DATASHEET

Revision: 1.04

Release date: 08/06/2014

RDA8810PL

GSM/GPRS/EDGE Smartphone Application Processor

RDA8810PL Smartphone Application Processor

Platform Features

AP MCU subsystem

- ARM Cortex-A5 32-bit RISC processor, with FPU and SIMD NEON engine, up to 1G Hz
- 32kB L1 I-cache and 32kB L1 D-cache; 256kB unified L2 cache
- In-house VOC DSP for voice and image process
- 64-bit AXI fabric, up to 400MHz
- 64kB on-chip ROM for system boot
- 64kB on-chip SRAM for internal use

Memory interface

- Integrated 2Gb LPDDR2 SDRAM
- Integrated 4Gb 8Bit 1.8V 4K SLC Nand Flash
- Two SDIO interfaces, for TF cards and WIFI
- support 4.4.1 EMMC interface
- support SPI-NAND and SPI-NOR

Connectivity

- Support 3 SIM cards and both 1.8V and 3.0V device
- USB2.0 high-speed OTG
- 3 UARTs for GPS, BT, FM, and debugging interfaces
- 3 SPI for peripheral devices
- 3 I2C for peripheral devices
- 1 I2S for peripheral devices
- 4 10-bit GPADC for sampling peripheral signals
- 64 GPIOs, includes 16 interruptable GPIOs
- Pulse Width Modulation (PWM) 1 PWL, 1 PWT and 1 LPG
- 2 memory card controller
- 8x8 Keypad scanner with multiple key detection, support ADC serial interface Keypad

Modem Features

System CPU(XCUP)

- RDA RISC Core 32x32 bits
- 16/32 bit instruction set
- 4KB I-cache and 4KB D-cache

Base Band CPU (BCPU)

- RDA RISC Core32x32 bits
- 16/32 bit instruction set
- 1KB I-cache and 1KB D-cache
- GSM/GPRS Dedicated accelerators
- RF Interface (RF IF)
- DigRF interface for communication with the integrated transceiver
- EGPRS class 12

RF Transceiver

- Quad bands integrated transceiver
- Integrated RF transceiver, Saw-Less, only needs external PA.
- Digital low-IF receiver
- Direct modulation transmitter
- Frequency synthesizer
- Integrated VCO, loop filters, etc.
- Digital AFC
- DigRF interface between digital baseband and transceiver
- RF FEM control

Voice and Modem codec

- Supports HR, FR, EFR, AMR voice codec
- Integrated microphone bias:
- DTMF and Comfort Tone generator
- 4 samples In and Out Fifos
- I2S / DAI Interface
- Serial Input / Output at 8/16 ks/s
- Can be used for test purpose in DAI mode
- Audio Interface To ABB
- 13 bit RX Data from audio ADC
- 16 bit TX Data to stereo DAC

MULTI-MEDIA FEATURES

- **LCD display**
  - Support LCD module interface with 8/16/24 bit parallel interface or SPI series interface
  - Support RGB565/666/888 interface
  - Typical image sizes: QCIF, QVGA, WQVGA, CIF,VGA,WVGA,FWVGA
  - Support image size up to qHD
  - 4-layers blending graphical engine capable of resizing and YUV2RGB conversion
  - Support 2chip select

- **Image signal processor**
  - Support 8-bit parallel or 1/2/4-bit series interface
  - Support CSI-2 interface, up to 1GHz
  - Support up to 2Mpix sensor

- **Video**
  - Separate video processor
  - Decodes up to 1080p 30fps
  - Encodes up to 1080p 30fps for H.264 format

- **Graphics**
  - Separate graphic processor, Vivante's GC860
  - support OpenGLES1.1/2.0
  - support OpenVG1.4
  - support DirectFB
  - support GDI/DirecShow
- 30M Triangle/s, 250M Pixel/s

- **Audio**
  - 2 channels voice ADC, 8kHz, 13 bits/sample for headset and on-board microphone
  - Voice DAC, 8kHz, 13 bits/sample for receiver
  - High fidelity Stereo DAC, up to 48kHz, 16 bits per sample
  - Stereo Audio speaker driver
  - 1.5W differential output stereo amplifier for loudspeaker, Class K
  - Stereo analog audio line input

POWER MANAGEMENT FEATURES

- **Power Management**
  - Power On reset control
  - Internal 32K OSC for standby/ shutoff/ sleep state
  - Integrated Li-ion battery charger (from USB or AC charger)
  - Integrated all internal voltages from VBAT
  - Provide all LDOs for external components
  - Integrated 4 DC-DC, 1 BOOST
  - Integrated LCD backlight controller
  - Integrated Touch screen controller
  - Integrated RGB, keypad backlight controller

1. General Description

A high performance, high integrated system-on-chip solution for low cost, low power GSM/GPRS/EDGE smartphone .

RDA8810PL is a high performance, highly integrated system-on-chip solution for low cost, low power, GSM/GPRS/EDGE smartphone.

Integrating all essential electronic components, including AP MCU, GSM/GPRS/EDGE Module baseband, multimedia accelerators, quad band RF transceiver, saw-less, power management, audio amplifier onto a single system on chip, RDA8810PL offers best in class bill of material, space requirement and cost/feature ratio for complete phone handsets.

RDA8810PL Is capable of running the ARM cortex-A5 32-bit RISC processor at up to 1.2GHz, support NAND flash memory, 32-bit mobile DDR for optimal performance. An extensive set of interfaces and connectivity peripherals are included to interface to cameras, touch-screen displays, MMC/SD cards, and external Bluetooth, WiFi, FM, GPS modules.

Built around a cost effective 32-bit XCPU RISC core running at up to 312MHz with 4kB of Instruction cache and 4k of Data cache, A high performance proprietary 16/32-bit digital signal processing engine can further improve overall performance and user experience when performing complex multimedia tasks.

RDA8810PL is GPRS Class 12 enabled, EDGE, and supports Full Rate (FR), Half Rate (HR), Enhanced Full Rate (EFR) and Adaptive Multi Rate (AMR) voice coders. It also integrates a SIM controller with integrated level shifters that can support three SIM cards. RDA8810PL is available in a small footprint, fine pitch, 14X16, 325 ball LFBGA package.

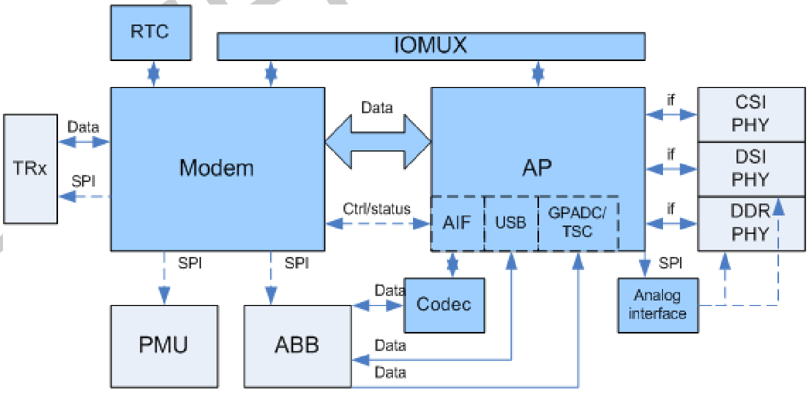


Figure 1.1: System Block Diagram

RDA8810PL provides a single-chip baseband solutions for GSM/GPRS/EDGE wireless smartphone.

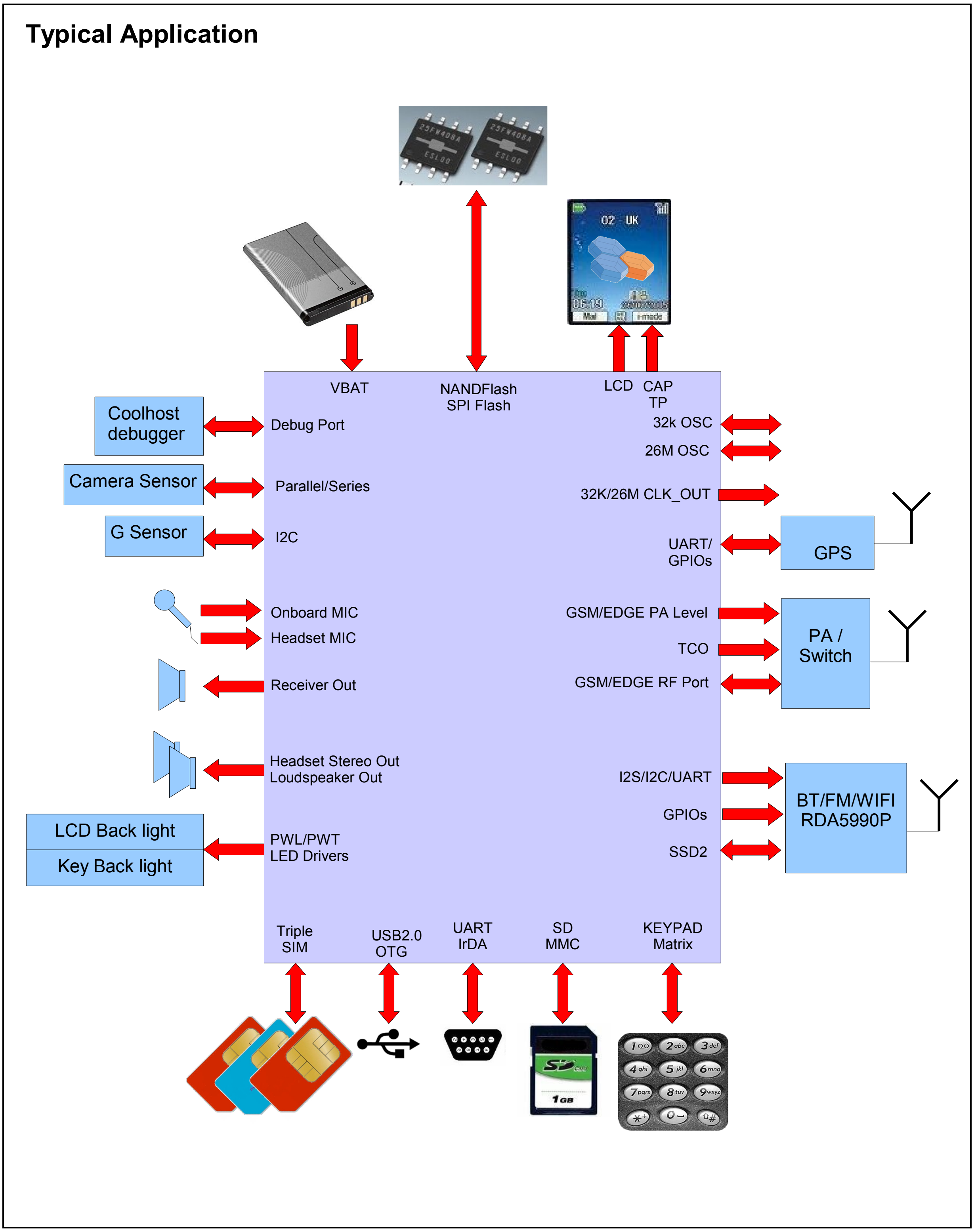


Figure 1.2: Typical Application

Table of Contents

Figure Index

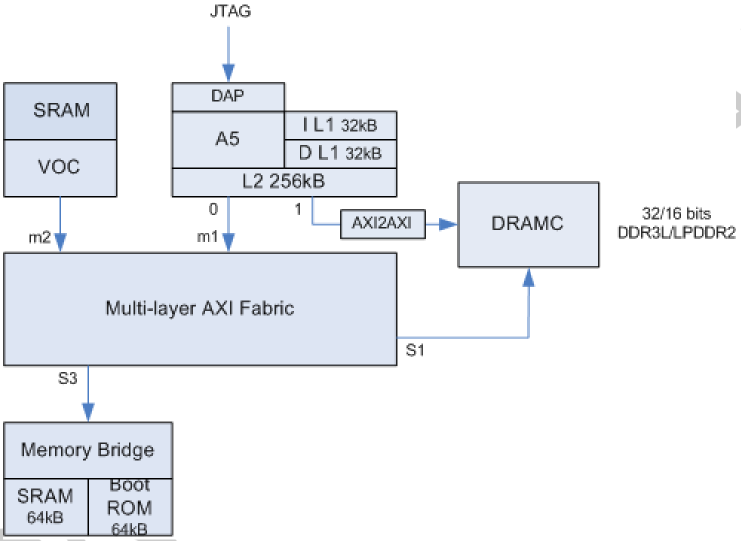
2. AP CPU/DSP Subsystem

2.1 Processor

2.1.1 Introduction

AP CPU/DSP subsystem supplys calculation capability for OS and applications. CPU and DSP are horsepower engines, more faster more power. Memory access latency and bandwidth are also important for system performance. DRAM controller acts as a key role on memory access. CPU's L1/L2 cache helps a lot on reducing memory access latency. Some on-chip SRAM and ROM are also integrated for some certain use, such as boot, .etc.

2.1.2 Block Diagram



2.1.3 ARM Cortex-A5 CPU

RDA8810PL AP uses ARM Cortex-A5 cpu, which is an ARMv7 instruction set architecture core. 8810 also integrated NEON, a SIMD engine, in this A5 to help multimedia efficiency. A5 has 8-stage pipeline. A5 has single 64-bit AXI interface.

A5 has L1 cache and L2 cache. L1 instruction cache and data cache are both 32kB. L2 unified cache is 16-way 256kB. L2 cache controller has two AXI interfaces, one is to DDR, the other is to system and peripherals.

2.1.4 In-house VOC DSP

In-house VOC DSP is used to process audio and image. It is developed as a target-specific DSP core, including basic function-call support, able to execute the code with very little (or no) control intervention from the CPU. It is controlled and configured by the CPU through the AMBA bus.

Features

- Bi-MAC, single test/logic Computational Unit with two 16x16 -> 32-bit multipliers
- Eight 16-bit general purpose registers, that can be combined in four 32-bit general purpose registers.
- All 16-bit registers can be used as pointers, four of them are incremental (for easy array addressing).
- Four 32-bit general purpose registers.
- 2 x 20 Kbytes data RAM + 20 Kbytes instruction RAM.
- Double stack with random access: for 32-bit & 16-bit values (push, pop).
- Function call support (jal, return).
- Two zero-cycle loop counters.
- Pointer & Direct addressing modes.
- DMA sub-module for block transfers between external memory and VoC memories.

2.1.5 Internal memories

8810 has 64kB internal SRAM and 64kB internal Boot ROM. Internal memories are used for system boot and other functions, for example, 64kB SRAM could also used as temporal data buffers for video processor. The access latency with internal SRAM is a lot shorter than with external DDR, so brings advantages in some functions. But internal SRAM is expensive.

2.1.6 DRAM controller

Smartphone need fast and huge memory, so choice DDR, the same memory type as PC. RDA8810PL supports 16/32-bit Mobile DDR/LPDDR2 up to 400MHz. If with 32-bit DDR at 400MHz, system could get 3.2GB/s memory bandwidth, this is enough for our system.

DRAMC is not only ddr controller, but also access arbitrator. DRAMC has 4 AXI system interfaces, it queues memory accesses from these system interfaces and services memory accesses according to priority or DDR interface usage efficiency.

2.1.7 Interrupt

Interrupt module supports 32 level interrupt inputs, and output an result to A5's nIRQ port. If more than 32 inputs in system, could OR them before send to interrupt module, and ISR (interrupt service routine) need to find interrupt source firstly.

In RDA8810PL, the interruptions are generated by the modules, they can be filtered at the module level (the mask must be configured in each corresponding module's registers), then they are filtered by the IRQ Controller (the mask is set in IRQ Control registers). The interrupts must be cleared in the corresponding module.

For power saving, one CPU can shut down its own clock. A special wakeup mask allows re-enabling the clock when an interruption enabled in this mask is triggered..

Features

- A mask at the module level.
- A mask at the IRQ Control level.
- Wakeup mask.
- Sleep Register

3. AP PERIPHERALS

3.1 DMA Controller

3.1.1 Overview

The DMA controller relieve the CPU from doing generic memory transfers. A data FIFO is integrated to allow burst transfers. It can generate an interrupt to the system CPU at transfer completion.

Read and Write transfers are supported from and to any memory mapped location (external memory, internal SRAM, modules FIFO…). The addresses can be word, halfword or byte aligned. The DMA can also be used in pattern mode. In this case, a 32-bit word will be used to fill the destination memory zone.

Features

- Support for linear memory transfers.
- Support for word, halfword and byte aligned addresses.
- Burst transfer mode and internal FIFO for best performances.
- Autonomous transfer up to 64K byte per transfer.
- support scatter/gather link list
- Interrupt generation at completion of the transaction.
- Can fill a part of the memory with a 32-bit pattern.

?

3.2 NAND Controller

3.2.1 Overview

This module implements a NAND-type flash memory controller with standard AMBA APB interface. It is compatible for all NAND-type flash memory from 512M bits to 64G bits. It is able to control up to 4 pieces of NAND Flash memory with write-protected ability to prevent writing data to flash memories and it also provides a security function interface to protect the stored data against forbid access.

NAND controller supports 8/16-bit ONFI IO interface, supports 2/4/8kB page. ECC engine used is 24-bit BCH/2kB for SLC, 96-bit BCH/1kB for MLC.

3.2.2 Registers

1. Flash Buffer(mapping address: 14’h0000~14’h2280)

Bits	Access	Default	Description
31:0	RW	0	Data read/write from/to controller buffer

Hardware ECC mode
Page data buffer (4K bytes, mapping address: 14’h0000 ~ 14’h0FFF), it could works in entry mode, and address in its mapping address range could be used as access entry.

Bits	Access	Default	Description
31:0	RW	0	Data read/write from/to controller page data buffer

Spare data buffer (1K bytes, mapping address: 14’h1000 ~ 14’h13FF), 8 bytes per page, supported maximum page number is 128 in multi-block/command operation.

Bits	Access	Default	Description
31:0	RW	0	Data read/write from/to controller buffer spare data area.

Software ECC mode
Page data buffer (4K bytes, mapping address: 14’h0000 ~ 14’h10D9), it could works in entry mode, and address in its mapping address range could be used as access entry.

Bits	Access	Default	Description
31:0	RW	0	Data read/write from/to controller page data buffer

2. 2Flash/SM Direct Command-Address Register (Offset: 14’h3000)

Bits	Access	Default	Description
31:24	RW	0x0	Direct command
23:0	RW	0x000	Row address. The meaning of row address must be referenced to the datasheet of NAND FLASH.

3. Flash/SM Column Address Register (Offset: 14’h300C)

Bits	Access	Default	Description
31:14	Unusable		
13:0	RW	0	Flash/SM column address register

4. Flash/SM Configuration Register A (Offset: 14’h3010)

Bits	Access	Default	Description
31:20	Unusable		

3:0	RW	3’b100	Flash/SM access cycle select:
			3’b010: sysclk/5;
			3’b011: sysclk/6
			3’b100: sysclk/7;
			3’b101: sysclk/8
			3’b101: sysclk/9;
			3’b111: sysclk/10;
			Note: the access cycle must be not less than the min Flash/SM access cycle

4	RW	0	Chip select 0, active Low
5	RW	1	Chip select 1, active Low
6	RW	1	Chip select 2, active Low
7	RW	1	Chip select 3, active Low
15:8	RW	8’h8c	nand flash page size mark. This register is related to register 0x2014 bit3 to bit0 8c : 4kByte page size, with 224Byte spare area 8b: 2kByte page size with 64Byte spare area 8d: 8kByte page size
16	RW	0	1: the data in cpu and in nand_ctrl sram are the same 0: the data in cpu and in nand_ctrl sram are reverse
17	RW	0	1: the data in nand flash and in nand_ctrl sram are reverse 0: the data in nand flash and in nand_ctrl sram are the same
18	RW	0	1: 16-bit io 0: 8-bit io
19	RW	0	1: synchronous io 0: asynchronous io

5. Flash/SM Configuration Register B (Offset: 14’h3014)

Bits	Access	Default	Description
31:8	Unusable		
3:0	RW	0	2’b00: 2K mode (for SLC), the message length of bch encoder is 2kByte 2’b10: 1K mode (for MLC), the message length of bch encoder is 1kByte Refer to 0x2010 bit15 to bit8. There are 3 working mode in the controller: (1){0x2010[15:8], 0x2014[3:0]} = 0x8b0, the bch will protect 2kByte data, without OOB (2){0x2010[15:8], 0x2014[3:0]} = 0x8c0, the bch will protect 4kByte data, without OOB (3){0x2010[15:8], 0x2014[3:0]} = 0x8c2, the bch will protect 2kByte data and OOB (3){0x2010[15:8], 0x2014[3:0]} = 0x8d2, the bch will protect 4kByte/6kByte data and OOB
7	RW	0	BCH enable active High 1 : bch encoder and decoder work 0 : bch encoder and decoder bypass

6. Flash/SM Busy Flag Register (Offset: 14’h301C)

Bits	Access	Default	Description

31:4	Unusable		
3	R		0 = decoder is idle; 1 = decoder is busy
2	R		0 = nand controller fsm is idle; 1 = nand controller fsm is busy
1	Ro	0	Flash Controller busy, active high
0	Ro	0	External Flash busy, active high

7. Flash/SM Controller Interrupt Mask Register (Offset: 14’h3020

Bits	Access	Default	Description
31:6	Unusable		
5	RW	0	CRC error interrupt enable, active high
4	RW	0	Operation done interrupt enable, active high
3	RW	0	Flash busy timeout interrupt enable, active high
2	RW	0	Erase error interrupt enable, active high
1	RW	0	Programming error interrupt enable, active high
0	RW	0	ECC (data) error interrupt enable, active high

8. Flash/SM Controller Interrupt Status Register (Offset: 14’h3024)

Bits	Access	Default	Description
31:27	Unusable		
26:20	R	0	error number, how many error bits detected by the decoder.
19	RW	0	CRC check fail 6, active high, cleared by CPU write 1
18	RW	0	CRC check fail 5, active high, cleared by CPU write 1
17	RW	0	CRC check fail 4, active high, cleared by CPU write 1
16	RW	0	CRC check fail 3, active high, cleared by CPU write 1
15	RW	0	CRC check fail 2, active high, cleared by CPU write 1
14	RW	0	CRC check fail 1, active high, cleared by CPU write 1
13	RW	0	CRC check fail 0, active high, cleared by CPU write 1
12	RW	0	ECC (data) error 6, active High, cleared by CPU write 1
11	RW	0	ECC (data) error 5, active High, cleared by CPU write 1
10	RW	0	ECC (data) error 4, active High, cleared by CPU write 1
9	RW	0	ECC (data) error 3, active High, cleared by CPU write 1
8	RW	0	ECC (data) error 2, active High, cleared by CPU write 1
7	RW	0	ECC (data) error 1, active High, cleared by CPU write 1
6	RW	0	ECC (data) error 0, active High, cleared by CPU write 1
5	RW	0	CRC check fail, active high, cleared by CPU write 1
4	RW	0	Operation done for multi-block/command operation, active high, cleared by CPU write 1
3	RW	0	Busy timeout, active high, cleared by CPU write 1
2	RW	0	Erase error, active high, cleared by CPU write 1
1	RW	0	Program error, active High, cleared by CPU write 1
0	RW	0	ECC (data) error, active High, cleared by CPU write 1

9. Flash/SM ID Code Register A (Offset: 14’h3028)

Bits	Access	Default	Description

31:24	R	0x00	Byte 8 of Flash/SM ID code
23:16	R	0x00	Byte 7 of Flash/SM ID code
15:8	R	0x00	Byte 6 of Flash/SM ID code
7:0	R	0x00	Byte 5 of Flash/SM ID code

10. Flash/SM ID Code Register B (Offset: 14’h302C)

Bits	Access	Default	Description
31:24	R	0x00	Byte 4 of Flash/SM ID code
23:16	R	0x00	Byte 3 of Flash/SM ID code
15:8	R	0x00	Byte 2 of Flash/SM ID code
7:0	R	0x00	Byte 1 of Flash/SM ID code

11. Flash/SM DMA REQ Enable Register(Offset: 14’h3030)

Bits	Access	Default	Description
31:2	Unusable		
1	W/R	0x0	Read DMA request enable 0:disable 1:enable
0	W/R	0x0	Write DMA request enable 0:disable 1:enable

12. Flash/SM COMMAND Definition RegisterA (Offset: 13’h3034)

Bits	Access	Default	Description
31:24	W/R	0x85	Copy back program first cycle command definition (For supporting SMIC copy back: 0x86)
23:16	W/R	0x10	Copy back program second cycle command definition (For supporting SMIC copy back: 0x16)
15:8	W/R	0x10	Program second cycle command definition (may use to support SMIC extended memory page program if necessary)
7:0	W/R	0xD0	Erase second cycle command definition (For supporting SMIC block erase:0xD1)

13. Flash/SM COMMAND Definition RegisterB Offset: 14’h3038)

Bits	Access	Default	Description
31:24	W/R	0x50	Single command 5C mapping definition (can be used to support other special command)
23:16	W/R	0x01	Single command 5B mapping definition (For supporting SMIC factory bad block table read: 0x58)
15:8	W/R	0x00	Single command 5A mapping definition (For supporting SMIC factory bad block table read: x5A)
7:0	W/R	0x30	Read second cycle command definition

14. Flash/SM NAND FLASH Status Register (Offset: 14’h303C)

Bits	Access	Default	Description
31:8	Unusable		
7:0	R	0	flash status

15. Flash/SM NAND FLASH Status Register (Offset: 14’h3040)

Bits	Access	Default	Description

31:16	Unusable		
15:8	RW	0x0	readid_addr : read id address
14:3	reserved		
2:0	RW	0x6	rdid_length : read id length

16. NAND FLASH Features Control Register(Offset: 14’h3044)
17. For activating the Synchronous Interface

Bits	Access	Default	Description
31:16	R	0x00	Reserved
15:8	W/R	0x01	features_addr : The address when activating the synchronous interface
7:3	W/R	0x00	unusable
2:0	W/R	0x3	features_length : Number of features

18. NAND FLASH Features Data(Offset: 14’h3048)

Bits	Access	Default	Description
31:24	W/R	0x00	Get features data: Features data P4
23:16	W/R	0x00	Get features data: Features data P3
15:8	W/R	0x00	Get features data: Features data P2
7:0	W/R	0x00	Get features data: Features data P1

19. NAND FLASH synchronous delay number(Offset: 14’h304C)

Bits	Access	Default	Description
31:8	R	0x00	Reserved
7:4	W/R	0x1	Dqs delay number
3	R	0x0	1: the interface is synchronous 0: the interface is asynchronous
2:0	W/R	0x1	Number of cycles to wait tadl

20. NAND FLASH last 2 bytes mark(Offset: 14’h3050)

Bits	Access	Default	Description
31:29	R	0x0	reserved
28:16	W/R	0x10DE	Mark the last 2 bytes in reading. In accordance with the page size.
15:13	R	0x0	Reserved
12:0	W/R	0x10DD	Mark the last 2 bytes in writing. In accordance with the page size.

