | | | |
|---|---|---|---|---|---|
| DAISYNC | CCI_ABB_SFSI | | | | |
| MCINS | A_FUNC_ABB_DOUT[3] | | | | |
| MCCK | A_FUNC_ABB_DOUT[2] | | | | |
| MCDA0 | A_FUNC_ABB_DOUT[1] | | | | |
| MCCM0 | A_FUNC_ABB_DOUT[0] | | | | |
| KCOL6 | A_FUNC_ABB_DOUT[7] | | | | |
| KCOL5 | A_FUNC_ABB_DOUT[6] | | | | |
| KROW5 | A_FUNC_ABB_DOUT[5] | | | | |
| SRCLKENAI | A_FUNC_ABB_DOUT[4] | | | | |

## 6.2.2 BBRX

MCU APB bus registers for BBRX ADC are listed as followings.

**83010300h ACIF_BBRX_CON** **ACIF_BBRX_CON**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | RG_EDGERFRX_DITHERDIS | RG_EDGERFRX_QSEL | | RG_EDGERFRX_ISEL | | | RG_EDGERFRX_GAIN | | | RG_EDGERFRX_CALI | | | |
| Type | | | | RW | RW | | RW | | | RW | | | RW | | | |
| Reset | | | | 0 | 0 | | 0 | | | 0 | | | 0 | | | |

Set this register for BBRX analog circuit configuration controls.

**RG_EDGERFRX_CALI[3:0]**

The register field is for control of biasing current in BBRX mixed-signal module. Biasing current in BBRX mixed-signal module has impact on the performance of A/D conversion. The larger the value of the register field, the larger the biasing current in BBRX mixed-signal module, and the larger the SNR.

**0100** 8/4x
**0011** 7/4x
**0010** 6/4x
**0001** 5/4x
**x000** 4/4x
**1001** 4/5x
**1010** 4/6x

**1011** 4/7x

**1100** 4/8x

**RG_EDGERFRX_GAIN[1:0]**

The register bits are for configuration of gain control of analog inputs in BBRX mixed-signal module.

**00** Input range is 0.80X AVDD for analog inputs in GSM RX mixed-signal module.

**01** Input range is 0.40X AVDD for analog inputs in GSM RX mixed-signal module.

**10** Input range is 0.57X AVDD for analog inputs in GSM RX mixed-signal module.

**11** Input range is 0.33X AVDD for analog inputs in GSM RX mixed-signal module.

**RG_EDGERFRX_ISEL[1:0]**

Input signal selection for I-channel in BBRX mixed-signal module.

**00** Select QS_BDLAIP / QS_BDLAIN to ICH ADC

**01** Select QS_BDLAIP / QS_BDLAIN to ICH ADC

**10** Select QS_BDLAQP / QS_BDLAQN to ICH ADC

**11** Select VCM to ICH ADC for self-test

**RG_EDGERFRX_QSEL[1:0]**

Input signal selection for Q-channel in BBRX mixed-signal module.

**00** Select QS_BDLAQP / QS_BDLAQN to QCH ADC

**01** Select QS_BDLAQP / QS_BDLAQN to QCH ADC

**10** Select QS_BDLAIP / QS_BDLAIN to QCH ADC

**11** Select VCM to QCH ADC for self-test

**RG_EDGERFRX_DITHERDIS**

Dither feature disable control register, which can effectively reduce the THD (total harmonic distortion) of the BBRX ADC.

**0** enable (default value)

**1** disable

## 6.2.3 APC DAC

MCU APB bus registers for APC DAC are listed as followings.

**83010600h** **ACIF_APC_CON0** **ACIF_APC_CON0**

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

Set this register for APC analog circuit configuration controls.

**RG_APC_TGSEL** APC_TG trigger edge select.

**0** Trigger the signal at rising edge.

**1** Trigger the signal at falling edge.

**83010604h** **ACIF_APC_CON1** **ACIF_APC_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | MIXEDSYS_APC_SETEN | | | | | MIXEDSYS_APC_SETTG | MIXEDSYS_APC_SETBUS | | | | | | | | | |
| Type | RW | | | | | RW | RW | | | | | | | | | |
| Reset | 0 | | | | | 0 | 0 | | | | | | | | | |

Set this register for APC analog circuit manual controls.

MIXEDSYS_APC_SETEN manual APC set enable

MIXEDSYS_APC_SETTG manual APC set toggle

MIXEDSYS_APC_SETBUS manual APC 10bits input

## 6.2.4 Auxiliary ADC

MCU APB bus registers for AUX ADC are listed as followings.

**83010700h** **ACIF_AUX_CON0** **ACIF_AUX_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | | | | | RG_TESTMODE_PLL | RG_AUX_CALI | |
| Type | | | | | | | | | | | | | | RW | RW | |
| Reset | | | | | | | | | | | | | | 0 | 01 | |

Set this register for AUX analog circuit configuration controls.

**RG_AUX_CALI** AUX ADC biasing current control

**00** 0.5x

**01** 1.0x

**10** 1.5x

**11** 2.0x

RG_TESTMODE_PLL pll testmode control

0: normal

1: AUX_IN4/AUX_IN5 channels can not be selected

## 6.2.5 Voice Front-end

MCU APB bus registers for speech are listed as followings.

**83010100h ACIF_VOICE_CON0 ACIF_VOICE_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RGS_ZCD_STATE (R) | | CCI_VUPG | | | | | | RG_VDPG | | | | RG_VBIRX_ZCD_HYS_EN | RG_VBIRX_ZCD_EN | RG_VBIRX_ZCD_CALI | |
| Type | R | | RW | | | | | | RW | | | | RW | RW | RW | |
| Reset | 0 | | 0 | | | | | | 0 | | | | 0 | 0 | 0 | |

Set this register for VOICE PGA gains analog circuit. CCI_VUPG is set for microphone input volume control. And RG_VDPG is set for output volume control.

**RG_ZCD_STATE** Indicates the ZCD current state,

0: ZCD is available for gain modification

1: ZCD is currently unavailable for gain modification

**RG_VDPG** Voice-band down-link PGA gain control bits.

| VDPG [3:0] | Gain | VDPG [3:0] | Gain | VDPG [3:0] | Gain | VDPG [3:0] | Gain |
|---|---|---|---|---|---|---|---|
| 1111 | 8dB | 1011 | 0dB | 0111 | -8dB | 0011 | -16dB |
| 1110 | 6dB | 1010 | -2dB | 0110 | -10dB | 0010 | N/A |
| 1101 | 4dB | 1001 | -4dB | 0101 | -12dB | 0001 | N/A |
| 1100 | 2dB | 1000 | -6dB | 0100 | -14dB | 0000 | N/A |

**CCI_VUPG** Voice-band up-link PGA gain control bits. For VCFG[3] = 1, it is only valid for INPUT 1.VIN0/VIN1 input path

| VUPG [5:0] | Gain | VUPG [5:0] | Gain | VUPG [5:0] | Gain | VUPG [5:0] | Gain |
|---|---|---|---|---|---|---|---|
| 111111 | 49 dB | 101111 | 33 dB | 011111 | 17 dB | 001111 | 1 dB |
| 111110 | 48 dB | 101110 | 32 dB | 011110 | 16 dB | 001110 | N/A- |
| 111101 | 47 dB | 101101 | 31 dB | 011101 | 15 dB | 001101 | N/A- |
| 111100 | 46 dB | 101100 | 30 dB | 011100 | 14 dB | 001100 | N/A- |
| 111011 | 45 dB | 101011 | 29 dB | 011011 | 13 dB | 001011 | N/A- |
| 111010 | 44 dB | 101010 | 28 dB | 011010 | 12 dB | 001010 | N/A- |
| 111001 | 43 dB | 101001 | 27 dB | 011001 | 11 dB | 001001 | N/A- |
| 111000 | 42 dB | 101000 | 26 dB | 011000 | 10 dB | 001000 | N/A- |
| 110111 | 41 dB | 100111 | 25 dB | 010111 | 9 dB | 000111 | N/A- |
| 110110 | 40 dB | 100110 | 24 dB | 010110 | 8 dB | 000110 | N/A- |
| 110101 | 39 dB | 100101 | 23 dB | 010101 | 7 dB | 000101 | N/A- |
| 110100 | 38 dB | 100100 | 22 dB | 010100 | 6 dB | 000100 | N/A- |
| 110011 | 37 dB | 100011 | 21 dB | 010011 | 5 dB | 000011 | N/A- |
| 110010 | 36 dB | 100010 | 20 dB | 010010 | 4 dB | 000010 | N/A- |
| 110001 | 35 dB | 100001 | 19 dB | 010001 | 3 dB | 000001 | N/A- |
| 110000 | 34 dB | 100000 | 18 dB | 010000 | 2 dB | 000000 | N/A- |
| FM input path | | | | | | | |
| VUPG [5:0] | Gain | VUPG [5:0] | Gain | VUPG [5:0] | Gain | VUPG [5:0] | Gain |
| 111111 | 22 dB | 101111 | 6 dB | 011111 | -10 dB | 001111 | -26 dB |
| 111110 | 21 dB | 101110 | 5 dB | 011110 | -11 dB | 001110 | N/A- |
| 111101 | 20 dB | 101101 | 4 dB | 011101 | -12 dB | 001101 | N/A- |
| 111100 | 19 dB | 101100 | 3 dB | 011100 | -13 dB | 001100 | N/A- |
| 111011 | 18 dB | 101011 | 2 dB | 011011 | -14 dB | 001011 | N/A- |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 111010 | 17 dB | 101010 | 1 dB | 011010 | -15 dB | 001010 | N/A- |
| 111001 | 16 dB | 101001 | 0 dB | 011001 | -16 dB | 001001 | N/A- |
| 111000 | 15 dB | 101000 | -1 dB | 011000 | -17 dB | 001000 | N/A- |
| 110111 | 14 dB | 100111 | -2 dB | 010111 | -18 dB | 000111 | N/A- |
| 110110 | 13 dB | 100110 | -3 dB | 010110 | -19 dB | 000110 | N/A- |
| 110101 | 12 dB | 100101 | -4 dB | 010101 | -20 dB | 000101 | N/A- |
| 110100 | 11 dB | 100100 | -5 dB | 010100 | -21 dB | 000100 | N/A- |
| 110011 | 10 dB | 100011 | -6 dB | 010011 | -22 dB | 000011 | N/A- |
| 110010 | 9 dB | 100010 | -7 dB | 010010 | -23 dB | 000010 | N/A- |
| 110001 | 8 dB | 100001 | -8 dB | 010001 | -24 dB | 000001 | N/A- |
| 110000 | 7 dB | 100000 | -9 dB | 010000 | -25 dB | 000000 | N/A- |

RG_VBIRX_ZCD_HYS_EN Voice buffer zero-detection hysteresis enable

0 : Disable
1 : Enable

RG_VBIRX_ZCD_EN Enable VBIRX Zero Detection Function

0 : Disable
1 : Enable

RG_VBIRX_ZCD_CALI Trim the hysterisis of ZDTC

00 : 13mV
01 : 26mV
10 : 40mV
11 : 56mV

**83010104h ACIF_VOICE_CON1** **ACIF_VOICE_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | RG_VMIC_VREF | | RG_LOWGAIN_EN | RG_VCFG | | | | | | RG_GLB_CALI | | RG_VCALI | | |
| Type | | | RW | | RW | RW | | | | | | RW | | RW | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Reset | | | 0 | 0 | 0 | | | 0 | 0 |

Set this register for VOICE analog circuit configuration controls.

RG_VMIC_VREF Tuning the dc level of micbias

00: 1.9V 01: 2.0V 10: 2.1V 11: 2.2V

RG_LOWGAIN_EN lower PGA gain 6dB

0: normal

1: enable additional -6dB (based on VUPG gain setting)

RG_VCFG Set PGA's input selection, AC/DC coupled and GAIN/ATT mode

| VCFG [3:0] | Configuration |
|---|---|
| 0000 | Select AU_VIN0 as input (Normal mode) |
| 0001 | Select AU_VIN1 as input (Normal mode) |
| 001X | Select AU_FMR/L as input (Normal mode) |
| 1001 | PGA bypass mode and ADC is AC couple, Input channel is select as AU_VIN1 (Test mode) |
| 1101 | PGA bypass mode and ADC is DC couple, Input channel is select as AU_VIN1 (Test mode) |

RG_GLB_CALI Set global bias current ratio

00: 1X
01: 1.2X
10: 0.67X
11: 0.8X

RG_VCALI Set VBI bias current ratio

000: X1 (represents nominal 1.75uA current bias)
001: X4/5
010: X4/6
011: X4/7
100: X5/4
101: X6/4
110: X7/4
111: X8/4

**83010108h ACIF_VOICE_CON2** **ACIF_VOICE_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | RG_VBUF_BIAS | | | | | | | RG_VBUF_FLOAT | | | RG_VADC_DVREF_CAL | RG_VADC_DENB | | RG_VADCINMODE | |
| Type | | RW | | | | | | | RW | | | RW | RW | | RW | |
| Reset | | 0 | | | | | | | 0 | | | 0 | 0 | | 0 | |

RG_VBUF_BIAS Voice BUF bias control

00: 3X
01: 4X
10: 1X
11: 2X

RG_VBUF_FLOAT VBI_DACBUF Output floating during power down

0: Output shorted during power down

1: Output floating during power down

RG_VADC_DVREF_CAL ADC Dither Reference Voltage Calibration

0: 2/15*AVDD
1: 3/15*AVDD

RG_VADC_DENB ADC Dither Enable

0: Enable
1: Disable

RG_VADCINMODE ADC input source selection

0 : from LNA

1: from DAC (internal test path)

**8301010Ch ACIF_VOICE_CON3** **ACIF_VOICE_CON3**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RG_VDEPOP | | | | | | | | RG_VREF24_EN | | RG_VBIAS_PWDB | RG_VLNA_PWDB | RG_VADC_PWDB | | | RG_VBUF_PWDB |
| Type | RW | | | | | | | | RW | | RW | RW | RW | | | RW |
| Reset | 0 | | | | | | | | 0 | | 0 | 0 | 0 | | | 0 |

RG_VDEPOP Anti-pop for buffer output enable

0:Disable
1:Enable

RG_VREF24_EN Enable 2.4V reference buffer

0: Disable
1: Enable

RG_VBIAS_PWDB Power-down bias

0: Power down
1: Active

RG_VLNA_PWDB Power down PGA and AGC

0: Power down
1: Active

RG_VADC_PWDB power down ADMOD

0: Power down
1: Active

RG_VBUF_PWDB Voice BUF Power down

0:Power down
1:Active

**83010110h ACIF_VOICE_CON4** **ACIF_VOICE_CON4**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | RG_VREF24_CON | | RG_VCM14_PWD | RG_RESV | | | |
| Type | | | | | | | | | | RW | | RW | RW | | | |
| Reset | | | | | | | | | | 1 | | 0 | 1100 | | | |

RG_VREF24_CON Reference Voltage Control (Necessary)

00 : 2.40V
01 : 2.27V
10 : 2.15V
11 : 2.03V

RG_VCM14_PWD Enable 1.4V common mode voltage (Necessary)

0: Enable
1: Disable

RG_RESV Reserved

## 6.2.6 Audio Front-end

MCU APB bus registers for audio are listed as followings.

**83010200h ACIF_AUDIO_CON0** **ACIF_AUDIO_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RG_DACREF | | | | | RG_ASDM_CK_INV | RG_AMUTER | RG_AMUTEL | RG_APGR | | | | RG_APGL | | | |
| Type | RW | | | | | RW | RW | RW | RW | | | | RW | | | |
| Reset | 0 | | | | | 0 | 0 | 0 | 111 | | | | 111 | | | |

RG_DACREF Select DAC reference as

0: VDD/GND (typically 2.8V)

1: REFP/REFN (Level is set by RG_VREF24_CON)

RG_ASDM_CK_INV Select ASDM clock inverse. 0: default phase; 1: inverse phase

RG_AMUTER Mute right channel

0: Normal

1: Mute

RG_AMUTEL Mute left channel

0: Normal

1: Mute

RG_APGR Audio right channel amplifier gain control

RG_APGL Audio left channel amplifier gain control

| APGR/L [3:0] | Gain | APGR/L [3:0] | Gain | APGR/L [3:0] | Gain | APGR/L [3:0] | Gain |
|---|---|---|---|---|---|---|---|
| 1111 | N/A | 1011 | 13dB | 0111 | -1dB | 0011 | -13dB |
| 1110 | N/A | 1010 | 8dB | 0110 | -4dB | 0010 | -16dB |
| 1101 | 17dB | 1001 | 5dB | 0101 | -7dB | 0001 | -19dB |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1100 | 14dB | 1000 | 2dB | 0100 | -10dB | 0000 | -22dB |

**83010204h** **ACIF_AUDIO_CON1** **ACIF_AUDIO_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | RG_ADEPOP | | RG_ABUFSELR | | | RG_ABUFSELL | | | RG_ACALI | | | | |
| Type | | | | RW | | RW | | | RW | | | RW | | | | |
| Reset | | | | 1 | | 0 | | | 0 | | | 0 | | | | |

Set this register for AUDIO analog circuit configuration controls.

**ACALI** Audio bias current control, in 2's complement format.

**00000** X1
**00001** X5/4
**00010** X6/4
**00011** X7/4
**00111** X8/4
**10000** X1
**10001** X4/5
**10010** X4/6
**10011** X4/7
**10111** X4/8
**OTHERS** Prohibited
**BIT3** Reserved

**ABUFSELL** Audio buffer L-channel input selection.

**00X** audio DAC L-channel output
**100** external FM R/L-channel radio output, stereo to mono
**101** external FM L-channel radio output
**others** reserved

**ABUFSELR** Audio buffer R-channel input selection.

**00X** audio DAC R-channel output
**100** external FM R/L-channel radio output, stereo to mono
**101** external FM R-channel radio output
**others** reserved

**ADEPOP** Audio de-pop noise control bit

**0** disable
**1** enable

83010208h ACIF_AUDIO_CON2 ACIF_AUDIO_CON2

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | RG_ADACCK_EN | RG_ABUFHALFV_EN | RG_ADEPOPX_EN | | | | | | RG_ABIAS_PWDB | RG_ADACR_PWDB | RG_ADACL_PWDB | RG_AOUTR_PWDB | RG_AOUTL_PWDB |
| Type | | | | RW | RW | RW | | | | | | RW | RW | RW | RW | RW |
| Reset | | | | 1 | 0 | 0 | | | | | | 0 | 0 | 0 | 0 | 0 |

Set this register for AUDIO analog circuit configuration controls.

RG_ADACCK_EN Gating the DAC clock (Necessary)0: Clock gated
1: Clock pass

RG_ABUFHALFV_EN Enable HalfV 1.4V common mode voltage (Necessary)
0: Disable
1: Enable

RG_ADEPOPX_EN Power on depop removal circuit
0: Disable
1: Enable

**AOUTL_PWDB** Power-down AUDIO L-channel output buffer.
**0** power-down
**1** power-up

**AOUTR_PWDB** Power-down AUDIO R-channel output buffer.
**0** power-down
**1** power-up

**ADACL_PWDB** Power-down AUDIO L-channel DAC.
**0** power-down
**1** power-up

**ADACR_PWDB** Power-down AUDIO R-channel DAC.
**0** power-down
**1** power-up

**ABIAS_PWDB** Power-down AUDIO bias current circuit.
**0** power-down
**1** power-up

8301020Ch ACIF_AUDIO_CON3 ACIF_AUDIO_CON3

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | RG_ABUF | SPARE_A | RG_A | RG_ABU | | SPARE_ | RG_AMUX | | RG_ADEPO | | |

| | | | | | | _INSHORT | UDIO_CON3_9_ | HFMODE | F_BIAS | AUDIO_CON3_5_ | | PX | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | RW | RW | RW | RW | RW | RW | RW | |
| Reset | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 10 | |

Set this register for AUDIO analog circuit configuration controls.

**AMUX** Mux audio DAC output to FM R/L pins.
- **00** FM input
- **01** FM input
- **10** L-channel DAC Output
- **11** R-channel DAC Output

**ABUF_BIAS** Audio buffer quasi-current select bits.
- **00** 1.0x (default)
- **01** 1.33x
- **10** 0.33x
- **11** 0.67x

**AHFMODE** Audio hand-free mode enable bit.
- **0** normal mode
- **1** hand-free mode

**ABUF_INSHORT** Audio buffer input short enable (during voice mode)
- **0** disable
- **1** enable

RG_ADEPOPX Power on depop removal circuit resistor calibration
- 00: 8k
- 01: 4k
- 10: 2k
- 11: 1k

## 6.2.7 Power Management Control

Power management unit, so called PMU, is integrated into analog part. To facilitate software control and interface design, PMU control share the CCI interface along with other analog parts, such as BBRX, VBI and ABI.

### 6.2.7.1 Low Dropout Regulators ( LDOs )

The PMU Integrates 13 LDOs that are optimized for their given functions by balancing quiescent current, dropout voltage, line/load regulation, ripple rejection, and output noise.

### RF LDO (Vrf)

The RF LDO is a linear regulator that could source 150mA (max) with 2.8V output voltage. It supplies the RF circuitry of the handset. The LDO is optimized for high performance and adequate quiescent current.