21. IRBN counter (Offset: 14’h3054)

Bits	Access	Default	Description
31:16	W/R	0x0004	rbn_cnt_reg2
15:0	W/R	0x0020	rbn_cnt_reg1

22. monitor selection (Offset: 14’h3058)

Bits	Access	Default	Description
31:4	R	0x0	Reserved
3:0	W/R	0xf	mon_reg, monitor selection

23. PAGE parameter registers, for 8-kbyte page only (Offset: 14’h305C)

24. conra[15:8] == 16’hxD, conrb[1:0] == 2’h2

Bits	Access	Default	Description
31:30	R/W	0x0	Reserved
29:16	R/W	0x227E	page_size : for 8-bit io flash, page_size = (true nand flash page size-1)- byte for 16-bit io flash, page_size = (true nand flash page size-2)- byte The inchip buffer size is 8832-byte
15:4	R/W	0x0	Reserved
3:0	W/R	0x5	msg_pkg_num: number of message package, the unit is 1kByte, for 8kByte page size only. This register can be 7, which means 6kByte data and 1kByte OOB; or be 5, which means 4kByte data and 1kByte OOB.

25. READ PAGE parameter registers (Offset: 14’h3060)

Bits	Access	Default	Description
31:24	R/W	0x0	Reserved
23:16	R/W	0x0	para_addr : read parameter address
15:10	R/W	0x0	Reserved
9:0	W/R	0x38E	page_para_length : the length of page parameter, the unit is byte. 16-bit IO, page_para_length = (true page parameter length -2)-byte 8-bit IO, page_para_length = (true page parameter length -1)-byte

26. TIME_0 registers (Offset: 14’h3064)

Bits	Access	Default	Description
31:16	R/W	0x2	tfeat: the time of flash goes to busy under the command of set/get features, the unit is micro-second
15:7	R/W	0x0	Reserved
6:0	W/R	0x3f	time_1us_num : the IO frequency, the unit is MHz

27. TIME_1 registers (Offset: 14’h3068)

Bits	Access	Default	Description
31:16	R/W	0x3e8	trst: the time of flash goes to busy under the command of reset, the unit is micro-second
15:0	W/R	0x4b	tread : the time of flash goes to busy under the command of read page or read parameter, the unit is micro-second

28. TIME_2 registers (Offset: 14’h306C)

Bits	Access	Default	Description
31:16	R/W	0x2710	tbers: the time of flash goes to busy under the command of block erase, the unit is micro-second
15:0	W/R	0xa28	tprog : the time of flash goes to busy under the command of program page, the unit is micro-second

3.3 SPINAND/SPINOR Controller

3.3.1 Overview

SPINOR controller supports SPI-NOR. This controller also supports SPI-NAND. SPI-NOR/NAND has lower pin-count compared to parallel NOR/NAND.

3.4 UART interface

3.4.1 Overview

RDA8810PL includes Universal Asynchronous Receiver Transmitter channels. UART can be used as a serial interface or as an IrDA interface. Each sample is sent serially, has 1 start bit (always zero), 7 or 8 data bits, 0 or 1 parity bit and 1 or 2 stop bits (always one). The words sent have a fully programmable format: size, stop bit number and parity bit. Breaks (data line held low) can be generated and detected allowing resynchronizing the two devices.

Features

- Fully programmable transfer word format.
    - 7 or 8 data bits.
    - 1 or 2 stop bits.
    - Odd, even, mark, space or none parity.
- Smooth stop feature (the UART stops after the end of the current word transfer).
- Break generation and detection.
    - Break length programmable.
    - Interrupt can be generated.
- *Supports Automatic Flow Control (CTS and RTS lines).
- *Support RI, DCD, DTR, DSR.
- Programmable receive(32 bytes deep) and transmit fifos (16 bytes deep).
- Supports low speed IrDA 1.0 SIR mode by adding external hardware.
- DMA capabilities (through the System IFC) to allow fully automated data transfers.
- Wide selection of programmable interrupts to allow interrupt driven data transfer management.
    - Rx or Tx Fifo trigger reached.
    - Timeout: No characters in or out of the Rx FIFO during the last 4 character times and there is at least one character in the Rx FIFO during this time.
    - DMA timeout: After Rx Sys IFC DMA is started, no characters in or out of the Rx Fifo during the last 4 character times.
    - End of the Sys IFC DMA transfer.
    - CTS detection.
    - Tx or Rx overflow, parity or framing (bad sample format received) error or break.
- Loop Back capabilities for test purposes.
- Up to 1843.2 Kbit/s for serial and 115.2 Kbit/s for IrDA.

Data FIFO Operation

The UART can operate in two modes: the DMA mode and the direct CPU use.

DMA operation (using the IFC):
When the UART Rx receives some data it stores them in its Rx FIFO. When at least one data is available in the Rx FIFO, the configured channel of the IFC reads the data and copies it in memory (the location is programmable). When the Tx FIFO has at least one available room, the configured channel of the IFC transfers the data from memory (the address is programmable) to the Tx FIFO.

Direct use (using only the CPU):
Interrupts can be generated when the level of the Tx FIFO or the Rx FIFO reaches a programmable threshold. This allows direct software control of the transfers through interrupt signaling.

Auto Flow Control Operation

When RDA8810PL's Uart_CTS input becomes inactive, the UART stops sending data (after the end of the character currently transferred). When the Rx FIFO current level reaches the programmable Auto Flow Control level, RDA8810PL's Uart_RTS output signal becomes inactive, in order for the remote UART to stop sending data.

IrDA SIR Operation

IrDA 1.0 SIR mode is available and can be activated when the user opens a UART. IrDA 1.0 SIR mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 KBaud.

Transmitting an infrared pulse corresponds to a zero, while one is represented by not sending any pulse. The width of a infrared pulse is 3/16 th of a normal serial bit time (depending on the baudrate). Each sample begins with a pulse (the start bit)

Received data is inverted (due to the IrDA physical purposes). Therefore, the Uart_Rx port, has the correct UART polarity. See the following figure for more details.

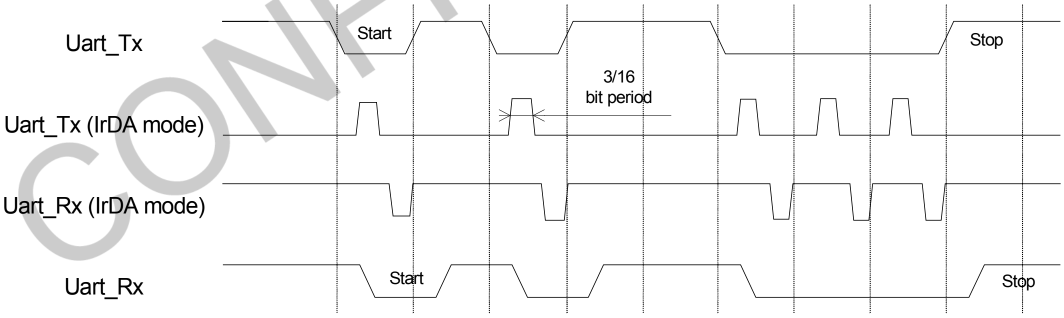


Figure 3.1: IrDA SIR Data Format

The UART module operation when IrDA SIR mode is enabled is similar to when the mode is disabled, with one exception: data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled. This is because the IrDA SIR physical layer specifies a minimum of 10ms delay between transmission and reception. This 10ms delay must be handled by the software.

3.4.2 Block Diagram

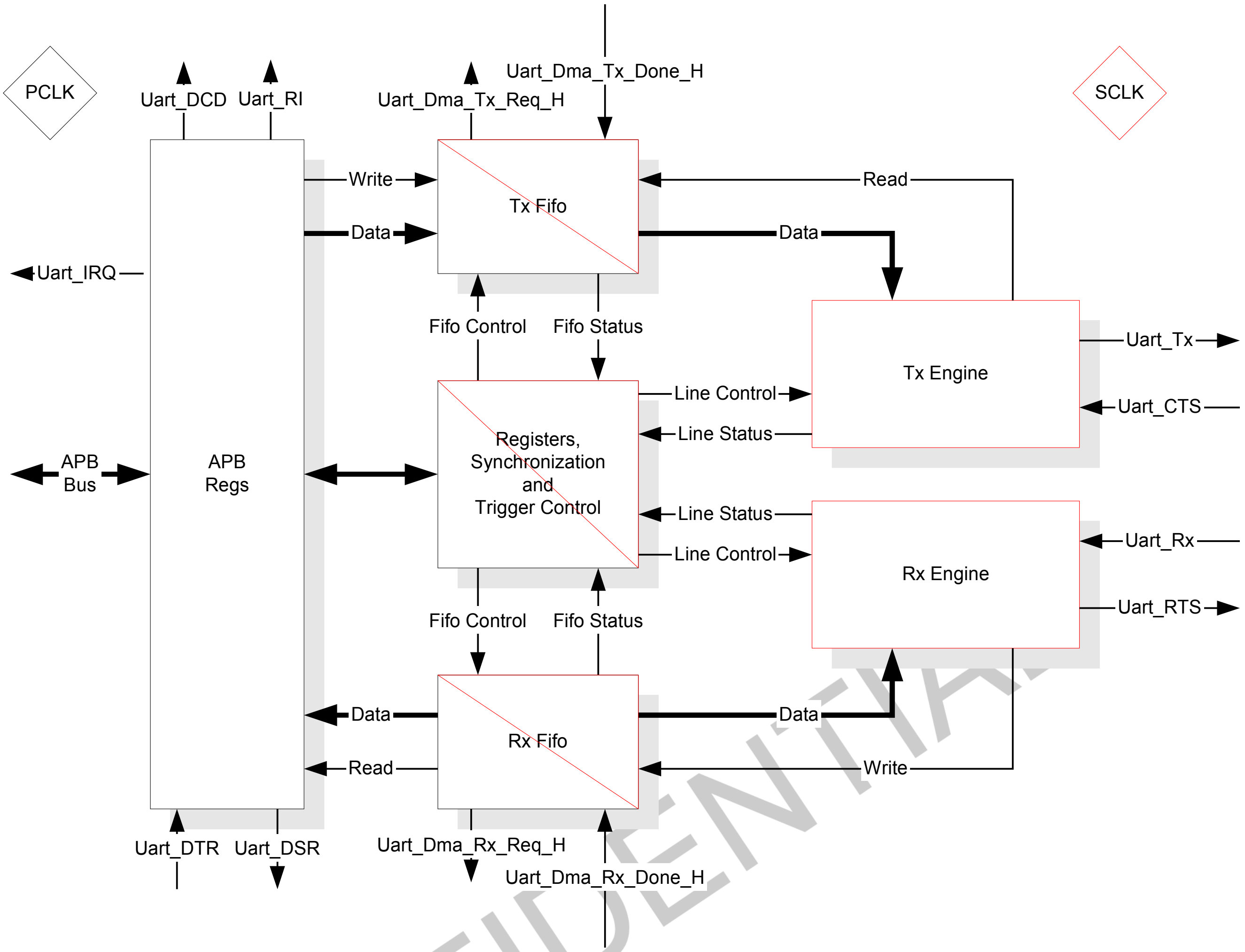


Figure 3.2: UART Block Diagram

3.4.3 Control Registers

UART Register Definitions

0x20A0_0000 ctrl

Bit	Name	Function	Type	Default
0	Enable	Allows to turn off the UART: 0 = Disable 1 = Enable	rw	0
1	Data Bits	Number of data bits per character (least significant bit first): 0 = 7 bits 1 = 8 bits	rw	0
2	Tx Stop Bits	Stop bits controls the number of stop bits transmitted. Can receive with one stop bit (more inaccuracy can be compensated with two stop bits when divisor mode is set to 0). 0 = one stop bit is transmitted in the serial data.	rw	0

Bit	Name	Function	Type	Default
		1 = two stop bits are generated and transmitted in the serial data out. 0 1_BIT 1 2_BITS		
3	Parity Enable	Parity is enabled when this bit is set. 0 NO 1 YES	rw	0
5:4	Parity Select	Controls the parity format when parity is enabled: 00 = an odd number of received 1 bits is checked, or transmitted (the parity bit is included). 01 = an even number of received 1 bits is checked or transmitted (the parity bit is included). 10 = a space is generated and received as parity bit. 11 = a mark is generated and received as parity bit. 0 ODD 1 EVEN 2 SPACE 3 MARK	rw	0
20	Divisor Mode	Selects the divisor value used to generate the baud rate frequency (BCLK) from the SCLK (see UART Operation for details). If IrDA is enable, this bit is ignored and the divisor used will be 16. 0 = (BCLK = SCLK / 4) 1 = (BCLK = SCLK / 16)	rw	0
21	IrDA Enable	When set, the UART is in IrDA mode and the baud rate divisor used is 16 (see UART Operation for details).	rw	0
22	DMA Mode	Enables the DMA signaling for the Uart_Dma_Tx_Req_H and Uart_Dma_Rx_Req_H to the IFC. 0 DISABLE 1 ENABLE	rw	0
23	Auto Flow Control	Enables the auto flow control. Uart_RTS is controlled by the Rx RTS bit and the UART Auto Control Flow System. If Uart_CTS become inactive high, the Tx data flow is stopped. 1 ENABLE 0 DISABLE	rw	0
24	Loop Back Mode	When set, data on the Uart_Tx line is held high, while the serial output is looped back to the serial input line, internally. In this mode all the interrupts are fully functional. This feature is used for diagnostic purposes. Also, in loop back mode, the modem control input Uart_CTS is disconnected and the modem control output Uart_RTS are looped back to the inputs, internally. In IrDA mode, Uart_Tx signal is inverted (see IrDA SIR Mode Support).	rw	0
25	Rx Lock Err	Allow to stop the data receiving when an error is detected (framing, parity or break). The data in the fifo are kept.	rw	0
31:28	Rx Break Length	Length of a break, in number of bits.	rw	all1

0x20A0_0004 status

Bit	Name	Function	Type	Default
5:0	Rx Fifo Level	Those bits indicate the number of data available in the Rx Fifo. Those data can be read.	r	0
12:8	Tx Fifo space	Those bits indicate the number of space available in the Tx Fifo.	r	0

Bit	Name	Function	Type	Default
14	Tx Active	This bit indicates that the UART is sending data. If no data is in the fifo, the UART is currently sending the last one through the serial interface.	r	0
15	Rx Active	This bit indicates that the UART is receiving a byte.	r	0
16	Rx Overflow Err	This bit indicates that the receiver received a new character when the fifo was already full. The new character is discarded. This bit is cleared when the UART_STATUS register is written with any value.	r	0
17	Tx Overflow Err"	This bit indicates that the user tried to write a character when fifo was already full. The written data will not be kept. This bit is cleared when the UART_STATUS register is written with any value.	r	0
18	Rx Parity Err	This bit is set if the parity is enabled and a parity error occurred in the received data. This bit is cleared when the UART_STATUS register is written with any value.	r	0
19	Rx Framing Err	This bit is set whenever there is a framing error occured. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. This bit is cleared when the UART_STATUS register is written with any value.	r	0
20	Rx Break Int	This bit is set whenever the serial input is held in a logic 0 state for longer than the length of x bits, where x is the value programmed Rx Break Length. A null word will be written in the Rx Fifo. This bit is cleared when the UART_STATUS register is written with any Value.	r	0
24	DCTS	This bit is set when the Uart_CTS line changed since the last time this register has been written. This bit is cleared when the UART_STATUS register is written with any value.	r	0
25	CTS	current value of the Uart_CTS line. '1' = Tx not allowed. '0' = Tx allowed.	r	1
28	DTR	Current value of the DTR line.	r	0
31	Clk Enabled	This bit is set when Uart Clk has been enabled and received by UART after Need Uart Clock becomes active. It serves to avoid enabling RTS too early.	r	0

0x20A0_0008 rxtx_buffer

Bit	Name	Function	Type	Default
7:0	Rx Data	The UART_RECEIVE_BUFFER register is a read-only register that contains the data byte received on the serial input port. This register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overflow error will also occur.	r	no
7:0	Tx Data	The UART_TRANSMIT_HOLDING register is a write-only register that contains data to be transmitted on the serial output port. 16 characters of data may be written to the UART_TRANSMIT_HOLDING register before the FIFO is full. Any attempt to write data when the FIFO is full results in the write data being lost.	w	no

0x20A0_000C irq_mask

Bit	Name	Function	Type	Default
0	Tx Modem Status	Clear to send signal change detected.	rw	0
1	Rx Data Available	Rx Fifo at or upper threshold level (current level &gt;= Rx Fifo trigger level).	rw	0
2	Tx Data Needed	Tx Fifo at or below threshold level (current level &lt;= Tx Fifo trigger level).	rw	0
3	Rx Timeout	No characters in or out of the Rx Fifo during the last 4 character times and there is at least 1 character in it during this time.	rw	0
4	Rx Line Err	Tx Overflow, Rx Overflow, Parity Error, Framing Error or Break Interrupt.	rw	0
5	Tx Dma Done	Pulse detected on Uart_Dma_Tx_Done_H signal.	rw	0
6	Rx Dma Done	Pulse detected on Uart_Dma_Rx_Done_H signal.	rw	0
7	Rx Dma Timeout	In DMA mode, there is at least 1 character that has been read in or out the Rx Fifo. Then before received Rx DMA Done, No characters in or out of the Rx Fifo during the last 4 character times.	rw	0
8	DTR RISE	Rising edge detected on the UART_DTR signal.	rw	0
9	DTR FALL	Falling edge detected on the UART_DTR signal.	rw	0

0x20A0_0010 irq_cause

Bit	Name	Function	Type	Default
0	Tx Modem Status	Clear to send signal detected. Reset control: This bit is cleared when the UART_STATUS register is written with any value.	r	0
1	Rx Data Available	Rx Fifo at or upper threshold level (current level &gt;= Rx Fifo trigger level). Reset control: Reading the UART_RECEIVE_BUFFER until the Fifo drops below the trigger level.	r	0
2	Tx Data Needed	Tx Fifo at or below threshold level (current level &lt;= Tx Fifo trigger level). Reset control: Writing into UART_TRANSMIT_HOLDING register above threshold level.	r	0
3	Rx Timeout	No characters in or out of the Rx Fifo during the last 4 character times and there is at least 1 character in it during this time. Reset control: Reading from the UART_RECEIVE_BUFFER register.	r	0
4	Rx Line Err	Tx Overflow, Rx Overflow, Parity Error, Framing Error or Break Interrupt. Reset control: This bit is cleared when the UART_STATUS register is written with any value.	r	0
5	Tx Dma Done	This interrupt is generated when a pulse is detected on the Uart_Dma_Tx_Done_H signal. Reset control: Write one in this register.	rc	0
6	Rx Dma Done	This interrupt is generated when a pulse is detected on the Uart_Dma_Rx_Done_H signal. Reset control: Write one in this register.	rc	0
7	Rx Dma Timeout	In DMA mode, there is at least 1 character that has been read in or out the Rx Fifo. Then before received Rx DMA Done, No characters in or out of the Rx Fifo during the last 4 character times.	rc	0
8	DTR RISE	This interrupt is generated when a rising edge is detected on the UART_DTR signal. Reset control: Write one in this register.	rc	0

Bit	Name	Function	Type	Default
9	DTR FALL	This interrupt is generated when a falling edge is detected on the UART_DTR signal. Reset control: Write one in this register.	rc	0
16	Tx Modem Status U	Same as previous, not masked.	r	0
17	Rx Data Available U	Same as previous, not masked.	r	0
18	Tx Data Needed U	Same as previous, not masked.	r	0
19	Rx Timeout U	Same as previous, not masked.	r	0
20	Rx Line Err U	Same as previous, not masked.	r	0
21	Tx Dma Done U	Same as previous, not masked.	r	0
22	Rx Dma Done U"	Same as previous, not masked.	r	0
23	Rx Dma Timeout U"	Same as previous, not masked.	r	0
24	DTR RISE U"	Same as previous, not masked.	r	0
25	DTR FALL U	Same as previous, not masked.	r	0

0x20A0_0014 triggers

Bit	Name	Function	Type	Default
4:0	Rx Trigger	Defines the empty threshold level at which the Data Available Interrupt will be generated. The Data Available interrupt is generated when quantity of data in Rx Fifo > Rx Trigger.	rw	0
11:8	Tx Trigger	Defines the empty threshold level at which the Data Needed Interrupt will be generated. The Data Needed Interrupt is generated when quantity of data in Tx Fifo <= Tx Trigger.	rw	0
20:16	AFC Level	Controls the Rx Fifo level at which the Uart_RTS Auto Flow Control will be set inactive high (see UART Operation for more details on AFC). The Uart_RTS Auto Flow Control will be set inactive high when quantity of data in Rx Fifo > AFC Level.	rw	0

0x20A0_0018 CMD_Set

Bit	Name	Function	Type	Default
0	RI	Ring indicator. When write '1', set RI bit. When read, get RI bit value.	rs	0
1	DCD	Data carrier detect. When write '1', set DCD bit. When read, get DCD bit value.	rs	0
2	DSR	Data set ready. When write '1', set RI bit. When read, get RI bit value.	rs	0
3	Tx Break Control	Sends a break signal by holding the Uart_Tx line low until this bit is cleared.	rs	0
4	Tx Finish n Wait	When this bit is set the Tx engine terminates to send the current byte and then it stops to send data.	rs	0

Bit	Name	Function	Type	Default
5	RTS	Controls the Uart_RTS output. 0 = the Uart_RTS will be inactive high (Rx not allowed). 1 = the Uart_RTS will be active low (Rx allowed).	rs	0
6	Rx Fifo Reset	Writing a 1 to this bit resets and flushes the Receive Fifo. This bit does not need to be cleared.	s	0
7	Tx Fifo Reset	Writing a 1 to this bit resets and flushes the Transmit Fifo. This bit does not need to be cleared.	s	0

0x20A0_001C CMD_Clr

Bit	Name	Function	Type	Default
0	RI	Ring indicator. When write '1', clear RI bit. When read, get RI bit value.	rc	0
1	DCD	Data carrier detect. When write '1', clear DCD bit. When read, get DCD bit value.	rc	0
2	DSR	Data set ready. When write '1', clear RI bit. When read, get RI bit value.	rc	0
3	Tx Break Control	Sends a break signal by holding the Uart_Tx line low until this bit is cleared.	rc	0
4	Tx Finish n Wait	When this bit is set the Tx engine terminates to send the current byte and then it stops to send data.	rc	0
5	RTS	Controls the Uart_RTS output. 0 = the Uart_RTS will be inactive high. 1 = the Uart_RTS will be active low.	rc	0

0x20A1_00xx UART2
0x20A9_00xx UART3

3.5 SPI interface

3.5.1 Overview

This module is a master interface for a synchronous serial link, it can be configured to be compatible with Motorola SPI or to comply with some various synchronous serial protocols.

RDA8810PL includes 2 general purpose SPIs and 1 SPI dedicated to PMU, ABB integrated modules.

Features

- SPI1 has 3 chip selects
- SPI2 has 2 chip selects
- 2 selectable data input.
- programmable clock polarity.

- programmable data frame size (from 4 to 32 bits).
- 16 x 8 bits transmit FIFO.
- 16 x 8 bits receive FIFO.
- a few delay options (time between CS, clocks and data).
- received pattern matching before filling RX FIFO (SD-MMC read block feature)
- transmit zero when TX FIFO empty (for generating dummy data during pattern matching read)
- can be controlled by LPS (an new frame is not started until LPS allows it)
- direct pin control to force value to 0, 1 or input.
- special read mode with output enable control of the DO pin (selected input should be multiplexed with DO pin to use this feature)

3.5.2 Block Diagram

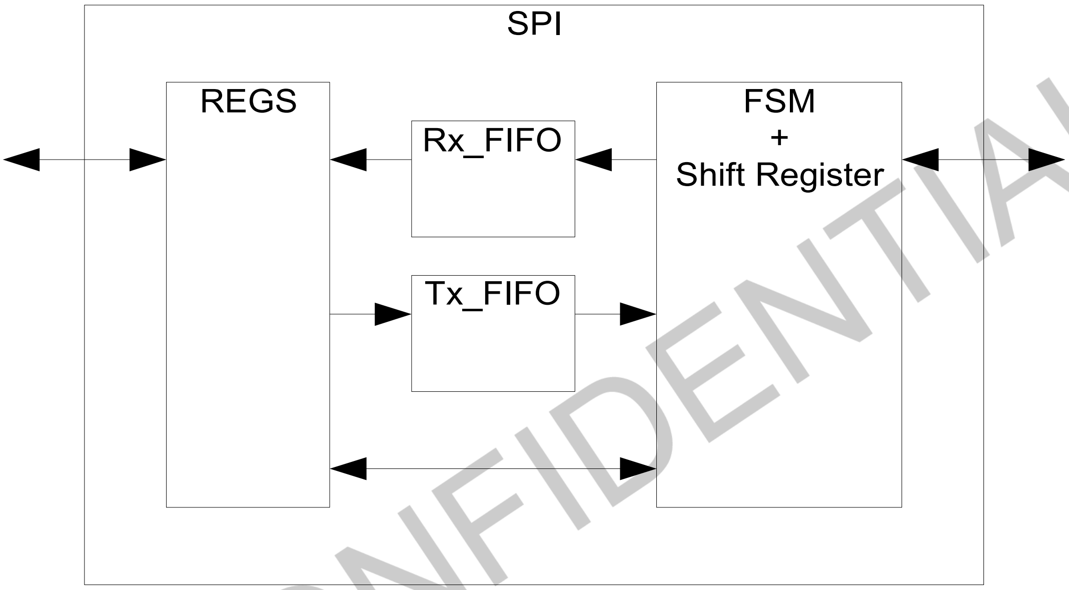


Figure 3.3 SPI Block Diagram

3.5.3 Control Registers

SPI Register Definitions

0x20A2_0000 ctrl

Bit	Name	Function	Type	Default
0	Enable	Enable the module and activate the chip select selected by CS_sel field.	rw	
2:1	CS_sel	Selects the active CS. 0 CS0 1 CS1 2 CS2 3 CS3	rw	
4	Input_mode	When set to 1 the inputs are activated, else only the output is	rw	1

Bit	Name	Function	Type	Default
		driven and no data are stored in the receive FIFO. Notes: The Input_mode bit status is also readable onto the bit rxtx_buffer[31].		
5	Clock_Polarity	The spi clock polarity when '0' the clock disabled level is low, and the first edge is a rising edge. When '1' the clock disabled level is high, and the first edge is a falling edge.	rw	1
7:6	Clock_Delay	Transfer start to first edge delay value from 0 to 2 is the number of spi clock half period between the CS activation and the first clock edge.	rw	11
9:8	DO_Delay	Transfer start to first data out delay value from 0 to 2 is the number of spi clock half period between the CS activation and the first data out	rw	11
11:10	DI_Delay	Transfer start to first data in sample delay value from 0 to 3 is the number of spi clock half period between the CS activation and the first data in sampled. NOTE: DI_Delay must be less or equal to DO_Delay + CS_Delay + 2. In other words DI_Delay can be 3 only if DO_Delay and CS_Delay are not both equal to 0.	rw	11
13:12	CS_Delay	Chip select deactivation to reactivation minimum delay value from 0 to 3 is the number of spi clock half period between the CS deactivation and a new CS activation (CS will activate only if more data are available in the transmit FIFO)	rw	11
15:14	CS_Pulse	Chip select deactivation to reactivation minimum delay value from 0 to 3 is the number of spi clock half period between the CS deactivation and a new CS activation (CS will activate only if more data are available in the transmit FIFO)	rw	11
20:16	Frame_Size	The frame size is the binary value of this register + 1 valid value are 3 to 31 (frame size 4 to 32bits)	rw	all1
28:24	OE_delay	When 0: regular mode, SPI_DO pin as output only. Value from 1 to 31 is the number of data out to transfert before the SPI_DO pin switch to input.	rw	all1
29	ctrl_data_mux_sel	Selects the active CS and Input_reg either from the ctrl or rxtx_buffer register. If SPI FIFO 8b or 32b, when set to "0": CS from CS_sel and INPUT from Input_mode in the register ctrl. Only if SPI FIFO 32b, when set to "1": CS and INPUT from SPI DATA.(Do not work for FIFO8b) 0 Ctrl_reg_sel 1 Data_reg_sel	rw	
31:30	Input_sel	Selects the input line to be used as SPI data in.(Not used for SPI3) when "00" the SPI_DI_0 is used. When "01" the SPI_DI_1 is used. When "10" the SPI_DI_2 is used. When "11" reserved.	rw	

0x20A2_0004 status

Bit	Name	Function	Type	Default
0	Active_Status	'1' when a transfer is in progress.	r	
3	Cause_Rx_Ovf_Irq	The receive FIFO overflow irq cause. Writing a '1' clear the receive overflow status and cause.	rw	
4	Cause_Tx_Th_Irq	The transmit FIFO threshold irq cause.	r	
5	Cause_Tx_Dma_Irq	The transmit Dma Done irq cause. Writing a '1' clear the transmit Dma Done status and cause.	rw	
6	Cause_Rx_Th_Irq	The receive FIFO threshold irq cause.	r	
7	Cause_Rx_Dma_Irq	The receive Dma Done irq cause. Writing a '1' clear the receive Dma Done status and cause.	rw	
9	Tx_Ovf	The transmit FIFO overflow status. Writing a '1' clear the transmit overflow status and cause.	rw	
10	Rx_Udf	The receive FIFO underflow status. Writing a '1' clear the receive underflow status and cause.	rw	
11	Rx_Ovf	The receive FIFO overflow status. Writing a '1' clear the receive overflow status and cause.	rw	
12	Tx_Th	The transmit FIFO threshold status.	r	
13	Tx_Dma_Done	The transmit Dma Done status. Writing a '1' clear the transmit Dma Done status and cause.	rw	
14	Rx_Th	The receive FIFO threshold status.	r	
15	Rx_Dma_Done	The receive Dma Done status. Writing a '1' clear the receive Dma Done status and cause.	rw	
20:16	Tx_Space	Transmit FIFO Space Number of empty spot in the FIFO	r	1000
28:24	Rx_Level	Receive FIFO level Number of DATA in the FIFO		
0	FIFO_Flush	Writing '1' flush both FIFO, don't do it when SPI is active (transfer in progress)	w	

0x20A2_0008 rxtx_buffer

Bit	Name	Function	Type	Default
0	DATA_IN	Write to the transmit FIFO.	w	
0	DATA_OUT	Read in the receive FIFO.	r	
0	CS	Chip Select on which write the data written in the Fifo.	w	
31	READ_ENA	Set this bit to one when the data received while sending this peculiar data are expected to be kept in the FIFO, otherwise no data is recorded in the FIFO.	w	

0x20A2_000C cfg

Bit	Name	Function	Type	Default
3:0	CS_Polarity	Chip select polarity 0 active_high chip select is active high 1 active_low chip select is active low	rw	all1

Bit	Name	Function	Type	Default
25:16	Clock_Divider	The state machine clock is generated by dividing the system clock by the value of this register + 1. So the output clock is divided by (register + 1)*2	rw	1023
28	Clock_Limiter	When enabled the clock input to the divider is not the system clock, but a limited version of it: It cannot be above 52MHz, so the output clock will never be above 26MHz. for system clock of 104Mhz the clock input to the divider is 52Mhz, for system clock of 78Mhz the clock input to the divider is 39Mhz, for lower system clock value, the input to the divider is the system clock.	rw	0

0x20A2_0010 pattern

Bit	Name	Function	Type	Default
7:0	pattern	MMC Pattern value for RX pattern match mode.	rw	
8	pattern_mode	Enable the pattern mode. 0 disabled Spi Behaviour. 1 enabled Pattern matching.	rw	0
9	pattern_selector	Select the RX pattern matching mode when the pattern_mode is enabled(set 1). Used for SD/MMC SPI mode. 0 UNTIL No datas are written into the RX FIFO UNTIL the received data is equal to the pattern. 1 WHILE No datas are written into the RX FIFO WHILE the received data is equal to the pattern.	rw	0

0x20A2_0014 stream

Bit	Name	Function	Type	Default
0	tx_stream_bit	When TX stream mode is enabled, once the TX fifo is empty, all new bits send have the value of this bit. 0 zero 1 one	rw	0
8	tx_stream_mode	Enable the TX stream mode. Used for SD/MMC SPI mode. When enabled, this mode provide infinite bit stream for sending, after fifo is empty the extra bits generated all have the same value. The value is in tx_stream_bit. 0 disabled 1 enabled	rw	0
16	tx_stream_stop_with_rx_dma_done	Allow to automatically clear the tx_stream_mode when Rx_Dma_Done is set. 0 disabled 1 enabled	rw	0

0x20A2_0018 pin_control

Bit	Name	Function	Type	Default
1:0	clk_ctrl	0 Spi_Ctrl The Spi_Clk pin is set OUTPUT(Basic SPI Behaviour). 1 Input_Ctrl The Spi_Clk pin is set INPUT (High Impedance). 2 Force_0_Ctrl The Spi_Clk pin is set OUTPUT and forced to 0. 3 Force_1_Ctrl The Spi_Clk pin is set OUTPUT and forced to 1.	rw	0
2:3	do_ctrl	0 Spi_Ctrl The Spi_DO pin is set OUTPUT(Basic SPI Behaviour). 1 Input_Ctrl The Spi_DO pin is set INPUT (High Impedance). 2 Force_0_Ctrl The Spi_DO pin is set OUTPUT and forced to 0. 3 Force_1_Ctrl The Spi_DO pin is set OUTPUT and forced to 1.	rw	0
4:5	cs0_ctrl	0 Spi_Ctrl The Spi_CSO pin is set OUTPUT(Basic SPI Behaviour). 1 Input_Ctrl The Spi_CSO pin is set INPUT (High Impedance). 2 Force_0_Ctrl The Spi_CSO pin is set OUTPUT and forced to 0. 3 Force_1_Ctrl The Spi_CSO pin is set OUTPUT and forced to 1.	rw	0
6:7	cs1_ctrl	0 Spi_Ctrl The Spi_CS1 pin is set OUTPUT(Basic SPI Behaviour). 1 Input_Ctrl The Spi_CS1 pin is set INPUT (High Impedance). 2 Force_0_Ctrl The Spi_CS1 pin is set OUTPUT and forced to 0. 3 Force_1_Ctrl The Spi_CS1 pin is set OUTPUT and forced to 1.	rw	0
8:9	cs2_ctrl	0 Spi_Ctrl The Spi_CS2 pin is set OUTPUT(Basic SPI Behaviour). 1 Input_Ctrl The Spi_CS2 pin is set INPUT (High Impedance). 2 Force_0_Ctrl The Spi_CS2 pin is set OUTPUT and forced to 0. 3 Force_1_Ctrl The Spi_CS2 pin is set OUTPUT and forced to 1.	rw	0
10:11	cs3_ctrl	0 Spi_Ctrl The Spi_CS3 pin is set OUTPUT(Basic SPI Behaviour). 1 Input_Ctrl The Spi_CS3 pin is set INPUT (High Impedance). 2 Force_0_Ctrl The Spi_CS3 pin is set OUTPUT and forced to 0. 3 Force_1_Ctrl The Spi_CS3 pin is set OUTPUT and forced to 1.	rw	0

0x20A2_001C irq

Bit	Name	Function	Type	Default
0	Mask_Rx_ovf_Irq	Mask the receive FIFO overflow irq	rw	
1	Mask_Tx_Th_Irq	Mask the transmit FIFO threshold irq	rw	

Bit	Name	Function	Type	Default
2	Mask_Tx_Dma_Irq	Mask the transmit Dma Done irq	rw	
3	Mask_Rx_Th_Irq	Mask the receive FIFO threshold irq	rw	
4	Mask_Rx_Dma_Irq	Mask the receive DMA Done irq	rw	
6:5	Tx_Threshold	Transmit FIFO threshold this threshold is used to generate the irq. Transmit FIFO threshold this threshold is used to generate the irq. 0 1_Empty_Slot 1 4_Empty_Slots 2 8_Empty_Slots 3 12_Empty_Slots	rw	11
8:7	Rx_Threshold	Receive FIFO threshold this threshold is used to generate the irq. 0 1_Valid_Data 1 4_Valid_Data 2 8_Valid_Data 3 12_Valid_Data	rw	11

0x20A3_00xx SPI2
0x20A4_00xx SPI3

3.6 Audio Interface Analog + I2S(AIF)

3.6.1 Overview

The Audio Interface (AIF) module is the audio interface between RDA8810PL system and internal codec (or external codec chip).

AIF can work in master mode in which it generates synchronization signals needed by codec, which contains the audio ADC and DAC to convert the serial digital output data to analog audio signal and the input analog signal to serial digital audio data. It can also works in slave mode which uses synchronization signals sent from external chip. Please see below for more details about serial data format.

AIF includes both serious and parallel interface. It is able to support multiple sample rate including 8 kHz, 11.025 kHz, 12kHz, 16 kHz, 22,05 kHz, 24kHz, 32kHz, 44,1 kHz, 48kHz and various data format. AIF can also generate common DTMF, comfort tones and side tone with a configurable gain.

Features

- Common features.
  - 4*32-bit RX and 4*32-bit TX FIFO.
  - All common DTMF and Comfort Tones can be generated and gained from -15 dB to 0 dB.
  - Side Tone fully configurable: Mute or amplification from -36 dB to +6 dB.
  - Loop back capabilities for test purposes.
- Serial Interfaces.

  - 16-bit mono or stereo samples.
  - MSB/LSB configurable.
  - Configurable as master or slave.
  - LRCK/BCLK ratio from 16 to 31.
  - Supports multiple sample rates (8 kHz; 11.025 kHz; 12kHz; 16 kHz; 22,05 kHz; 24kHz; 32kHz; 44,1 kHz; 48kHz).
  - Fully configurable clock polarity: Data can be sampled or sent independently at either rising edge or falling edge of the clock.
  - Configurable TX/RX data delays: Digital audio data in can be aligned or 1/2/3 cycles delayed to LRCK edge. Digital audio data out can be aligned or 1 cycle delayed to LRCK edge. 1 cycle supplementary delay is possible for TX data in slave mode.
  - Support Audio mode: I2S compatible but can have more configurations.
  - Support voice mode: One cycle strobe pulse at the beginning of each new sample. Compatible to Maxim 9851/9853 voice mode.
- DAI interface.
  - 13-bit mono sample.
  - Generate 104 kHz master clock.
  - Starts after a pulse on the DAI_RST line. See *GSM 44014-500* page 40 for more details about DAI.
- Parallel Interface
  - 13-bit mono samples from ADC.
  - 16-bit x 2 stereo samples to stereo DAC.
  - Separate TX and RX strobe lines for synchronization.

3.7 I2C interface

3.7.1 Overview

I2C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. It is most suitable for applications requiring occasional communication over a short distance between many devices. The I2C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously.

The interface defines 3 transmission speeds:

- Normal: 100Kbps
- Fast: 400Kbps
- High speed: 3.5Mbps

Only 100Kbps and 400Kbps modes are supported directly.

RDA8810PL Chip includes 3xI2C masters.

Features

- Compatible with Philips I2C standard
- Multi Master Operation
- Software programmable clock frequency
- Clock Stretching and Wait state generation
- Software programmable acknowledge bit

- Interrupt or bit-polling driven byte-by-byte data-transfers
- Arbitration lost interrupt, with automatic transfer cancellation
- Start/Stop/Repeated Start/Acknowledge generation
- Start/Stop/Repeated Start detection
- Bus busy detection
- Supports 7 and 10bit addressing mode
- Operates from a wide range of input clock frequencies

Operations

- **I2C System configuration**

The I2C system uses a serial data line (SDA) and a serial clock line (SCL) for data transfers. All devices connected to these two signals must have open drain or open collector outputs. The logic AND function is exercised on both lines with external pull-up resistors.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8 bits long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (see START and STOP signals).

- **I2C protocol**

START signal
When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a high-to-low transition of SDA while SCL is high. The START signal denotes the beginning of a new data transfer. A Repeated START is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

The I2C master generates a START signal when the STA-bit in the Command Register is set and the RD or WR bits are set. Depending on the current status of the SCL line, a START or Repeated START is generated.

Slave address transfer
The first byte of data transferred by the master immediately after the START signal is the slave address. This is a seven-bits calling address followed by a RW bit. The RW bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

Note: The I2C master supports 10bit slave addresses by generating two address transfers. See the Philips I2C specifications for more details.

The I2C master treats a Slave Address Transfer as any other write action. Store the slave device’s address in the Transmit Register and set the WR bit. The I2C master will then transfer the slave address on the bus.

Data Transfer
Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a No Acknowledge, the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does not acknowledge the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

To write data to a slave, store the data to be transmitted in the Transmit Register and set the WR bit. To read data from a slave, set the RD bit. During a transfer the core set the TIP flag, indicating that a Transfer is In Progress. When the transfer is done the TIP flag is reset, the IF flag set and, when enabled, an interrupt generated. The Receive Register contains valid data after the IF flag has been set. The user may issue a new write or read command when the TIP flag is reset.

Stop signal
The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the P-bit, is defined as a low-to-high transition of SDA while SCL is at logical ‘1’.

- **Arbitration Procedure**

Clock Synchronization
The I2C bus is a true multi-master bus that allows more than one master to be connected on it. If two or more masters simultaneously try to control the bus, a clock synchronization procedure determines the bus clock. Because of the wired-AND connection of the I2C signals a high to low transition affects all devices connected to the bus. Therefore a high to low transition on the SCL line causes all concerned devices to count off their low period. Once a device clock has gone low it will hold the SCL line in that state until the clock high state is reached. Due to the wired-AND connection the SCL line will therefore be held low by the device with the longest low period, and held high by the device with the shortest high period.

Clock stretching

Slave devices can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave’s SCL low period is greater than the master’s SCL low period, the resulting SCL bus signal low period is stretched, thus inserting wait-states.

3.7.2 Block Diagram

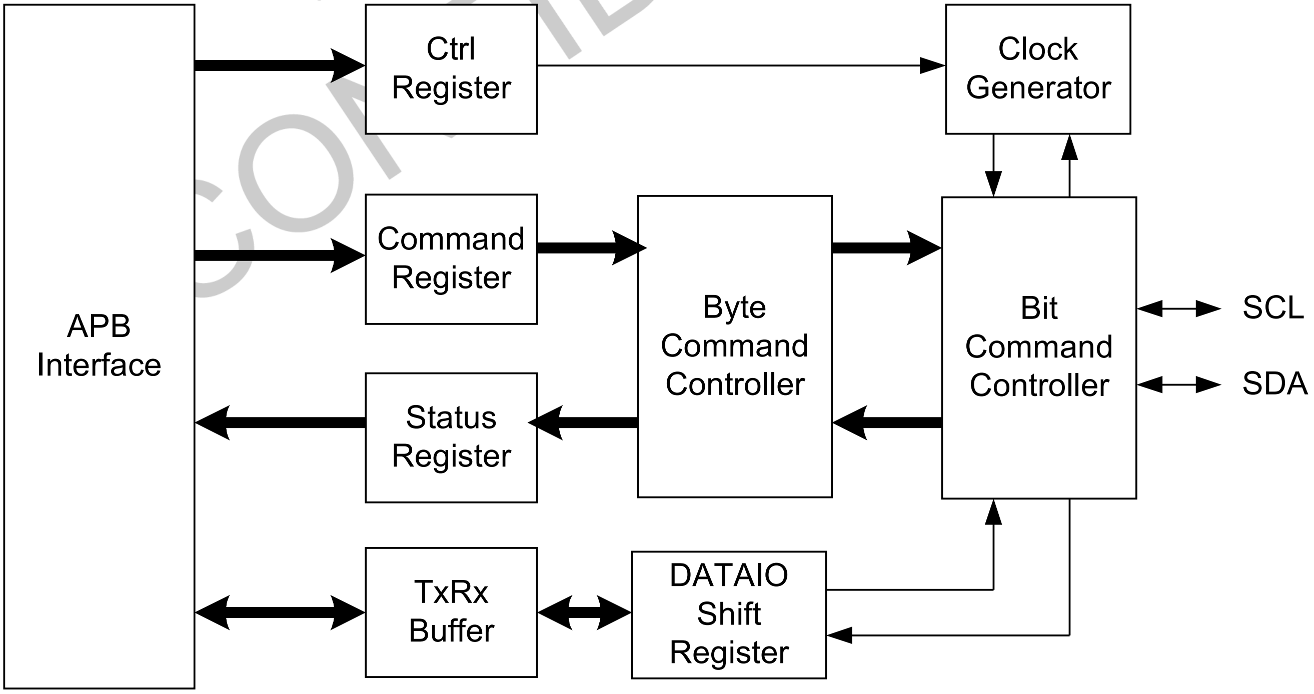


Figure 3.4 I2C Block Diagram

3.7.3 Control Registers

I2C Register Definitions

0x2095_0000 CTRL

Bit	Name	Function	Type	Default
0	EN	I2C master enable, high active.	rw	0
8	IRQ_MASK	I2C master interrupt enable, high active.	rw	0
31:16	Clock_Prescale	This register is used to prescale the SCL clock line. Due to the structure of I2C interface, this module uses a 5*SCL clock frequency. Clock_Prescale must be programmed to this 5*SCL clock frequency (minus 1). Change the value of Clock_Prescale only when bit EN is cleared. Example: PCLK_MOD is 52 MHz, desired SCL is 100 KHz. Prescale = 52MHz / (5 * 100KHz) -1 = 103.	rw	all1

0x2095_0004 STATUS

Bit	Name	Function	Type	Default
0	IRQ_Cause	IRQ Cause bit. This bit is set when one byte transfer has been completed or arbitration is lost, this bit is generated by bit IRQ_Status AND bit IRQ_MASK.	r	0
4	IRQ_Status	IRQ status bit.	r	0
8	TIP	TIP, Transfer in progress. '1' when transferring data. '0' when transfer complete.	r	0
12	AL	AL,Arbitration lost. This bit is set when the I2C master lost arbitration.	r	0
16	Busy	Busy,I2C bus busy. '1' after START signal detected. '0' after STOP signal detected.	r	0
0	RxACK	RxACK, Received acknowledge from slave. '1'= "No ACK" received. '0'= ACK received.	r	0

0x2095_0008 TXRX_BUFFER

Bit	Name	Function	Type	Default
0	TX_DATA	Byte to transmit via I2C. for Bit 0, In case of a data transfer this bit represents the data's LSB. In case of a slave address transfer this bit represents the RW bit. '1' = reading from slave. '0' = writing to slave.	w	-
0	RX_DATA	Last byte received via I2C.	r	-

0x2095_000C CMD

Bit	Name	Function	Type	Default
0	ACK	ACK,when master works as a receiver,sent ACK(ACK='0') or NACK(ACK='1').	w	0
4	RD	RD,read from slave, this bit is auto cleared.	w	0
8	STO	STO,generate stop condition, this bit is auto cleared.	w	0
12	WR	WR,write to slave, this bit is auto cleared.	w	0
16	STA	STA,generate (repeated) start condition, this bit is auto cleared.	w	0

0x2095_0010 IRQ_CLR

Bit	Name	Function	Type	Default
0	IRQ_Clr	When write '1', clears a pending I2C interrupt.	c	0

0x2096_00xx I2C2
0x2097_00xx I2C3

3.8 Keypad interface

3.8.1 Overview

The RDA8810PL keypad is a matrix of 8 rows and 8 columns. Keypad module provides key press, key release detecting mechanism. Each time a key is pressed, keypad module will sense it and will begin to scan the keypad by alternatively driving each column until all keys are released. Besides the keypad matrix, the module also detects key ON, which is an independent input.

The Keypad will send IRQ when either key press or key release event occurs. It can also send IRQ repeatedly when one or several keys are pressed long time. Debounce mechanism is provided to avoid glitches, the time of debounce is programmable.

Any combination of two keys pressed can be detected simultaneously, if certain rules are expected, up to 10 keys can be detected at the same time. However, the following case should be avoided: Note the key at column i1 and row i2 as key(i1, i2). If key(a,b) and key(c,d) is pressed at the same time, then press either key(a,d) or key(b,c) will cause a detection error. The key scanner will consider all of the four keys have been pressed.