### Digital Core LDO (Vcore)

The digital core regulator is a linear regulator that could source 200mA (max) with 0.8V to 1.35V (25mv/step ) output voltage. It supplies the power for baseband circuitry of the SoC. The LDO is optimized for very low quiescent current.

### Digital IO LDO (Vio)

The digital IO LDO is a linear regulator that could source 200mA (max) with 2.8V output voltage. It supplies the power for baseband circuitry of the SoC. The LDO is optimized for very low quiescent current and turns on automatically together with Va LDOs.

### Analog LDO (Va)

The analog LDO is a linear regulator that could source 100mA (max) with 2.8V output voltage. It supplies the analog sections of the SoC. The LDO is optimized for low frequency ripple rejection in order to reject the ripple coming from the burst at 217Hz of RF power amplifier.

### TCXO LDO (Vtcxo)

The TCXO LDO is a linear regulator that could source 40mA (max) with 2.8V output voltage. It supplies the temperature compensated crystal oscillator, which needs ultra low noise supply with very good ripple rejection.

### Single-Step RTC LDO (Vrtc)

The single-step RTC LDO is a linear regulator that can charge up a capacitor-type backup coin cell to 2.8V, and also supply the RTC module even at the absence of the main battery. The single-step LDO features the reverse current protection and is optimized for ultra low quiescent current while sustaining the RTC function as long as possible.

### Memory LDO (Vm)

The memory LDO is a linear regulator that could source 150mA (max) with 1.8V or 2.9V output voltage selection based on the supply specification of memory chips. It supplies the memory circuitry in the handset. The LDO is optimized for very low quiescent current with wide output loading range.

### SIM LDO (Vsim)

The SIM LDO is a linear regulator that could source 30mA (max) with 1.8V or 3.0V output voltage selection based on the supply specs of subscriber identity modules (SIM) card. It supplies the SIM card and SIM level shifter circuitry in the handset. The Vsim LDO is controlled independently by the register named RG_VSIM1_EN.

### SIM2 LDO (Vsim2)

The SIM2 LDO is a linear regulator that could source 30mA (max) with 1.3V, 1.5V, 1.8V, 2.5V, 2.8V, 3.0V or 3.3V output voltage selection based on the supply specs of the 2nd subscriber identity modules (SIM) card. It supplies the 2nd SIM card and SIM level shifter circuitry in the handset. The Vsim2 LDO is controlled independently by the register named RG_VSIM2_EN.

### USB LDO (Vusb)

The USB LDO is a linear regulator that could source 50mA (max) with 3.3V output dedicated for USB circuitry.

### Camera Analog LDO (Vcama)

The Vcama LDO is a linear regulator that could source 150mA (max) with 1.5V, 1.8V, 2.5V or 2.8V output which is selected by the register named RG_VCAMA_VOSEL[1:0]. It supplies the analog power of the camera module. Vcama is controlled independently by the register named RG_VCAMA_EN.

### Camera Digital LDO (Vcamd)

The Vcamd LDO is a linear regulator that could source 100mA (max) with 1.3V, 1.5V, 1.8V, 2.5V, 2.8V, 3.0V or 3.3V output which is selected by the register named RG_VCAMD_VOSEL[2:0]. It supplies the digital power of the camera module. Vcamd is controlled independently by the register named RG_VCAMD_EN.

### VIBR LDO (Vibr)

The Vibr LDO is a linear regulator that could source 150mA (max) with 1.3V, 1.5V, 1.8V, 2.5V, 2.8V, 3.0V or 3.3V output voltage which is selected by the register named RG_VIBR _VOSEL[2:0]. It supplies the power of the vibrator module. Vibr is controlled independently by the register named RG_VIBR_EN.

## 6.2.7.2 SIM Card Interface

There are two SIM card interface modules to support two SIM cards simultaneously. The SIM card interface circuitry of PMU meets all ETSI and IMT-2000 SIM interface requirements. It provides level shifting needs for low voltage GSM controller to communicate with either 1.8V or 3V SIM cards. All SIM cards contain a clock input, a reset input, and a bi-directional data input/output. The clock and reset inputs to SIM cards are level shifted from the supply of digital

CORE (Vcore) of baseband to the SIM supply (Vsim). The bi-directional data bus is internal pull high to Vsim via 5KΩ resistor.

The 2nd SIM card interface can be used for supporting another SIM card or mobile TV. The interface pins such as SIO2, SRST2, SCLK2, can be configured as GPIO when there is no need to use the 2nd SIM card interface.

All pins that connect to the SIM card (Vsim, SRST, SCLK, SIO) withstand over 2kV HBM (human body mode) ESD. In order to ensure proper ESD protection, careful board layout is required.

#### 6.2.7.3 Keypad LED/Current Sink Drivers

Built-in open-drain output switches drive the Keypad LED in the handset. This driver is controlled by baseband with enable registers (RG_KPLED_EN), and the output is high impedance when disabled. It can sink 60mA. Four current controlled open drain drivers (Isink0~3) are also implemented to drive LCM backlight module and each provides 6 current level step up to 24mA..

#### 6.2.7.4 Speaker Amplifier

The speaker amplifier is a class AB audio amplifier which is a highly integrated design with built-in output stages. The audio amplifier can provide good linearity and low EMI, also help to reduce the component cost. It has programmable gain setting from 0dB to 22.5dB with 1.5dB per step. Current limit circuit is also implemented to detect over current and shut down automatically so as to prevent devastation.

#### 6.2.7.5 Power-on Sequence and Protection Logic

The PMU handles the powering ON and OFF of the handset. There are three ways to power-on the handset system :

1. Push PWRKEY (Pull the PWRKEY pin to the low level)

Pulling PWRKEY low is the typical way to turn on the handset. The Vcore LDO will be turned-on first, and then Vm and followed by Va/Vio LDOs turn-on in turn. The supplies for the baseband are ready and then the system reset ends at the moment when the Vcore/Va/Vio/Vm are fully turned-on to ensure the correct timing and function. After that, baseband would send the PWRBB signal back to PMU for acknowledgement. To successfully power-on the handset, PWRKEY should be kept low until PMU receives the PWRBB from baseband.

2. RTC module generate PWRBB to wakeup the system

If the RTC module is scheduled to wakeup the handset at some time, the PWRBB signal will directly send to the PMU. In this case, PWRBB becomes high at the specific moment and let PMU power-on just like the sequence described above. This is the case named RTC alarm.

3. Valid charger plug-in (CHRIN voltage is within the valid range)

Charger plugging-in will also turn on the handset if the charger is valid (no OVP take place). However, if the battery voltage is too low to power-on the handset (UVLO state), the system won't be turned-on by any of these three ways. In this case, charger will charge the battery first and the handset will be powered-on automatically as long as the battery voltage is high enough.

| Phone State | CHRON | UVLO | PWRKEY && (~PWRBB) | Vrtc | Vcore, Vio, Vm, Va | Vtcxo, Vrf |
|---|---|---|---|---|---|---|
| No Battery or Vbat < 2.5V | X | H | X | Off | Off | Off |
| 2.5V < Vbat < 3.2V | L | H | X | On | Off | Off |
| Pre-Charging | H | H | X | On | Off | Off |
| Charger-on (Vbat>3.2V) | H | L | X | On | On | On |
| Switched off | L | L | H | On | Off | Off |
| Stand-by | L | L | L | On | On | Off |
| Active | L | L | L | On | On | On |

Table 13 States of mobile handset and regulator

### 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 can ensure the smooth power-on sequence. In addition, when the battery voltage is getting lower and lower, it will enter UVLO state and the PMU will be turned-off by itself, except for Vrtc LDO, to prevent further discharging. Once the PMU enters UVLO state, it draws low quiescent current. The RTC LDO is still working until the DDLO disables it.

### Deep Discharge Lockout (DDLO)

PMU will enter to the deep discharge lockout (DDLO) state when the battery voltage drops below 2.5V. In this state, the Vrtc LDO will be shutdown. Otherwise, it draws very low quiescent current to prevent further discharging or even 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 at 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 use clock from internal ring-oscillator. At power-off, **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 the Vrtc. Once the over-temperature state is resolved, a new power on sequence is required to enable the LDOs.

### 6.2.7.6 Battery Charger

The battery charger is optimized for the Li-ion batteries. The typical charging procedure can be divided into three phases: pre-charging mode, the constant current charging mode, and the full voltage charging mode. Figure 2 shows the flow chart of the charging procedure. Most of the charger circuits are integrated in the PMU except for one PNP, NMOS and one accurate resistor for current sensing. These components should be applied externally.

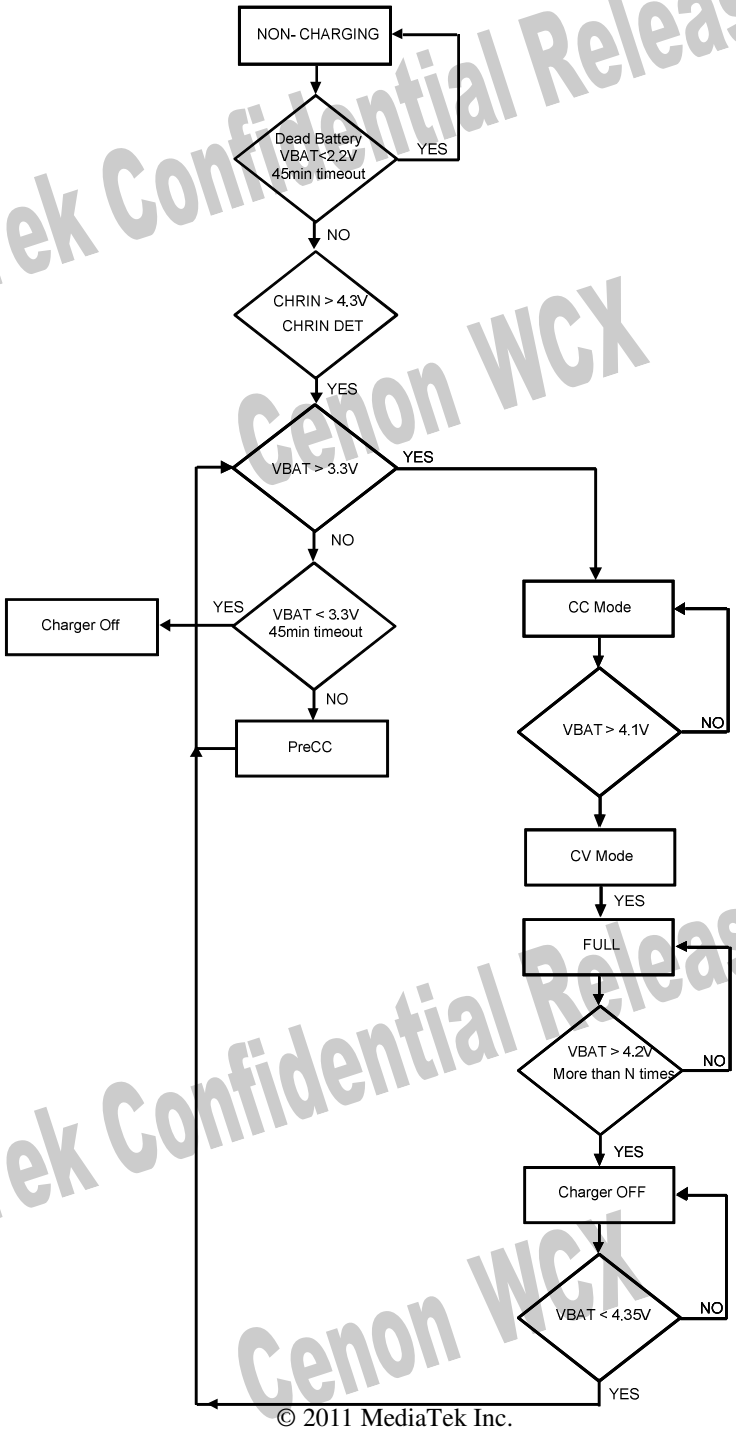


**Figure 2. Battery Charger Flow Chart**

## 1. Charge Detection

The PMU charger block has a detection circuit that senses the charger plug-in/out and provides the correct information to the baseband. If CHRIN is over 4.3V, charger detection will be report to baseband and charger circuit will be enabled. If the CHRIN voltage is over 7V, charger will send an invalid signal to baseband for further indication. The stop of charging when CHRIN is over 7V could be achieved by external component.

## 2. Pre-Charging mode

When the battery voltage is below the CC threshold, the charging status is in the pre-charging mode. There are two steps in this mode. While the battery voltage is deeply discharged below 2.7V, a 50mA trickle current is used for charging the battery. This is the pre-CC1 state. When the battery voltage exceeds 2.7V, the self-calibrated pre-charge mode is enabled, which allows 20mV (typically) voltage drop across the external current sense resistor. This is the pre-CC2 state. The pre-charge current in this state can be calculated as:

$$I_{PRE\_CC} = \frac{V_{SENSE}}{R_{SENSE}} = \frac{20mV}{R_{SENSE}} \qquad (2)$$

Typically, $I_{CONST}$=100mA with $V_{SENSE}$=20mV and $R_{SEN}$=0.2Ω.

## 3. Constant Current Charging Mode

Once the battery voltage has exceeded the CC threshold, a constant current is used for periodical charging. With periodical charging, charger circuit could detect CHRIN state and battery state in non-charging period. This is called the constant current charging mode. An up-to-800mA constant charging current could be programmed via the register setting. The relation between the voltage drop across the external current sense resistor and the charging current is as follows,

$$I_{CONSTANT} = \frac{V_{SENSE}}{R_{SENSE}} \qquad (3)$$

Typically, $I_{CONST}$=800mA with $V_{SENSE}$=160mV and $R_{SEN}$=0.2Ω.

Before the battery voltage reaches 4.1V, the charger will be in the constant current charging mode.

## 4. Full/Constant Voltage Charging Mode

While the battery voltage reaches 4.1V, a constant current with much shorter period is used for charging. It could allow more often full battery detection in non charging period. This is called full voltage charging mode or constant voltage charging mode in correspondence to a linear charger. While the battery voltage reaches 4.2V more than the pre-setting times within the limited charging cycles, the end-of-charging process starts. It may prolong the charging and detecting

period for getting the optimized the full charging volume. This end of charging process is fully controlled by the baseband and could be easily optimized for different battery pack.

#### 5. Over-Voltage Protection

Once the battery voltage exceeds 4.35V, a hardware over voltage protection (OV) should be activated and turn off the charger immediately.

#### 6. Watchdog Timer

An internal watchdog timer is used as a protection for charging period control. In the constant current charging mode or the full voltage charging mode, the baseband must refresh the timer periodically to keep the charging alive. Once, the watchdog timer out, charger will stop charging. This provides the time domain protection for charging control.

#### 7. CSDAC

CSDAC is an 8-bit current DAC for current sink. Typically, the step for 1 LSB current sink is 55µA. Hence, the controlled charging current could be calculated as $I_{CHR} = \beta$ x 55µA x CSDAC_DATA[7:0].

#### 8. Current Sense

A current sense circuit measures the voltage difference between VSEN and VBAT, which could be used as a feedback signal for CSDAC driving control.

### 6.2.7.7 External Components Selection

#### Input Capacitor Selection

For each of input pins (VBAT) of PMU, a local bypass capacitor is recommended. Use a 10µF, low ESR capacitor. MLCC capacitors provide the best combination of low ESR and small size. Using a 10µF Tantalum capacitor with a small (1µF or 2.2µF) ceramic in parallel is an alternative low cost solution.

For charger input pin (CHRIN), a bypass 1µF ceramic capacitor is recommended.

#### LDO Capacitor Selection

The VCAMA, VRF, VIO LDOs require a 2.2µF capacitor, and the other LDOs require a 1µF capacitor. Large value capacitor may be used for desired noise or PSRR requirement. But the acceptable settling time should be taken into consideration. The MLCC X5R type capacitors must be used with VRF, VTCXO, VCAM_A and VA LDOs for good system performance. For other LDOs, MLCC X5R type capacitors are also recommended to use.

#### Setting the Charge Current

PMU is capable of charging battery. The charging current is controlled with an external sense resistor, *Rsense*. It is calculated as the Eq.(3). If the charge current is pre-defined, Rsense can be determined.

Accurate sense resistors are available from the following vendors: Vishay Dale, IRC, Panasonic.

### Charger FET Selection

The PMOS FET selection used in charger should consider the minimum drain-source breakdown voltage (BVDS), the minimum turn-on threshold voltage (VGS), and heat-dissipating ability.

These specifications can be calculated as below:

$V_{GS} = V_{CHRIN} - V_{GATEDRV}$

$V_{DS} = V_{CHRIN} - V_{DIODE} - V_{SENSE} - VBAT$

$$R_{DS(ON)} = \frac{V_{DS}}{I_{CHR}}$$

$P_{DISS} = ( V_{CHRIN}-V_{DIODE}-V_{SENSE}-VBAT) \times I_{CHR}$

Appropriate PMOS FETs are available from the following vendors: Siliconix, IR, Fairchild.

### Charger Diode Selection

The diode is used to prevent the battery from discharging through the PMOS's body diode into the charger's internal circuits. Choose a diode with sufficient current rating to handle the battery charging current and voltage rating greater than Vbat.

### Layout Guideline

Use the general guidelines listed below to design the printed circuit boards:

1. Split battery connection to the AVDD43_SPK, VBAT_ANALOG, VBAT_RF and VBAT_DIGITAL pins for PMU. Place the input capacitor as close to the power pins as possible.
2. Va and Vtcxo capacitors should be returned to AGND. Vrf capacitor should be returned to AGND_RF.
3. Split the ground connection. Use separate traces or planes for the analog, digital, and power grounds (i.e. AGND, pin of PMU, respectively) and tie them together at a single point, preferably close to battery return.
4. Place a separate trace from the BATSNS pin to the battery input to prevent voltage drop error when sensing the battery voltage.
5. Kelvin-connect the charge current sense resistor by placing separate traces to the BATSNS and ISENSE pins. Make sure that the traces are terminated as close to the resistor's body as possible.
6. Careful use of copper area, weight, and multi-layer construction will help to improve thermal performance.

### 6.2.7.8 Functional Specification

#### 6.2.7.8.1 Electrical Characteristics

VBAT = 3 V ~ 4.3 V, minimum loads applied on all outputs, unless other noted. Typical values are at $T_A$ = 25 °C.

| Parameter | Conditions | Min. | Typical | Max. | Unit |
|---|---|---|---|---|---|
| **Switch-Off Mode: Supply Current** | | | | | |
| VBAT < 2.5 V | RTC LDO OFF | | TBD | | µA |
| 2.5 V < VBAT < 3.3 V | VBAT=3.3V | | TBD | | µA |
| 3.3 V < VBAT | VBAT=4.2V | | TBD | | µA |
| **Operation: Supply Current** | | | | | |
| All outputs on | VBAT=4.2V | | TBD | | µA |
| VSIM, VSIM2, VTXCO, VRF, VUSB, VCAM_A, VCAM_D, VBT off; all others on | VBAT=4.2V | | TBD | | µA |
| **Under Voltage (UV)** | | | | | |
| Under voltage falling threshold 1 | UV_SEL[1:0] = 00 | 2.85 | 2.9 | 2.95 | V |
| Under voltage falling threshold 2 | UV_SEL[1:0] = 01 | 2.7 | 2.75 | 2.8 | V |
| Under voltage falling threshold 3 | UV_SEL[1:0] = 10 | 2.55 | 2.6 | 2.65 | V |
| Under voltage falling threshold 4 | UV_SEL[1:0] = 11 | 2.35 | 2.5 | 2.65 | V |
| Under voltage rising threshold | UV_SEL[1:0] = xx | 3.1 | 3.2 | 3.3 | V |
| **Reset Generator** | | | | | |
| Output High | | $V_{IO}$-0.5 | | | V |
| Output Low | | | | 0.2 | V |
| Output Current | | | TBD | | mA |
| **Power Key Input/VMSEL Input** | | | | | |
| High Voltage | | 0.7*VBAT | | | V |
| Low Voltage | | | | 0.3*VBAT | V |
| **Thermal Shutdown** | | | | | |
| Threshold | | | 150 | | degree |

| | | | | | |
|---|---|---|---|---|---|
| Hysteresis | | | 40 | | degree |
| **LDO Enable Response Time** | | | 250 | | µs |

#### 6.2.7.8.2 Regulator Output

| Parameter | Conditions | Min. | Typical | Max. | Unit |
|---|---|---|---|---|---|
| **Digital Core Voltage** | | | | | |
| Output voltage (V_D) | Register VOSEL=00000 | 1.1 | 1.2 | 1.3 | V |
| | Register VOSEL=00001 | 1.125 | 1.225 | 1.325 | |
| | Register VOSEL=00010 | 1.15 | 1.25 | 1.35 | |
| | Register VOSEL=00011 | 1.175 | 1.275 | 1.375 | |
| | Register VOSEL=00100 | 1.2 | 1.3 | 1.4 | |
| | Register VOSEL=00101 | 1.225 | 1.325 | 1.425 | |
| | Register VOSEL=00110 | 1.25 | 1.35 | 1.45 | |
| | Register VOSEL=10000 | 0.7 | 0.8 | 0.9 | |
| | Register VOSEL=10001 | 0.725 | 0.825 | 0.925 | |
| | Register VOSEL=10010 | 0.75 | 0.85 | 0.95 | |
| | Register VOSEL=10011 | 0.775 | 0.875 | 0.975 | |
| | Register VOSEL=10100 | 0.8 | 0.9 | 1.0 | |
| | Register VOSEL=10101 | 0.825 | 0.925 | 1.025 | |
| | Register VOSEL=10110 | 0.85 | 0.95 | 1.05 | |
| | Register VOSEL=10111 | 0.875 | 0.975 | 1.075 | |
| | Register VOSEL=11000 | 0.9 | 1 | 1.1 | |
| | Register VOSEL=11001 | 0.925 | 1.025 | 1.125 | |
| | Register VOSEL=11010 | 0.95 | 1.05 | 1.15 | |
| | Register VOSEL=11011 | 0.975 | 1.075 | 1.175 | |
| | Register VOSEL=11100 | 1.0 | 1.1 | 1.2 | |
| | Register VOSEL=11101 | 1.025 | 1.125 | 1.225 | |

| | | | | | |
|---|---|---|---|---|---|
| | Register VOSEL=11110 | 1.05 | 1. 15 | 1.25 | |
| | Register VOSEL=11111 | 1.075 | 1. 175 | 1.275 | |
| Output current (Id_max) | | | 200 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |
| **Digital IO Voltage** | | | | | |
| Output voltage (V_IO) | | 2.7 | 2.8 | 2.9 | V |
| Output current (Iio_max) | | | 200 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |
| **RF Voltage** | | | | | |
| Output voltage (V_RF) | | 2.7 | 2.8 | 2.9 | V |
| Output current (Ia_max) | | | 150 | | mA |
| Line regulation | | | | 0.2 | % |
| Output noise voltage | f =10Hz to 80 kHz | | 90 | | uVrms |
| Ripple rejection | 10 Hz < freq. < 3 kHz | | 65 | | dB |
| Current limit | | 1.2x | | 5x | Id_max |
| **Analog Voltage** | | | | | |
| Output voltage (V_A) | | 2.7 | 2.8 | 2.9 | V |
| Output current (Ia_max) | | | 100 | | mA |
| Line regulation | | | | 0.2 | % |
| Output noise voltage | f = 10 Hz to 80 kHz | | 90 | | uVrms |
| Ripple rejection | 10 Hz < freq. < 3 kHz | | 65 | | dB |
| Current limit | | 1.2x | | 5x | Id_max |
| **VTCXO Voltage** | | | | | |
| Output voltage (V_TCXO) | | 2.7 | 2.8 | 2.9 | V |
| Output current (Itcxo_max) | | | 40 | | mA |

| | | | | | |
|---|---|---|---|---|---|
| Line regulation | | | | 0.2 | % |
| Output noise voltage | f = 10 Hz to 80 kHz | | 90 | | µVrms |
| Ripple rejection | 10 Hz < freq. < 3 kHz | | 65 | | dB |
| Current limit | | 1.2x | | 5x | Id_max |
| **RTC Voltage** | | | | | |
| Output voltage(@5uA loading) | | 2. 6 | 2. 8 | 3.0 | V |
| Output current (Irtc_max) | | | 2 | | mA |
| Off reverse input current | | | 1 | | µA |
| **External Memory Voltage** | | | | | |
| Output voltage (V_M) | VMSEL=L | 1.7 | 1.8 | 1.9 | V |
| | VMSEL=H | 2.8 | 2.9 | 3.0 | V |
| Output current (Im_max) | | | 150 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |
| **SIM Voltage** | | | | | |
| Output voltage (V_SIM) | Register RG_VSIM_SEL=0 | 1.71 | 1.8 | 1.89 | V |
| | Register RG_VSIM_SEL=1 | 2.85 | 3.0 | 3.15 | V |
| Output current (Isim_max) | | | 30 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |
| **SIM2 Voltage** | | | | | |
| Output voltage (V_SIM2) | Register RG_VSIM2_V0SEL=000 | 1.235 | 1.3 | 1.365 | V |
| | Register RG_VSIM2_V0SEL=001 | 1.425 | 1.5 | 1.575 | V |
| | RG_VSIM2_V0SEL=010 | 1.71 | 1.8 | 1.89 | |
| | RG_VSIM2_V0SEL=011 | 2.375 | 2.5 | 2.625 | |
| | RG_VSIM2_V0SEL=100 | 2.66 | 2.8 | 2.94 | |

| | | | | | |
|---|---|---|---|---|---|
| | RG_VSIM2_VOSEL=101 | 2.85 | 3.0 | 3.15 | |
| | RG_VSIM2_VOSEL=110 | 3.135 | 3.3 | 3.465 | |
| Output current (Isim2_max) | | | 30 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |
| **USB Voltage** | | | | | |
| Output voltage (V_USB) | | 3.135 | 3.3 | 3.465 | V |
| Output current (Iusb_max) | | | 50 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |
| **Digital Camera Voltage** | | | | | |
| Output voltage (V_CAM_D) | Register VCAM_D_VOSEL=000 | 1.235 | 1.3 | 1.365 | V |
| | Register VCAM_D_VOSEL=001 | 1.425 | 1.5 | 1.575 | V |
| | Register VCAM_D_VOSEL=010 | 1.71 | 1.8 | 1.89 | V |
| | Register VCAM_D_VOSEL=011 | 2.375 | 2.5 | 2.625 | V |
| | Register VCAM_D_VOSEL=100 | 2.66 | 2.8 | 2.94 | V |
| | Register VCAM_D_VOSEL=101 | 2.85 | 3.0 | 3.15 | V |
| | Register VCAM_D_VOSEL=110 | 3.135 | 3.3 | 3.465 | V |
| Output current (Icamerd_max) | | | 100 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 2x | | 8x | Id_max |
| **Analog Camera Voltage** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Output voltage (V_CAM_A) | Register VCAM_A_SEL=00 | 1.425 | 1.5 | 1. 575 | V |
| | Register VCAM_A_SEL=01 | 1.71 | 1.8 | 1.89 | V |
| | Register VCAM_A_SEL=10 | 2.375 | 2.5 | 2.625 | V |
| | Register VCAM_A_SEL=11 | 2. 66 | 2.8 | 2.94 | V |
| Output current (Icamera_max) | | | 150 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |
| **KEYPAD LED Driver** | | | | | |
| Sink Current of Key-Pad LED Driver | Von<0.5V | 60 | | | mA |
| **Vibrator Driver** | | | | | |
| Output voltage (V_VIBR) | Register VIBR_VOSEL=000 | 1.235 | 1.3 | 1.365 | V |
| | Register VIBR_VOSEL=001 | 1.425 | 1.5 | 1.575 | V |
| | Register VIBR_VOSEL=010 | 1.71 | 1.8 | 1.89 | V |
| | Register VIBR_VOSEL=011 | 2.375 | 2.5 | 2.625 | V |
| | Register VIBR_VOSEL=100 | 2.66 | 2.8 | 2.94 | V |
| | Register VIBR_VOSEL=101 | 2.85 | 3.0 | 3.15 | V |
| | Register VIBR_VOSEL=110 | 3.135 | 3.3 | 3.465 | V |
| Output current (Ivibr_max) | | | 150 | | mA |
| Line regulation | | | | 0.2 | % |
| Current limit | | 1.2x | | 5x | Id_max |

### 6.2.7.8.3 SIM interface

| Parameter | Conditions | Min. | Typical | Max. | Unit |
|---|---|---|---|---|---|
| **Interface to 3 V SIM Card** | | | | | |
| Volrst | I = 200 µA | | | 0.36 | V |

| | | | | | |
|---|---|---|---|---|---|
| Vohrst | I = -200 µA | 0.9*VSIM | | | V |
| Volclk | I = 100 µA | | | 0.4 | V |
| Vohclk | I = -200 µA | 0.9*VSIM | | | V |
| Vihsio , Vohsio | I = ±20 µA | VSIM-0.4 | | | V |
| Iil | Vil = 0 V | | | -1 | mA |
| Vol | Iol = 1 mA | | | 0.4 | V |
| **Interface to 1.8 V SIM Card** | | | | | |
| Volrst2 | I = 200 µA | | | 0.2*VSIM | V |
| Vohrst | I = -200 µA | 0.9*VSIM | | | V |
| Volclk | I = 100 µA | | | 0.12*VSIM | V |
| Vohclk | I = -200 µA | 0.9*VSIM | | | V |
| Vil | | | | 0.15*VSIM | V |
| Vihsio , Vohsio | I = ±20 µA | VSIM-0.4 | | | V |
| Iil | Vil = 0 V | | | -1 | mA |
| Vol | Iol = 1 mA | | | 0.15*VSIM | V |
| **SIM Card Interface Timing** | | | | | |
| SIO pull-up resistance to VSIM | | | 5 | | kΩ |
| SRST, SIO rise/fall times | VSIM = 3, 1.8 V, load with 30 pF | | | 1 | µs |
| SCLK rise/fall times | VSIM = 3 V, CLK load with 30 pF | | | 18 | ns |
| | VSIM = 1.8 V, CLK load with 30 pF | | | 50 | ns |
| SCLK frequency | CLK load with 30 pF | | | 5 | MHz |

| | | | | | |
|---|---|---|---|---|---|
| SCLK duty cycle | SIMCLK Duty = 50%, fsimclk = 5 MHz | 47 | | 53 | % |
| SCLK propagation delay | | | 30 | 50 | ns |

#### 6.2.7.8.4 Charger Circuit

| Parameter | Conditions | Min. | Typical | Max. | Unit |
|---|---|---|---|---|---|
| AC charger input voltage | | 4.2 | | 8 | V |
| AC charger detect on threshold (Vchg_on) | VBAT<3.2V | 4.2 | | 7 | V |
| | VBAT>=3.2V | VBAT +120mV | | 7 | V |
| Maximum charging current (AC charging) | VBAT>=3.2V | | 0.16 / $R_{sense}$ | | A |
| Pre-charging current | VBAT<2.2V | | 50 | | mA |
| | 2.2V<VBAT<3.3V<br>(1) USB HOST<br>(2) Dedicated Charger | TBD | 70<br>200 | TBD | mA |
| Pre-charging off threshold | | | 3.2 | | V |
| Pre-charging off hysteresis | | | 0.3 | | V |
| CC mode to CV mode threshold | | 4.05 | 4.1 | 4.15 | V |
| BAT_ON (Vih) | | 2.33 | 2.47 | 2.6 | V |
| GATEDRV rising time ($T_r$) | BAT_ON, or OV | 1 | | 5 | µs |
| Over voltage protection threshold (OV) | | | 4.35 | | V |

#### 6.2.7.8.5 Regulators and Drivers

| Item | LDO | Voltage | Current | Description |
|---|---|---|---|---|
| 1 | VCORE | 0.8V~1.35V(25mv/step) | 200 mA | Digital core |
| 2 | VIO | 2.8V | 200 mA | Digital IO |
| 3 | VRF | 2.8V | 150 mA | RF chip |

| | | | | |
|---|---|---|---|---|
| 4 | VA | 2.8V | 100 mA | Analog baseband |
| 5 | VRTC | 2.8V | 2 mA | Real-time clock |
| 6 | VM | 1.8V / 2.9V | 150 mA | External memory, selectable |
| 7 | VSIM | 1.8V / 3.0V | 30 mA | SIM card, selectable |
| 8 | VTCXO | 2.8V | 40 mA | 13/26 MHz reference clock |
| 9 | VSIM2 | 1.3V/1.5V/1.8V/2.5V/2.8V/3.0V/3.3V | 30 mA | SIM2 card, selectable |
| 10 | VUSB | 3.3V | 50 mA | USB |
| 11 | VIBR | 1.3V/1.5V/1.8V/2.5V/2.8V/3.0V/3.3V | 150 mA | Vibrator |
| 12 | VCAM_A | 1.5V / 1.8V / 2.5V / 2.8V | 150 mA | Analog camera power |
| 13 | VCAM_D | 1.3V/1.5V/1.8V/2.5V/2.8V/3.0V/3.3V | 100 mA | Digital camera power |

| Driver | Type | Current | Description |
|---|---|---|---|
| KPLED | Open-drain NMOS switch | 60 mA | Drive the keypad LEDs |
| ISINK0 | Open-drain NMOS switch | Up tp 24mA | Drive LCM backlight LED |
| ISINK1 | Open-drain NMOS switch | Up tp 24mA | Drive LCM backlight LED |
| ISINK2 | Open-drain NMOS switch | Up tp 24mA | Drive LCM backlight LED |
| ISINK3 | Open-drain NMOS switch | Up tp 24mA | Drive LCM backlight LED |

#### 6.2.7.8.6 Class AB Audio Amplifier

| Parameter | Conditions | Min. | Typical | Max. | Unit |
|---|---|---|---|---|---|
| Output Power (8 Ohm) | 4.3V, THD+N=1% | 800 | | | mW |
| | 3.7V, THD+N=1% | 700 | | | mW |
| | 3.2V, THD+N=1% | 500 | | | mW |
| PSRR | VBAT=3.2/3.7/4.3V<br>Vin=200mVpp | | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Input AC to Ground | | | | |
| | 217Hz | 50 | 67 | | dB |
| | 1KHz | 50 | 67 | | dB |
| | 3KHz | 50 | 67 | | dB |
| | 30KHz | 50 | 53 | | dB |
| THD+N (8Ohm) | 4.3V, 800mW, 1KHz, 25° C | | 0.02 | 1 | % |
| | 3.7V, 700mW, 1KHz, 25° C | | 0.02 | 1 | % |
| | 3.2V, 500mW, 1KHz, 25° C | | 0.02 | 1 | % |
| SNR | VBAT=3.2V/4.3V<br>0.5W/0.8W, 8Ohm,<br>20Hz to 22KHz, A-weighted | | | | |
| | 3.2V, A-weighted | 93 | | | dB |
| | 4.3V, A-wieghted | 93 | | | dB |

## 6.3 PMU Registers Definition

| ADDRESS | TITLE | DESCRIPTION |
|---|---|---|
| 83010800 | PMIC_VRF_CON0 | |
| 83010804 | PMIC_VRF_CON1 | |
| 83010808 | PMIC_VRF_CON2 | |
| 83010810 | PMIC_VTCXO_CON0 | |
| 83010814 | PMIC_VTCXO_CON1 | |
| 83010818 | PMIC_VTCXO_CON2 | |
| 83010820 | PMIC_VA_CON0 | |
| 83010824 | PMIC_VA_CON1 | |

| | | |
|---|---|---|
| 83010828 | PMIC_VA_CON2 | |
| 83010830 | PMIC_VCAMA_CON0 | |
| 83010834 | PMIC_VCAMA_CON1 | |
| 83010838 | PMIC_VCAMA_CON2 | |
| 83010840 | PMIC_VCAMD_CON0 | |
| 83010844 | PMIC_VCAMD_CON1 | |
| 83010848 | PMIC_VCAMD_CON2 | |
| 83010850 | PMIC_VIO_CON0 | |
| 83010854 | PMIC_VIO_CON1 | |
| 83010858 | PMIC_VIO_CON2 | |
| 83010860 | PMIC_VUSB_CON0 | |
| 83010864 | PMIC_VUSB_CON1 | |
| 83010868 | PMIC_VUSB_CON2 | |
| 83010880 | PMIC_VSIM1_CON0 | |
| 83010884 | PMIC_VSIM1_CON1 | |
| 83010888 | PMIC_VSIM1_CON2 | |
| 8301088C | PMIC_VSIM1_CON3 | |
| 83010890 | PMIC_VSIM2_CON0 | |
| 83010894 | PMIC_VSIM2_CON1 | |
| 83010898 | PMIC_VSIM2_CON2 | |

| | | |
|---|---|---|
| 8301089C | PMIC_VSIM2_CON3 | |
| 830108A0 | PMIC_VRTC_CON0 | |
| 830108A4 | PMIC_VRTC_CON1 | |
| 830108B0 | PMIC_VIBR_CON0 | |
| 830108B4 | PMIC_VIBR_CON1 | |
| 830108B8 | PMIC_VIBR_CON2 | |
| 830108C0 | PMIC_VM_CON0 | |
| 830108C4 | PMIC_VM_CON1 | |
| 830108C8 | PMIC_VM_CON2 | |
| 830108D0 | PMIC_VCORE_CON0 | |
| 830108D4 | PMIC_VCORE_CON1 | |
| 830108D8 | PMIC_VCORE_CON2 | |
| 830108E0 | PMIC_LDOS_CON | |
| 830108F0 | PMIC_INT_EN0 | |
| 830108F4 | PMIC_INT_EN1 | |
| 830108F8 | PMIC_OC_STATUS | |
| 830108FC | PMIC_OC_FLAG | |
| 83010900 | PMIC_STARTUP_CON0 | |
| 83010904 | PMIC_STARTUP_CON1 | |
| 83010908 | PMIC_STARTUP_CON2 | |

| | | |
|---|---|---|
| 8301090C | PMIC_STARTUP_CON3 | |
| 83010980 | PMIC_ISINK0_CON0 | |
| 83010984 | PMIC_ISINK0_CON1 | |
| 83010988 | PMIC_ISINK0_CON2 | |
| 83010990 | PMIC_ISINK1_CON0 | |
| 830109A0 | PMIC_ISINK2_CON0 | |
| 830109B0 | PMIC_ISINK3_CON0 | |
| 830109C0 | PMIC_KPLED_CON0 | |
| 830109D0 | PMIC_CLASSAB_CON0 | |
| 830109D4 | PMIC_CLASSAB_CON1 | |
| 83010A00 | PMIC_CHR_CON0 | |
| 83010A04 | PMIC_CHR_CON1 | |
| 83010A08 | PMIC_CHR_CON2 | |
| 83010A0C | PMIC_CHR_CON3 | |
| 83010A10 | PMIC_CHR_CON4 | |
| 83010A14 | PMIC_CHR_CON5 | |
| 83010A18 | PMIC_CHR_CON6 | |
| 83010A1C | PMIC_CHR_CON7 | |
| 83010A20 | PMIC_CHR_CON8 | |
| 83010A24 | PMIC_CHR_CON9 | |

| | | |
|---|---|---|
| 83010A28 | PMIC_CHR_CON10 | |
| 83010A2C | PMIC_CHR_CON11 | |
| 83010A30 | PMIC_CHR_CON12 | |
| 83010A34 | PMIC_CHR_CON13 | |
| 83010A38 | PMIC_CHR_CON14 | |
| 83010A3C | PMIC_CHR_CON15 | |
| 83010A40 | PMIC_CHR_CON16 | |
| 83010F00 | FMTR_CON0 | |
| 83010F04 | FMTR_CON1 | |
| 83010F08 | FMTR_DATA | |
| 83010F10 | MIXEDSYS_MON_CON0 | |
| 83010F14 | MIXEDSYS_MON_CON1 | |
| 83010F18 | MIXEDSYS_MON_CON2 | |
| 83010F20 | ABB_MON_CON0 | |
| 83010F30 | SIM_MON_CON0 | |
| 83010F80 | ACIF_RES1_AC_CON0 | |
| 83010F84 | ACIF_RES1_AC_CON1 | |
| 83010F88 | ACIF_RES1_STATUS | |

### 6.3.1 PMU Register Setting

83010800h PMIC_VRF_CON0 PMIC_VRF_CON0

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VRF_OC_FLAG(R) | | CCI_VRF_OC_AUTO_OFF | | RG_VRF_PDNMOS_EN | | | | | | | | RG_VRF_RS_EN | VRF_ON_SEL | RG_VRF_EN |
| Type | | R | | RW | | RW | | | | | | | | RW | RW | RW |
| Reset | | 0 | | 0 | | 1 | | | | | | | | 0 | 0 | 0 |

VRF_OC_FLAG(R) Deglitched signal for VRF_CL_STATUS

CCI_VRF_OC_AUTO_OFF Automatically turn off VRF LDO if VRF_OC_FLAG

RG_VRF_PDNMOS_EN Vrf output power down Enable

(1'b1: enable output powerdown; 1'b0: disable output powerdown)

RG_VRF_RS_EN Vrf remote sense Enable

(1'b1: enable; 1'b0: disable)

VRF_ON_SEL VRF on manual mode

0: auto mode , 1: manual mode

RG_VRF_EN Vrf Enable while VRF_ON_SEL =1

(1'b1: enable; 1'b0: disable)

**83010804h PMIC_VRF_CON1** **PMIC_VRF_CON1**

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

RG_VRF_CAL Vrf Voltage Calibration

| RG_VRF_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VRF_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010808h PMIC_VRF_CON2** **PMIC_VRF_CON2**

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

| Name | | | | | | | | | CCI_VRF_STBTD | CCI_VRF_OC_GEAR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | RW | RW | | | | |
| Reset | | | | | | | | | 0 | 0 | | | | |

CCI_VRF_STBTD : Delay time for STB after VRF_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VRF_OC_GEAR : VRF OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**83010810h PMIC_VTCXO_CON0** **PMIC_VTCXO_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VTCXO_OC_FLAG(R) | | CCI_VTCXO_OC_AUTO_OFF | | RG_VTCXO_PDNMOS_EN | | | | | | | | | VTCXO_ON_SEL | RG_VTCXO_EN |
| Type | | R | | RW | | RW | | | | | | | | | RW | RW |
| Reset | | 0 | | 0 | | 1 | | | | | | | | | 0 | 0 |