Features

- Generate IRQ for both key press and release events.
- Generate IRQ if one or several keys are pressed long time. The IRQ is repeated with a configurable interval from 1*debounce time to 64*debounce time.
- provide debounce mechanism. Configurable debounce time maximum 480 ms.
- Detect two simultaneous key press with any combination. Detect up to 10 keys at the same time if certain patterns are excluded. (i.e. Keypad module can detect entire one row and one column without crossed key).
- Key ON detection.

- Possible to disable some rows and columns of keypad matrix.

3.8.2 Block Diagram

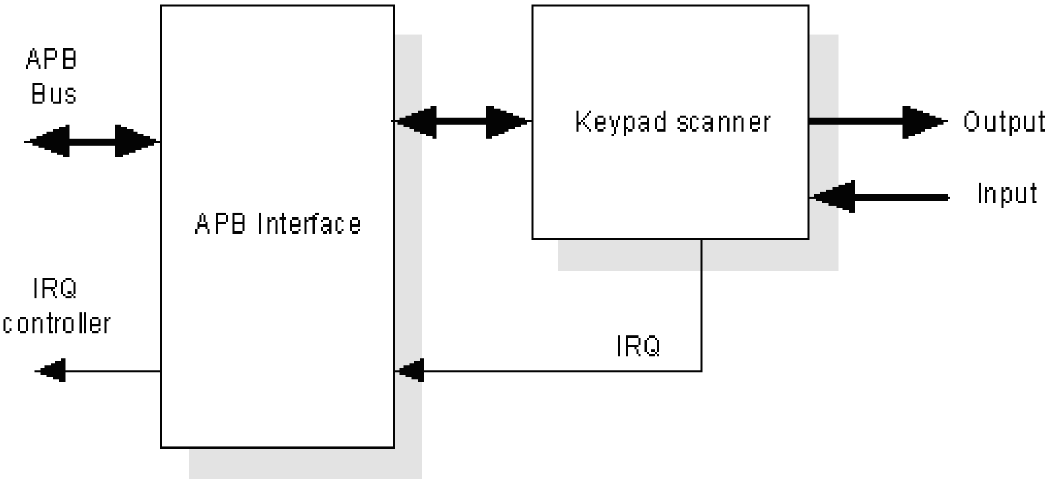


Figure 3.5: Keypad Block Diagram

3.8.3 Control Registers

Keypad Register Definitions

0x2092_0000 KP_DATA_L

Bit	Name	Function	Type	Default
0	KP_DATA_L	For keys in column Idx_KeyOut(from 0 to 3) and in line Idx_KeyIn(from 0 to 7), the pressing status are stored in KP_DATA_L(Idx_KeyOut*8+Idx_KeyIn) :<br />0 = Released<br />1 = Pressed	r	0

0x2092_0004 KP_DATA_H

Bit	Name	Function	Type	Default
31:0	KP_DATA_H	For keys in column Idx_KeyOut(from 4 to 7) and line Idx_KeyIn(from 0 to 7), the pressing status are stored in KP_DATA_H(Idx_KeyIn*8-32+Idx_KeyIn):<br />0 = Released<br />1 = Pressed	r	0

0x2092_0008 KP_STATUS

Bit	Name	Function	Type	Default
0	KEYIN_STATUS	For keys in lines status 0 = Released 1 = Pressed	r	0
31	KP_ON	Indicate Key ON pressing status : 0 = Release 1 = Pressed	r	0

0x2092_000C KP_CTRL

Bit	Name	Function	Type	Default
0	KP_En	This bit enables key detection. If this bit is '0', the key detection function is disabled. Key ON is an exception, it can be still detected and generate key interrupt even if KP_En = '0', however in this case, the debouncing time configuration in key control register is ignored and the key ON state is considerred to be stable if it keeps same in consecutive 2 cycles of 16KHz clock. 0 = keypad disable 1 = keypad enable	rw	0
9:2	KP_DBN_Time	De-bounce time = (KP_DBN_TIME + 1) * SCAN_TIME, SCAN_TIME = 0.3125 ms * Number of Enabled KeyOut (determined by KP_OUT_MASK). For example, if KP_DBN_TIME = 7, KP_OUT_MASK = "111111", then De-bounce time = (7+1)*0.3125*6=15 ms. The maximum debounce time is 480 ms.	rw	0
0	KP_ITV_Time	Configure interval of generating an IRQ if one key or several keys are pressed long time. Interval of IRQ generation = (KP_ITV_Time + 1) * (KP_DBN_TIME + 1) * SCAN_TIME. SCAN_TIME = 0.3125 ms * Number of Enabled KeyOut (determined by KP_OUT_MASK). For example, if KP_ITV_TIME = 7, KP_DBN_TIME = 7, KP_OUT_MASK = "111111", then De-bounce time = (7+1)*(7+1)*0.3125*6=120 ms.	rw	63
23:16	KP_IN_MASK	each bit masks one input lines. '1' = enabled '0' = disabled The Key In pins 0 to 5 are muxed with the boot mode pins, latched during Reset. Key_In 0: BOOT_MODE_NO_AUTO_PU. Key_In 1: BOOT_MODE_FORCE_MONITOR. Key_In 2: BOOT_MODE_UART_MONITOR_ENABLE. Key_In 3: BOOT_MODE_USB_MONITOR_DISABLE. Key_In 4: reserved	rw	63
31:24	KP_OUT_MASK	each bit masks one output lines. '1' = enabled '0' = disabled		63

0x2092_0010 KP_IRQ_MASK

Bit	Name	Function	Type	Default
0	KP_EVT0_IRQ_MASK	This bit mask keypad irq generated by event0 (key press or key release event, not including all keys release event which is event1). 0 = keypad event irq disable 1 = keypad event irq enable	rw	0
1	KP_EVT1_IRQ_M	This bit mask keypad irq generated by event1 (all keys release	rw	0

Bit	Name	Function	Type	Default
	ASK	event). 0 = keypad event irq disable 1 = keypad event irq enable		
2	KP_ITV_IRQ_MASK	This bit mask keypad irq generated by key pressed long time (generated each interval configured in KP_ITV_Time. 0 = keypad interval irq disable 1 = keypad interval irq enable	rw	0

0x2092_0014 KP_IRQ_CAUSE

Bit	Name	Function	Type	Default
0	KP_EVT0_IRQ_CAUSE	keypad event0(key press or key release event, not including all keys release which is event1) IRQ cause.	r	0
1	KP_EVT1_IRQ_CAUSE	keypad event1(all keys release event) IRQ cause.	r	0
2	KP_ITV_IRQ_CAUSE	keypad interval irq cause.	r	0
16	KP_EVT0_IRQ_STATUS	keypad event0(key press or key release event, not including all keys release which is event1) irq status.	r	0
17	KP_EVT1_IRQ_STATUS	keypad event1(all keys release event) irq status.	r	0
18	KP_ITV_IRQ_STATUS	keypad interval irq status.	r	0

0x2092_0018 KP_IRQ_CLR

Bit	Name	Function	Type	Default
0	KP_IRQ_CLR	Write '1' to this bit clears key IRQ.	c	0

3.9 Pulse Width Modulation

3.9.1 Overview

The PWM module generates four independent PWM outputs, utilizing three specialized modulation schemes. There are two Pulse Width Light (PWL) outputs designed for driving LEDs with varying brightness, one Pulse Width Tone (PWT) output for generating tones, and one Light Pulse Generator (LPG) output for blinking an LED at visibly recognizable frequencies.

Features

- Pulse Width Tone (PWT)
    - Generates square wave output capable of driving piezo electric speaker

  - ○ Variable frequency between 349Hz and 5276Hz with 12 half-tone frequencies per octave
  - ○ Volume control (coarse)
- Light Pulse Generation (LPG)
  - ○ Adjustable blinking period from 125 ms to 3 seconds
  - ○ Adjustable on-time from 15 ms to ¼ second.
- Two separate Pulse Width Light (PWL)
  - ○ Pseudo random bit sequence with output on-time proportional to a programmed threshold value
  - ○ Minimizes flicker

Typical Operation

The PWT module generates a 'symmetric' square wave whose duty cycle is variable between 0 and 50% (as opposed to the pseudo-random output in the case of PWL). The generated tones have been tested for half-tone frequency increments between 349Hz and 5276Hz. Basic volume control can be achieved by varying the duty cycle, but the actual values depend on the final frequency and the buzzer being used. Also, changing the duty cycle will change the harmonics, and thus will have an effect on the pitch of the note, but for very coarse operation, this mechanism should suffice.

The Light Pulse Generator (LPG) can be used for blinking a LED at adjustable periods for various signaling purposes. The period is variable between 125 ms through 3 seconds. The length of time the LED will stay on is also programmable multiples of 256/16kHz up to ¼ of a second.

The two PWL modules are optimized for driving back lights and LEDs where pseudo-random pulses with a programmable average on-time are desired. The resulting output will be non-symmetric and will help reduce flicker effects. A low-pass filter can be added to convert the PWL output to an analog value.

3.9.2 Control Registers

PWM Register Definitions

0x2094_0000 PWT_Config

Bit	Name	Function	Type	Default
0	PWT_Enable	Enables the Pulse Width Tone output 1 = Enable PWT output 0 = Disable PWT output	rw	0
13:4	PWT_Duty	The PWT_Duty value can be used to set the approximate volume of the tone. The PWT_Duty value must be less than or equal to half the PWT_Period value and must be at least a value of 8, otherwise no tone will be generated.	rw	all1
26:16	PWT_Period	PWT_Period is the divider value to produce a tone of a given frequency. To calculate the PWT_Period value, Use the following formula: PWT_Period = FBASE/FNOTE where FBASE is the frequency of the PWM module clock (it is based on the system frequency, 26, 39, 52, 78 or 104 MHz divided by 5). FNOTE is the frequency of the desired tone.	rw	all1

0x2094_0004 LPG_Config

Bit	Name	Function	Type	Default
0	LPG_Reset_L	Setting this bit to '0' will reset the Light Pulse Generator internal counters.	rw	
1	LPG_Test	Setting this bit to '1' will enter LPG test mode.	rw	
7:4	LPG_OnTime	Configures the duty cycle for the Light Pulse Generator by setting the ontime for the LPG output. The actual on-time is calculated as: Tick Period * LPG_OnTime * 256 where the Tick Period is nominally 1/16kHz. 0 undefined 1 15_6mS 2 31_2mS 3 46_8mS 4 62mS 5 78mS 6 94mS 7 110mS 8 125mS 9 140mS 10 156mS 11 172mS 12 188mS 13 200mS 14 218mS 15 234mS	rw	0xf
18:16	LPG_Period	Configures the main period of the light pulse generator. The period is calculated based on the following configurations: with the Tick Period ~ 1/16kHz 0 0_125s Tick Period * 2048 1 0_25s Tick Period * 4096 2 0_5s Tick Period * 8192 3 0_75s Tick Period * 12288 4 1s Tick Period * 16384 5 1_25s Tick Period * 20480 6 1_5s Tick Period * 24576 7 1_75s Tick Period * 28672	rw	

0x2094_0008 PWT_Config

Bit	Name	Function	Type	Default
7:0	PWL_Min	Sets the lower boundary for PWL pulse. When pulse mode is not used, this is the threshold value for the PWL0. Reading this value will return the current value used for the threshold.	rw	all1
15:8	PWL_Max	Sets the upper boundary for PWL pulse. When pulse mode is not used, this value is ignored. Reading this value will return the LFSR value used for generating the PWL outputs.	rw	all1
16	PWL0_En_H	When this bit is written with '1', the PWL 0 is enabled and the output is a PRBS whose average on-time is proportional to	rs	0

Bit	Name	Function	Type	Default
		PWL_Min. This bit is cleared when either of the Force bits are written. Reading this bit will return the current state of the PWL0 enable.		
17	PWL0_Force_L	Writing a '1' to this bit will force the PWL0 to output a low value. If the PWL0 was previously enabled, this will clear the bit.	rc	0
18	PWL0_Force_H	Writing a '1' to this bit will force the PWL0 to output a high value. If the PWL0 was previously enabled, this will clear the bit.	s	no
19	PWL_Pulse_En	This will enable the PWL pulse mode. The threshold will dynamically sweep between PWL_Min and PWL_Max at a rate depending on PWL_Pulse_Per.	rw	0
20	PWL0_Set_OE	Writing '1' to this bit will set the output enable. Reading this bit will return the current status.	rs	0
21	PWL0_Clr_OE	Writing '1' to this bit will clear the output enable.	c	no
22	PWL_Set_Mux	Writing a '1' to this bit will swap the PWL0 and PWL1 outputs. Reading this bit will return the current status	rs	0
23	PWL_Clr_Mux	Writing a '1' to this bit will unswap the PWL0/PWL1 outputs.	c	no
31:24	PWL_Pulse_Per	This value will adjust the pulse period when pulsing is enabled.	w	all1

0x2094_000C PWL1_Config

Bit	Name	Function	Type	Default
7:0	PWL1_Threshold	Average duty cycle for the Pulse Width Light 1 output. The average duty cycle is calculated as PWL1_Threshold/256.	rw	all1
15:8	LFSR_Reg	LFSR value for PWL.	r	0xa1
16	PWL1_En_H	When this bit is written with '1', the PWL 1 is enabled and the output is a PRBS whose average on-time is proportional to PWL1_Threshold. This bit is cleared when either of the Force bits are written. Reading this bit will return the current state of the PWL1 enable.	rs	0
17	PWL1_Force_L	Writing a '1' to this bit will force the PWL1 to output a low value. If the PWL1 was previously enabled, this will clear the bit.	rc	0
18	PWL1_Force_H	Writing a '1' to this bit will force the PWL1 to output a high value. If the PWL1 was previously enabled, this will clear the bit.	s	no
20	PWL1_Set_OE	Writing '1' to this bit will set the output enable. Reading this bit will return the current status.	rs	0
21	PWL1_Clr_OE	Writing '1' to this bit will clear the output enable.	c	no

3.10 GPIO

3.10.1 Overview

RDA8810PL GPIO modules has configurable number of General Purpose Input or Output ports (GPIO) and General Purpose Output port(GPO). GPIO are used to control external chips, or to get external events. They can be used as inputs or outputs. As input, 21 of them can be configured to trigger an interruption (GPIO0 to GPIO7). The IRQ could be generated on edge or level.

Features

- GPIOs configurable as input or output.
- GPOs.
- 21 of the GPIOs can generate interrupt. Various Interrupt triggered mode.
  - Rising/Falling edge.
  - High/Low level.

*Maybe not all of the GPIOs/GPOs can be used because a few are not packaged to the pin pad due to the limitation of the total pin numbers. Please refer to the details of reference design.

3.10.2 Block Diagram

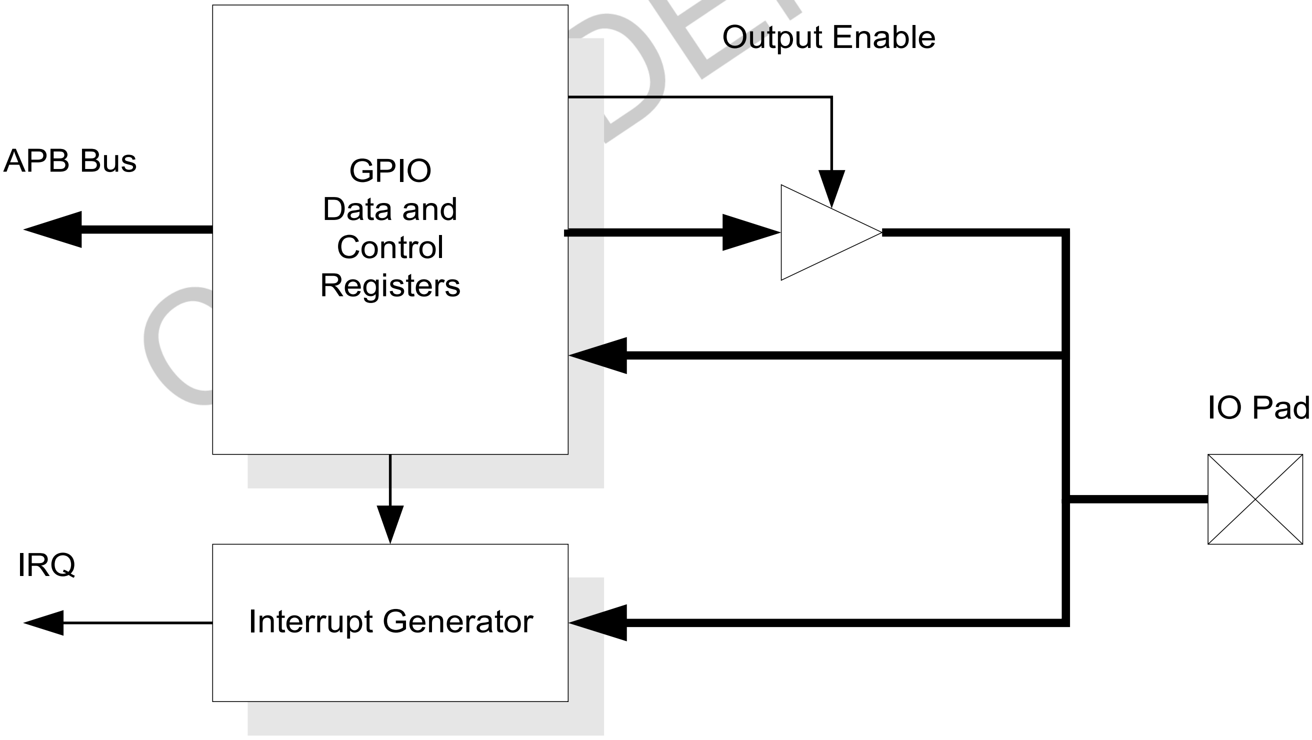


Figure 3.6 GPIO Block Diagram

3.10.3 Control Registers

GPIO Register Definitions

0x2093_0000 gpio_oen_val

Bit	Name	Function	Type	Default
31:0	oen_val	Set the direction of the GPIO n. 0 = output 1 = input	rw	all1

0x2093_0004 gpio_oen_set_out

Bit	Name	Function	Type	Default
31:0	oen_set_out	Write '1' sets the corresponding GPIO pin as output.	rc	0

0x2093_0008 gpio_oen_set_in

Bit	Name	Function	Type	Default
31:0	oen_set_in	Write '1' sets the corresponding GPIO pin as input.	rs	all1

0x2093_000C gpio_val

Bit	Name	Function	Type	Default
31:0	gpio_val	When write, update the output value. When read, get the input	rw	all1

0x2093_0010 gpio_set

Bit	Name	Function	Type	Default
31:0	gpio_set	Write '1' will set GPIO output value. When read, get the GPIO	rs	0

0x2093_0014 gpio_clr

Bit	Name	Function	Type	Default
31:0	gpio_clr	Write '1' clears corresponding GPIO output value. When read, get the GPIO	rc	0

0x2093_0018 gpint_ctrl_set

Bit	Name	Function	Type	Default
0	gpint_r_set	Write '1' will set GPIO interrupt mask for rising edge and level high. When read, get the GPIO interrupt mask for rising edge and level high.	rs	0

Bit	Name	Function	Type	Default
15:8	gpint_f_set	Write '1' will set GPIO interrupt mask for falling edge and level low. When read, get the GPIO interrupt mask for falling edge and level low.	rs	0
23:16	dbn_en_set	Write '1' will enable debounce mechanism.	rs	0
31:24	gpint_mode_set	Write '1' will set interruption mode to level.	rs	0

0x2093_001C gpint_ctrl_clr

Bit	Name	Function	Type	Default
0	gpint_r_clr	'Write '1' will clear GPIO interrupt mask for rising edge and level high.	rc	0
15:8	gpint_f_clr	Write '1' will clear GPIO interrupt mask for falling edge and level low.	rc	0
23:16	dbn_en_clr	Write '1' will disable debounce mechanism.	rc	0
31:24	gpint_mode_clr	Write '1' will set interruption mode to edge triggered.	rc	0

0x2093_0020 int_clr

Bit	Name	Function	Type	Default
7:0	gpint_clr	'Write '1' will clear GPIO interrupt.	c	0

0x2093_0024 int_status

Bit	Name	Function	Type	Default
7:0	gpint_status	Each bit represents if there is a GPIO interrupt	r	0

0x2093_0028 chg_ctrl

Bit	Name	Function	Type	Default
0	out_time	time for which GPIO0 is set to output mode, after a start read DCON command is issued. The output time = (OUT_TIME+1)*30.5us.	rw	all1
9:4	wait_time	time for which GPIO0 should wait before reading DC_ON, after a start read DCON command is issued. The wait time = (WAIT_TIME+1)*30.5us. NOTE: wait_time must be strictly greater than out_time;	rw	all1
17:16	int_mode	interruption mode of GPIO0 in mode DC_ON detection. 0x0 L2H "00" = send IRQ if last read DCON is '0' and now is '1'. 0x1 H2L "01" = send IRQ if last read DCON is '1' and now is '0'. 0x3 RR "11" = send IRQ every time read is ready.	rw	11

0x2093_002C chg_cmd

Bit	Name	Function	Type	Default
0	dcon_mode_set	Write '1' to set GPIO0 to charger DCON detect mode.	s	0
4	chg_mode_set	Write '1' to set GPO0 to charger watchdog mode.	s	0
8	dcon_mode_clr	Write '1' to clear charger DCON detect mode of GPIO0.	c	0
12	chg_mode_clr	Write '1' to clear the charger watchdog mode of GPO0.	c	0
24	chg_down	Write '1' to generate a pulse of '0' on GPO0 for 16 CLK_OSC cycles.	s	0

0x2093_0030 gpo_set

Bit	Name	Function	Type	Default
11:0	gpo_set	Write '1' will set GPO output value. When read, get the GPO	rs	0xaaaaaaaa

0x2093_0034 gpo_clr

Bit	Name	Function	Type	Default
11:0	gpo_clr	Write '1' will clear GPO output value. When read, get the GPO	rc	0xaaaaaaaa

3.11 GPADC

3.11.1 Overview

The GPADC module controls the analog module of general purpose analog to digital converter, which is capable of measuring one of 5 analog external inputs of the chip. The 5 external inputs can be connected to sensors to measure battery voltage, battery temperature, handset temperature, accessory ID, remote control, or other low frequency analog signal within the specified voltage range.

The GPADC is active periodically at a variable rate if at least one channel is enabled. The value of each channel is readable through the APB interface.

Features

- Up to 5 analog external channels.
- Automatic measurement at variable rate.
- Interrupt when programmed threshold is passed.

Operation

By default, GPADC works in functional mode, GPADC is active periodically at a variable rate and measures all the enabled channel (up to 5) each time.

There is also a test mode for debug purpose, in which GPADC receives a start measure command from APB, and do a measurement once.

Usually, the one GPADC is used for mic detect, and another is for user defined function.

3.11.2 Block Diagram

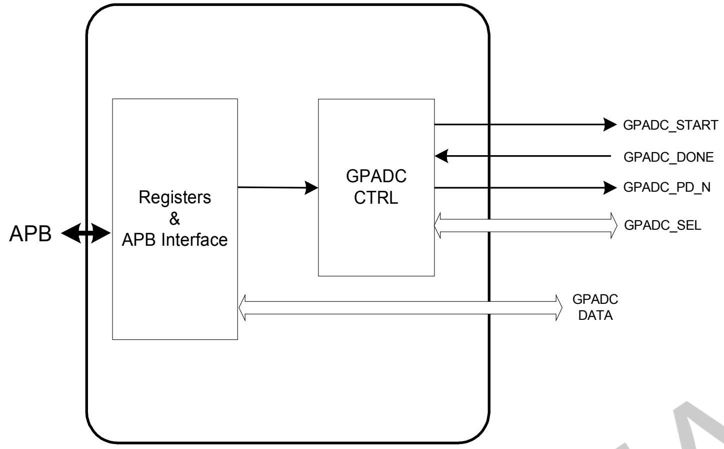


Figure 3.7 GPADC Block Diagram

3.11.3 Interface Timing Diagrams

If at least one channel is enabled, the GPADC will be active and begins to measure the enabled channel's value. After one measurement is finished, a “GPADC EOC” will be set in the APB register, and the GPADC will be disabled. The measurements are repeated with a period defined in register “GPADC_ATP”.

The following diagram shows the case in which all 5 analog channels are enabled:

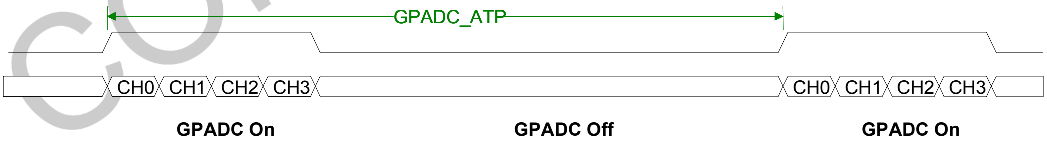


Figure 3.8 GPADC timing Diagram

Note: Acco gpadc is powered down after each measurement is done and power up at least one cycle of 16K clock before the next measurement.