VTCXO_OC_FLAG(R) Deglitched signal for VTCXO_CL_STATUS

CCI_VTCXO_OC_AUTO_OFF Automatically turn off VTCXO LDO if VTCXO_OC_FLAG

RG_VTCXO_PDNMOS_EN Vtcxo output power down Enable

(1'b1: enable output powerdown; 1'b0: disable output powerdown)

VTCXO_ON_SEL Vtcxo on manual mode

0: auto mode , 1: manual mode

RG_VTCXO_EN Vtcxo Enable while VRF_ON_SEL =1

(1'b1: enable; 1'b0: disable)

**83010814h PMIC_VTCXO_CON1** **PMIC_VTCXO_CON1**

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

RG_VTCXO_CAL Vtcxo Voltage Calibration

| RG_VTCXO_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VTCXO_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010818h PMIC_VTCXO_CON2** **PMIC_VTCXO_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | CCI_VTCXO_STBTD | | CCI_VTCXO_OC_GEAR | | | | CCI_SRCLKEN | |
| Type | | | | | | | | | RW | | RW | | | | RW | |
| Reset | | | | | | | | | 0 | | 0 | | | | 1 | |

CCI_VTCXO_STBTD : Delay time for STB after VTCXO_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VTCXO_OC_GEAR : VTCXO OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

CCI_SRCLKEN SRCLKEN to PMIC force enable control signal

**0** disable (****controlled by sleep control module)

**1** enable

**83010820h** **PMIC_VA_CON0** **PMIC_VA_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VA_OC_FLAG(R) | | CCI_VA_OC_AUTO_OFF | | RG_VA_PDNMOS_EN | | | | | | | | RG_VA_RS_EN | | RG_VA_EN |
| Type | | R | | RW | | RW | | | | | | | | RW | | RW |
| Reset | | 0 | | 0 | | 1 | | | | | | | | 0 | | 1 |

VA_OC_FLAG(R) Deglitched signal for VA_CL_STATUS

CCI_VA_OC_AUTO_OFF Automatically turn off VA LDO if VA_OC_FLAG

RG_VA_PDNMOS_EN Va output power down Enable

(1'b1: enable output powerdown; 1'b0: disable output powerdown)

RG_VA_RS_EN Va remote sense Enable

(1'b1: enable; 1'b0: disable)

RG_VA_EN Va enable for testing

(1'b1: enable; 1'b0: disable)

**83010824h** **PMIC_VA_CON1** **PMIC_VA_CON1**

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

RG_VA_CAL Va Voltage Calibration

| RG_VA_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |

| RG_VA_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
|---|---|---|---|---|---|---|---|---|
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010828h PMIC_VA_CON2** **PMIC_VA_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | CCI_VA_STBTD | | CCI_VA_OC_GEAR | | | | QI_VA_LP_EN | |
| Type | | | | | | | | | RW | | RW | | | | RW | |
| Reset | | | | | | | | | 0 | | 0 | | | | 0 | |

CCI_VA_STBTD : Delay time for STB after VA_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VA_OC_GEAR : VA OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

QI_VA_LP_EN Va low power mode enable

(1'b1: low power mode; 1'b0: typical mode)

**83010830h PMIC_VCAMA_CON0** **PMIC_VCAMA_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VCAMA_OC_FLAG(R) | | CCI_VCAMA_OC_AUTO_OFF | | RG_VCAMA_PDNMOS_EN | | | | | RG_VCAMA_VOSEL | | | | | RG_VCAMA_EN |
| Type | | R | | RW | | RW | | | | | RW | | | | | RW |
| Reset | | 0 | | 0 | | 1 | | | | | 0 | | | | | 0 |

VCAMA_OC_FLAG(R) Deglitched signal for VCAMA_CL_STATUS

CCI_VCAMA_OC_AUTO_OFF Automatically turn off VCAMA LDO if VCAMA_OC_FLAG

RG_VCAMA_PDNMOS_EN Vcama output power down Enable

(1'b1: enable output powerdown; 1'b0: disable output powerdown)

| RG_VCAMA_VOSEL<br>Vcama Output Voltage Selection | 00 | 01 | 10 | 11 |
|---|---|---|---|---|
| Vout | 1.5V | 1.8V | 2.5V | 2.8V |

RG_VCAMA_EN Vcama Enable

(1'b1: enable; 1'b0: disable)

**83010834h** **PMIC_VCAMA_CON1** **PMIC_VCAMA_CON1**

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

RG_VCAMA_CAL Vcama Voltage Calibration

| RG_VCAMA_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VCAMA_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010838h** **PMIC_VCAMA_CON2** **PMIC_VCAMA_CON2**

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

CCI_VCAMA_STBTD: Delay time for STB after VCAMA_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VCAMA_OC_GEAR : VCAMA OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**83010840h PMIC_VCAMD_CON0** **PMIC_VCAMD_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VCAMD_OC_FLAG(R) | | CCI_VCAMD_OC_AUTO_OFF | | RG_VCAMD_PDNMOS_EN | | | | RG_VCAMD_VOSEL | | | | | | RG_VCAMD_EN |
| Type | | R | | RW | | RW | | | | RW | | | | | | RW |
| Reset | | 0 | | 0 | | 1 | | | | 100 | | | | | | 0 |

VCAMD_OC_FLAG(R) Deglitched signal for VCAMD_CL_STATUS

CCI_VCAMD_OC_AUTO_OFF Automatically turn off VCAMD LDO if VCAMD_OC_FLAG

RG_VCAMD_PDNMOS_EN VCAMD POWER DOWN NMOS Enable

(1'b1: enable; 1'b0: disable)

| RG_VCAMD_VOSEL<br>VCAMD output selection signal | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| Vout | 1.3V | 1.5V | 1.8V | 2.5V | 2.8V | 3.0V | 3.3V | 3.3V |

RG_VCAMD_EN VCAMD Enable

(1'b1: enable; 1'b0: disable)

**83010844h PMIC_VCAMD_CON1** **PMIC_VCAMD_CON1**

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

RG_VCAMD_CAL VCAMD Voltage Calibration

(Notes: when the output is 1.3V, the calibration range is -60mV~+160mV.)

| RG_VCAMD_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VCAMD_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010848h PMIC_VCAMD_CON2** **PMIC_VCAMD_CON2**

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

CCI_VCAMD_STBTD : Delay time for STB after VCAMD_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VCAMD_OC_GEAR : VCAMD OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**83010850h PMIC_VIO_CON0** **PMIC_VIO_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VIO_OC_FLAG | | CCI_VIO_OC_AUTO_OFF | | RG_VIO_PDNMOS_EN | | | | | | | | | | RG_VIO_EN |
| Type | | R | | RW | | RW | | | | | | | | | | RW |
| Reset | | 0 | | 0 | | 1 | | | | | | | | | | 1 |

VIO_OC_FLAG(R) Deglitched signal for VIO_CL_STATUS

CCI_VIO_OC_AUTO_OFF Automatically turn off VIO LDO if VIO_OC_FLAG

RG_VIO_PDNMOS_EN VIO POWER DOWN NMOS Enable

(1'b1: enable; 1'b0: disable)

RG_VIO_EN VIO enable for testing

(1'b1: enable; 1'b0: disable)

**83010854h PMIC_VIO_CON1** **PMIC_VIO_CON1**

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

RG_VIO_CAL VIO Voltage Calibration

| RG_VIO_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VIO_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010858h PMIC_VIO_CON2** **PMIC_VIO_CON2**

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

CCI_VIO_OC_GEAR : VIO OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**83010860h PMIC_VUSB_CON0** **PMIC_VUSB_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VUSB_OC_FLAG | | CCI_VUSB_OC_AUTO_OFF | | RG_VUSB_PDNMOS_EN | | | | | | | | | | RG_VUSB_EN |
| Type | | R | | RW | | RW | | | | | | | | | | RW |
| Reset | | 0 | | 0 | | 1 | | | | | | | | | | 1 |

VUSB_OC_FLAG(R) Deglitched signal for VUSB_CL_STATUS

CCI_VUSB_OC_AUTO_OFF Automatically turn off VUSB LDO if VUSB_OC_FLAG

RG_VUSB_PDNMOS_EN VUSB output power down enable

RG_VUSB_EN Vusb enable for testing
(1'b1: enable; 1'b0: disable)

**83010864h PMIC_VUSB_CON1** **PMIC_VUSB_CON1**

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

RG_VUSB_CAL VUSB Voltage Calibration

| RG_VUSB_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VUSB_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |
|---|---|---|---|---|---|---|---|---|

**83010868h PMIC_VUSB_CON2** **PMIC_VUSB_CON2**

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

CCI_VUSB_STBTD : Delay time for STB after VUSB_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VUSB_OC_GEAR : VUSB OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**83010880h PMIC_VSIM1_CON0** **PMIC_VSIM1_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VSIM1_OC_FLAG(R) | | CCI_VSIM1_OC_AUTO_OFF | | RG_VSIM1_PDN_MOS_EN | | | | | | CCI_VSIM1_VOSEL | | | | CCI_VSIM1_EN |
| Type | | R | | RW | | RW | | | | | | RW | | | | RW |
| Reset | | 0 | | 0 | | 1 | | | | | | 0 | | | | 0 |

VSIM1_OC_FLAG(R) Deglitched signal for VSIM1_CL_STATUS

CCI_VSIM1_OC_AUTO_OFF Automatically turn off VSIM1 LDO if VSIM1_OC_FLAG

RG_VSIM1_PDNMOS_EN VSIM1 output power down enable

CCI_VSIM1_VOSEL VSIM1 Output Voltage Selection Signal

**==> only works at VSIM1_GPLDO_EN(@VSIM1_CON2) = 1**

| CCI_VSIM1_VOSEL | 0 | 1 |
|---|---|---|
| Vout | 1.8V | 3.0V |

CCI_VSIM1_ENcci mode VSIM1 Enable

**==> only works at VSIM1_GPLDO_EN(@VSIM1_CON2) = 1**

(1'b1: enable; 1'b0: disable)

**83010884h PMIC_VSIM1_CON1** **PMIC_VSIM1_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | RG_VSIM1_CAL | | | | | | | CCI_VSIM1_CSTOP |
| Type | | | | | | | | | RW | | | | | | | W |
| Reset | | | | | | | | | 0 | | | | | | | 0 |

| RG_VSIM1_CAL VSIM1 Voltage Calibration | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VSIM1_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010888h PMIC_VSIM1_CON2** **PMIC_VSIM1_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | CCI_VSIM1_ | | CCI_VSIM1_ | | | | VSIM1_G | |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | STBTD | OC_GEAR | | | PLDO_EN | |
| Type | | | | | | | | | RW | RW | | | RW | |
| Reset | | | | | | | | | 0 | 0 | | | 0 | |

CCI_VSIM1_STBTD : Delay time for STB after VSIM1_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VSIM1_OC_GEAR : VSIM1 OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**VSIM1_GPLDO_EN** The control selection of VSIM1 LDO enable & voltage, which is controlled by SIM controller circuit or VSIM1_CON0

**0** VSIM1 LDO is controlled by SIM controller

**1** VSIM1 LDO is controlled by VSIM_CON0

**8301088Ch** **PMIC_VSIM1_CON3** **PMIC_VSIM1_CON3**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | RG_SIM_SRP | | RG_SIM_SRN | | RG_SIM_BIAS | | RG_SIMIO_DRV | | | |
| Type | | | | | | | RW | | RW | | RW | | RW | | | |
| Reset | | | | | | | 0 | | 0 | | 10 | | 0 | | | |

RG_SIM_SRP SIM SCLK slew rate control. PMOS path speed tuning

| RG_SIM_SRP[1:0] | 00 (default) | 01 | 10 | 11 |
|---|---|---|---|---|
| SCLK slew rate | 1X | 1.1X | 0.9X | 1X |

RG_SIM_SRN SIM SCLK slew rate control. NMOS path speed tuning

| RG_SIM_SRN[1:0] | 00 (default) | 01 | 10 | 11 |
|---|---|---|---|---|
| SCLK slew rate | 1X | 1.1X | 0.9X | 1X |

RG_SIM_BIAS SIM slew rate control. Bias current adjustment bits

| RG_SIM_BIAS[1:0] | 00 | 01 | 10 (default) | 11 |
|---|---|---|---|---|
| Bias current | 1X | 1.2X | 1.5X | 2X |

RG_SIMIO_DRV SIO input pull-up resistor adjustment

| RG_SIMIO_DRV[2:0] | 1XX | 000 | 001 | 010 | 011 |
|---|---|---|---|---|---|
| Pull up resistor (kohm) | disable | 5 | 3.3 | 20 | 6.7 |

**83010890h PMIC_VSIM2_CON0** **PMIC_VSIM2_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VSIM2_OC_FLAG(R) | | CCI_VSIM2_OC_AUTO_OFF | | RG_VSIM2_PDNMOS_EN | | | | CCI_VSIM2_VOSEL | | | | | | CCI_VSIM2_EN |
| Type | | R | | RW | | RW | | | | RW | | | | | | RW |
| Reset | | 0 | | 0 | | 0 | | | | 0 | | | | | | 0 |

VSIM2_OC_FLAG(R) Deglitched signal for VSIM2_CL_STATUS

CCI_VSIM2_OC_AUTO_OFF Automatically turn off VSIM2 LDO if VSIM2_OC_FLAG

RG_VSIM2_PDNMOS_EN VSIM2 output power down enable

CCI_VSIM2_VOSEL VSIM2 Output Voltage Selection Signal

**==> only works at VSIM2_GPLDO_EN(@VSIM2_CON2) = 1**

| RG_VSIM2_VOSEL<2:0> | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| Vout | 1.3V | 1.5V | 1.8V | 2.5V | 2.8V | 3.0V | 3.3V | 3.3V |

CCI_VSIM2_EN cci mode VSIM2 Enable

**==> only works at VSIM2_GPLDO_EN(@VSIM2_CON2) = 1**
(1'b1: enable; 1'b0: disable)

**83010894h PMIC_VSIM2_CON1** **PMIC_VSIM2_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | RG_VSIM2_CAL | | | | | | | CCI_VSIM2_CSTOP |
| Type | | | | | | | | | RW | | | | | | | W |
| Reset | | | | | | | | | 0 | | | | | | | 0 |

RG_VSIM2_CAL VSIM2 Voltage Calibration

(Notes: when the output is 1.3V, the calibration range is -60mV~+160mV.)

| RG_VSIM2_CAL<br>VSIM2 Voltage Calibration | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VSIM2_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**83010898h PMIC_VSIM2_CON2** **PMIC_VSIM2_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | CCI_VSIM2_STBTD | | CCI_VSIM2_OC_GEAR | | | | VSIM2_GPLDO_EN | |
| Type | | | | | | | | | RW | | RW | | | | RW | |
| Reset | | | | | | | | | 0 | | 0 | | | | 0 | |

CCI_VSIM2_STBTD : Delay time for STB after VSIM2_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VSIM2_OC_GEAR : VSIM2 OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**VSIM2_GPLDO_EN** The control selection of VSIM2 LDO enable & voltage, which is controlled by SIM controller circuit or VSIM2_CON0

**0** VSIM2 LDO is controlled by SIM controller

**1** VSIM2 LDO is controlled by VSIM2_CON0

**8301089Ch PMIC_VSIM2_CON3** **PMIC_VSIM2_CON3**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | RG_SIM2_SRP | | RG_SIM2_SRN | | RG_SIM2_BIAS | | RG_SIMIO2_DRV | | | RG_SIM2_MODE |
| Type | | | | | | | RW | | RW | | RW | | RW | | | RW |
| Reset | | | | | | | 0 | | 0 | | 10 | | 0 | | | 0 |

RG_SIM2_SRP SIM2 SCLK slew rate control. PMOS path speed tuning

| RG_SIM2_SRP[1:0] | 00 (default) | 01 | 10 | 11 |
|---|---|---|---|---|
| SCLK2 slew rate | 1X | 1.1X | 0.9X | 1X |

RG_SIM2_SRN SIM2 SCLK slew rate control. NMOS path speed tuning

| RG_SIM2_SRN[1:0] | 00 (default) | 01 | 10 | 11 |
|---|---|---|---|---|
| SCLK2 slew rate | 1X | 1.1X | 0.9X | 1X |

RG_SIM2_BIAS SIM2 slew rate control. Bias current adjustment bits

| RG_SIM2_BIAS[1:0] | 00 | 01 | 10 (default) | 11 |
|---|---|---|---|---|

| Bias current | 1X | 1.2X | 1.5X | 2X |
|---|---|---|---|---|

RG_SIMIO2_DRV SIO2 input pull-up resistor adjustment

| RG_SIMIO2_DRV[2:0] | 1XX | 000 | 001 | 010 | 011 |
|---|---|---|---|---|---|
| Pull up resistor (kohm) | disable | 5 | 3.3 | 20 | 6.7 |

RG_SIM2_MODE Enable SIM2 GPIO mode

0: SIM mode

1: GPIO mode

**830108A0h PMIC_VRTC_CON0** **PMIC_VRTC_CON0**

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

RG_VRTC_EN Vrtc enable for testing

(1'b1: enable; 1'b0: disable)

**830108A4h PMIC_VRTC_CON1** **PMIC_VRTC_CON1**

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

RG_VRTC_CAL Vrtc Voltage Calibration

| RG_VRTC_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VRTC_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**830108B0h PMIC_VIBR_CON0** **PMIC_VIBR_CON0**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | VIBR_OC_FLAG(R) | | CCI_VIBR_OC_AUTO_OFF | | RG_VIBR_PDNMOS_EN | | | | RG_VIBR_VOSEL | | | | | | CCI_VIBR_EN |
| | R | | RW | | RW | | | | RW | | | | | | RW |
| | 0 | | 0 | | 1 | | | | 100 | | | | | | 0 |

VIBR_OC_FLAG(R) Deglitched signal for VIBR_CL_STATUS

CCI_VIBR_OC_AUTO_OFF Automatically turn off VIBR LDO if VIBR_OC_FLAG

RG_VIBR_PDNMOS_EN VIBR POWER DOWN NMOS Enable

(1'b1: enable; 1'b0: disable)

RG_VIBR_VOSEL VIBR output selection signal

| RG_VIBR_VOSEL<2:0> | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| Vout | 1.3V | 1.5V | 1.8V | 2.5V | 2.8V | 3.0V | 3.3V | 3.3V |

CCI_VIBR_EN cci mode VIBR Enable

(1'b1: enable; 1'b0: disable)

**830108B4h PMIC_VIBR_CON1** **PMIC_VIBR_CON1**

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

RG_VIBR_CAL VIBR Voltage Calibration

(Notes: when the output is 1.3V, the calibration range is -60mV~+160mV.)

| RG_VIBR_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VIBR_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**830108B8h** **PMIC_VIBR_CON2** **PMIC_VIBR_CON2**

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

CCI_VIBR_STBTD : Delay time for STB after VIBR_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VIBR_OC_GEAR : VIBR OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**830108C0h** **PMIC_VM_CON0** **PMIC_VM_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VM_OC_FLAG | | CCI_VM_OC_AUTO_OFF | | RG_VM_PDNMOS_EN | | | | | | | | | | RG_VM_EN |
| Type | | R | | RW | | RW | | | | | | | | | | RW |
| Reset | | 0 | | 0 | | 1 | | | | | | | | | | 1 |

VM_OC_FLAG(R) Deglitched signal for VM_CL_STATUS

CCI_VM_OC_AUTO_OFF Automatically turn off VM LDO if VM_OC_FLAG

RG_VM_PDNMOS_EN VM POWER DOWN NMOS Enable
(1'b1: enable; 1'b0: disable)

RG_VM_EN Vm enable for testing
(1'b1: enable; 1'b0: disable)

**830108C4h PMIC_VM_CON1** **PMIC_VM_CON1**

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

RG_VM_CAL VM Voltage Calibration

| RG_VM_CAL<3:0> | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
|---|---|---|---|---|---|---|---|---|
| Vout | ±0 | -20mV | -40mV | -60mV | -80mV | -100mV | -120mV | -140mV |
| RG_VM_CAL<3:0> | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Vout | +160mV | +140mV | +120mV | +100mV | +80mV | +60mV | +40mV | +20mV |

**830108C8h PMIC_VM_CON2** **PMIC_VM_CON2**

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

CCI_VM_STBTD : Delay time for STB after VM_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VM_OC_GEAR : VM OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**830108D0h PMIC_VCORE_CON0** **PMIC_VCORE_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | VCORE_OC_FLAG(R) | | CCI_VCORE_OC_AUTO_OFF | | RG_VCORE_PDNMOS_EN | | RG_VCORE_VOSEL | | | | | MUST BE 0 | MUST BE 0 | | RG_VCORE_EN |
| Type | | R | | RW | | RW | | RW | | | | | RW | RW | | RW |
| Reset | | 0 | | 0 | | 1 | | 0 | | | | | 0 | 0 | | 1 |

VCORE_OC_FLAG(R) Deglitched signal for VCORE_CL_STATUS

CCI_VCORE_OC_AUTO_OFF Automatically turn off VCORE LDO if VCORE_OC_FLAG

RG_VCORE_PDNMOS_EN VCORE POWER DOWN NMOS Enable
(1'b1: enable; 1'b0: disable)