3.12 Timer

3.12.1 Overview

RDA8810PL includes 1 OS timer, and 1 hardware delay timers. The OS timer serves for OS, and has Ticks of 16384 Hz or 2M Hz. It is a decrement timer and the initial value can be programmed. When the OS timer reaches 0, it will cause an interruption. After that, the OS timer will either stop or restart with the initial value (loop mode) or wrap to maximum value (wrap mode).

The hardware delay timer is an incremental timer. It has Ticks of 16384 Hz or 2M Hz. This timer is free running, and the value will be reset to 0. The timer value will wrap to 0 after 0xfffffff, and will generate an IRQ when wrap. The hardware delay timer can also generate an interval IRQ which can be configured to every 1/8 second,1/4 second, 1/2 second, or 1 second.

The timers module has two IRQs. One dedicated for OS timer, the other is used by all the other timers. (including watchdog timer, hardware delay timer)

Features

- 1 56-bit decremental timer for OS.
- 1 64-bit incremental hardware delay timer.
- Multiple IRQ sources: timers wrap, interval arrives.

3.12.2 Block Diagram

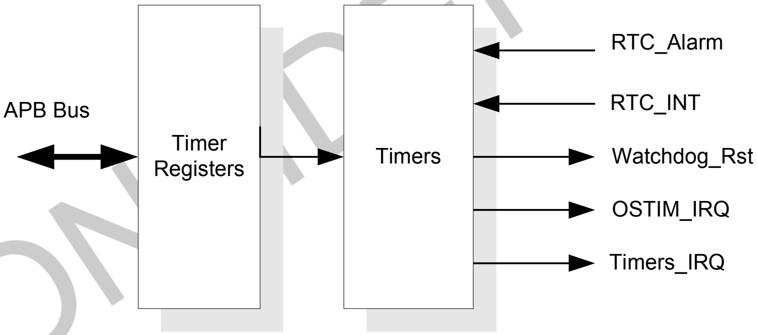


Figure 3.9 Timer Block Diagram

3.12.3 Control Registers

TIMER Register Definitions

0x01A0_2000 OSTimer_Ctrl

Bit	Name	Function	Type	Default
23:0	LoadVal	Value loaded to OS timer.	rw	0
24	Enable	Write '1' to this bit will enable OS timer. When read, the value is what we have written to this bit, it changes immediately after been written.	rw	0
25	Enabled	Read this bit will get the information if OS timer is really	r	0

Bit	Name	Function	Type	Default
		enabled or not. This bit will change only after the next front of 16 KHz system clock. '1' indicates OS timer enabled. '0' indicates OS timer not enabled.		
26	Cleared	Read this bit will get the information if OS timer interruption clear operation is finished or not. '1' indicates OS timer interruption clear operation is on going. '0' indicates no OS timer interruption clear operation is on going.	r	0
28	Repeat	Write '1' to this bit will set OS timer to repeat mode. When read, get the information if OS timer is in repeat mode. '1' indicates OS timer in repeat mode. '0' indicates OS timer not in repeat mode.	rw	0
29	Wrap	Write '1' to this bit will set OS timer to wrap mode. When read, get the information if OS timer is in wrap mode. '1' indicates OS timer in wrap mode. '0' indicates OS timer not in wrap mode.	rw	0
30	Load	Write '1' to this bit will load the initial value to OS timer.	rw	0

0x01A0_2004 OSTimer_CurVal

Bit	Name	Function	Type	Default
31:0	CurVal	Current value of OS timer. The value is 24 bits and the first 8 bits are sign extension of the most important bit. A negative value indicates that the timer has wraped.	r	-

0x01A0_2008 WDTimer_Ctrl

Bit	Name	Function	Type	Default
0	Start	Write '1' to this bit will enable watchdog timer and Load it with WDTimer_LoadVal.	s	0
4	Stop	Write '1' to this bit will stop watchdog timer.	c	0
16	ReLoad	Write '1' to this bit will load WDTimer_LoadVal value to watchdog timer. Use this bit to implement the watchog keep alive.	w	0
8	WDEnabled	Read this bit will get the information if watchdog timer is really enabled or not. This bit will change only after the next front of 32 KHz system clock. '1' indicates watchdog timer is enabled, if current watchdog timer value reaches 0, the system will be reseted. '0' indicates watchdog timer is not enabled.	r	0

0x01A0_200C WDTimer_LoadVal

Bit	Name	Function	Type	Default
23:0	LoadVal	Load value of watchdog timer. Number of 32kHz Clock before Reset.	r	-

0x01A0_2010 HWTimer_Ctrl

Bit	Name	Function	Type	Default
1:0	Interval	interval of generating an HwTimer IRQ. "00": interval of 1/8 second. "01": interval of 1/4 second. "10": interval of 1/2 second. "11": interval of 1 second.	rw	00
8	Interval_En	This bit enables interval IRQ mode. '0': hw delay timer does not generate interval IRQ. '1': hw delay timer generate an IRQ each interval.	rw	0

0x01A0_2014 HWTimer_CurVal

Bit	Name	Function	Type	Default
31:0	CurVal	Current value of the hardware delay timer. The value is incremented every 61 us. This timer is running all the time and wrap at value 0xFFFFFFFF.	r	0

0x01A0_2018 Timer_Irq_Mask_Set

Bit	Name	Function	Type	Default
0	OSTimer_Mask	Set mask for OS timer IRQ.	rs	0
1	HWTimer_Wrap_Mask	Set mask for hardwre delay timer wrap IRQ.	rs	0
2	HWTimer_Itv_Mask	Set mask for hardwre delay timer interval IRQ.	rs	0

0x01A0_201C Timer_Irq_Mask_Clr

Bit	Name	Function	Type	Default
0	OSTimer_Mask	Clear mask for OS timer IRQ.	rc	0
1	HWTimer_Wrap_Mask	Clear mask for hardwre delay timer wrap IRQ.	rc	0
2	HWTimer_Itv_Mask	Clear mask for hardwre delay timer interval IRQ.	rc	0

0x01A0_2020 Timer_Irq_Clr

Bit	Name	Function	Type	Default
0	OSTimer_Mask	Clear OS timer IRQ.	c	
1	HWTimer_Wrap_Mask	Clear hardwre delay timer wrap IRQ.	c	
2	HWTimer_Itv_Mask	Clear hardwre delay timer interval IRQ.	c	

0x01A0_2024 Timer_Irq_Cause

Bit	Name	Function	Type	Default
0	OSTimer_Cause	OS timer IRQ cause.	r	0
1	HWTimer_Wrap_Cause	hardware delay timer wrap IRQ cause.	r	0
2	HWTimer_Itv_Cause	hardware delay timer interval IRQ cause.	r	0
16	OSTimer_Status	OS timer IRQ status.	r	0
17	HWTimer_Wrap_Status	hardware delay timer wrap IRQ status.	r	0
18	HWTimer_Itv_Status	hardware delay timer interval IRQ status.	r	0

3.13 Calendar

3.13.1 Overview

The Calendar module provides date and time information. It works on the 32.768 KHz oscillator with independent power supply. Calendar module is able to generate an IRQ repetitively at certain interval, which can be programmed to 1 second, 1 minute or 1 hour. In addition to provide timing data, alarm interrupt is generated and it is also used to power-up the baseband core by sending wakeup signal. The year span is supported from 2000 to 2127, the maxim day of each month is stored in the calendar block, which depends on the leap year condition.

Features

- Independent power supply.
- Counters for second, minute, hour, day, month, year and day of week.
- Maxim day of each month stored in module, leap year supported.
- Alarm generate, wakeup triggered by alarm. Alarm IRQ.
- Periodical IRQ for certain intervals.

3.13.2 Block Diagram

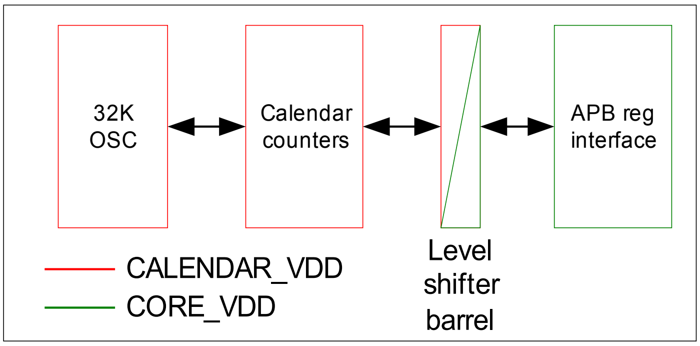


Figure 3.10 Calendar Block Diagram

3.13.3 Control Registers

Calender Register Definitions

0x01A0_6000 Ctrl

Bit	Name	Function	Type	Default
0	Interval	These 2 bits configure the interval of generating an IRQ status. 0 DISABLE 1 PER_SEC 2 PER_MIN 3 PER_HOUR	rw	00

0x01A0_6004 Cmd

Bit	Name	Function	Type	Default
0	Calendar_Load	When write, command to program calendar with a new value (sec, min, hour, day, month, year, day of week) previously written in registers Calendar_LoadVal_H and Calendar_LoadVal_L. This bit is auto cleared. '1' = load calendar timer. When read, Calendar timer load status. '1' = Calendar load has not finished. '0' = Calendar load has finished.	rs	0
4	Alarm_Load	When write, command to program alarm with a new value (sec, min, hour, day, month, year, day of week) prviously written in registers AlarmVal_H and AlarmVal_L. This bit is auto cleared. '1' = load alarm. When read, alarm load status. '1' = alarm load has not finished. '0' = alarm load has finished.	rs	0
5	Alarm_Enable_Set	command to enable alarm. When alarm is triggered, it will generate a wakup. '1' = enable alarm. When read, alarm enable status. '1' = alarm enable operation is on going, not finished. '0' = alarm is enabled.	rs	0
6	Alarm_Enable_Clr	command to disable alarm. '1' = disable alarm. When read, alarm enable status. '1' = alarm disable operation is on going, not finished. '0' = alarm is disabled.	rc	0
8	Alarm_Clr	writing '1', clear Alarm triggered signal (connect to wakeup) and alarm triggered IRQ. When read, get alarm clear status. '1' = alarm clear operation is on going, not finished. '0' = alarm is cleared.	rc	0
9	Itv_Irq_Clr	writing '1', clear interval IRQ.	c	0
16	Itv_Irq_Mask_Set	When write '1', Set interval Irq Mask. When read, get interval Irq mask.	rs	0
17	Itv_Irq_Mask_Clr	When write '1', Clear interval Irq Mask. When read, get inteval Irq mask.	rc	0

Bit	Name	Function	Type	Default
31	Calendar_Not_Valid	When write '1', mark calendar value to be not valid. When read, Indicate if the Calendar value is valid or not. The calendar value is not valid in case of mismatch between the calendar counter and the APB register, which is the case of wakeup the phone after shut down. This mismatch disappear after one RTC cycle or after re-porgramming a new calendar value. '1' = not valid.	rs	0

0x01A0_6008 Status

Bit	Name	Function	Type	Default
0	Itv_Irq_Cause	Interval Irq Cause	r	0
1	Alarm_Irq_Cause	Alarm Irq Cause.	r	0
8	Force_Wakeup	Force Wakeup status. After set "Force_Wakeup" to '1' in sys_ctrl, the real force_wakeup is not set immediatly, this bit indicates when the force wakeup is really set. This bits also indicates if the interface between Calendar domain and Core domain is enabled. '1': force wakeup set.	r	0
12	Chg_Mask	Charger Mask status. After set "Chg_Mask" to '1' in sys_ctrl, the real Chg_Mask line is not set immediatly, this bit indicates when the Chg_Mask line is really set. '1': Chg_Mask line set.	r	0
16	Itv_Irq_Status	Interval Irq Status.	r	0
20	Alarm_Enable	Alarm Enable Status. Note: When calendar is not programmed, Alarm can be enabled or not. It is suggested to clear Alarm Enable when program RTC.	r	0
31	Calendar_Not_Prog	'1' = Calendar has not been programmed. This bit keep value '0' after the calendar is programmed once.	r	0

0x01A0_600C Calendar_LoadVal_L

Bit	Name	Function	Type	Default
0	Sec	Second value loaded to calendar, ranged from 0 to 59.	rw	-
13:8	Min	Minute value loaded to calendar, ranged from 0 to 59.	rw	-
0	Hour	Hour value loaded to calendar, ranged from 0 to 23.	rw	-

0x01A0_6010 Calendar_LoadVal_H

Bit	Name	Function	Type	Default
0	Day	Day value loaded to calendar, ranged from 1 to 31.	rw	-
11:8	Mon	Month value loaded to calendar, ranged from 1 to 12.	rw	-
22:16	Year	Year value loaded to calendar, ranged from 0 to 127. Represent year 2000 to 2127.	rw	-
26:24	WeekDay	Day of the week value loaded to calendar, ranged from 1 to 7. Represent Monday, Tuesday etc.	rw	-

0x01A0_6014 Calendar_CurVal_L

Bit	Name	Function	Type	Default
0	Sec	Second value loaded to calendar, ranged from 0 to 59.	rw	-
13:8	Min	Minute value loaded to calendar, ranged from 0 to 59.	rw	-
0	Hour	Hour value loaded to calendar, ranged from 0 to 23.	rw	-

0x01A0_6018 Calendar_CurVal_H

Bit	Name	Function	Type	Default
0	Day	Day value loaded to calendar, ranged from 1 to 31.Current Day value of calendar, ranged from 1 to 31. Maximum number of days in each month are stored in the module, and leap year is supported, so February can have 28 or 29 days.	r	-
11:8	Mon	Month value loaded to calendar, ranged from 1 to 12. Current Month value of calendar, ranged from 1 to 12.	r	-
22:16	Year	Current Year value of calendar, ranged from 0 to 127. Represent year 2000 to 2127.	r	-
26:24	WeekDay	Current Day of the week value of calendar, ranged from 1 to 7. Represent Monday, Tuesday etc.	r	-

0x01A0_601C AlarmVal_L

Bit	Name	Function	Type	Default
0	Sec	Second value loaded to calendar, ranged from 0 to 59.	rw	-
13:8	Min	Minute value loaded to calendar, ranged from 0 to 59.	rw	-
0	Hour	Hour value loaded to calendar, ranged from 0 to 23.	rw	-

0x01A0_6020 AlarmVal_H

Bit	Name	Function	Type	Default
0	Day	Day value loaded to calendar, ranged from 1 to 31.	rw	-
11:8	Mon	Month value loaded to calendar, ranged from 1 to 12.	rw	-
22:16	Year	Year value loaded to calendar, ranged from 0 to 127. Represent year 2000 to 2127.	rw	-

3.14 SDIO controller

3.14.1 Overview

This module connects inner bus and outer SD or MMC card. It receives the inner command and data, transfers it to outer SD or MMC card, and transfer response or data back.

Features

- SD Card Specification Version 2.0
- SDIO Version 1.10
- MMC specification Version 3.1
- Hot insertion and removal of media cards will be considered by GPIO module

3.14.2 Block Diagram

The SD/MMC controller block diagram has two parts, the inner bus interface part, the SD/MMC interface part. The inner bus interface connects with the APBI module, and it includes three parts, the APB interface, the FIFO interface and the interrupt signal. The SD/MMC interface is the bridge between the inner bus interface and the SD/MMC bus, and it is in charge of receiving or transferring the data between FIFO and SD/MMC card.

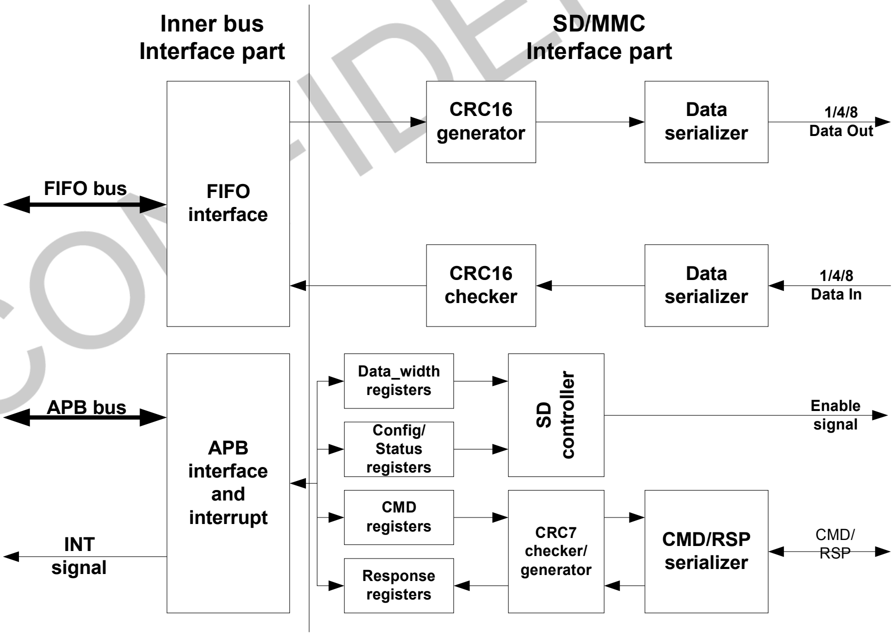


Figure 3.11: SD/MMC Controller Block Diagram

3.14.3 Control Registers

SD_MMC Register Definitions

0x20A5_0000 apbi_ctrl_sdmmc

Bit	Name	Function	Type	Default
0	L_Endian	Controls the big endian or little endian of the FIFO data. Take 32 bit data 0X0A0B0C0D for Example,bit[31:24]=Byte3,bit[23:16]=Byte2,bit[15:8]=Byte1,bit[7:0]= Byte0. "000": the order is not changed. Byte3="0A",Byte2="0B",Byte1="0C",Byte0="0D". "001": reversed on byte. Byte3="0D",Byte2="0C,Byte1="0B",Byte0="0A". "010": reversed on half word. Byte3="0C",Byte2="0D,Byte1="0A",Byte0="0B". "010": reversed on bit. Byte3="B0",Byte2="30,Byte1="D0",Byte0="50". "100": reversed on bit. Byte3="0A",Byte2="0X,Byte1="0D",Byte0="0C".	rw	0
3	Soft_rst_L	For the software to clear FIFO in case there is an error in communication with SD controller and some data are left behind. Active Low.	rw	1

0x20A5_0008 APBI_FIFO_TxRx

Bit	Name	Function	Type	Default
0	DATA_IN	Write to the transmit FIFO	w	
0	DATA_OUT	Read in the receive FIFO	r	

0x20A5_0800 SDMMC_CONFIG

Bit	Name	Function	Type	Default
0	SDMMC_SENDCMD	SD/MMC operation begin register, active high. When '1', the controller finishes the last command and goes into suspend status. At suspend status, the controller will not execute the next command until the bit is set '0'.	rw	0
1	SDMMC_SUSPEND	SD/MMC operation suspend register, active high.	rw	1
4	RSP_EN	'1'indicates having a response,'0'indicates no response.	rw	0
6:5	RSP_SEL	Response select register,"10" means R2 response, "01" means R3 response, "00" means others response, "11" is reserved. 2 R2 1 R3 0 OTHER		
8	RD_WT_EN	'1' indicates data operation, which includes read and write.	rw	0
9	RD_WT_EN	'1' means write operation,'0' means read operation. 0 READ	rw	0

Bit	Name	Function	Type	Default
		1 WRITE		
10	S_M_SEL	'1'means multiple block data operation. 0 SIMPLE 1 MULTIPLE	rw	0

0x20A5_0804 SDMMC_STATUS

Bit	Name	Function	Type	Default
0	Not_SDMMC_OVER	'1' means the SD/MMC operation is not over.	r	0
1	BUSY	'1' means SD/MMC is busy.	r	0
2	DL_BUSY	'1' means the data line is busy.	r	0
3	SUSPEND	'1' means the controller will not perform the new command when SDMMC_SENDCMD= '1'	r	0
8	RSP_ERROR	Response CRC checks error register '1' means response CRC check error.	r	0
9	NO_RSP_ERROR	'1' means the card has no response to command.	r	0
14:12	CRC_STATUS	CRC check for SD/MMC write operation "101" transmission error "010" transmission right "111" flash programming error	r	0
23:16	DATA_ERROR	8 bits data CRC check, "00000000" means no data error, "00000001" means DATA0 CRC check error, "10000000" means DATA7 CRC check error, each bit match one data line.	r	0
24	DAT3_VAL	SDMMC DATA 3 value.	r	-

0x20A5_0808 SDMMC_CMD_INDEX

Bit	Name	Function	Type	Default
5:0	COMMAND	SD/MMC command register.	rw	0

0x20A5_080C SDMMC_CMD_ARG

Bit	Name	Function	Type	Default
31:0	ARGUMENT	SD/MMC command argument register, write data to the SD/MMC card.	rw	0

0x20A5_0810 SDMMC_RESP_INDEX

Bit	Name	Function	Type	Default
0	RESPONSE	SD/MMC response index register.	r	0

0x20A5_0814 SDMMC_RESP_ARG3

Bit	Name	Function	Type	Default
31:0	ARGUMENT3	Response argument of R1, R3 and R6, or 127 to 96 bit response argument of R2.	r	0

0x20A5_0818 SDMMC_RESP_ARG2

Bit	Name	Function	Type	Default
31:0	ARGUMENT2	95 to 64 bit response argument of R2.	r	0

0x20A5_081C SDMMC_RESP_ARG1

Bit	Name	Function	Type	Default
31:0	ARGUMENT3	63 to 32 bit response argument of R2.	r	0

0x20A5_0820 SDMMC_RESP_ARG0

Bit	Name	Function	Type	Default
31:0	ARGUMENT0	31 to 0 bit response argument of R2.	r	0

0x20A5_0824 SDMMC_DATA_WIDTH

Bit	Name	Function	Type	Default
0	SDMMC_DATA_WIDTH	SD/MMC data width: 0x1: 1 data line 0x2: 2 reserved 0x4: 4 data lines 0x8: 8 data lines	rw	0

0x20A5_0828 SDMMC_BLOCK_SIZE

Bit	Name	Function	Type	Default
0	SDMMC_BLOCK_SIZE	SD/MMC size of one block: 0-1:reserved 2: 1 word 3: 2 words 4: 4 words 5: 8 words 6: 16 words 11: 512 words 12-15 reserved	rw	0

0x20A5_082C SDMMC_BLOCK_CNT

Bit	Name	Function	Type	Default
0	SDMMC_BLOCK_CNT	Block number that wants to transfer.	rw	0

0x20A5_0830 SDMMC_INT_STATUS

Bit	Name	Function	Type	Default
0	NO_RSP_INT	'1' means no response.	r	0
1	RSP_ERR_INT	'1' means CRC error of response.	r	0
2	RD_ERR_INT	'1' means CRC error of reading data.	r	0
3	WR_ERR_INT	'1' means CRC error of writing data.	r	0
4	DAT_OVER_INT	'1' means data transmission is over.	r	0
5	TXDMA_DONE_INT	'1' means tx dma done.	r	0
6	RXDMA_DONE_INT	'1' means rx dma done.	r	0
8	NO_RSP_SC	'1' means no response is the source of interrupt.	r	0
9	RSP_ERR_SC	'1' means CRC error of response is the source of interrupt.	r	0
10	RD_ERR_SC	'1' means CRC error of reading data is the source of interrupt.	r	0
11	WR_ERR_SC	'1' means CRC error of writing data is the source of interrupt.	r	0
12	DAT_OVER_SC	'1' means the end of data transmission is the source of interrupt.	r	0
13	TXDMA_DONE_SC	'1' means tx dma done is the source of interrupt.	r	0
14	RXDMA_DONE_SC	'1' means rx dma done is the source of interrupt.	r	0

0x20A5_0834 SDMMC_INT_MASK

Bit	Name	Function	Type	Default
0	NO_RSP_MK	When no response, '1' means INT is disable.	rw	0
1	RSP_ERR_MK	When CRC error of response, '1' means INT is disable.	rw	0
2	RD_ERR_MK	When CRC error of reading data, '1' means INT is disable.	rw	0
3	WR_ERR_MK	When CRC error of writing data, '1' means INT is disable.	rw	0
4	DAT_OVER_MK	When data transmission is over, '1' means INT is disable.	rw	0
5	TXDMA_DONE_MK	when tx dma done, '1' means INT is disabled.	rw	0
6	RXDMA_DONE_MK	'1' means rx dma done, '1' means INT is disabled.	rw	0

0x20A5_0838 SDMMC_INT_CLEAR

Bit	Name	Function	Type	Default
0	NO_RSP_CL	Write a '1' to this bit to clear the source of interrupt in	w	0

Bit	Name	Function	Type	Default
		NO_RSP_SC.		
1	RSP_ERR_CL	Write a '1' to this bit to clear the source of interrupt in RSP_ERR_SC.	w	0
2	RD_ERR_CL	Write a '1' to this bit to clear the source of interrupt in RD_ERR_SC.	w	0
3	WR_ERR_CL	Write a '1' to this bit to clear the source of interrupt in WR_ERR_SC.	w	0
4	DAT_OVER_CL	Write a '1' to this bit to clear the source of interrupt in DAT_OVER_SC.	w	0
5	TXDMA_DONE_CL	Write a '1' to this bit to clear the source of interrupt in TXDMA_DONE_SC.	w	0
6	RXDMA_DONE_CL	Write a '1' to this bit to clear the source of interrupt in RXDMA_DONE_SC.	w	0

0x20A5_083C SDMMC_TRANS_SPEED

Bit	Name	Function	Type	Default
0	SDMMC_TRANS_SPEED	Mclk = Pclk/(2*(SDMMC_TRANS_SPEED +1)).	rw	0

0x20A5_0840 SDMMC_MCLK_ADJUST

Bit	Name	Function	Type	Default
0	SDMMC_MCLK_ADJUST	This register may delay the mclk output. When MCLK_ADJUSTER = n, Mclk is outputted with n Pclk.	rw	0
4	CLK_INV	Invert Mclk.	rw	0

0x20A6_0xxx SDMMC2
0x20A7_0xxx SDMMC3

3.15 EMMC controller

3.16 USB controller

3.16.1 Overview

USB controller primarily provides a ‘Dual-role’ USB controller for use as either the host or the peripheral in point-topoint communications with another USB function (which may be either high-speed, full-speed or low-speed).

It is compatible with the USB standard for high-/full-speed functions and with the *On-The-Go* supplement to the USB 2.0 specification. The USB *On-The-Go* specification has been introduced to provide a low-cost connectivity solution for consumer portable devices such as mobile phones and tablet. Devices that are solely peripherals initiate USB traffic through a Session Request Protocol (SRP) while Dual-role devices support both SRP and Host Negotiation Protocol (HNP).

USB controller supports up to 15 ‘Transmit’ endpoints and/or up to 15 ‘Receive’ endpoints in addition to Endpoint 0. (The use of these endpoints for IN transactions and OUT transactions depends on whether the controller is being used as a peripheral or as a host. When used as a peripheral, IN transactions are processed through Tx endpoints and OUT transactions are processed through Rx endpoints. When used as a host, IN transactions are processed through Rx endpoints and OUT transactions are processed through Tx endpoints.) These additional endpoints can be individually configured in software to handle either Bulk transfers (which also allows them to handle Interrupt transfers) or Isochronous transfers.

Each endpoint requires a FIFO to be associated with it. USBC has a RAM interface for connecting to a single block of synchronous single-port RAM which is used for all the endpoint FIFOs.

The FIFO for Endpoint 0 is required to be 64 bytes deep and will buffer 1 packet. The RAM interface is configurable with regard to the other endpoint FIFOs, which is 8192 bytes in size and can buffer either 1 or 2 packets.

USBC is offered with a 32-bit synchronous CPU interface designed for connection to an AMBA™ AHB bus. The interface supports use with an AHB bus running at a wide range of bus speeds. USBC provides all the encoding, decoding and checking needed in sending and receiving USB packets – interrupting the CPU only when endpoint data has been successfully transferred.

When acting as the host for point-to-point communications, USBC additionally maintains a frame counter and automatically schedules SOF, Isochronous, Interrupt and Bulk transfers. It also includes support for the Session Request and the Host Negotiation Protocols used in point-to-point communications, details of which are given in the *USB On-The-Go* supplement to the USB 2.0 specification.

Features

- Operates either as a function controller for a USB peripheral or as the host/peripheral in point-to-point communications with another USB function
- compatible with the USB 2.0 standard for high-speed (480 Mbps) functions and with the *On-The-Go* supplement
- Supports point-to-point communications with one high-, full- or low-speed device
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports Suspend and Resume signaling
- Supports High-Bandwidth Isochronous & Interrupt transfers
- supports up to 15 additional Transmit endpoints and up to 15 additional Receive endpoints
- Synchronous RAM interface for FIFOs
- Support for DMA access to FIFOs
- High-level 32-bit AMBA AHB-compatible CPU interface (works with a wide range of bus speeds)
- Soft connect/disconnect option
- Performs all transaction scheduling in hardware

3.17 IFC

3.17.1 Overview

The Intelligent Flow Controller is a bridge between the system bus and the peripheral bus. The IFC also provides DMA capabilities to allow data transfer from or to peripherals. It supports 7 DMA standard channels for 8-bit or 32-bit.

Features

- 7 independent DMA channels. internal FIFO of four 32-bit words per channel.
- Burst mode on AHB bus to enhance transfer rate
- Support 2 types of transfer: memory to peripheral and peripheral to memory
- Incremental address for AHB master access and non-incremental address for APB access.
- Dynamic allocation of the 7 DMA channels, request lines among peripherals.
- Hardware semaphore registers which indicate to the CPU which channel must be used and a global status register indicating which channel is free.
- AHB to APB bridge, write buffer for a single write access the master isn't stalled, no wait state inserted

?

4. Modem Subsystem

4.1 Overview

RDA8810 modem supports GSM/GPRS/EDGE. Modem communicates with AP through DPRAM mailbox and AHB interface. Modem could share ddr memory on AP side, or use separate PSRAM. Modem includes XCPU and BCPU. XCPU is mainly in charge of communication protocol, BCPU is mainly in charge of modulation/demodulation. XCPU and BCPU share memory bridge to access modem's internal sram/rom or external PSRAM.

4.2 XCPU side

XCPU side includes XCPU, DMA, interrupt, RFSPI, TCU, SIMIF and bus connect. XCPU is core of this side.

The XCPU RISC is a 16/32-bits processor. Using a Reduced Instruction Set Architecture, an efficient 6-stage instruction pipeline and separated Instruction and Data caches, it provides high performance to the system. The Pipeline Stages are as follows:

- **PC**. Program Counter. Calculate the address of the next instruction and send it to the instruction cache.
- **IF**. Instruction Fetch. In this stage the instruction cache is being accessed and the instruction information is retrieved.
- **RF**. Register File. The register file is being accessed and the instruction is decoded.
- **EX**. Execution. The instruction is executed
- **DC**. Data Memory read and write access.
- **WB**. Write Back. Results are written back to the register file.

Features

- RDA RISC Core.

  - ○ 32x32-bit Multiplier.
  - ○ 32x32-bit -> 64-bit Multiplier Accumulator (MAC) in 2 cycles (pipelined).
  - ○ Read / Write Buffer.
  - ○ 16/32 bit instruction set.
- 32 interrupt sources.
- 4 kByte Instruction Cache.
- 4 kByte Data Cache.
- 16 byte streaming buffer to accelerate uncached instruction accesses.

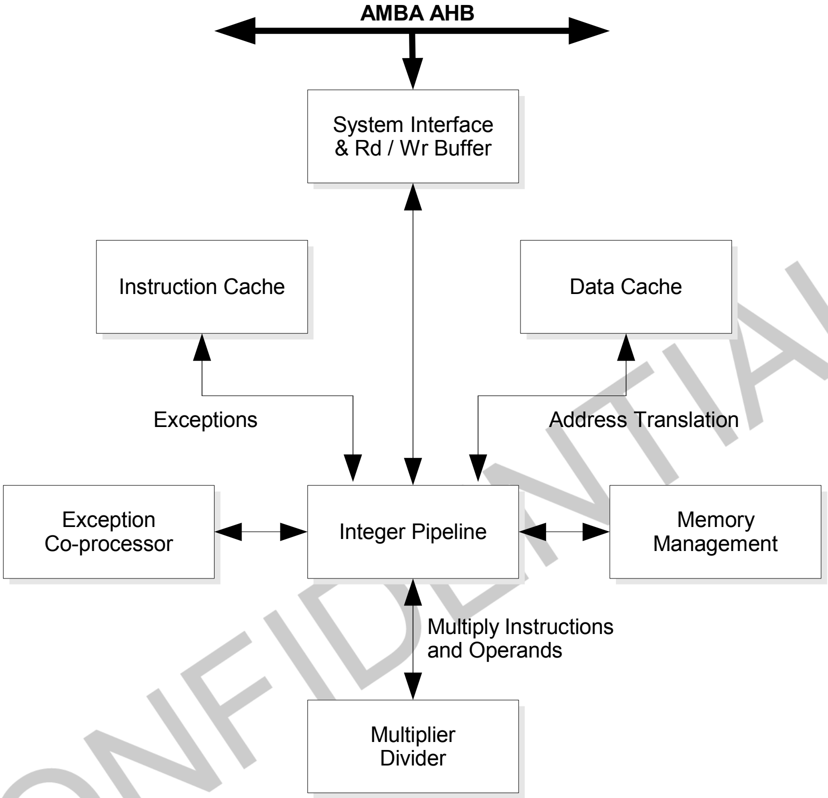


Figure 4.1: XCPU Block Diagram

Block Diagram

4.3 BCPU side

BCPU side works on GSM/GPRS/EDGE modulation/demodulation. It's made up of two parts: the software code running on BCPU and several hardware acceleration modules. It's based on an AHB and APB bus interfaces. BCPU side is connected to XCPU side by the bridge AHB2AHB, and memory bridge is connected to both XCPU and BCPU side.

BCPU uses the same 32-bit RISC core as XCPU, and has 1kB I cache, 1kB D cache. BCPU side also has veterbi, interleaving, correlation, cipher, and cordic accelerator to help BCPU on modulation/demodulation works.

4.4 Memory bridge

The memory bridge is the interface to general memory used by modem subsystem, including internal rom, internal sram and access to external memory.

Features

- Dual AHB Slave
  - A rom/sram controller
  - An Asynchronous FIFO to external controller
- The bridge is implemented as a crossbar between the 2 AHB and the 2 controller (rom/sram and FIFO)
- The rom/sram controller can insert wait cycle to the AHB Slave (to manage read/write conflicts and to allow using slower instance if needed).
- All AHB burst size are supported
  - Wrap will be split at wrap address
  - Burst longer than the data buffer size of the FIFO will be split
  - Read INCR will read a fixed data size
- FIFO data buffer can store 2 requests either read or write (each 4x32 bytes) from either AHB slave interface.
- APB slave for configuration

External Bus Controller Features

- The controller handles 16 bits data bus width only, however 8 bit memory chips can still be used; by groups of 2. 8 bit peripherals must be connected to the 8 LSBs of data and accessed through even addresses only.
- Manage page mode SRAM or FLASH.
- Manage burst mode PseudoSRAM.
- Manage AD-Mux and AD-Mux burst mode PseudoSRAM.
- 1 Control Register Enable (M_CRE) Output pin in same power domain than other memory IO for PseudoSRAM register control.
- FIFO interface for address/data path
- APB interface for configuration (subset of the mem bridge APB address space)

Operations

Dual AHB Operation

The Slave allow simultaneous access to rom/sram controller and FIFO from the two slaves.
Access to the rom/sram controller are sequenced using WAIT on AHB (HReady low).
Access to the FIFO are sequenced using split when the FIFO busy on the other AHB slave (other split conditions are described bellow in FIFO Operation section).

Arbitration in case of simultaneous access to the same controller: Should try to serve each side (BB and Sys) alternatively.
Simultaneous access to rom/sram controller: memorize the last accepted burst. when the two arrives exactly simultaneously reply wait to the memorized side and process the other side.
Simultaneous access to FIFO: First serve read of data if data are available for this read. Serve the side that did not enter the FIFO on last Burst in priority.

FIFO Operation

The FIFO can store 2 requests either read or write from either AHB Slave interface. But only one at a time.
For each request there is an associated Write Data Buffer and a Read Data Buffer.

If the FIFO is full the master is split, this master will be released when one spot is free in the FIFO (all masters split because FIFO was full are released at the same time).

Write
A burst is stored to the Data buffer until the burst is finished. If the Data buffer is full or the Wrap address of a wrap burst is reached, the Split response is send (the master will be released in the same condition as above).

Read
The request is stored (address, length ...) and the master is split.
The master will be released when the Data buffer has been filled by the external controller (EBC or AHB Master). When the Master comes back, the Data are provided, if the master end the burst the transfer is complete (even if there is still some data in the buffer, in this case they are lost) if the master request more than the available Data it receive the Split response and will be released in the same conditions as for FIFO full: the request is not stored.

FIFO Flush status: Read in FIFO at special address space does not impact the external controller but returns when the command has reached the end of the FIFO (so the read returns only when previous writes are done, no pooling is required)

If external controller has error (disabled space for EBC or AHB Error response for master) the FIFO data reads as and error code. “0xD15AB1ED”

EBC Operation

FIFO access are translated to external memory access.

Configuration of Chip select are validated only between access (atomic change)

- Flash block address remapping and M_CRE control registers are placed in the FIFO space to keep accesses in sequence and avoid using FIFO Flush each time.

4.5 TCU

The Timing Control Unit handle the scheduling of individual events with a quarter bit precision.

The TCU is based on a quarter bit counter (13/12 MHz = 1.083 MHz), which wraps at a programmable value. An interruption is generated when the counter wraps (Frame interrupt). The current value of this counter is available at any time. The wrap value can be set at any time. To assure a predictable behavior this value should only be set by the programmer “far” from the wrap time, to make sure this new value will be taken into account the next time.

The programmer can define events associated to particular counter values (dates). When the counter equals the given date, the action associated to the event will take place.

The event are contained in a table of events split into a programming table and an active table. Only the events in the active table are compared to the counter value. The programming table table allows to prepare a full sequence of events which will be transfered to the active table when the counter wraps.

A force signal allows to transfer events from the programming table to the active table immediately.

The LPS Low Power Synchronizer is a part of the TCU, it is responsible for maintaining the time base while running at a slower clock (32kHz) for power saving.

TCU Features

- Quarter bit precision.
- 60 entries event table.
- Programmable counter wrap value.

LPS Features

- Allow skipping frames by masking the TCU Frame Interrupt.
- Calibration of the 32kHz clock against the System Clock with an accuracy of 4 System clocks during a programmable time to reach wanted rate accuracy.
- Low power counters (32kHz) can maintain timebase while System clock is suspended.
- A hardware state machine can assist the CPU for the power up after each exit of the low power mode.

4.6 SIM Card interface

4.6.1 Overview

The SIM Card Interface is a module that automates the communication between the Baseband and the SIM card and works with software drivers to interface with the SIM protocol code. At the physical level, the communication protocol is a character-based, half-duplex, asynchronous, serial protocol, much like half-duplex UART. The RDA8810PL's SIM Card Interface is compatible USIM (T0 only).

Features

- One transmit channel with 5 byte FIFO (5 bytes is the standard command size)
- One receive channel with 4 byte FIFO
- Receive character count flag and interrupt
- Parity bit generation & checking (with inversion)
- NULL character filtering
- Programmable receive timeout
- Automatic reset sequence and protocol format detection
- Hardware assisted direct/inverse convention encoding and decoding
- Programmable guard time and turnaround guard time
- Automatic clock-stop mechanism

4.6.2 Block Diagram

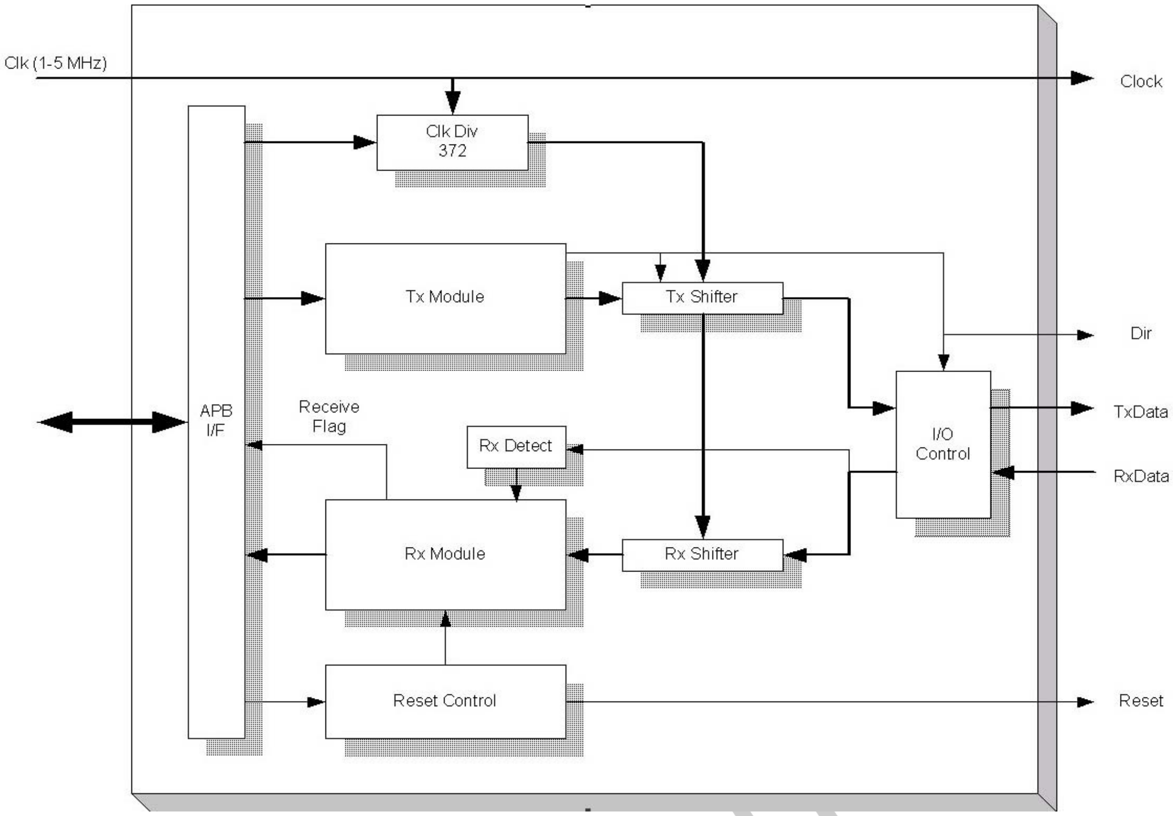


Figure 4.2 SCI Block Diagram

4.6.3 Control Registers

SCI Register Definitions

0x01A1_0000 Config

Bit	Name	Function	Type	Default
0	Enable	Enables the SIM Card IF module	rw	0
1	Parity	Selects the parity generation/detection 0 Even_parity 1 Odd_parity	rw	0
2	PERF	Parity Error Receive Feed-through 0 = Don't store bytes with detected parity errors 1 = Feed-through bytes with detected parity errors	rw	0
3	Filter_Disable	Enable or disable NULL (0x60) character filtering when SIM card sends NULL to reset WWT timer. 0 = Enable NULL character filtering, NULL characters are not reported if not data. 1 = Disable NULL character filtering. NULL characters (0x60) are transferred to the SCI data buffer.	rw	0
4	ClockStop	Manual SCI Clock Stop control. Manually starts and stops the SCI clock. This bit must be set to '1' when Autostop mode is enabled. 0 = Enable the SCI clock 1 = Disable SCI clock	rw	1

Bit	Name	Function	Type	Default
5	AutoStop_En_H	Enables automatic clock shutdown when command is complete. Enabling this will generate the necessary startup and shutdown delays required by the SIM protocol. 0 = Auto clock control not enabled. SCI clock controlled by SCI_Clockstop bit 1 = Auto clock control enabled.	rw	0
6	MSBH_LSBL	Sets the transmission and reception bit order: 0 = LSB is sent/recieved first (Direct convention) 1 = MSB is sent/received first (Inverse convention)	rw	1
7	LLI	Logic Level Invert: 0 = Logic level 0 data is sent/received as '0' or 'A' which is the same as the start bit. (Direct convention) 1 = Logic level 0 data is sent/received as '1' or 'Z' which is the opposite of the start bit. (Inverse convention)	rw	1
8	PEGen_Len	Parity Error signal length. This configuration bit can be used to extend the duration of the parity error signal generation from 1 ETU to 1.5 ETU 0 = Parity Error signal duration is 1 ETU starting at 10.5 ETU 1 = Parity Error signal duration is 1.5 ETU starting at 10.5 ETU	rw	0
9	Parity_En	Enable or disable parity error checking on the receive data 0 = Disable parity error checking 1 = Enable parity error checking	rw	0
10	Stop_Level	Logical value of the clock signal when SCI clock is stopped (either due to automatic shutdown or manual shutdown) 0 = Stop clock at low level 1 = Stop clock at high level	rw	1
16	ARG_H	Automatic Reset Generator. Write a '1' to this bit to initiate an automatic reset procedure on the SIM. Write '0' to switch back to SCI_Reset control (bit 20). An ARG interrupt will be generated if the ARG process succeeded or failed. The ARG status bit (ARG_Det) must be read to determine if a reset response from the card was detected. This bit needs to be cleared between ARG attempts.	rw	0
17	AFD_En_H	Automatic format detection. This bit is generally set in conjunction with the ARG_H bit to enable automatic detection of the data convention. 1 = Enable TS detection and automatic convention settings programming 0 = disable automatic settings and use the register bits (MSBH_LSBL and LLI) to control the convention	rw	0
18	Tx_Resend_En_H	1 = Enable automatic resend of characters when Tx parity error is detected 0 = Disable automatic resend	rw	1
20	Reset	Direct connection to the SIM card reset pin. This is overridden when ARG_H is enabled 0 = SCI_Reset low voltage 1 = SCI Reset high voltage	rw	0
21	Dly_Sel	This selects between two delay times for the automatic clock stop startup and shutdown: 0 = short delay Startup/Shutdown : 744 SCI clocks / 1860 SCI clocks 1 = long delay Startup/Shutdown : (2 x 744) SCI clocks / (2 x 1860) SCI clocks	rw	0
29:24	Par_Chk_Offset	Allows fine control of the parity check position during the parity error time period.	rw	0xe

Bit	Name	Function	Type	Default
31:30	Reserved	These bits are reserved and must be written as '00' for the SCI module to work properly: "11" = Ser In &lt;- Ser Out loopback "10" = Ser In &lt;- Ser In (unmasked) others = Ser In &lt;- Ser In masked with Txing_H (normal mode)	rw	0

0x01A1_0004 Status

Bit	Name	Function	Type	Default
0	RxData_Rdy	Returns the status of the Rx FIFO: 0 = Rx FIFO empty 1 = There is at least 1 character in the Rx FIFO	r	
1	Tx_FIFO_Rdy	Returns the status of the Tx FIFO: 0 = Tx FIFO is full 1 = There is at least 1 free spot in the Tx FIFO	r	
2	Format_Det	Returns the status of the automatic format detection after reset: 0 = TS character has not been detected in the ATR 1 = TS character has been detected and SCI module is using the automatic convention settings This bit is cleared when the AFD_En bit is cleared	r	
3	ARG_Det	Returns the status of the automatic reset procedure: 0 = ARG detection has failed 1 = ARG detection has detected that the SIM has responded to the reset This bit is used in conjunction with the ARG interrupt. The ARG interrupt will be generated at the successful or unsuccessful termination of the ARG process. This bit can be used to determine the success or failure.	r	
4	Reset_Det	This is the status of the Reset pin when automatic reset generation is enabled. This bit can be used to discover whether the SIM card that has successfully responded to an ARG procedure has an active high or active low reset. (Det means 'Detection')	r	
5	Clk_Rdy_H	Status of the control signal to the clock control module. This bit respects the startup and shutdown phases, so during these times, the clock may actually be on, but it is not considered to be 'ready' 0 = SCI clock may be on or off but is not ready for use 1 = SCI clock is on and ready for use	r	
6	Clk_Off	Status bit of the Sci clock. 0 = Sci clock is ON 1 = Sci clock is OFF	r	
8	Rx_Err	A receive parity error was detected. Reading this register clears the bit.	r	
9	Tx_Err	A transmit parity error was detected. Reading this register clears the bit.	r	
10	RxOverflow	The internal receive FIFO has reached an overflow condition. Reading this register clears the bit.	r	
11	TxOverflow	The internal transmit FIFO has reached an overflow condition. Reading this register clears the bit.	r	
31:30	AutoStop_State	Returns the state of the clock management state machine when AutoStop mode is enabled. This value is '00' when manual mode is selected. 0 Startup_phase	r	

Bit	Name	Function	Type	Default
		Clock is on, but not ready to be used. 1 Auto_on Clock is on and ready to be used 2 Shutdown_phase Clock is still on, but should not be used. 3 Clock_off Clock is off.		