RG_VCORE_VOSEL

| RG_VCORE_VOSEL<4:0> | 00000 | 00001 | 00010 | 00011 | 00100 | 00101 | 00110 | 10000 |
|---|---|---|---|---|---|---|---|---|
| Vout | 1.2V | 1.225V | 1.25V | 1.275V | 1.3V | 1.325V | 1.35V | 0.8V |
| RG_VCORE_VOSEL<4:0> | 10001 | 10010 | 10011 | 10100 | 10101 | 10110 | 10111 | 11000 |
| Vout | 0.825V | 0.85V | 0.875V | 0.9V | 0.925V | 0.95V | 0.975V | 1V |
| RG_VCORE_VOSEL<4:0> | 11001 | 11010 | 11011 | 11100 | 11101 | 11110 | 11111 | |
| Vout | 1.025V | 1.05V | 1.075V | 1.1V | 1.125V | 1.15V | 1.175V | |

RG_VCORE_EN Vcore enable for testing
(1'b1: enable; 1'b0: disable)

(Note :RG_VCORE_EN=0 is only for test use.Vcore can't be disabled in system application. )

**830108D4h PMIC_VCORE_CON1** **PMIC_VCORE_CON1**

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

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | RW | | | | | | | | |
| Reset | | | | | | | | 0 | | | | | | | | |

RG_VCORE_VOSEL_SLEEP Sleep mode VOSEL

**830108D8h PMIC_VCORE_CON2** **PMIC_VCORE_CON2**

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

CCI_VCORE_STBTD: Delay time for STB after VCORE_EN

00 : 213us

01: 427us

10: 610us

11: 823us

CCI_VCORE_OC_GEAR : VCORE OC Flag deglitch time

00 : 91us

01: 213us

10: 427us

11: 823us

**830108E0h PMIC_LDOS_CON** **PMIC_LDOS_CON**

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

RG_LDOS_RESER reserve for LDOS

**830108F0h PMIC_INT_EN0** **PMIC_INT_EN0**

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

| Name | CCI_VA_OC_INT_EN | CCI_VM_OC_INT_EN | CCI_VIO_OC_INT_EN | CCI_VRF_OC_INT_EN | CCI_VTCXO_OC_INT_EN | | | | CCI_VCAMA_OC_INT_EN | CCI_VCAMD_OC_INT_EN | CCI_VUSB_OC_INT_EN | | CCI_VSIM1_OC_INT_EN | CCI_VSIM2_OC_INT_EN | CCI_VIBR_OC_INT_EN | CCI_SPK_OC_INT_EN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | RW | RW | RW | RW | RW | | | | RW | RW | RW | | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | | | | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

CCI_VA_OC_INT_EN Enable bit of VA over-current interrupt if the VA_OC_FLAG been asserted

CCI_VM_OC_INT_EN Enable bit of VM over-current interrupt if the VM_OC_FLAG been asserted

CCI_VIO_OC_INT_EN Enable bit of VIO over-current interrupt if the VIO_OC_FLAG been asserted

CCI_VRF_OC_INT_EN Enable bit of VRF over-current interrupt if the VRF_OC_FLAG been asserted

CCI_VTCXO_OC_INT_EN Enable bit of VTCXO over-current interrupt if the VTCXO_OC_FLAG been asserted

CCI_VCAMA_OC_INT_EN Enable bit of VCAMA over-current interrupt if the VCAMA_OC_FLAG been asserted

CCI_VCAMD_OC_INT_EN Enable bit of VCAMD over-current interrupt if the VCAMD_OC_FLAG been asserted

CCI_VUSB_OC_INT_EN Enable bit of VUSB over-current interrupt if the VUSB_OC_FLAG been asserted

CCI_VSIM1_OC_INT_EN Enable bit of VSIM1 over-current interrupt if the VSIM1_OC_FLAG been asserted

CCI_VSIM2_OC_INT_EN Enable bit of VSIM2 over-current interrupt if the VSIM2_OC_FLAG been asserted

CCI_VIBR_OC_INT_EN Enable bit of VIBR over-current interrupt if the VIBR_OC_FLAG been asserted

CCI_SPK_OC_INT_EN Enable bit of SPK over-current interrupt if the SPK_OC_FLAG been asserted

**830108F4h PMIC_INT_EN1** **PMIC_INT_EN1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | RG_BATON_UNDET2_INT_INV | RG_CHRDET_INT_INV | RG_BATON_UNDET_INT_INV | RG_CHRLDO_INT_INV | RG_CHROVP_INT_INV | | | | RG_BATON_UNDET2_INT_EN | RG_CHRDET_INT_EN | RG_BATON_UNDET_INT_EN | RG_CHRLDO_INT_EN | RG_CHROVP_INT_EN |
| Type | | | | RW | RW | RW | RW | RW | | | | RW | RW | RW | RW | RW |
| Reset | | | | 0 | 0 | 0 | 0 | 0 | | | | 0 | 0 | 0 | 0 | 0 |

RG_BATON_UNDET2_INT_INV Inverse polarity of interrupt BATON_UNDET2

RG_CHRDET_INT_INV Inverse polarity of interrupt CHRDET

RG_BATON_UNDET_INT_INV Inverse polarity of interrupt BATON_UNDET

RG_CHRLDO_INT_INV Inverse polarity of interrupt CHRLDO

RG_CHROVP_INT_INV Inverse polarity of interrupt CHROVP

RG_BATON_UNDET2_INT_EN Enable of interrupt BATON_UNDET2

RG_CHRDET_INT_EN Enable of interrupt CHRDET

RG_BATON_UNDET_INT_EN Enable of interrupt BATON_UNDET

RG_CHRLDO_INT_EN Enable of interrupt CHRLDO

RG_CHROVP_INT_EN Enable of interruptCHROVP

**830108F8h PMIC_OC_STATUS** **PMIC_OC_STATUS**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | VA_OC_STA | VM_OC_STA | VIO_OC_STA | VRF_OC_ST | VTCXO_OC_ | VCORE_OC_ | SPK_OC_P_S | SPK_OC_N_S | VCAMA_OC_ | VCAMD_OC_ | VUSB_OC_S | | VSIM1_OC_STA | VSIM2_OC_STA | VIBR_OC | SPK_OC_STAT |

| | TUS | TUS | TUS | ATUS | STATUS | STATUS | TATUS | TATUS | STATUS | STATUS | TATUS | | TUS | TUS | _STATUS | US |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | 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 |

VA_OC_STATUS VA over current flag

(1'b1: over current; 1'b0: no over current )

VM_OC_STATUS VM over current flag

(1'b1: over current; 1'b0: no over current )

VIO_OC_STATUS * VIO over current flag

(1'b1: over current; 1'b0: no over current )

VRF_OC_STATUS * VRF over current flag

(1'b1: over current; 1'b0: no over current )

VTCXO_OC_STATUS * VTCXO over current flag

(1'b1: over current; 1'b0: no over current )

VCORE_OC_STATUS VCORE over current flag

(1'b1: over current; 1'b0: no over current )

SPK_OCP_STATUS classAB PMOS over current flag

(1'b1: over current; 1'b0: no over current )

SPK_OCN_STATUS classAB NMOS over current flag

(1'b1: over current; 1'b0: no over current )

VCAMA_OC_STATUS * VCAMA over current flag

(1'b1: over current; 1'b0: no over current )

VCAMD_OC_STATUS * VCAMD over current flag

(1'b1: over current; 1'b0: no over current )

VUSB_OC_STATUS * VUSB over current flag

(1'b1: over current; 1'b0: no over current )

VSIM1_OC_STATUS * VSIM1 over current flag

(1'b1: over current; 1'b0: no over current )

VSIM2_OC_STATUS * VSIM2 over current flag

(1'b1: over current; 1'b0: no over current )

VIBR_OC_STATUS * VIBR over current flag

(1'b1: over current; 1'b0: no over current )

SPK_OC_STATUS classAB over current flog

(1'b1: over current; 1'b0: no over current )

**830108FCh PMIC_OC_FLAG** **PMIC_OC_FLAG**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | VA_OC_FLG | VM_OC_FLG | VIO_OC_FLG | VRF_OC_FLG | VTCXO_OC_FLG | VCORE_OC_FLG | | | VCAMA_OC_FLG | VCAMD_OC_FLG | VUSB_OC_FLG | | VSIM1_OC_FLG | VSIM2_OC_FLG | VIBR_OC_FLG | SPK_OC_FLG |
| Type | RW | RW | RW | RW | RW | RW | | | RW | RW | RW | | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

VA_OC_FLG R : Deglitched signal for VA_CL_STATUS

W: Clear VA_OC_FLG

VM_OC_FLG R : Deglitched signal for VM_CL_STATUS

W: Clear VM_OC_FLG

VIO_OC_FLG R: Deglitched signal for VIO_CL_STATUS

W: Clear VIO_OC_FLG

VRF_OC_FLG R: Deglitched signal for VRF_CL_STATUS

W: Clear VRF_OC_FLG

VTCXO_OC_FLG R: Deglitched signal for VTCXO_CL_STATUS

W: Clear VTCXO_OC_FLG

VCORE_OC_FLG R: Deglitched signal for VCORE_CL_STATUS

W: Clear VCORE_OC_FLG

VCAMA_OC_FLG R: Deglitched signal for VCAMA_CL_STATUS

W: Clear VCAMA_OC_FLG

VCAMD_OC_FLG R: Deglitched signal for VCAMD_CL_STATUS

W: Clear VCAMD_OC_FLG

VUSB_OC_FLG R: Deglitched signal for VUSB_CL_STATUS

W: Clear VUSB_OC_FLG

VSIM1_OC_FLG R: Deglitched signal for VSIM1_CL_STATUS

W: Clear VSIM1_OC_FLG

VSIM2_OC_FLG R: Deglitched signal for VSIM2_CL_STATUS

W: Clear VSIM2_OC_FLG

VIBR_OC_FLG R: Deglitched signal for VIBR_CL_STATUS

W: Clear VIBR_OC_FLG

SPK_OC_FLG R: Deglitched signal for SPK_OC_STATUS

W: Clear SPK_OC_FLG

**83010900h PMIC_STARTUP_CON0** **PMIC_STARTUP_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | PWRKEY_DEB(R) | PWRKEY_VCORE(R) | QI_TEST_MODE_POR(R) | | PMU_THR_PWROFF(R) | QI_PMU_THR_STATUS(R) | | | | | | RG_USBDL_EN | RG_THR_HWPDN_EN | RG_THERMAL_DIS | RG_THR_SEL | |
| Type | R | R | R | | R | R | | | | | | RW | RW | RW | RW | |
| Reset | 0 | 0 | 0 | | 0 | 0 | | | | | | 1 | 0 | 0 | 0 | |

PWRKEY_DEB(R) Debounced PWRKEY signal

PWRKEY_VCORE(R) PWRKEY signal before debounced

QI_TEST_MODE_POR(R) Test mode por signal

PMU_THR_PWROFF(R) PMU thermal power-off Indicator

**0** Normal operation

1 Power-off

QI_PMU_THR_STATUS(R) Startup monitor signal thermal status

RG_USBDL_EN Reserved (no used)

RG_THR_HWPDN_EN

0: Enable thermal hardware powerdown function
1: Disable thermal hardware powerdown function

RG_THERMAL_DIS Thermal shut-down disable control
0: enable thermal shutdown protection
1 : disable thermal shutdown protection

RG_THR_SEL Thermal shut-down threshold fine tuning (Internal use)
00 : Initial setting
01 : +10oC
10 : -20oC
11 : -10oC

**83010904h PMIC_STARTUP_CON1** **PMIC_STARTUP_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | RG_BIAS_GEN_FORCE | | | | | | RG_PMU_LEV_UNGATE | | | RG_RST_DRVSEL | RG_STRUP_TEST | | RG_VREF_BG | | |
| Type | | RW | | | | | | RW | | | RW | RW | | RW | | |
| Reset | | 0 | | | | | | 0 | | | 0 | 0 | | 0 | | |

RG_BIAS_GEN_FORCE Force the IBIAS/VBIAS Generator ON in the Testmode
0 : Normal
1 : Force ON

RG_PMU_LEV_UNGATE Use to ungate PMU related level shifters, active high

0 : Gated
1 : Ungated

RG_RST_DRVSEL Reset pin output driving capability Selection
0: 7.5mA (default)
1: 15mA

RG_STRUP_TEST Startup macro test clock enable signal

0 : Normal clock source
1 : Test clock source to speedup FT

RG_VREF_BG Reference voltage fine tuning according to VBG
000 : initial setting
001 : minus 1 step
010 : minus 2 step
011 : minus 3 step
100 : plus 4 step
101 : plus 3 step
110 : plus 2 step
111 : plus 1 step

**83010908h PMIC_STARTUP_CON2** **PMIC_STARTUP_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | QI_STRUP_FLAG_OUT(R) | | | | QI_PMU_RSTB(R) | QI_PMU_DCXO26M_ON(R) | QI_PMU_DCXO26M_DLY | RG_THRDET_SEL | RG_TP_LED | | | | RG_IBIAS_TRIM | | | |
| Type | R | | | | R | R | R | RW | RW | | | | RW | | | |
| Reset | 0 | | | | 0 | 0 | 0 | 0 | 0 | | | | 0 | | | |

QI_STRUP_FLAG_OUT(R) Startup monitor siganls

QI_PMU_RSTB(R) System Reset Disable Signal

QI_PMU_DCXO26M_ON(R) enable signal for DCXO

QI_PMU_DCXO26M_DLY delayed enable signal for DCXO

RG_THRDET_SEL Thermal detection method:
1: 4-state polling (<110oC/110oC~130oC/110oC~150oC/>150oC)
0: 3-state hysteresis (<110oC/110oC~150oC/>150oC)

RG_TP_LED RG_TP_LED[3:0]: PMU test mode but need to set PAD_PMU_TESTMODE = VBAT

| RG_TP_LED[2:0] | KP_LED | ISINK3 | ISINK2 | ISINK1 | ISINK0 |
|---|---|---|---|---|---|
| 000 | DISABLE STARTUP DEBUG MODE @ PMU_TESTMODE=1 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| 001 | UVLO | DDLO | THERMAL | CHRDET | RTC_PWRBB |
| 010 | CLK75K | PWRKEY | PWRKEY_DEB | PWRON | BGR_READYB |
| 011 | DCXO26M_ON | DCXO26M_DLY | THR_STATUS[2] | THR_STATUS[1] | THR_STATUS[0] |
| 100 | VRF_PG_STATUS | VTCXO_PG_STATUS | VTHR_POL[2] | VTHR_POL[1] | VTHR_POL[0] |
| 101 | VCORE_PG_STATUS | VM_PG_STATUS | VIO_PG_STATUS | VA_PG_STATUS | GND |
| 110 | GND | GND | GND | GND | GND |
| 111 | GND | GND | GND | GND | GND |

RG_TP_LED[3]: reserved

RG_IBIAS_TRIM V/I source bias triming

| | | | |
|---|---|---|---|
| 0000 | +0% | 1000 | -50% |
| 0001 | +6.25% | 1001 | -43.75% |
| 0010 | +12.5% | 1010 | -37.5% |
| 0011 | +18.75% | 1011 | -31.25% |
| 0100 | +25% | 1100 | -25% |
| 0101 | +31.25% | 1101 | -18.75% |
| 0110 | +37.5% | 1110 | -12.5% |
| 0111 | +43.75% | 1111 | -6.25% |

**8301090Ch** **PMIC_STARTUP_CON3** **PMIC_STARTUP_CON3**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RG_IBI | RG_T | RG_SV | RG_STRUP_ | RG_STRUP_FLAG_SE | | | | RG_STRUP_RSV | | | | | | | |

| | AS_TRIM_EN | HR_TMODE | 12_TMODE | FLAG_EN | L | |
|---|---|---|---|---|---|---|
| Type | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 |

RG_IBIAS_TRIM_EN Reserve (Not used)

RG_THR_TMODE

0: Disable thermal threshould input from KP_LED

1: Enalbe thermal threshould input from KP_LED

RG_SV12_TMODE Startup internal 1.2v capless LDO test mode

0: disable test mode
1 : enable test mode to output 1.2v to KP_LED

RG_STRUP_FLAG_EN

0: Disable startup digital macro flag output function
1: Enable startup digital macro flag output function

RG_STRUP_FLAG_SEL Startup digital macro flag output selection control

| RG_STRUP_FLAG_SEL[3:0] | STRUP_FLAG_OUT[3] | STRUP_FLAG_OUT[2] | STRUP_FLAG_OUT[1] | STRUP_FLAG_OUT[0] |
|---|---|---|---|---|
| 0 | PMU_DDLO | uvlo_deb | chrdet_deb | pmu_thr_pwroff |
| 1 | RTC_PWRBB | PMU_PWRKEY_DEB | PMU_PWRKEY | PMU_UVLO |
| 2 | mon_pmu_ck75k_pre | mon_pmu_ck75k | mon_pmu_ck256us | mon_pmu_ck1ms |
| 3 | PMU_TEST_CK | PCHR_CHRDET | 0 | 0 |
| 4 | vcore_pg_deb | vio18_pg_deb | va_pg_deb | 0 |
| 5 | PMU_DCXO26M_ON | PMU_DCXO26M_DLY | 0 | PMU_PWRON |
| 6 | PMU_THERMAL | PMU_THR_STATUS[2] | PMU_THR_STATUS[1] | PMU_THR_STATUS[0] |
| 7 | 0 | VTHR_POL[2] | VTHR_POL[1] | VTHR_POL[0] |
| 8 | 0 | vio28_pg_deb | vtcxo_pg_deb | vrf_pg_deb |
| 9 | 0 | 0 | 0 | 0 |
| 10 | VUSB_STB | VRF_STB | VTCXO_STB | VCORE_STB |
| 11 | VIO18_STB | VIO28_STB | VA_STB | 0 |
| 12 | PMU_TESTMODE | VCORE_PG_STATUS | VIO18_PG_STATUS | VIO28_PG_STATUS |
| 13 | VA_PG_STATUS | VRF_PG_STATUS | VTCXO_PG_STATUS | 0 |
| 14 | 0 | 0 | 0 | 0 |
| 15 | 0 | 0 | 0 | 0 |

RG_STRUP_RSV Reserved

**83010980h PMIC_ISINK0_CON0** **PMIC_ISINK0_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | RG_ISINK0_STEP | | | | | RG_ISINK0_MOD | RG_ISINK0_E |

| | | | | | | | | | | | | | | E | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Type | | | | | | | | | | RW | | | | RW | RW |
| Reset | | | | | | | | | | 0 | | | | 0 | 0 |

RG_ISINK0_STEP Coarse 6 step current level for ISINK CH0, 000:4mA ~ 101:24mA, 4mA per step

| RG_ISINK0_STEP[2:0] | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| ISINK0 current (mA) | 4 | 8 | 12 | 16 | 20 | 24 | - | - |

RG_ISINK0_MODE ISINK0 PWM MODE SEL

1 : Register control mode (see the RG_ISINK0_EN to turn-on)

0 : PWM mode, (controlled by PWM3)

RG_ISINK0_EN Turn on ISINK Channel 0

**83010984h PMIC_ISINK0_CON1** **PMIC_ISINK0_CON1**

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

RG_ISINKS_VREF_CAL Fine Tune reference voltage level for ISINK (0.0675V~0.13V,default [00000]=0.1V).