0x01A1_0008 Data

Bit	Name	Function	Type	Default
0	Data_IN	Writing to this register will send the data to the SIM card. If automatic clock shutdown is enabled, the appropriate delay will be applied before the data is actually sent.	w	
7:0	Data_OUT	Reading this register will read from the receive data FIFO.	r	

0x01A1_000c ClkDiv

Bit	Name	Function	Type	Default
8:0	ClkDiv	Clock divider for generating the baud clock from the SCI clock. This value must match the value used by the SIM card whose default value is 0x174.	rw	0x174
23:16	ClkDiv_16	Secondary clock divider for generating 16x baud clock.	rw	0x18
30:24	MainDiv	Main clock divider to generate the SCI clock. This value should be calculated as follows: MainDiv = Clk_Sys/(2xSCI_Clk) - 1 where SCI_Clk is in the range of 3-5 MHz as specified in the SIM specification.	rw	0x4
31	Clk_Inv	Inverts the polarity of the SCI clock to the SIM card. 0 = No inversion 1 = Invert external SCI clock	rw	0

0x01A1_0010 RxCnt

Bit	Name	Function	Type	Default
0	RxCnt	This value should be programmed with the number of expected characters to receive. It will be decremented each time a character is <strong>actually</strong> received and should be 0 when the transfer is complete. If a character is sent after the RxCnt reaches zero, the extra character flag will be set but this value will stay at zero.	rw	0
31	Clk_Persist	When in automatic clock shutdown mode, this bit can prevent the clock from entering shutdown mode when the transfer is complete. This should be used for multi-transfer commands where the clock must not be shut down until the command is complete. This bit must be programmed for each transfer. 1 = Keep clock on 0 = Allow clock shutdown when transfer is complete	rw	0

0x01A1_0014 Times

Bit	Name	Function	Type	Default
0	ChGuard	This is the extra guard time that can be added to the 2 ETU minimum (and default) guard time between successive transmitted characters. This should be programmed depending on the SIM's ATR. The total ETU guard time will be ChGuard + 1.	rw	1
11:8	TurnaroundGuard	Turnaround guard time configuration. This value can be used to adjust the delay between the leading edge of a received character and the leading edge of the next transmitted character. The minimum time specified in the SIM recommendation is 16 ETU. The number of ETUs can be calculated using the following formula: Total Turnaround Time (in ETUs) = 11 + TurnaroundGuard	rw	0x6
23:16	WI	Work Waiting Time factor. A timeout will be generated when the WWT is exceeded. The WWT is calculated by: WWT = 960 x WI x (F/Fi) where Fi is the main SCI clock frequency (3-5 MHz) and F is 372 before an enhanced PPS and 512 after an enhanced PPS. The SCI_WI value must be calculated as follows: SCI_WI = WI * D Thus, by default (WI = 10) this value needs to be set to 10 before an EPPS, but needs to be scaled to WI*D=80 after the EPPS procedure.	rw	0x0A
31:24	Tx_PERT	Number of times to try resending character when the SIM indicates a parity error.	rw	0xFF

0x01A1_0018 Ch_Filt

Bit	Name	Function	Type	Default
0	Ch_Filt	Value of the character to be filtered. 0x60 is the NULL character in the SIM protocol. If character filtering is enabled, the <strong>first</strong> 0x60 character that is received by the SIM during a transfer will <strong>not</strong> be recorded. The purpose of this character is to enable the SIM to reset the WWT counter when the SIM is not ready to send the data. This filter has no effect on characters within the datastream.	rw	0x60

0x01A1_0020 Int_Cause

This register is a **READ ONLY** register that returns the logical **and** of the SCI_INT_STATUS register and the SCI_INT_MASK. If any of these bits is '1', the SCI module will generate an interrupt. Bits 21:16 return the status of the interrupt which is the interrupt state before the mask is applied. These bits should only be used for debugging.

Bit	Name	Function	Type	Default
0	Rx_Done	Number of expected Rx characters, as programmed in the RxCnt register, has been received.	r	0
1	Rx_Half	Receiver FIFO is half full.	r	0
2	WWT_Timeout	No Tx character has been sent NOR any Rx character detected within the WWT timeout.	r	0
3	Extra_Rx	An extra character has been received after the number of characters in RxCnt has been received.	r	0
4	Resend_Ovfl	The automatic re-transmit of parity error characters has exceeded	r	0

Bit	Name	Function	Type	Default
		the threshold specified in the Tx_PERT field.		
5	ARG_End	End of the ARG sequence. The status register must be read to determine whether the ARG sequence was successful or not.	r	0

0x01A1_0024 Int_Clr

This is a WRITE ONLY register that is used to clear an SCI interrupt. Write a '1' to the interrupt that is to be cleared. Writing '0' has no effect.

Bit	Name	Function	Type	Default
0	Rx_Done	Number of expected Rx characters, as programmed in the SCI_RxCnt register, has been received.	c	0
1	Rx_Half	Receiver FIFO is half full.	c	0
2	WWT_Timeout	No Tx character has been sent NOR any Rx character detected within the WWT timeout.	c	0
3	Extra_Rx	An extra character has been received after the number of characters in SCI_RxCnt has been received.	c	0
4	Resend_Ovfl	The automatic re-transmit of parity error characters has exceeded the threshold specified in the SCI_Tx_PERT field.	c	0
5	ARG_End	End of the ARG sequence. The status register must be read to determine whether the ARG sequence was successful or not.	c	0

0x01A1_0028 Int_Mask

This register is READ/WRITE register that enables the desired interrupt. A '1' in a bit position indicates that the corresponding interrupt is enabled and if the interrupt occurs, the SCI will generate a hardware interrupt.

Bit	Name	Function	Type	Default
0	Rx_Done	Number of expected Rx characters, as programmed in the SCI_RxCnt register, has been received.	rw	0
1	Rx_Half	Receiver FIFO is half full.	rw	0
2	WWT_Timeout	No Tx character has been sent NOR any Rx character detected within the WWT timeout.	rw	0
3	Extra_Rx	An extra character has been received after the number of characters in SCI_RxCnt has been received.	rw	0
4	Resend_Ovfl	The automatic re-transmit of parity error characters has exceeded the threshold specified in the SCI_Tx_PERT field.	rw	0
5	ARG_End	End of the ARG sequence. The status register must be read to determine whether the ARG sequence was successful or not.	rw	0

0x01A1_10xx SIM2
0x01A1_20xx SIM3

4.7 RFIF

The RF modules (RFIF) is the control interface of the DigRF interface which is a standard digital interface between digital baseband and the integrated RF transceiver.

Features

- Rx and Tx analog front end interface
- DigRF interface with
  Stream mode and block mode supported for Tx
  16-bit per sample, 2 samples per symbol or 1 sample per symbol supported for Rx
- 2 words of 32 bits Tx FIFO
- 16 words of 32 bits Rx FIFO
- DMA operation supported on Rx and Tx

5. Multi-media Subsystem

5.1 LCDC

5.2 Camera

5.2.1 Overview

The Camera module is an interface between system and camera sensor. It supports 8-bit parallel interface, CSI-2 MIPI interface and SPI interface.

The controller only handles the data and synchronization signals of the sensor. The control is usually done through an I2C interface which is not integrated in the camera controller module.

Features

- On the fly cropping and decimation.
- Supports up to 2M pixel sensor.
- Supports any arbitrary size scaling down from 4096x4096
- latch Ram FIFO Size 160*16 bit
- Support data format: RGB565, YUV422 and compression data
- Support 8-bit parallel interface
- Support CSI-2 MIPI interface
- Support SPI interface with 1/2/4 data mode.

Data Formats

Following are some data format examples:

- YUV422 (YUYV)

Data	First pixel	First Pixel	Second Pixel	Second Pixel	Third Pixel	Third Pixel
Even/Odd	Y[7:0]	U[7:0]	Y[7:0]	V[7:0]	Y[7:0]	U[7:0]

- YUV422 (YVYU)

Data	First pixel	First Pixel	Second Pixel	Second Pixel	Third Pixel	Third Pixel
Even/Odd	Y[7:0]	V[7:0]	Y[7:0]	U[7:0]	Y[7:0]	V[7:0]

- YUV422 (UYVY)

Data	First pixel	First Pixel	Second Pixel	Second Pixel	Third Pixel	Third Pixel
Even/Odd	U[7:0]	Y[7:0]	V[7:0]	Y[7:0]	U[7:0]	Y[7:0]

- YUV422 (VYUY)

Data	First pixel	First Pixel	Second Pixel	Second Pixel	Third Pixel	Third Pixel
Even/Odd	V[7:0]	Y[7:0]	U[7:0]	Y[7:0]	V[7:0]	Y[7:0]

- RGB565

Bytes	D7	D6	D5	D4	D3	D2	D1	D0
even	R7	R6	R5	R4	R3	G7	G6	G5
odd	G4	G3	G2	B7	B6	B5	B4	B3

Cropping & Decimation

Input data from CMOS sensors are first cropped with specified start and end point, which may then followed by a decimation. Decimation can be done in both horizontal and vertical directions. For horizontal decimation, it depends on whether data format is RGB or YUV.

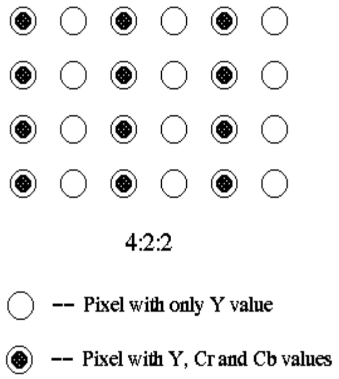

4:2:2
○ -- Pixel with only Y value
◉ -- Pixel with Y, Cr and Cb values

Figure 5.1: YUV 4:2:2 Subsampling

If data format is RGB565, horizontal decimation is done at pixel level. For example, if 1:2 rate is chosen:

Src Data: $R_0G_0B_0$ $R_1G_1B_1$ $R_2G_2B_2$ $R_3G_3B_3$ $R_4G_4B_4$ $R_5G_5B_5$

Dst Data: $R_0G_0B_0$ $R_2G_2B_2$ $R_4G_4B_4$

If data format is YUV422, horizontal decimation is should regroup some pixels. For example, if 1:2 rate is chosen:

Src Data: $Y_0U_0Y_1V_0$ $Y_2U_1Y_3V_1$ $Y_4U_2Y_5V_2$ $Y_6U_3Y_7V_3$ $Y_8U_4Y_9V_4$ $Y_{10}U_5Y_{11}V_5$

Dst Data: $Y_0U_0Y_2V_0$ $Y_4U_2Y_6V_2$ $Y_8U_4Y_{10}V_4$

Another example, if YUV422 1:3 rate is chosen:

Src Data: $Y_0U_0Y_1V_0$ $Y_2U_1Y_3V_1$ $Y_4U_2Y_5V_2$ $Y_6U_3Y_7V_3$ $Y_8U_4Y_9V_4$ $Y_{10}U_5Y_{11}V_5$

Dst Data: $Y_0U_0Y_3V_0$ $Y_6U_3Y_9V_3$

For Vertical decimation, data format RGB565/555/444 or YUV422 will be the same, vertical decimation is done at line level. For example, if 1:2 rate is chosen:

Src Data: $Line_0$ $Line_1$ $Line_2$ $Line_3$ $Line_4$ $Line_5$

Dst Data: $Line_0$ $Line_2$ $Line_4$

FIFO

Rx Data FIFO is in APB domain. So FIFO_Write signal from CAM_PIXCLK domain will be synchronized to APB domain. This requires APB clock frequency faster than CAM_PIXCLK frequency * 0.75. For example, if RGB565 QXGA 15 frame/s is required, APB CLK freq should > 3.1M pixel * 2 byte per pixel * 15 frame per sec * 0.75 = 70 MHz.

Interface

The Camera module provides two types of interface for camera sensor. One is the traditional 8-bit parallel interface, the other is SPI interface which uses less signal. The two interfaces are share the same pin so that it does not increase the total amounts of chip pins.

The typical connect of 1 data series interface is as following, here the GC6113 and PA6167/6175 are chose for examples.

Parallel Interface	SPI interface	GC6113 (1 data mode)	PAS6167/6175
CAM_D0			
CAM_D1	SPI_CAM_CTS		CTS
CAM_D2			
CAM_D3			
CAM_D4	SPI_CAM_OF		OVERFLOW_M
CAM_D5	SPI_CAM_RD		OVERFLOW_S
CAM_D6	SPI_CAM_SSN		SSN
CAM_D7	SPI_CAM_SCK	SCK	SCLK

CAM_PCLK	SPI_CAM_DI	SDO0	SDO
CAM_VSYNC			
CAM_HREF			
CAM_CLK	CAM_CLK	MCLK	SYS_CLK
CAM_RST	I2C_SCL	SBCL	I2C_SCL
CAM_PDN	I2C_SDA	SBDA	I2C_SDA

5.2.2 Block Diagram

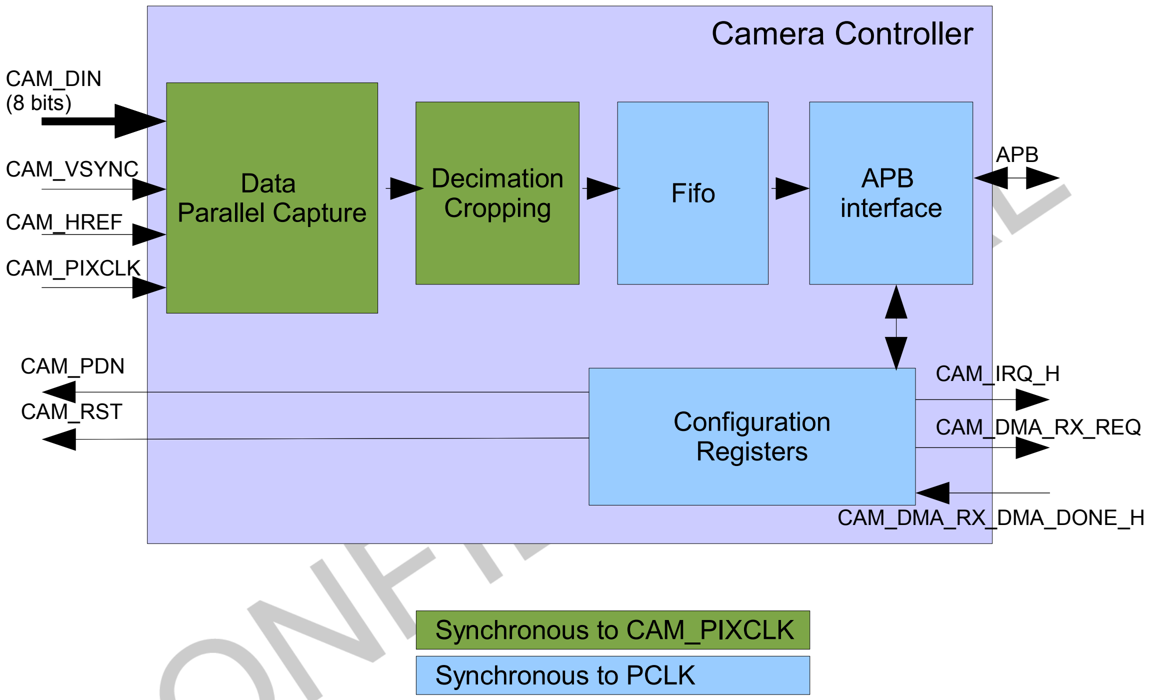


Figure 5.2: Camera Controller

5.2.3 Control Registers

Camera Register Definitions

0x2000_0000 CTRL

Bit	Name	Function	Type	Default
0	Enable	Enable camera controller,high active. 1 ENABLE 0 DISABLE	rw	0
1	DctEnable	Enable Dct module,high active. 1 DctENABLE 0 DctDISABLE	rw	0
2	BufEnable	Enable pre-cal-average two neighbor pix,high active. 1 BufENABLE 0 BufDISABLE	rw	0
5:4	DataFormat	"0" = RGB565. "1" = YUV422. "2" = Compressed Data. "3" = Reserved. 0 RGB565 1 YUV422 2 JPEG 3 RESERVE	rw	0
8	RESET_Pol	'0' = keep output camera reset polarity. '1' = invert output camera reset polarity. 1 INVERT 0 NORMAL	rw	1
9	PWDN_Pol	'0' = keep output camera power down polarity. '1' = invert output camera power down polarity. 1 INVERT 0 NORMAL	rw	0
10	VSYNC_Pol1	'0' = keep input VSYNC polarity. '1' = invert input VSYNC polarity. 1 INVERT 0 NORMAL	rw	0
11	HREF_Pol	'0' = keep input HREF polarity so data is sampled when HREF high. '1' = invert input HREF polarity so data is sampled when HREF low. 1 INVERT 0 NORMAL	rw	0
12	PIXCLK_Pol	'0' = keep pix clk polarity. '1' = invert pix clk polarity. 1 INVERT 0 NORMAL	rw	0
14	VSYNC_Drop	'0' = VSYNC irq always exists when Frame decimation is enabled. '1' = VSYNC irq will drop when Frame data are dropped in decipation.	rw	1

Bit	Name	Function	Type	Default
		1 DROP 0 NORMAL		
17:16	DecimFrm	"0"= All frame data will be sent. "1"= only one frame out of two (1/2) will be sent. "2"= only one frame out of three (1/3) will be sent. "3"= only one frame out of four (1/4) will be sent. 0 ORIGINAL 1 DIV_2 2 DIV_3 3 DIV_4	rw	0
19:18	DecimCol	"0"= Pixel Decimation Disabled. "1"= Pixel Decimation 1/2. "2"= Pixel Decimation 1/3. "3"= Pixel Decimation 1/4. 0 ORIGINAL 1 DIV_2 2 DIV_3 3 DIV_4	rw	0
21:20	DecimRow	"0"= line Decimation Disabled. "1"= line Decimation 1/2. "2"= line Decimation 1/3. "3"= line Decimation 1/4. 0 ORIGINAL 1 DIV_2 2 DIV_3 3 DIV_4	rw	0
26:24	Reorder	Controls the Re-ordering of the FIFO data. In following table, for input data, right comes before left. So YUYV means V comes first. for output data, right data is the LSB. So YUYV means V is stored in low 8-bit (byte0) of 32-bit word. If Bit 26 is '1', byte2 and byte0 is Y. If Bit 25 is '1', both byte2/byte3 and byte1/byte0 interchange. If Bit 24 is '1', byte U and V should interchange. (UV bytes can be decided using bit 26). input YUYV, output YUYV: "000" input YVYU, output YUYV: "001" input UYVY, output YUYV: "110" input VYUY, output YUYV: "111" input YUYV, output UYVY: "010" input YVYU, output UYVY: "011" input UYVY, output UYVY: "100" input VYUY, output UYVY: "101" input YUYV, output YVYU: "001" input YVYU, output YVYU: "000" input UYVY, output YVYU: "111" input VYUY, output YVYU: "110" input YUYV, output VYUY: "011" input YVYU, output VYUY: "010" input UYVY, output VYUY: "101" input VYUY, output VYUY: "100" Decimation will reorder data flow also. Input UYVY becomes YUVY	rw	0

Bit	Name	Function	Type	Default
		after decimation. This reorder is corrected using Bit 26 infomation.		
28	CropEn	"0"= Cropping Disabled. "1"= Cropping Enabled. Note: this bit should set to '0' when bit field "DataFormat" is "10" (compressed data) 1 ENABLE 0 DISABLE	rw	0
30	Bist Mode	In Bist Mode, FIFO RAM are read and write by its address, FIFO mode is disabled. 1 BIST 0 NORMAL	rw	0
31	TEST	Debug only. A RGB565 test card is sent to system bus instead of real data from sensor. 1 TEST 0 NORMAL	rw	0

0x2000_0004 STATUS

Bit	Name	Function	Type	Default
0	OVFL	'1' = FIFO over-write IRQ status. Write to corresponding bit in IRQ CLEAR register will clear this bit.	ro	0
1	VSYNC_R	'1' = VSYNC rising edge IRQ status Write to corresponding bit in IRQ CLEAR register will clear this bit.	ro	0
2	VSYNC_F	'1' = VSYNC falling edge IRQ status Write to corresponding bit in IRQ CLEAR register will clear this bit.	ro	0
3	DMA DONE	'1' = DMA Done IRQ status Write to corresponding bit in IRQ CLEAR register will clear this bit.	ro	0
4	FIFO EMPTY	'1' = FIFO Empty status, not clear-able.	ro	1

0x2000_0008 DATA

Bit	Name	Function	Type	Default
0	RX_DATA	Read in the receive FIFO	r	

0x2000_000C IRQ MASK

Bit	Name	Function	Type	Default
0	OVFL	'1' = FIFO over-write enable	rw	0
1	VSYNC_R	'1' = VSYNC rising edge enable	rw	0
2	VSYNC_F	'1' = VSYNC falling edge enable	rw	0
3	DMA DONE	'1' = DMA Done enable	rw	0

0x2000_000C IRQ MASK

Bit	Name	Function	Type	Default
0	OVFL	'1' = FIFO over-write enableWrite '1' to clear FIFO over-write interrupt	wo	0
1	VSYNC_R	Write '1' to clear VSYNC rising edge interrupt	wo	0
2	VSYNC_F	Write '1' to clear VSYNC falling edge interrupt	wo	0
3	DMA DONE	Write '1' to clear DMA Done interrupt	wo	0

0x2000_0014 IRQ CAUSE

Bit	Name	Function	Type	Default
0	OVFL	'1' = FIFO over-write cause	ro	0
1	VSYNC_R	'1' = VSYNC rising edge cause	ro	0
2	VSYNC_F	'1' = VSYNC falling edge cause	ro	0
3	DMA DONE	'1' = DMA Done cause	ro	0

0x2000_0018 CMD SET

Bit	Name	Function	Type	Default
0	PWDN	Power down pin of CMOS sensor .	rs	1
4	RESET	Reset pin of CMOS sensor. Active Low.	rs	1
8	FIFO RESET	For the software to clear FIFO. This bit is auto-reset to 0.	s	0

0x2000_001C CMD CLR

Bit	Name	Function	Type	Default
0	PWDN	Power down pin of CMOS sensor .	rc	1
4	RESET	Reset pin of CMOS sensor.	rc	1

0x2000_0020 DSTWINCOL

Bit	Name	Function	Type	Default
0	DstWinColStart	start pixel of cropped window.	wr	0
27:16	DstWinColEnd	end pixel of cropped window.	wr	0

0x2000_0024 DSTWINROW

Bit	Name	Function	Type	Default
0	DstWinRowStart	start line of cropped window.	wr	0
27:16	DstWinRowEnd	end line of cropped window.	wr	0

0x2000_0028 SCALE_CONFIG

Bit	Name	Function	Type	Default
0	SCALE_EN	Enable camera scaler,high active. 1 SCALEENABLE 0 SCALEDISABLE	wr	0
9:8	SCALE_COL	scaler col. 0 ORIGINAL 1 DIV_2 2 NOUSE 3 DIV_4	wr	0
17:16	SCALE_ROW	scaler row. 0 ORIGINAL 1 DIV_2 2 NOUSE 3 DIV_4	rw	0

0x2000_002C SPI_CAMERA_REG0

Bit	Name	Function	Type	Default
0	SPI_CAMERA_REGISTER0	spi camera config resiger0	wr	0

0x2000_0030 SPI_CAMERA_REG1

Bit	Name	Function	Type	Default
0	SPI_CAMERA_REGISTER1	spi camera config resiger1	wr	0

0x2000_0034 SPI_CAMERA_REG2

Bit	Name	Function	Type	Default
0	SPI_CAMERA_REGISTER2	spi camera config resiger2	wr	0

0x2000_0038 SPI_CAMERA_REG3

Bit	Name	Function	Type	Default
0	SPI_CAMERA_REGISTER3	spi camera config resiger3	wr	0

0x2000_003C SPI_CAMERA_REG4

Bit	Name	Function	Type	Default
0	SPI_CAMERA_REGISTER4	spi camera config resiger4	wr	0

0x2000_0040 SPI_CAMERA_REG5

Bit	Name	Function	Type	Default
0	SPI_CAMERA_REGISTER5	spi camera config resiger5	wr	0

0x2000_0044 SPI_CAMERA_REG6

Bit	Name	Function	Type	Default
0	SPI_CAMERA_REGISTER6	spi camera config resiger6	wr	0

0x2000_0048 SPI_CAMERA_OBS0

Bit	Name	Function	Type	Default
0	SPI_CAMERA_OBS_REGISTER0	spi camera obs resiger0	wr	0

0x2000_004C SPI_CAMERA_OBS1

Bit	Name	Function	Type	Default
0	SPI_CAMERA_OBS_REGISTER1	spi camera obs resiger1	wr	0

5.3 GOUDA

5.3.1 Overview

Features

The GOUDA module has two functions: 2D graphic engine and LCD controller. It handles various graphic operations used to compose and display images on LCD.

These operations : colour space conversion, resizing, landscape/portrait rotation and layer blending and overlay are very CPU-cycle consuming, and also especially bandwidth hungry as for every single operation, one or several images have be loaded, processed and temporary rewritten into memory before being actually sent to the LCD display.

GOUDA module relieve the CPU by both handling the pixel processing itself and also by using the minimum memory bandwidth as it only loads once needed data and outputs the final result directly.

GOUDA can handle 4 video layers and output the resulting image composition directly to the LCD interface. The graphic subsystem have the following capabilities:

- 4 distinct video sources can be blended to produce final output :
  - Layer 0 (aka video layer) :
    - Input formats : YUV4:2:0 planar (IYUV, YV12), YUV4:2:2 pixel-packed (UYVY, YUY2), RGB565
    - YUV to RGB565 conversion (CCIR601 component video standard definition)
    - Image resizing (separate H and V ratios, from 1/4th to 4x each direction, 8bit fractional step)
    - adjustable layer depth
    - per-layer alpha blending
    - chroma-key with simple range handling (bit-masking based)
  - Layer 1-3 (aka overlay layers):
    - Input formats : RGB565, ARGB8888
    - fixed depth
    - per-layer alpha blending
    - chroma-key with simple range handling (bit-masking based)
    - per-pixel alpha blending (for ARGB)
- Region Of Interest (ROI) handling for partial refresh of the LCD screen.
- Background colour (colour filled where no layer is defined)
- Embedded LCD controller for direct output (LCD commands buffer in dedicated SRAM).