2.5mV per step.

| VREF_CAL[4:0] | 00000 | 00001 | 00010 | 00011 | 00100 | 00101 | 00110 | 00111 | 01000 | 01001 | 01010 | 01011 | 01100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Vref (mV) | 100.0 | 102.5 | 105.0 | 107.5 | 110.0 | 112.5 | 115.0 | 117.5 | 120.0 | 122.5 | 125.0 | 127.5 | 130.0 |
| VREF_CAL[4:0] | 10011 | 10100 | 10101 | 10110 | 10111 | 11000 | 11001 | 11010 | 11011 | 11100 | 11101 | 11110 | 11111 |
| Vref (mV) | 67.5 | 70.0 | 72.5 | 75.0 | 77.5 | 80.0 | 82.5 | 85.0 | 87.5 | 90.0 | 92.5 | 95.0 | 97.5 |

**83010988h PMIC_ISINK0_CON2** **PMIC_ISINK0_CON2**

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

RG_ISINKS_RSV

RG_ISINKS_RSV[0] : current trimming enable
0: normal operation, 1: bias trimming enable

RG_ISINKS_RSV[2:1] : Reserved register for future use

RG_ISINKS_RSV[3] : ClassAB dis shoot through enable

0: enable dis shoot through, 1: normal operation

**83010990h PMIC_ISINK1_CON0** **PMIC_ISINK1_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | RG_ISINK1_STEP | | | | | RG_ISINK1_MODE | RG_ISINK1_EN |
| Type | | | | | | | | | | RW | | | | | RW | RW |
| Reset | | | | | | | | | | 0 | | | | | 0 | 0 |

RG_ISINK1_STEP Coarse 6 step current level for ISINK CH1, 000:4mA ~ 101:24mA, 4mA per step

| RG_ISINK1_STEP[2:0] | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| ISINK1 current (mA) | 4 | 8 | 12 | 16 | 20 | 24 | - | - |

RG_ISINK1_MODE ISINK1 PWM MODE SEL

1 : Register control mode (see the RG_ISINK1_EN to turn-on)

0 : PWM mode, (controlled by PWM3)

RG_ISINK1_EN Turn on ISINK Channel 1

**830109A0h PMIC_ISINK2_CON0** **PMIC_ISINK2_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | RG_ISINK2_STEP | | | | | RG_ISINK2_MODE | RG_ISINK2_EN |
| Type | | | | | | | | | | RW | | | | | RW | RW |
| Reset | | | | | | | | | | 0 | | | | | 0 | 0 |

RG_ISINK2_STEP Coarse 6 step current level for ISINK CH2, 000:4mA ~ 101:24mA, 4mA per step

| RG_ISINK2_STEP[2:0] | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| ISINK2 current (mA) | 4 | 8 | 12 | 16 | 20 | 24 | - | - |

RG_ISINK2_MODE ISINK2 PWM MODE SEL

1 : Register control mode (see the RG_ISINK2_EN to turn-on)

0 : PWM mode, (controlled by PWM3)

RG_ISINK2_EN Turn on ISINK Channel 2

**830109B0h PMIC_ISINK3_CON0 PMIC_ISINK3_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | | RG_ISINK3_STEP | | | | | RG_ISINK3_MODE | RG_ISINK3_EN |
| Type | | | | | | | | | | RW | | | | | RW | RW |
| Reset | | | | | | | | | | 0 | | | | | 0 | 0 |

RG_ISINK3_STEP Coarse 6 step current level for ISINK CH3, 000:4mA ~ 101:24mA, 4mA per step

| RG_ISINK3_STEP[2:0] | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| ISINK3 current (mA) | 4 | 8 | 12 | 16 | 20 | 24 | - | - |

RG_ISINK3_MODE ISINK3 PWM MODE SEL

1 : Register control mode (see the RG_ISINK3_EN to turn-on)

0 : PWM mode, (controlled by PWM3)

RG_ISINK3_EN Turn on ISINK Channel 3

**830109C0h PMIC_KPLED_CON0 PMIC_KPLED_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | MUST BE 0 | MUST BE 0 | | | RG_KPLED_SEL | | | | | RG_KPLED_MODE | RG_KPLED_EN |
| Type | | | | | | RW | RW | | | RW | | | | | RW | RW |
| Reset | | | | | | 0 | 0 | | | 0 | | | | | 0 | 0 |

RG_KPLED_SEL 3 bits for KPLED current adjustment. 8 steps in total. The minimal current should be no less than 60mA at <111> step.

| RG_KPLED_SEL[2:0] | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| KPLED current | 1X | 2X | 3X | 4X | 5X | 6X | 7X | 8X |

**RG_KPLED_MODE** KPLED enable mode select

**0** pwm mode, controlled by hardware **PWM1** output signal

**1** Register control mode (see the KPLED_EN to turn-on)

RG_KPLED_EN Turn on KeyPAD

**830109D0h** **PMIC_CLASSAB_CON0** **PMIC_CLASSAB_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | RG_SPK_MINUS_GAIN | | RG_SPK_IN_FLOAT_B | RG_SPK_IN_TIE_HIGH | RG_SPK_TESTIN_SEL | RG_SPK_OBIAS | | RG_SPK_OC_EN | RG_SPK_OUT_FLOAT_B | RG_SPK_VOL | | | | RG_SPK_EN |
| Type | | | RW | | RW | RW | RW | RW | | RW | RW | RW | | | | RW |
| Reset | | | 0 | | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | | | | 0 |

RG_SPK_MINUS_GAIN class AB minus gain enable

00: disable

01: -3dB gain

1011 -6dB gain

RG_SPK_IN_FLOAT_B Let OP's input floating when inactive

0 : input floating
1 : input may be tied H/L

RG_SPK_IN_TIE_HIGH Let OP's input tied to high when inactive(functional when RG_SPK_IN_FLOAT_B=1)

0 : input tie to H
1 : input tie to L

RG_SPK_TESTIN_SEL Input source select

0 : normal path
1 : test path

RG_SPK_OBIAS class AB mode output stage bias current select

00: 6mA

01: 9mA
10:12mA
11:15mA

RG_SPK_OC_EN class AB mode Over-current protection enable

0:disable
1:enable

RG_SPK_OUT_FLOAT_B class AB mode Output floating during power down

0:output floating
1:not floating, tie H

RG_SPK_VOL class AB volume control 1.5dB/step

| RG_SPK_VOL[3:0] | Gain [dB] | RG_SPK_VOL[3:0] | Gain [dB] | RG_SPK_VOL[3:0] | Gain [dB] | RG_SPK_VOL[3:0] | Gain [dB] |
|---|---|---|---|---|---|---|---|
| 0000 | 0 | 0100 | 6 | 1000 | 12 | 1100 | 18 |
| 0001 | 1.5 | 0101 | 7.5 | 1001 | 13.5 | 1101 | 19.5 |
| 0010 | 3 | 0110 | 9 | 1010 | 15 | 1110 | 21 |
| 0011 | 4.5 | 0111 | 10.5 | 1011 | 16.5 | 1111 | 22.5 |

RG_SPK_EN SPKDRV Powering Up/Down Control

0:Disable
1:Enable

**830109D4h** **PMIC_CLASSAB_CON1** **PMIC_CLASSAB_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | SPK_OC_FLAG (R) | | | | | | | | | CCI_SPK_OC_AUTO_OFF | | | SPK_OC_WND | | SPK_OC_TRG | |
| Type | R | | | | | | | | | RW | | | RW | | RW | |
| Reset | 0 | | | | | | | | | 0 | | | 0 | | 0 | |

**SPK_OC_FLAG** The flag informs some SPK over current status have been asserted.

**0** no over current status

**1** some over current status

**CCI_SPK_OC_AUTOFF** Enable to power-off the SPK automatically if the SPK_OC_FLAG been asserted.

**0** disable

**1** enable

**SPK_OC_WND** Decision window setting for SPK over current status.

**0** 16uS

**1** 32uS

**2** 64uS

**3** 128uS

**SPK_OC_TRG** Threshold setting in the decision window for SPK over current status.

**0** 4/8

**1** 3/8

**2** 2/8

**3** 1/8

**83010A00h** **PMIC_CHR_CON0** **PMIC_CHR_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | VCDT_HV_DET(R) | VCDT_LV_DET(R) | CHRDET(R) | CHR_EN | CSDAC_EN | PCHR_AUTO | CHR_LDO_DET | VCDT_HV_EN | VCDT_HV_VTH | | | | VCDT_LV_VTH | | | |
| Type | R | R | R | RW | RW | RW | R | RW | RW | | | | RW | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1011 | | | | 10 | | | |

VCDT_HV_DET(R) ChargerIn high threshould detection (1: > vth, 0 < vth)

VCDT_LV_DET(R) ChargerIn low threshould detection (1: > vth, 0 < vth)

CHRDET(R) Charger detection output (1: detected, 0: not detected)

CHR_EN Enable current DAC & charger auto charing control

CSDAC_EN Current DAC driver enable

PCHR_AUTO Charger auto charging control

CHR_LDO_DET Charger LDO detection (1: detected, 0: not detected)

VCDT_HV_EN ChargerIn detection high threshold enable

VCDT_HV_VTH ChargerIn detection high threshold

0000:4.2V / 0001:4.25V / 0010:4.30V / 0011:4.35V / 0100:4.40V / 0101:4.45V / 0110:4.50V / 0111:4.55V

1000:4.60V / 1001:6.00V / 1010:6.50V / 1011:7.00V / 1100:7.50V / 1101:8.50V / 1110:9.50V / 1111:10.0V /

VCDT_LV_VTH ChargerIn detection low threshold

0000:4.2V / 0001:4.25V / 0010:4.30V / 0011:4.35V / 0100:4.40V / 0101:4.45V / 0110:4.50V / 0111:4.55V

1000:4.60V / 1001:6.00V / 1010:6.50V / 1011:7.00V / 1100:7.50V / 1101:8.50V / 1110:9.50V / 1111:10.0V /

**83010A04h PMIC_CHR_CON1** **PMIC_CHR_CON1**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | VBAT_CC_DET(R) | VBAT_CV_DET(R) | | | | | VBAT_CC_EN | VBAT_CV_EN | VBAT_CC_VTH | | | VBAT_CV_VTH | | | | |
| Type | R | R | | | | | RW | RW | RW | | | RW | | | | |
| Reset | 0 | 0 | | | | | 1 | 0 | 10 | | | 0 | | | | |

VBAT_CC_DET(R) Battery CC-voltage detection output (1: > vth, 0 < vth)

VBAT_CV_DET(R) Battery CV-voltage detection output (1: > vth, 0 < vth)

VBAT_CC_EN Battery CC detection enable

VBAT_CV_EN Battery CV detection enable

VBAT_CC_VTH Battery CC detection threshold

00: 3.250V

01: 3.275V

10: 3.300V

11: 3.325V

VBAT_CV_VTH Battery CV detection threshold

00000:4.000V ; 00001:4.0125 ; 00010:4.0250V ; 00011:4.0375 ; 00100:4.0500V ; 00101:4.0625 ;

00110:4.0675V ; 00111:4.0875 ; 01000:4.1000V ; 01001:4.1125 ; 01010:4.1250V ; 01011:4.1375 ; 01100:4.1500V ; 01101:4.1625 ; 01110:4.1750V ; 01111:4.1875 ; 1X000:4.2000V ; 1X001:4.2125 ; 1X010:4.2250V ; 1X011:4.2375 ; 1X100:4.2500V ; 1X101:4.2625 ; 1X110:4.2750V ; 1X111:4.2875 ; 11111:2.2000V ;

**83010A08h PMIC_CHR_CON2** **PMIC_CHR_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CS_DET(R) | | | CS_EN | | CS_VTH | | | | TOLTC | | | | TOHTC | | |
| Type | R | | | RW | | RW | | | | RW | | | | RW | | |
| Reset | 0 | | | 0 | | 111 | | | | 0 | | | | 0 | | |

CS_DET(R) Current sense comparator output (1: > Vth, 0 < Vth)

CS_EN Current sense enable

CS_VTH Current sense threshold.

000: 800mA ; 001:700mA ; 010:650mA ; 011:550mA ; 100:450mA ; 101:400mA ; 110:200mA ; 111:70mA

TOLTC Charger control cycles for driving low

TOHTC Charger control cycles for driving high

**83010A0Ch PMIC_CHR_CON3** **PMIC_CHR_CON3**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | BATON_UNDET(R) | VBAT_OV_DET(R) | | | VBAT_OV_DEG | BATON_HT_EN | BATON_EN | VBAT_OV_EN | VBAT_OV_VTH | | CSDAC_DLY | | | | CSDAC_STP | |
| Type | R | R | | | RW | RW | RW | RW | RW | | RW | | | | RW | |
| Reset | 0 | 0 | | | 0 | 0 | 1 | 1 | 1 | | 11 | | | | 1 | |

BATON_UNDET(R) Battery-On undetected (1: not detected, 0: detected)

VBAT_OV_DET(R) Battery over-voltage detection output (1: > vth, 0 < vth)

VBAT_OV_DEG Battery over-voltag deglitch (1: enable, 0: disable)

BATON_HT_EN Battery-On high temperature detection (1: enable, 0: disable)

BATON_EN Battery-On detection for drving protection (1: enable, 0: disable)

VBAT_OV_EN Battery over-voltag for driving protection (1: enable, 0: disable)

VBAT_OV_VTH Battery over-voltag detection threshold

OV_VTH_H: 00:4.325V ; 01:4.350V ; 10: 4.375V ; 11:4.1150V

OV_VTH_L: 00:4.275V ; 01:4.300V ; 10: 4.325V ; 11:4.0400V (Hysteresis)

CSDAC_DLY Current DAC output step timer

CSDAC_STP Current DAC output step

**83010A10h PMIC_CHR_CON4** **PMIC_CHR_CON4**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CSDAC_DAT | | | | | | | | | | | | | PCHR_RST | CSDAC_TEST | PCHR_TEST |
| Type | RW | | | | | | | | | | | | | RW | RW | RW |
| Reset | 0 | | | | | | | | | | | | | 0 | 0 | 0 |

CSDAC_DAT Current DAC driver data

PCHR_RST Charger control reset

CSDAC_TEST Current DAC driver test mode

PCHR_TEST Charger control test mode

**83010A14h PMIC_CHR_CON5** **PMIC_CHR_CON5**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | OTG_BVALID(R) | | | OTG_BVALID_EN | | | | | PCHR_FLAG_EN | | | | | | | |
| Type | R | | | RW | | | | | RW | | | | | | | |
| Reset | 0 | | | 0 | | | | | 0 | | | | | | | |

OTG_BVALID(R) OTG BValid detected (1: detected, 0: not detected)

OTG_BVALID_EN OTG BValid detection enable (1: enable, 0: disable)

PCHR_FLAG_EN Charger control debug_flag_en

**83010A18h PMIC_CHR_CON6** **PMIC_CHR_CON6**

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

CHRWDT_EN Charger control watchdog enable

CHRWDT_TD Charger control watchdog delay

**83010A1Ch PMIC_CHR_CON7** **PMIC_CHR_CON7**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CHRWDT_OUT | | | | | | | | | | | | | | CHRWDT_FLAG | CHRWDT_INT_EN |
| Type | R | | | | | | | | | | | | | | R | RW |
| Reset | 0 | | | | | | | | | | | | | | 0 | 0 |

CHRWDT_OUT Charger watchdog output

CHRWDT_FLAG Charger watchdog flag

CHRWDT_INT_EN Charger watchdog interrupt enable

**83010A20h PMIC_CHR_CON8** **PMIC_CHR_CON8**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | ADCIN_VCHR_EN | ADCIN_VSEN_EN | ADCIN_VBAT_EN | USBDL_SET | USBDL_RST | UVLO_VTHL | | | | BGR_UNCHOP | BGR_UNCHOP_PH | | BGR_RSEL | | |
| Type | | RW | RW | RW | RW | RW | RW | | | | RW | RW | | RW | | |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | | | | 0 | 0 | | 0 | | |

ADCIN_VCHR_EN AUXADC input source enable for CHR (1: enable, 0: disable)

ADCIN_VSEN_EN AUXADC input source enable for VSEN (1: enable, 0: disable)

ADCIN_VBAT_EN AUXADC input source enable for VBAT (1: enable, 0: disable)

USBDL_SET USBDL model set

USBDL_RST USBDL model reset

UVLO_VTHL UVLO low threshold selection

00:2.5V ; 01:2.6V ; 10:2.75V ; 11:2.9V

BGR_UNCHOP BGR unchop mode (1: unchop, 0: chop)

BGR_UNCHOP_PH BGR unchop mode phase selection

BGR_RSEL BGR resistor selection

**83010A24h** **PMIC_CHR_CON9** **PMIC_CHR_CON9**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RGS_BC11_CMP_OUT(R) | | | | RG_BC11_VSRC_EN | | BC11_RST | BC11_BB_CTRL | BC11_BIAS_EN | BC11_IPU_EN | | BC11_IPD_EN | | BC11_CMP_EN | | BC11_VREF_VTH |
| Type | R | | | | RW | | RW | RW | RW | RW | | RW | | RW | | RW |
| Reset | 0 | | | | 0 | | 0 | 0 | 0 | 0 | | 0 | | 0 | | 0 |

RGS_BC11_CMP_OUT(R) BC1.1 comparator output logical signal stored and can be monitored.

RG_BC11_VSRC_EN BB to CGR-BC1.1 signal, Enable bits of Voltage buffer of USB_DP,USB_DM.

BC11_RST BC 1.1 control reset signal (rising edge).

BC11_BB_CTRL BB to CGR-BC1.1 signal. The control bit enable the BC1.1 circuit control from BB.

BC11_BIAS_EN BB to CGR-BC1.1 signal. The bit is to enable the bias circuits of BC1.1 circuit.
0:disable/1:enable.

BC11_IPU_EN BB to CGR-BC1.1 signal.

The bit is to enable the IPU10u current source of BC1.1 circuit. 0:disable/1:enable.

BC11_IPD_EN BB to CGR-BC1.1 signal.

The bit is to enable the IPD100u current source of BC1.1 circuit. 0:disable /1:enable.

BC11_CMP_EN BB to CGR-BC1.1 signal.

The bit is to enable the comparator circuit of BC1.1 circuit. 0:disable /1:enable.

BC11_VREF_VTH BB to CGR-BC1.1 signal

The bit is to select the threshold voltage of BC1.1 comparator input. 0:0.330V / 1:1.146V.

**83010A28h PMIC_CHR_CON10** **PMIC_CHR_CON10**

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

RESERVED Reserved, do not write anything to this register

**83010A2Ch PMIC_CHR_CON11** **PMIC_CHR_CON11**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | PCHR_FLAG_OUT(R) | | | | | | | | PCHR_FLAG_SEL | | | | | | | |
| Type | R | | | | | | | | RW | | | | | | | |
| Reset | 0 | | | | | | | | 0 | | | | | | | |

PCHR_FLAG_OUT(R) Charger digital controller internal state flags

PCHR_FLAG_SEL Charger control debug flag signal selection. Note: PCHR_FLAG_EN should be enabled

| PCHR_FLAG_SEL | MON[3] | MON[2] | MON[1] | MON[0] |
|---|---|---|---|---|
| 5'h00 | QI_CS_DET | QI_VBAT_CV_DET | QI_VBAT_CC_DET | QI_VCDT_DET |
| 5'h01 | QI_DDLO_DET | QI_UVLO_DET | QI_BATON_UNDET | QI_VBAT_OV_DET |
| 5'h02 | CHRWDT_OUT | PCHR_STATE[2] | PCHR_STATE[1] | PCHR_STATE[0] |
| 5'h03 | QI_DRV_EN | SFSTR_EN | SFSTR_STATE[1] | SFSTR_STATE[0] |
| 5'h04 | BBCTL_FLAG | VBAT_CC_FLAG | VCDT_HV_FLAG | VCDT_LV_FLAG |
| 5'h05 | QI_CS_EN | QI_VCDT_EN | QI_VBAT_CV_EN | QI_VBAT_CC_EN |
| 5'h06 | DDLO_DEB | UVLO_DEB | BATON_UNDET_DEB | VBAT_OV_DEB |
| 5'h07 | 1'b0 | RGS_USBDL_MODE | QI_USBDL_MODE | USBDL_4scnt_out |
| 5'h08 | PCHR_RSTB | PCHR_CK1MS | PCHR_CK1US | PCHR_CK16US |
| 5'h09 | DDLO_CALI_DAT[3] | DDLO_CALI_DAT[2] | DDLO_CALI_DAT[1] | DDLO_CALI_DAT[0] |
| 5'h0A | QI_DRV_D[7] | QI_DRV_D[6] | QI_DRV_D[5] | QI_DRV_D[4] |
| 5'h0B | QI_DRV_D[3] | QI_DRV_D[2] | QI_DRV_D[1] | QI_DRV_D[0] |
| 5'h0C | QI_VBAT_CV_VTH[4] | QI_CS_VTH[2] | QI_CS_VTH[1] | QI_CS_VTH[0] |
| 5'h0D | QI_VBAT_CV_VTH[3] | QI_VBAT_CV_VTH[2] | QI_VBAT_CV_VTH[1] | QI_VBAT_CV_VTH[0] |
| 5'h0E | DDLO_CALI_OK | VBAT_CV_FLAG | QI_VBAT_CC_VTH[1] | QI_VBAT_CC_VTH[0] |
| 5'h0F | QI_VCDT_VTH[3] | QI_VCDT_VTH[2] | QI_VCDT_VTH[1] | QI_VCDT_VTH[0] |
| 5'h10 | mon_pchr_ckrtc | mon_bc11_rtc_ck1sec | bc11_rtc_rpen | bc11_rtc_timeout |
| 5'h11 | bc11_500ms_rpen | bc11_500ms_timeout | bc11_1M_rpen | bc11_1M_timeout |
| 5'h12 | v22_deb | bc11_rpen | bc11_charger_det | bc11_det_finish |
| 5'h13 | pchr_bc11_bias_en | pchr_bc11_vref_vth | usbldo_force_en | csdac_isusp |
| 5'h14 | pchr_bc11_ipd_en[1] | pchr_bc11_ipd_en[0] | pchr_bc11_ipu_en[1] | pchr_bc11_ipu_en[0] |
| 5'h15 | pchr_bc11_vsrc_en[1] | pchr_bc11_vsrc_en[0] | pchr_bc11_cmp_en[1] | pchr_bc11_cmp_en[0] |
| 5'h16 | bc11_rstb | mon_bc11_ck1us | dead_lat | QI_PCHR_BC11_CMP_OUT |
| 5'h17 | bc11_cnt[10] | bc11_cnt[9] | bc11_cnt[3] | bc11_cnt[2] |
| 5'h18 | bc11_500ms_rpcnt[9] | bc11_500ms_rpcnt[8] | bc11_500ms_rpcnt[1] | bc11_500ms_rpcnt[0] |
| 5'h19 | bc11_1024ms_rpcnt[9] | bc11_1024ms_rpcnt[8] | bc11_1024ms_rpcnt[1] | bc11_1024ms_rpcnt[0] |
| 5'h1A | bc11_720sec_rpcnt[9] | bc11_720sec_rpcnt[8] | bc11_720sec_rpcnt[1] | bc11_720sec_rpcnt[0] |
| 5'h1B | bc11_rtc_32768_rpcnt[14] | bc11_rtc_32768_rpcnt[13] | bc11_rtc_32768_rpcnt[1] | bc11_rtc_32768_rpcnt[0] |
| 5'h1C | bc11_rtc_1500sec_rpcnt[10] | bc11_rtc_1500sec_rpcnt[9] | bc11_rtc_1500sec_rpcnt[1] | bc11_rtc_1500sec_rpcnt[0] |
| 5'h1D | bc11_state[7] | bc11_state[6] | bc11_state[5] | bc11_state[4] |
| 5'h1E | bc11_state[3] | bc11_state[2] | bc11_state[1] | bc11_state[0] |
| 5'h1F | bc11_512ms_timeout | detect_stop | 1'b1 | 1'b0 |

**83010A30h PMIC_CHR_CON12** **PMIC_CHR_CON12**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | BAT | | | | | | | | | | | | PCHR_FT_CTRL | | | |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ON_UNDET2(R) | | | | | | | | | | | | | | | |
| Type | R | | | | | | | | | | | | RW | | | |
| Reset | 0 | | | | | | | | | | | | 0 | | | |

BATON_UNDET2(R) Reserved read only bit of battery-on detection function.

PCHR_FT_CTRL Charger FT mode control bits

bit 2 extend the charger detection duration 10ms
bit 1 speed up detection, force TOH/TOL = 3ms/1ms
bit 0 always turn on charger detection (VCDT)

**83010A34h PMIC_CHR_CON13** **PMIC_CHR_CON13**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | PCHR_TRIMDATA1(R) | | | | PCHR_TRIMDATA0(R) | | | | - | | | | - | | | |
| Type | R | | | | R | | | | - | | | | - | | | |
| Reset | 0 | | | | 0 | | | | 0 | | | | 0 | | | |

PCHR_TRIMDATA1(R) Reserved read only of trimmed data.

PCHR_TRIMDATA0(R) Reserved read only of trimmed data.

**83010A40h PMIC_CHR_CON16** **PMIC_CHR_CON16**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | RG_DRV_ITUNE | | RG_OVP_TRIM | | | | RG_PCHR_RV | | | | | | | |
| Type | | | RW | | RW | | | | RW | | | | | | | |
| Reset | | | 1 | | 1000 | | | | 0 | | | | | | | |

RG_DRV_ITUNE : IDAC 1LSB current resolution option. 00:1X , 01:0.5X, 10:0.5X, 11:1/3X

RG_OVP_TRIM[3:0] : 4bit OV trimming level selection. These registers mapping can calibrate -3.5%~4% error.

RG_PCHR_RV Reserved for pchr

**83010F00h FMTR_CON0** **FMTR_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FQMTR_EN | FQMTR_RST | | | | | FQMTR_WINSET | | | | | | | | | |
| Type | RW | RW | | | | | RW | | | | | | | | | |
| Reset | 0 | 0 | | | | | 0 | | | | | | | | | |

**FQMTR_EN** Frequency-meter enable control signal

**0** disable

**1** enable

**FQMTR_RST** Frequency-meter reset control signal

**0** normal operation

**1** reset

**FQMTR_WINSET** Frequency-meter measurement window setting (= numbers of FIXED clock cycles)

**83010F04h FMTR_CON1** **FMTR_CON1**

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

**FQMTR_FCKSEL** Frequency-meter FIXED clock selection

**0** CLKSQ 13MHz clock

**1** RTC XOSC 32KHz clock

**FQMTR_TCKSEL** Frequency-meter TESTED clock selection

**0** idle

**1** CLKSQ 26MHz clock

**2** MPLL clock

**3** UPLL 104M clock

**4** USB PHY 30MHz clock

**5** SPLL clock

**6** UPLL 78M cock

**7** RTC XOSC 32KHz clock

**8** USB clock

9 48M clock

**10~15** reserved

**83010F08h** **FMTR_DATA** **FMTR_DATA**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FQMTR_BUSY | | | | FQMTR_DATA | | | | | | | | | | | |
| Type | R | | | | R | | | | | | | | | | | |
| Reset | 0 | | | | 0 | | | | | | | | | | | |

Set this register for Frequency-Meter Data registers.

**FQMTR_BUSY** Frequency-meter busy status

**0** FQMTR is ready

**1** FQMTR is busy

**FQMTR_DATA** Frequency-meter measurement data

**** Frequency(TESTED clock) = Frequency(FIXED clock) * FQMTR_DATA[11:0]/FQMTR_WINSET[9:0]

**83010F10h** **MIXEDSYS_MON_CON0** **MIXEDSYS_MON_CON0**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | MIXEDSYS_MON_CFG | | | MIXEDSYS_MON_SEL | | | | |
| Type | | | | | | | | | RW | | | RW | | | | |
| Reset | | | | | | | | | 0 | | | 1Fh | | | | |

**MIXEDSYS_MON_CFG** monitor flags output mode select

**0** Normal monitor signals output mode

**1** MCU programmable debug mode (output MIXEDSYS_MON_OUT[7:0])

**MIXEDSYS_MON_SEL** monitor flags output select

| X | MON[7] | MON[6] | MON[5] | MON[4] |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | MON_ESDM_CK |
| 01 | 0 | 0 | 0 | 0 |
| 02 | DA_AYDL[7] | DA_AYDL[6] | DA_AYDL[5] | DA_AYDL[4] |
| 03 | DA_AYDR[7] | DA_AYDR[6] | DA_AYDR[5] | DA_AYDR[4] |
| 04 | 0 | 0 | QI_EDGERFRX_PWDB | 0 |
| 05 | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 06 | 0 | 0 | 0 | VBAT_OV_DET_TMUX |
| 07 | 0 | 0 | AUX_COMP_TMUX | AUX_PENIRQ_TMUX |
| 08 | 0 | 0 | RG_PCHR_AUTOMODE | NI_PCHR_TEST_CK |
| 09 | 0 | 0 | OTG_BVALID_TMUX | RG_CSDAC_EN |
| 10 | DA_APC_BUS[7] | DA_APC_BUS[6] | DA_APC_BUS[5] | DA_APC_BUS[4] |
| 11 | AUX_ST | AUX_CS_B | AUX_DIN | NI_IRQ_PWDB |
| 12 | AUX_COMP_TMUX | AUX_PENIRQ_TMUX | NI_ADC_PWDB | MON_AUX_SCLK |
| 13 | GPIO_SRST2_IN | RG_VSIM1_EN | RG_VSIM1_VOSEL | QI_SIMRST |
| 14 | GPIO_SCLK2_IN | RG_VSIM2_EN | C2A_SIM2SEL | QI_SIMRST2_OUT |
| 15 | 0 | 0 | EN_VTCXO | EN_BB |
| 16 | 0 | 0 | 0 | 0 |
| 17 | PMU_THR_STATUS_TMUX[2] | PMU_THR_STATUS_TMUX[1] | PMU_THR_STATUS_TMUX[0] | VA_OC_STATUS_TMUX |
| 18 | VCAMA_OC_STATUS_TMUX | VCAMD_OC_STATUS_TMUX | VUSB_OC_STATUS_TMUX | VCORE_OC_STATUS_TMUX |
| 19 | C2A_INTERNAL_PWM | C2A_INTERNAL_PWM2 | PMU_OC_INT | VA_OC_INT |
| 20 | VCAMA_OC_INT | VCAMD_OC_INT | VUSB_OC_INT | 0 |
| 21 | 0 | PWRKEY_VCORE_TMUX | PWRKEY_DEB_TMUX | TEST_MODE_POR_TMUX |
| 22 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | QI_VCAMA_SOFT_STB | QI_VCAMD_SOFT_STB |
| 24 | 0 | 0 | MON_USB_CLK_INT | MON_FMCU_CK |
| 25 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 |
| 27 | STRUP_FLAG_OUT_TMUX[3] | STRUP_FLAG_OUT_TMUX[2] | STRUP_FLAG_OUT_TMUX[1] | STRUP_FLAG_OUT_TMUX[0] |
| 28 | RG_CHRWDT_WR | QI_UPLL_PWD | QI_VUPG[5] | QI_VUPG[4] |
| 29 | QI_VRF_EN | QI_VTCXO_EN | RG_VCAMA_EN | RG_VCAMD_EN |

| | | | | |
|---|---|---|---|---|
| 30 | RG_MPLL_FBDIV[7] | RG_MPLL_FBDIV[6] | RG_MPLL_FBDIV[5] | RG_MPLL_FBDIV[4] |
| 31 | 0 | 0 | RG_ISINKS_CH0_EN | RG_ISINKS_CH1_EN |

| X | MON[3] | MON[2] | MON[1] | MON[0] |
|---|---|---|---|---|
| 00 | MON_VSDM_CK | MON_ASDM_CK | A2T_VTXSOUT | NI_ZCD_STATE_TMUX |
| 01 | 0 | 0 | DA_AYDL[8] | DA_AYDR[8] |
| 02 | DA_AYDL[3] | DA_AYDL[2] | DA_AYDL[1] | DA_AYDL[0] |
| 03 | DA_AYDR[3] | DA_AYDR[2] | DA_AYDR[1] | DA_AYDR[0] |
| 04 | A2T_EDGERFRX_BNI | A2T_EDGERFRX_BPI | A2T_EDGERFRX_BNQ | A2T_EDGERFRX_BPQ |
| 05 | CHRWDT_OUT_TMUX | CHRDET_TMUX | CHR_LDO_DET_TMUX | CHR_EN |
| 06 | VBAT_CV_DET_TMUX | VBAT_CC_DET_TMUX | VCDT_HV_DET_TMUX | VCDT_LV_DET_TMUX |
| 07 | AUX_ST | AUX_CS_B | CS_DET_TMUX | BATON_UNDET_TMUX |
| 08 | PCHR_FLAG_OUT_TMUX[3] | PCHR_FLAG_OUT_TMUX[2] | PCHR_FLAG_OUT_TMUX[1] | PCHR_FLAG_OUT_TMUX[0] |
| 09 | NI_APC_PWDB | NI_APC_TG | DA_APC_BUS[9] | DA_APC_BUS[8] |
| 10 | DA_APC_BUS[3] | DA_APC_BUS[2] | DA_APC_BUS[1] | DA_APC_BUS[0] |
| 11 | NI_SEL[3] | NI_SEL[2] | NI_SEL[1] | NI_SEL[0] |
| 12 | AUX_COMP_TMUX | AUX_PENIRQ_TMUX | A2T_AUX_SFSO | A2T_AUX_SDO |
| 13 | NI_SIMCLK | NI_SIMDATA_OE | A2C_SIMDATA_IN | NI_SIMDATA_OUT |
| 14 | NI_SIMCLK2_OUT | NI_SIMDATA2_OE | A2C_SIM2DATA_IN | NI_SIMDATA2_OUT |
| 15 | EN_EXT | DCXO_DELAY | PLL_PWDB | EN_DCXO |
| 16 | SPK_OCP_DET_TMUX | SPK_OCN_DET_TMUX | SPK_OC_DET_TMUX | PMU_THR_PWROFF |
| 17 | VM_OC_STATUS_TMUX | VIO_OC_STATUS_TMUX | VRF_OC_STATUS_TMUX | VTCXO_OC_STATUS_TMUX |
| 18 | VSIM1_OC_STATUS_TMUX | VSIM2_OC_STATUS_TMUX | VIBR_OC_STATUS_TMUX | SPK_OC_DET_TMUX |
| 19 | VM_OC_INT | VIO_OC_INT | VRF_OC_INT | VTCXO_OC_INT |
| 20 | VSIM1_OC_INT | VSIM2_OC_INT | VIBR_OC_INT | SPK_OC_INT |

| | | | | |
|---|---|---|---|---|
| 21 | PMU_RSTB_TMUX | CHRDET_TMUX | PMU_DCXO26M_ON_TMUX | PMU_DCXO26M_DLY_TMUX |
| 22 | 0 | QI_UPLL_PWD | SPK_OC_DET_TMUX | SPK_OC_DEG |
| 23 | QI_VUSB_SOFT_STB | QI_VSIM1_SOFT_STB | QI_VSIM2_SOFT_STB | QI_VIBR_SOFT_STB |
| 24 | MON_FDSP_CK | MON_FUSB_CK | MON_FGSM_CK | MON_F48M_CK |
| 25 | MON_CLKSQ_26M_CK | MON_FMCU_CK | MPLL_FHPRD | MPLL_CLKSWPRD |
| 26 | MON_F32K_CK | FQMTR_BUSY | MON_FQMTR_CK | MON_CLK26M_CK |
| 27 | QI_CLKSQ_CKSEL | DA_KPLED_EN | RG_VIBR_EN | QI_CLKSQ_DIFF_PWDB |
| 28 | QI_VUPG[3] | QI_VUPG[2] | QI_VUPG[1] | QI_VUPG[0] |
| 29 | RG_VUSB_EN | QI_CLKSQ_SIN_PWDB | QI_MPLL_PWD | RG_SPK_EN |
| 30 | RG_MPLL_FBDIV[3] | RG_MPLL_FBDIV[2] | RG_MPLL_FBDIV[1] | RG_MPLL_FBDIV[0] |
| 31 | RG_ISINKS_CH2_EN | RG_ISINKS_CH3_EN | CHR_EN | DA_SPK_EN |

**83010F14h MIXEDSYS_MON_CON1** **MIXEDSYS_MON_CON1**

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

Set this register for Monitor circuit configuration controls.

**MIXEDSYS_MON_OUT** monitor debug output

**83010F18h MIXEDSYS_MON_CON2** **MIXEDSYS_MON_CON2**

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | | | | | | | | | MIXEDSYS_MON_DATA | | | | | | | |
| Type | | | | | | | | | R | | | | | | | |
| Reset | | | | | | | | | 0 | | | | | | | |

Set this register for MIXEDSYS Monitor Data configuration controls.

**MIXEDSYS_MON_DATA** monitor debug output

**83010F20h ABB_MON_CON0** **ABB_MON_CON0**

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

| Type | | | | | | | | | | | | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reset | | | | | | | | | | | | 1Fh |

**ABB_MON_SEL** monitor flags output select

| X | MON[7] | MON[6] | MON[5] | MON[4] |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | MON_ESDM_CK |
| 01 | 0 | 0 | 0 | 0 |
| 02 | DA_AYDL[7] | DA_AYDL[6] | DA_AYDL[5] | DA_AYDL[4] |
| 03 | DA_AYDR[7] | DA_AYDR[6] | DA_AYDR[5] | DA_AYDR[4] |
| 04 | 0 | 0 | QI_EDGERFRX_PWDB | 0 |
| 05 | 0 | 0 | 0 | 0 |
| 06 | 0 | 0 | 0 | VBAT_OV_DET_TMUX |
| 07 | 0 | 0 | AUX_COMP_TMUX | AUX_PENIRQ_TMUX |
| 08 | 0 | 0 | RG_PCHR_AUTOMODE | NI_PCHR_TEST_CK |
| 09 | 0 | 0 | OTG_BVALID_TMUX | RG_CSDAC_EN |
| 10 | DA_APC_BUS[7] | DA_APC_BUS[6] | DA_APC_BUS[5] | DA_APC_BUS[4] |
| 11 | AUX_ST | AUX_CS_B | AUX_DIN | NI_IRQ_PWDB |
| 12 | AUX_COMP_TMUX | AUX_PENIRQ_TMUX | NI_ADC_PWDB | MON_AUX_SCLK |
| 13 | GPIO_SRST2_IN | RG_VSIM1_EN | RG_VSIM1_VOSEL | QI_SIMRST |
| 14 | GPIO_SCLK2_IN | RG_VSIM2_EN | C2A_SIM2SEL | QI_SIMRST2_OUT |
| 15 | 0 | 0 | EN_VTCXO | EN_BB |
| 16 | 0 | 0 | 0 | 0 |
| 17 | PMU_THR_STATUS_TMUX[2] | PMU_THR_STATUS_TMUX[1] | PMU_THR_STATUS_TMUX[0] | VA_OC_STATUS_TMUX |
| 18 | VCAMA_OC_STATUS_TMUX | VCAMD_OC_STATUS_TMUX | VUSB_OC_STATUS_TMUX | VCORE_OC_STATUS_TMUX |
| 19 | C2A_INTERNAL_PWM | C2A_INTERNAL_PWM2 | PMU_OC_INT | VA_OC_INT |
| 20 | VCAMA_OC_INT | VCAMD_OC_INT | VUSB_OC_INT | 0 |

| | | | | |
|---|---|---|---|---|
| 21 | 0 | PWRKEY_VCORE_TMUX | PWRKEY_DEB_TMUX | TEST_MODE_POR_TMUX |
| 22 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | QI_VCAMA_SOFT_STB | QI_VCAMD_SOFT_STB |
| 24 | 0 | 0 | MON_USB_CLK_INT | MON_FMCU_CK |
| 25 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 |
| 27 | STRUP_FLAG_OUT_TMUX[3] | STRUP_FLAG_OUT_TMUX[2] | STRUP_FLAG_OUT_TMUX[1] | STRUP_FLAG_OUT_TMUX[0] |
| 28 | RG_CHRWDT_WR | QI_UPLL_PWD | QI_VUPG[5] | QI_VUPG[4] |
| 29 | QI_VRF_EN | QI_VTCXO_EN | RG_VCAMA_EN | RG_VCAMD_EN |
| 30 | RG_MPLL_FBDIV[7] | RG_MPLL_FBDIV[6] | RG_MPLL_FBDIV[5] | RG_MPLL_FBDIV[4] |
| 31 | 0 | 0 | RG_ISINKS_CH0_EN | RG_ISINKS_CH1_EN |

| X | MON[3] | MON[2] | MON[1] | MON[0] |
|---|---|---|---|---|
| 00 | MON_VSDM_CK | MON_ASDM_CK | A2T_VTXSOUT | NI_ZCD_STATE_TMUX |
| 01 | 0 | 0 | DA_AYDL[8] | DA_AYDR[8] |
| 02 | DA_AYDL[3] | DA_AYDL[2] | DA_AYDL[1] | DA_AYDL[0] |
| 03 | DA_AYDR[3] | DA_AYDR[2] | DA_AYDR[1] | DA_AYDR[0] |
| 04 | A2T_EDGERFRX_BNI | A2T_EDGERFRX_BPI | A2T_EDGERFRX_BNQ | A2T_EDGERFRX_BPQ |
| 05 | CHRWDT_OUT_TMUX | CHRDET_TMUX | CHR_LDO_DET_TMUX | CHR_EN |
| 06 | VBAT_CV_DET_TMUX | VBAT_CC_DET_TMUX | VCDT_HV_DET_TMUX | VCDT_LV_DET_TMUX |
| 07 | AUX_ST | AUX_CS_B | CS_DET_TMUX | BATON_UNDET_TMUX |
| 08 | PCHR_FLAG_OUT_TMUX[3] | PCHR_FLAG_OUT_TMUX[2] | PCHR_FLAG_OUT_TMUX[1] | PCHR_FLAG_OUT_TMUX[0] |
| 09 | NI_APC_PWDB | NI_APC_TG | DA_APC_BUS[9] | DA_APC_BUS[8] |
| 10 | DA_APC_BUS[3] | DA_APC_BUS[2] | DA_APC_BUS[1] | DA_APC_BUS[0] |
| 11 | NI_SEL[3] | NI_SEL[2] | NI_SEL[1] | NI_SEL[0] |

| | | | | |
|---|---|---|---|---|
| 12 | AUX_COMP_TMUX | AUX_PENIRQ_TMUX | A2T_AUX_SFSO | A2T_AUX_SDO |
| 13 | NI_SIMCLK | NI_SIMDATA_OE | A2C_SIMDATA_IN | NI_SIMDATA_OUT |
| 14 | NI_SIMCLK2_OUT | NI_SIMDATA2_OE | A2C_SIM2DATA_IN | NI_SIMDATA2_OUT |
| 15 | EN_EXT | DCXO_DELAY | PLL_PWDB | EN_DCXO |
| 16 | SPK_OCP_DET_TMUX | SPK_OCN_DET_TMUX | SPK_OC_DET_TMUX | PMU_THR_PWROFF |
| 17 | VM_OC_STATUS_TMUX | VIO_OC_STATUS_TMUX | VRF_OC_STATUS_TMUX | VTCXO_OC_STATUS_TMUX |
| 18 | VSIM1_OC_STATUS_TMUX | VSIM2_OC_STATUS_TMUX | VIBR_OC_STATUS_TMUX | SPK_OC_DET_TMUX |
| 19 | VM_OC_INT | VIO_OC_INT | VRF_OC_INT | VTCXO_OC_INT |
| 20 | VSIM1_OC_INT | VSIM2_OC_INT | VIBR_OC_INT | SPK_OC_INT |
| 21 | PMU_RSTB_TMUX | CHRDET_TMUX | PMU_DCXO26M_ON_TMUX | PMU_DCXO26M_DLY_TMUX |
| 22 | 0 | QI_UPLL_PWD | SPK_OC_DET_TMUX | SPK_OC_DEG |
| 23 | QI_VUSB_SOFT_STB | QI_VSIM1_SOFT_STB | QI_VSIM2_SOFT_STB | QI_VIBR_SOFT_STB |
| 24 | MON_FDSP_CK | MON_FUSB_CK | MON_FGSM_CK | MON_F48M_CK |
| 25 | MON_CLKSQ_26M_CK | MON_FMCU_CK | MPLL_FHPRD | MPLL_CLKSWPRD |
| 26 | MON_F32K_CK | FQMTR_BUSY | MON_FQMTR_CK | MON_CLK26M_CK |
| 27 | QI_CLKSQ_CKSEL | DA_KPLED_EN | RG_VIBR_EN | QI_CLKSQ_DIFF_PWDB |
| 28 | QI_VUPG[3] | QI_VUPG[2] | QI_VUPG[1] | QI_VUPG[0] |
| 29 | RG_VUSB_EN | QI_CLKSQ_SIN_PWDB | QI_MPLL_PWD | RG_SPK_EN |
| 30 | RG_MPLL_FBDIV[3] | RG_MPLL_FBDIV[2] | RG_MPLL_FBDIV[1] | RG_MPLL_FBDIV[0] |
| 31 | RG_ISINKS_CH2_EN | RG_ISINKS_CH3_EN | CHR_EN | DA_SPK_EN |

**83010F30h SIM_MON_CON0** **SIM_MON_CON0**

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

**SIM_MON_SEL** monitor flags output select

| X | MON[1] | MON[0] |
|---|---|---|
| 00 | GPIO_SRST2_IN | RG_VSIM1_EN |
| 01 | RG_VSIM1_VOSEL | QI_SIMRST |
| 02 | NI_SIMCLK | NI_SIMDATA_OE |
| 03 | A2C_SIMDATA_IN | NI_SIMDATA_OUT |
| 04 | GPIO_SCLK2_IN | RG_VSIM2_EN |
| 05 | C2A_SIM2SEL | QI_SIMRST2_OUT |
| 06 | NI_SIMCLK2_OUT | NI_SIMDATA2_OE |
| 07 | A2C_SIM2DATA_IN | NI_SIMDATA2_OUT |

## 6.4 Programming Guide

### 6.4.1 BBRX Register Setup

The register used to control analog base-band receiver is ACIF_BBRX_CON.

#### 6.4.1.1 Programmable Biasing Current

To maximize the yield in modern digital process, the receiver features providing 4-bit 9-level programmable current to bias internal analog blocks. The 5-bits registers CALBIAS[3:0] is coded with 2's complement format.

#### 6.4.1.2 Offset Calibration

The base-band downlink receiver (RX) provides analog hardware for DSP algorithm to correct the offset error. The connection for measurement of RX offset error is shown in **Figure 58**, and the corresponding calibration procedure is described below.

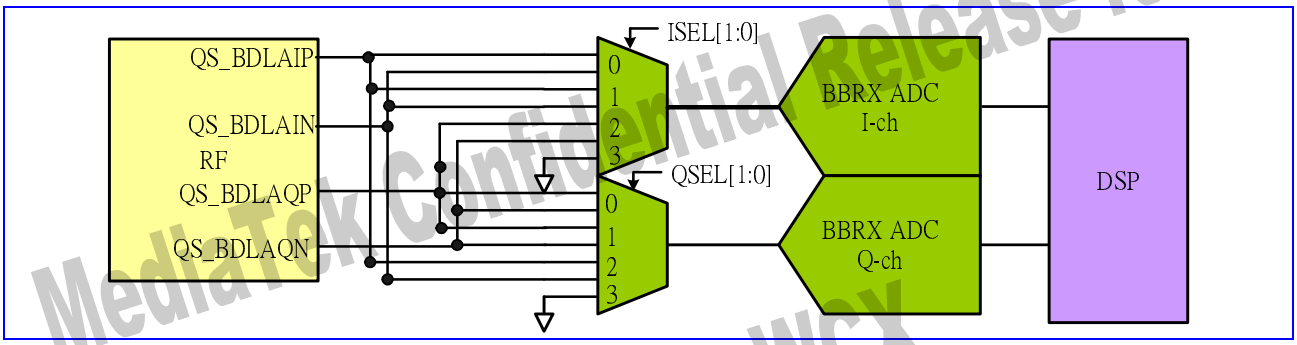



**Figure 58** Base-band A/D Offset Calibration

#### 6.4.1.3 Downlink RX Offset Error Calibration

The RX offset measurement is achieved by selecting grounded input to A/D converter (set ISEL [1:0] ='11' and QSEL [1:0] ='11' to select channel 3 of the analog input multiplexer, as shown in **Figure 59**. The output of the ADC is sent to DSP for further offset cancellation. The offset cancellation accuracy depends on the number of samples being converted. That is, more accurate measurement can be obtained by collecting more samples followed by averaging algorithm.

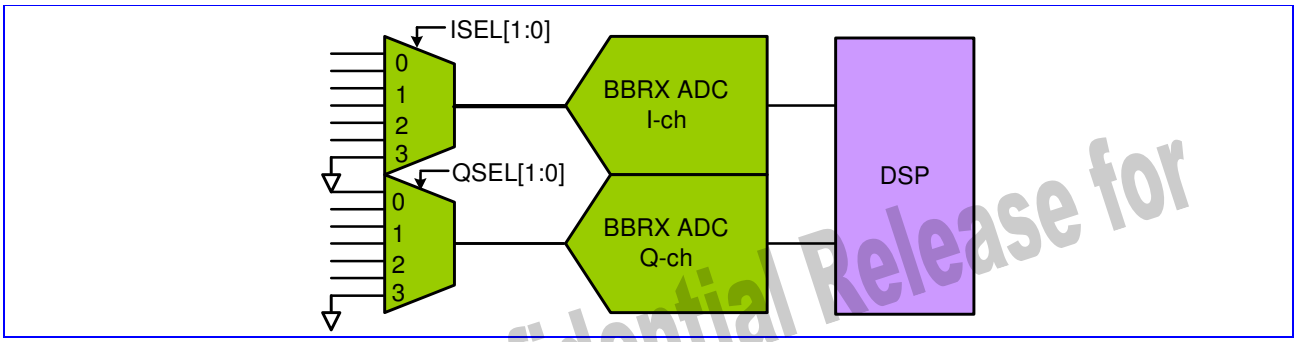



**Figure 59** Downlink ADC Offset Error Measurement

### 6.4.2 APC-DAC Register Setup

The register used to control the APC DAC is ACIF_APC_CON, which providing 4-bit 9-level programmable current to bias internal analog blocks. The 4-bits registers APC_CALI [3:0] is coded with 2's complement format.

### 6.4.3 Auxiliary A/D Conversion Register Setup

The register used to control the Aux-ADC is ACIF_AUX_CON0. For this register, which providing 2-bit 4-level programmable current to bias internal analog blocks.

### 6.4.4 Voice-band Blocks Register Setup

The registers used to control AMB are ACIF_VOICE_CON0, ACIF_VOICE_CON1, ACIF_VOICE_CON2, and ACIF_VOICE_CON3. For these registers, please refer to chapter "Analog Chip Interface".

#### 6.4.4.1 Reference Circuit

The voice-band blocks include internal bias circuits, a differential voltage reference circuit and a single-end microphone bias circuit. Internal bias current could be calibrated by varying VCALI[3:0] (coded with 2's complement format).

For proper operation, there should be an external 1uF capacitor connected to output pin AU_VCM. The VCM voltage (~1.4V, typical)**.** The following table illustrates typical 0dBm0 voltage when uplink/downlink programmable gains are unity. For other gain setting, 0dBm0 reference level should be scaled accordingly.

| Symbol | Parameter | Min | Typical | Max | Unit |
|---|---|---|---|---|---|
| $V_{0dBm0,UP}$ | 0dBm0 Voltage for Uplink Path, Applied Differentially Between Positive and Negative Microphone Input Pins | | 0.2V | | V-rms |
| $V_{0dBm0,Dn}$ | 0dBm0 voltage for Downlink Path, Appeared Differentially Between Positive and Negative Power Amplifier Output Pins | | 0.6V | | V-rms |

**Table 73** 0dBm0 reference level for unity uplink/downlink gain

The microphone bias circuit generates a single-ended output voltage on AU_MICBIAS_P for external electret type microphone. Typical output voltage is 1.9 V. The max current supplied by microphone bias circuit is 2mA.

#### 6.4.4.2 Uplink Path

Uplink path of voice-band blocks includes an uplink programmable gain amplifier and a sigma-delta modulator.

#### 6.4.4.2.1 Uplink Programmable Gain Amplifier

Input to the PGA is a multiplexer controlled by VCFG [3:0], as described in the following table. In normal operation, only AC coupling is suggested if amplification of input signal is desired

| Control Signal | Function | Descriptions |
|---|---|---|
| VCFG[0] | Input Selector | 0: Input 0 (From AU_VIN0_P / AU_VIN0_N) selected<br>1: Input 1 (From AU_VIN1_P / AU_VIN1_N) selected |
| VCFG[1] | Input Selector | 1: Input FM (From AU_FMINL / AU_FMINR) Is Selected |
| VCFG[2] | Coupling Mode | 0: AC Coupling (for testing purpose)<br>1: DC Coupling |
| VCFG[3] | Gain Mode | 0: Amplification Mode (gain range 1~49 dB)<br>1: Bypass Mode |

**Table 74** Uplink PGA input configuration setting

The PGA itself provides programmable gain (through VUPG[5:0]) with step of 1.0 dB, as listed in the following table.

| VUPG [5:0] | Gain | VUPG [5:0] | Gain |
|---|---|---|---|
| 111111 | 49 dB | 011111 | 17 dB |
| 111110 | 48 dB | 011110 | 16 dB |
| 111101 | 47 dB | 011101 | 15 dB |
| 111100 | 46 dB | 011100 | 14 dB |
| 111011 | 45 dB | 011011 | 13 dB |
| 111010 | 44 dB | 011010 | 12 dB |
| 111001 | 43 dB | 011001 | 11 dB |
| 111000 | 42 dB | 011000 | 10 dB |
| 110111 | 41 dB | 010111 | 9 dB |
| 110110 | 40 dB | 010110 | 8 dB |
| 110101 | 39 dB | 010101 | 7 dB |
| 110100 | 38 dB | 010100 | 6 dB |
| 110011 | 37 dB | 010011 | 5 dB |
| 110010 | 36 dB | 010010 | 4 dB |

| | | | |
|---|---|---|---|
| 110001 | 35 dB | 010001 | 3 dB |
| 110000 | 34 dB | 010000 | 2 dB |
| 101111 | 33 dB | 001111 | 1 dB |
| 101110 | 32 dB | 001110 | N/A- |
| 101101 | 31dB | 001101 | N/A- |
| 101100 | 30 dB | 001100 | N/A- |
| 101011 | 29 dB | 001011 | N/A- |
| 101010 | 28 dB | 001010 | N/A- |
| 101001 | 27 dB | 001001 | N/A- |
| 101000 | 26 dB | 001000 | N/A- |
| 100111 | 25 dB | 000111 | N/A- |
| 100110 | 24 dB | 000110 | N/A- |
| 100101 | 23 dB | 000101 | N/A- |
| 100100 | 22 dB | 000100 | N/A- |
| 100011 | 21 dB | 000011 | N/A- |
| 100010 | 20 dB | 000010 | N/A- |
| 100001 | 19 dB | 000001 | N/A- |
| 100000 | 18 dB | 000000 | N/A- |

**Table 75** Uplink PGA gain setting (VUPG[5:0])

The following table illustrates typically the 0dBm0 voltage applied at the microphone inputs, differentially, for several gain settings.

| VCFG [3] ='0' | | VCFG [3] ='1' (only valid for input 1) | |
|---|---|---|---|
| VUPG [5:0] | 0dBm0 (V-rms) | VUPG [5:0] | 0dBm0 (V-rms) |
| 111100 | 2mV | XXXXXX | 0.2V |
| 101000 | 20mV | | |
| 100000 | 50mV | | |
| 010100 | 0.2V | | |

**Table 76** 0dBm0 voltage at microphone input pins

#### 6.4.4.2.2 Sigma-Delta Modulator

Analog-to-digital conversion in uplink path is made with a second-order sigma-delta modulator (SDM) whose sampling rate is 6500KHz. For test purpose, one can set VADCINMODE to HI to form a look-back path from downlink DAC output to SDM input. The default value of VADCINMODE is zero.

### 6.4.4.3 Downlink Path

Downlink path of voice-band blocks includes a programmable output power amplifiers.

#### 6.4.4.3.1 Downlink Programmable Power Amplifier

Voice-band analog blocks include two identical output power amplifiers with programmable gain.

For the amplifier itself, programmable gain setting is described in the following table.

| VDPG0[3:0] / VDPG1[3:0] | Gain |
|---|---|
| 1111 | 8dB |
| 1110 | 6dB |
| 1101 | 4dB |
| 1100 | 2dB |
| 1011 | 0dB |
| 1010 | -2dB |
| 1001 | -4dB |
| 1000 | -6dB |
| 0111 | -8dB |
| 0110 | -10dB |
| 0101 | -12dB |
| 0100 | -14dB |
| 0011 | -16dB |
| 0010 | N/A |
| 0001 | N/A |
| 0000 | N/A |

**Table 77** Downlink power amplifier gain setting

Control signal VFLOAT, when set to 'HI', is used to make output nodes totally floating in power down mode. If VFLOAT is set to 'LOW" in power down mode, there will be a resistor of 50k ohm (typical) between AU_VOUT0_P and AU_VOUT0_N, as well as between AU_VOUT0_P and AU_VOUT0_N.

The amplifiers deliver signal power to drive external earphone. The minimum resistive load is 28 ohm and the upper limit of the output current is 50mA. On the basis that 3.14dBm0 digital input signal into downlink path produces DAC output differential voltage of 0.87V-rms (typical), the following table illustrates the power amplifier output signal level (in V-rms) and signal power for an external 32 ohm resistive load.

| VDPG[3:0] | Output Signal Level (V-rms) | Output Signal Power (mW / dBm) |
|---|---|---|
| 0010 | 0.11 | 0.37/-4.3 |
| 0110 | 0.27 | 2.28/3.6 |
| 1010 | 0.69 | 14.8/11.7 |
| 1110 | 1.74 | 94.6/19.8 |

**Table 78** Output signal level/power for 3.14dBm0 input. External resistive load = 32 ohm

The following table illustrates the output signal level and power for different resistive load when VDPG =1110.

| RLOAD | Output Signal Level (V-rms) | Output Signal Power (mW / dBm) |
|---|---|---|
| 30 | 1.74 | 101/20 |
| 100 | 1.74 | 30.3/14.8 |
| 600 | 1.74 | 5/7 |

**Table 79** Output signal level/power for 3.14dBm0 input, VDPG =1110

### 6.4.4.4 Power Down Control

Each block inside audio mixed-signal blocks features dedicated power-down control, as illustrated in the following table.

| Control Signal | Descriptions |
|---|---|
| VBIAS_PWDB | Power Down Reference Circuits (Active Low) |
| VLNA_PWDB | Power Down Uplink PGA (Active Low) |
| VADC_PWDB | Power Down Uplink SDM (Active Low) |
| VDAC_PWDB | Power Down DAC (Active Low) |
| VOUT0_PWDB | Power Down Downlink Power Amp 0 (Active Low) |

**Table 80** Voice-band blocks power down control

### 6.4.5 Audio-band Blocks Register Setup

The registers used to control audio blocks are ACIF_AUDIO_CON0, ACIF_AUDIO_CON1, ACIF_AUDIO_CON2 and ACIF_AUDIO_CON3. For these registers, please refer to chapter "Analog Chip Interface"

#### 6.4.5.1 Output Gain Control

Audio blocks include stereo audio DACs and programmable output power amplifiers. The DACs convert input bit-stream to analog signal by sampling rate of Fs*128 where Fs could be 32kHz, 44.1kHz, or 48kHz. Besides, it performs a 2nd-order butterworth filtering. The two identical output power amplifiers with programmable gain are designed to driving external AC-coupled single-end speaker. The minimum resistor load is 16 ohm and the maximum driving current is 50mA. The programmable gain setting, controlled by APGR[3:0] and APGL[3:0], is the same as that of the voice-band amplifiers.

Unlike voice signals, 0dBFS defines the full-scale audio signals amplitude. Based on bandgap reference voltage again, the following table illustrates the power amplifier output signal level (in V-rms) and signal power for an external 16 ohm resistive load.

| APGR[3:0]/ APGL[3:0] | Output Signal Level (V-rms) | Output Signal Power (mW / dBm) |
|---|---|---|
| 0010 | 0.055 | 0.19/-7.2 |
| 0110 | 0.135 | 1.14/0.6 |
| 1010 | 0.345 | 7.44/8.7 |
| 1110 | 0.87 | 47.3/16.7 |

**Table 81** Output signal level/power for 0dBFS input. External resistive load = 16 ohm

#### 6.4.5.2 Mute Function and Power Down Control

By setting AMUTER (AMUTEL) to high, right (left) channel output will be muted.

Each block inside audio mixed-signal blocks features dedicated power-down control, as illustrated in the following table.

| Control Signal | Descriptions |
|---|---|
| ABIAS_PWDB | Power Down Reference Circuits (Active Low) |
| ADACL_PWDB | Power Down L-Channel DAC (Active Low) |
| ADACR_PWDB | Power Down R-Channel DAC (Active Low) |
| AOUTL_PWDB | Power Down L-Channel Audio Amplifier (Active Low) |
| AOUTR_PWDB | Power Down R-Channel Audio Amplifier (Active Low) |

**Table 82** Audio-band blocks power down control

### 6.4.6 Multiplexers for Audio and Voice Amplifiers

- The audio/voice amplifiers feature accepting signals from various signal sources including AU_FMINR/AU_FMINL pins, that aimed to receive stereo AM/FM signal from external radio chip:

1) Voice-band amplifier 0 accepts signals from voice DAC output only.
2) Audio left/right channel amplifiers receive signals from either voice DAC, audio DAC, or AM/FM radio input pins (controlled by registers ABUFSELL[2:0] and ABUFSELR[2:0] ), too. Left and right channel amplifiers will produce identical output waveforms when receiving mono signals from voice DAC.

### 6.4.7 Preferred Microphone and Earphone Connections

In this section, preferred microphone and earphone connections are discussed.

Differential connection of microphone is shown below. This is the application circuits compatible with previous products. C1 and Rin form an AC coupling and high-pass network. C1*Rin should be chosen such that the in-band signal will not be attenuated too much. For differential minimum resistance of 13k ohm, minimum value of C1 is 170nF for less than 1dB attenuation at 300Hz. R2 is determined by microphone sensitivity. C2 and R2 form another low-pass filter to filtering noise coming from microphone bias pins. Pole frequency less than 50Hz is recommended.

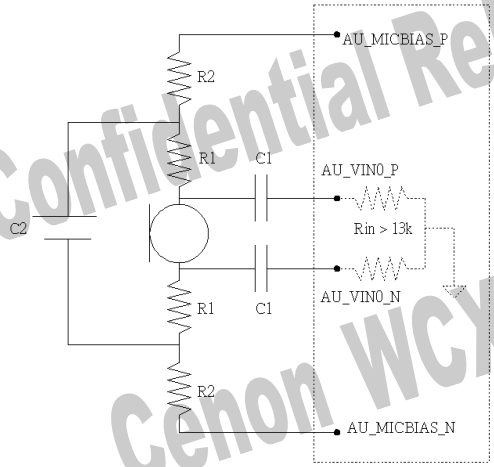



**Figure 6** Differential Microphone Connection

Another suggested connection method of microphone is shown below. R1 is chosen based on microphone sensitivity requirement. C1 and Rin form an AC coupling and high-pass network. R2 needs proper adjustment to obtain the best noise performance on the voice uplink input terminals.

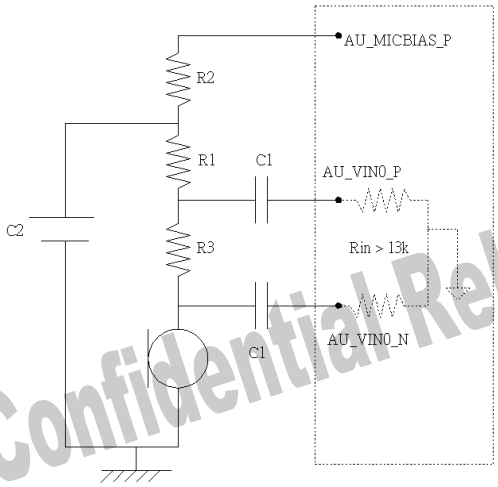



**Figure 7** Single-ended Microphone Connection

For earphone, both connections can be used. The application circuit shown in Figure 7 is highly recommended to achieve the better performance.

### 6.4.8 Clock Squarer Register Setup

The register used to control clock squarer is CLKSQ_CON. For this register, please refer to chapter "Clocks".

### 6.4.9 Phase-Locked Loop Register Setup

For registers control the PLL, please refer to chapter "Clocks" and "Software Power Down Control"

#### 6.4.9.1 Frequency Setup

The DSP/MCU PLL itself could be programmable to output either 52MHz or 104MHz clocks. Accompanied with additional digital dividers, 13/26/39/52/65/78/104 MHz clock outputs are supported.

#### 6.4.9.2 Programmable Biasing Current

The PLLs feature providing 5-bit 32-level programmable current to bias internal analog blocks. The 5-bits registers CALI [4:0] is coded with 2's complement format.

### 6.4.10 32-khz Crystal Oscillator Register Setup

For registers that control the oscillator, please refer to chapter "Real Time Clock" and "Software Power Down Control".