The GOUDA module provides three sets of interface for supporting various LCD interfaces.
parallel LCD interface; RGB565/666/888 LCD interface; DSI MIPI LCD interface.

5.3.2 Control Registers

GOUDA Register Definitions

0x2004_0000 gd_command

Bit	Name	Function	Type	Default
0	start	Starts the image transfer. Autoreset	rw	0

0x2004_0004 gd_status

Bit	Name	Function	Type	Default
0	ia_busy	High while image accelerator is busy	r	0
4	lcd_busy	High while LCD controller is busy	r	0

0x2004_0008 gd_eof_irq

Bit	Name	Function	Type	Default
0	eof_cause	High when End Of Frame IRQ has been generated. To clear it, write 1 in this bit or in eof_status.	rc	0
16	eof_status	Unmasked version of eof_cause. To clear it, write 1 in this bit or In eof_status.	rc	0

0x2004_000C gd_eof_irq_mask

Bit	Name	Function	Type	Default
0	eof_mask	EOF interrupt generation mask: 0: EOF IRQ disabled 1: EOF IRQ enabled	rw	0

0x2004_0010 gd_roi_tl_ppos

Bit	Name	Function	Type	Default
9:0	x0	LCD Region Of Interest Top-Left pixel x-axis	rw	0
25:16	y0	LCD Region Of Interest Top-Left pixel y-axis	rw	0

0x2004_0014 gd_roi_br_ppos

Bit	Name	Function	Type	Default
9:0	x1	LCD Region Of Interest Bottom-Right pixel x-axis	rw	0
25:16	y1	LCD Region Of Interest Bottom-Right pixel y-axis	rw	0

0x2004_0018 gd_roi_bg_color

Bit	Name	Function	Type	Default
4:0	b	Blue component of the ROI background color		0
10:5	g	Green component of the ROI background color		0
15:11	r	Red component of the ROI background color		0

0x2004_001C gd_vl_input_fmt

Bit	Name	Function	Type	Default
1:0	format	Input image format 00b: RGB565 pixel packed 01b: YUV4:2:2 pixel packed (UYVY) 10b: YUV4:2:2 pixel packed (YUYV) 11b: YUV4:2:0 planar (IYUV)	rw	0
14:2	stride	Image stride in bytes (of Y component for planar formats). This is the length from the beginning of a line to the beginning of the next line (can be different from image width * pixel size).	rw	0
31	active	Defines Layer's activity: 0: Layer disabled 1: Layer active	rw	0

0x2004_0020 gd_vl_tl_ppos

Bit	Name	Function	Type	Default
9:0	x0	Video Layer (layer 0) Top-Left pixel x-axis position	rw	0
25:16	y0	Video Layer (layer 0) Top-Left pixel y-axis position	rw	0

0x2004_0024 gd_vl_br_ppos

Bit	Name	Function	Type	Default
9:0	x1	Video Layer (layer 0) Bottom-Right pixel x-axis position	rw	0

Bit	Name	Function	Type	Default
25:16	y1	Video Layer (layer 0) Bottom-Right pixel y-axis position	rw	0

0x2004_0028 gd_vl_extents

Bit	Name	Function	Type	Default
9:0	max_line	Number of lines of source image (idem gd_vl_br_ppos.y1 when vertical scaling factor is one).	rw	0
25:16	max_col	Number of columns of source image (idem gd_vl_br_ppos.x1 when vertical scaling factor is one).	rw	0

0x2004_002C gd_vl_blend_opt

Bit	Name	Function	Type	Default
4:0	chroma key b	Blue component of the Chroma Key	rw	0
10:5	chroma key g	Green component of the Chroma Key	rw	0
15:11	chroma key r	Red component of the Chroma Key	rw	0
16	chroma key enable	Enables the Chroma Keying	rw	0
19:17	chroma key mask	Allows a range of color for the Chroma Keying: 000b: exact color match 001b: disregard 1 LSBit of each color component for matching 011b: disregard 2 LSBit of each color component for matching 111b: disregard 3 LSBit of each color component for matching	rw	0
27:20	alpha	Layer Alpha blending coefficient	rw	0
29:28	rotation	Layer rotation selection 00b: No rotation 01b: 90 degrees rotation (clockwise) 10b: reserved 11b: reserved	rw	0
31:30	depth	Layer depth 00b: Video layer behind all Overlay layers 01b: Video layer between Overlay layers 1 and 0 10b: Video layer between Overlay layers 2 and 1 11b: Video layer on top of all Overlay layers	rw	0

0x2004_0030 gd_vl_y_src

Bit	Name	Function	Type	Default
25:2	addr	Dword-aligned address of the Y component (or RGB) of the source image	rw	0

0x2004_0034 gd_vl_u_src

Bit	Name	Function	Type	Default
25:2	addr	Dword-aligned address of the U component of the source image	rw	0

0x2004_0038 gd_vl_v_src

Bit	Name	Function	Type	Default
25:2	addr	Dword-aligned address of the V component of the source image	rw	0

0x2004_003C gd_vl_resc_ratio

Bit	Name	Function	Type	Default
10:0	xPitch	Video layer rescaling ratio upon x-axis. This is a 3.8 fixed point number representing the input/output width ratio.	rw	0
26:16	yPitch	Video layer rescaling ratio upon y-axis. This is a 3.8 fixed point number representing the input/output height ratio.	rw	0
30	DctEnable	Enable Dct module,high active. 1 DctENABLE 0 DctDISABLE	rw	0
31	Yinterpolen	Enable row Interpolation,high active. 1 row_Interpolation_ENABLE 0 row_Interpolation_DISABLE	rw	0

0x2004_0040 Overlay_Layer[0]
0x2004_0054 Overlay_Layer[1]
0x2004_0068 Overlay_Layer[2]
0x2004_0040 gd_ol_input_fmt

Bit	Name	Function	Type	Default
1:0	format	Input image format 0: RGB565 pixel packed 1: ARGB8888 pixel packed others: reserved	rw	0
14:2	stride	Image stride in 16-bits word. >This is the length from the beginning of a line to the beginning of the next line (can be different from image width * pixel size).	rw	0
31	active	Defines Layer's activity: 0: Layer disabled 1: Layer active	rw	0

0x2004_0044 gd_ol_tl_ppos

Bit	Name	Function	Type	Default
9:0	x0	Overlay Layer (layer X+1) Top-Left pixel x-axis position	rw	0
25:16	y0	Overlay Layer (layer X+1) Top-Left pixel y-axis position	rw	0

0x2004_0048 gd_ol_br_ppos

Bit	Name	Function	Type	Default
9:0	x0	Overlay Layer (layer X+1) Bottom-Right pixel x-axis position	rw	0
25:16	y0	Overlay Layer (layer X+1) Bottom-Right pixel y-axis position	rw	0

0x2004_004C gd_ol_blend_opt

Bit	Name	Function	Type	Default
4:0	chroma key b	Blue component of the Chroma Key	rw	0
10:5	chroma key g	Green component of the Chroma Key	rw	0
15:11	chroma key r	Red component of the Chroma Key	rw	0

Bit	Name	Function	Type	Default
16	chroma key enable	Enables the Chroma Keying	rw	0
19:17	chroma key mask	Allows a range of color for the Chroma Keying: 000b: exact color match 001b: disregard 1 LSBit of each color component for matching 011b: disregard 2 LSBit of each color component for matching 111b: disregard 3 LSBit of each color component for matching	rw	0
27:20	alpha	Layer Alpha blending coefficient	rw	0

0x2004_0050 gd_ol_rgb_src

Bit	Name	Function	Type	Default
25:2	addr	Dword-aligned address of the source image	rw	0

0x2004_007C gd_lcd_ctrl

Bit	Name	Function	Type	Default
1:0	Destination	Destination Selection 0 LCD CS 0 1 LCD CS 1 2 Memory LCD 3 Memory RAM	rw	0
6:4	Output Format	Output format 000b: 8-bit - RGB3:3:2 - 1cycle/1pixel - RRRGGGBB 001b: 8-bit - RGB4:4:4 - 3cycle/2pixel - RRRRGGGG/BBBBRRRR/GGGGBBBB 010b: 8-bit - RGB5:6:5 - 2cycle/1pixel - RRRRRGGG/GGGBBBBB 011b: reserved 100b: 16-bit - RGB3:3:2 - 1cycle/2pixel - RRRGGGBBRRRGGGBB 101b: 16-bit - RGB4:4:4 - 1cycle/1pixel - XXXXRRRRGGGGBBBB 110b: 16-bit - RGB5:6:5 - 1cycle/1pixel - RRRRRGGGGGGBBBBB 111b: reserved The MSB select also the AHB access size (8-bit or 16-bit) when Memory destination is selected. Must set to RGB565 when RAM type destination selected 0 8_bit_RGB332 1 8_bit_RGB444 2 8_bit_RGB565 4 16_bit_RGB332 5 16_bit_RGB444 6 16_bit_RGB565	rw	0
7	High byte	Exchange the high byte and low byte of lcd output data 0: no change 1: change	rw	0
8	CS0 Polarity	Change Polarity of CS0 signal 0: no change 1: Inverted	rw	0
9	CS1 Polarity	Change Polarity of CS1 signal 0: no change 1: Inverted	rw	0
10	RS Polarity	Change Polarity of RS signal	rw	0

Bit	Name	Function	Type	Default
		0: no change 1: Inverted		
11	WR Polarity	Change Polarity of WR signal 0: no change 1: Inverted	rw	0
12	RD Polarity	Change Polarity of RD signa 0: no change 1: Inverted	rw	0
20:16	Nb Command	Number of command to be send to the LCD command (up to 31)	rw	0
24	Start command	Start command transfer only. Autoreset	w	0

0x2004_0080 gd_lcd_timing

All value are in cycle number of system clock

Bit	Name	Function	Type	Default
2:0	TAS	Address setup time (RS to WR, RS to RD)	rw	0
6:4	TAH	Adress hold time	rw	0
13:8	PWL	Pulse Width Low level, between 2 and 63.	rw	0
21:16	PWH	Pulse Width High level, between 2 and 63 (must be > (TAH+TAS)).	rw	0

0x2004_0084 gd_lcd_mem_address

Bit	Name	Function	Type	Default
25:2	addr_dst	Address destination pointer when memory destination is selected. The addr_dst[1] which correspond to the M_A[0] on the memory interface is used to select between command/data.	rw	all0

0x2004_0088 gd_lcd_stride_offset

Bit	Name	Function	Type	Default
9:0	stride_offset	Address offset (in Bytes) skipped at the end of each line when memory destination is selected. This 2D feature allows for in-memory image compositing.	rw	all0

0x2004_008C gd_lcd_single_access

Bit	Name	Function	Type	Default
15:0	lcd_data	data to write or data readen (the readen data is ready when the lcd is not busy)	rw	all0
16	type	Acesss type selection 0: Command 1: Data	rw	0
17	start_write	Start a single write access. Autoreset	w	0
18	start_read	Start a single read access (only when LCD output selected). Autoreset.	w	0

0x2004_0090 gd_spilcd_config

Bit	Name	Function	Type	Default
0	spi_lcd_select	spi lcd enable 0 normal lcd 1 spi lcd	rw	0
6:1	spi_device_id	spi lcd device id	rw	all0
14:7	spi_clk_divider	spi lcd clk divider		all0
17:15	spi_dummy_cycle	spi lcd device id	rw	all0
19:18	spi_line	spi line 0 4-line spi 1 3-line spi 2 4-line spi 3 no use	rw	all0
22:20	spi_rx_byte	spi lcd desired rx data	rw	all0
23	spi_rw	spi lcd read/write 0 spi write 1 spi read	rw	0

0x2004_0094 gd_spilcd_rd

Bit	Name	Function	Type	Default
31:0	spi_lcd_readback	spi lcd readback data	r	all0

0x2004_0098 gd_vl_fix_ratio

Bit	Name	Function	Type	Default
7:0	xratio	gouda col scale ratio	rw	all0
15:8	yratio	gouda row scale ratio	rw	all0
16	xfixen	fix gouda col scale ratio	rw	0
17	yfixen	fix gouda row scale ratio	rw	0

5.4 GPU

5.4.1 Overview

The GPU module is 3D graphics engine. RDA8810 uses Vivante's GC860 GPU IP.

Today’s smartphone and tablet feature rich, graphical user interfaces and run interactive applications like games and mobile web tools. Vivante GC860 graphics processing unit (GPU) IP defines a high-performance cores that deliver hardware acceleration for 2D and 3D graphics displays on these devices.

GC860 supports following graphics APIs:
□ OpenGL ES 2.0
□ OpenGL ES 1.1
□ OpenVG 1.1
□ DirectFB

□ GDI/DirectDraw

5.5 VPU

5.5.1 Overview

The VPU module is video decoder/encoder engine.

The VPU is a full HD multi-standard video IP for smartphone and tablet. It can decode compressed video in a format of H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-1/2, MPEG-4 SP/ASP, H.263P3, DivX/XviD, VP8, MVC, Theora, AVS, RV-8/9/10, and JPEG up to Full-HD 1920x1088 resolution. It can also perform H.264, MPEG-4, and H.263 encoding up to Full-HD 1920x1088 (max. 8192x8192 JPEG) resolution. The VPU can perform simultaneous multiple realtime encoding, decoding, or both encoding and decoding of different format video streams at multiple resolutions.

The VPU contains a 16-bit DSP called BIT processor. The BIT processor communicates with Cortex-A5 through a APB interface and controls the other sub-blocks of the VPU. Cortex-A5 requires low resources under 1 MIPS, because all of the functions such as bitstream parsing, video hardware sub- blocks control and error resilience are implemented in the BIT processor. Moreover it is designed to optimally share most of the sub-blocks that are used in common for video processing, which contributes to the ultra low power and low gate count.

It is connected with Cortex-A5 via 32-bit AMBA 3 APB bus for system control and 64-bit AMBA3 AXI for data. There are two 64-bit AXI buses: primary and secondary. The secondary bus is connected to internal SRAM to achieve high performance.
?

6. Analog Modules

6.1 SPI Interface for Analog IP control

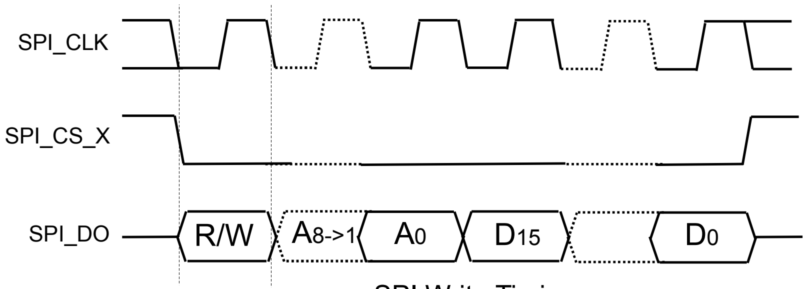
All the IPs connected to DBB feature an SPI interface dedicated to IP control.

The Power Management Unit(PMU) and the Analog Baseband(ABB) are controlled by the same SPI of DBB (with different chip-select). On the other hand the RF Transceiver has it's dedicated SPI as part of DigRF interface.

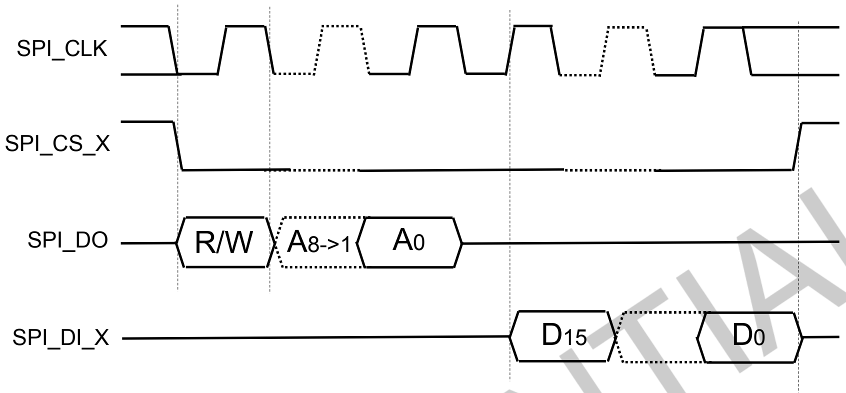
For simplicity sake and ease of use, the SPI protocol settings for PMU and ABB.

The chosen SPI setting is the following:

- Data from DBB are issued on falling edge of clock and can be latched on rising edge by the slave.
- Data from the slave are issued on rising edge of the clock and will be latched on falling edge of clock.
- the SPI frame is 26 bits long : first bit is the R/W selection, then 9 bits of address A_8 to A_0, then 16 bits of data D_{15} to D_0.
- Maximum SPI_CLK frequency : 26 MHz



SPI Write Timing



SPI Read Timing

Figure 6.1: SPI Write & Read Timing

6.2 Power Management Unit

The PMU is controlled from the DBB through SPI

6.2.1 Power System Management

The PMU performs a POR either when POWKEY is asserted (i.e. Power-ON/OFF button pressed) or when then WAKEUP line is raised (RTC Alarm) or when the charger (or USB) is plugged.

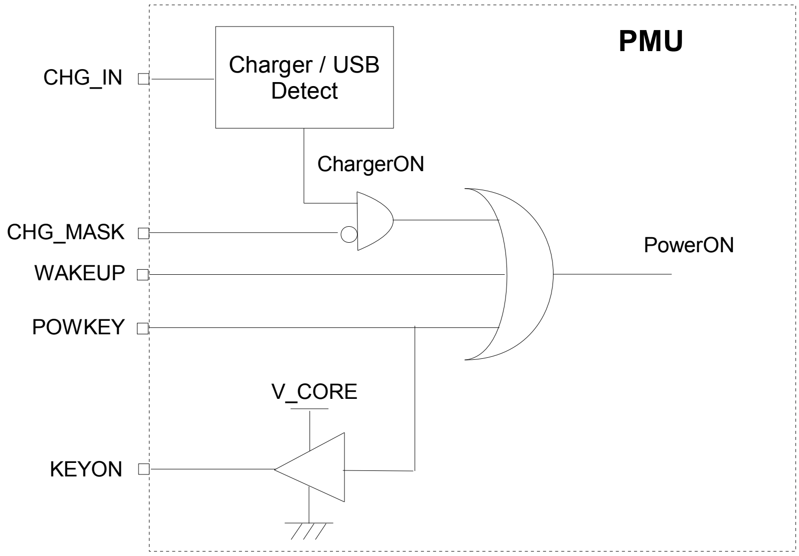


Figure 6.2: PMU Power ON

In the presented principle schematic, PowerON is an internal signal of the PMU that reflects that the system needs power. It can either be raised when the Power-ON key is pressed, when the WAKEUP line from DBB is raised or when the charger (or USB) is plugged. When a rising edge of PowerON is detected, the PMU should go through its POR sequence.

The POR sequence consists in turning ON the relevant power supplies then wait for RST delay that is be long enough to ensure that all LDOs are in steady state and that all the chip is correctly powered, then raise the RESETB signal.

In case the POR is trigged by POWKEY, It's DBB responsibility to set the WAKEUP line high when it considers that POWKEY has been pressed long enough. As POWKEY is a VBAT level input, a V_CORE version of POWKEY (named KEYON) is generated for DBB.

POR can also be initiated when the WAKEUP line is being raised directly because of an RTC Alarm (the POWKEY is not pressed).

As long as PowerON is high, the PMU provides voltages supplies that are specified by the current Power-Profile that describes the LDO and DC/DC settings.

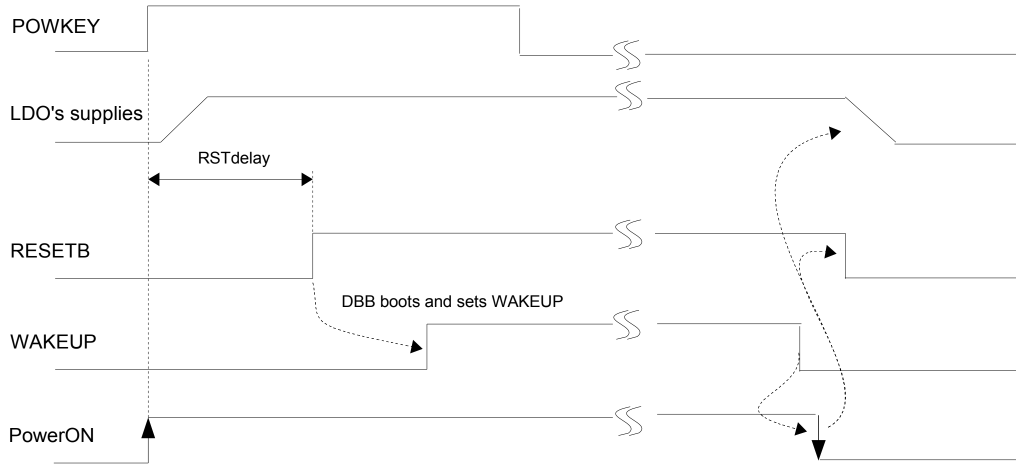


Figure 6.3: POR triggered by POWKEY press

6.2.2

The PMU implements multiple power profiles defining the LDOs and DC/DC activation in various modes.

- Default_powerOFF_profile: used when the system is OFF(system has been shut-off or first time battery is plugged...). In this case only V_RTC is provided.
- Default_powerON_profile: used when PowerON is asserted and the POR is trigged, all LDOs that have “reset state ON” are activated.
- Active_power_profile: used once system has booted and decided to switch from Default_powerON_profile. This profile is programmable.
- LP_power_profile: used when the system goes to low-power mode (switch from Active to Low-power through LP_MODE line). This profile is programmable.

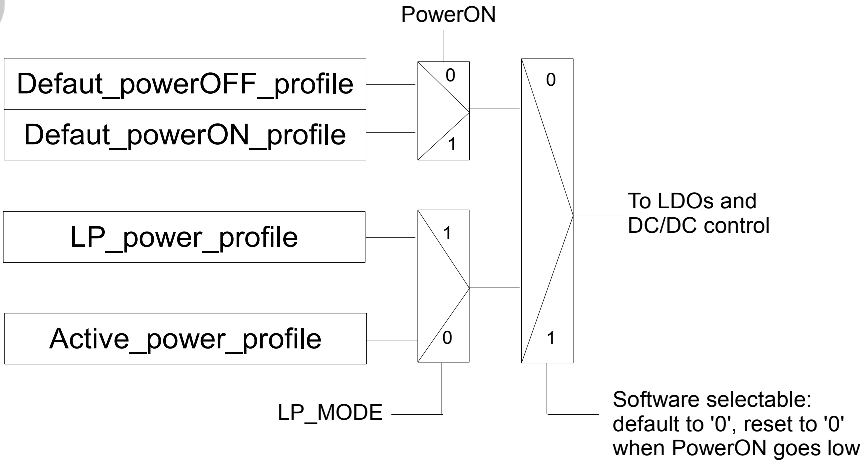


Figure 6.4: Principle schematic for Power-Profiles usage

If the PowerON line goes low, the RESETB signal will be set to Low level then all power supplies excepted V_RTC are switched OFF.

It can be useful to be able to go back to Default_powerOFF_profile (i.e. shut-off all supplies except RTC) even when a charger is plugged. The CHG_MASK (RTC domain) line is used for this purpose, by masking the ChargerON line.

The RF Transceiver IP integrate it's own LDOs and are powered directly with VBAT.

V_USB is not power-on automatically when USB is detected. The USB PHY remains OFF until the software decides to turn it ON.

6.2.3 LDOs

Name	Type	Voltage	Max Current	Usage (power domain)
V_CORE	DC-DC	1.2V	300mA	Core
V_CORE	LDO	1.2V	20mA	Core
V_BUCK_2V4	DC-DC		300mA	Internal transceiver
AVDD_2V4	LDO		200mA	Internal transceiver
V_BOOST	DC-DC		600mA	LCD backlight, VBAT_ABB, or Audio PA
V_PAD	LDO	1.8V / 2.8V	200mA	Digital IO
V_ANA	LDO		200mA	Analog module
V_MEM	LDO	1.8V / 2.8V	200mA	External Combo flash / ram
V_SPIMEM	LDO	1.8V / 2.8V	200mA	External SPI Memory
V_ASW	LDO	1.8V / 2.8V	200mA	Supplies periphery device
V_CAM	LDO	1.8V / 2.8V	200mA	Camera
V_LCD	LDO	1.8V / 2.8V	200mA	LCD module
V_MMC	LDO	2.8V / 3.2V	150mA	Memory Card
V_VIB	LDO	2.3V / 3.3V	300mA	Vibrator
V_USB	LDO	3.3V	300mA	USB PHY
V_MIC	LDO	1.4V/1.75V	200mA	Microphone
V_RTC	LDO	1.2V	0.6mA	Real time clock
V_SIM (all)	LDO	1.8V / 3.0V	80mA	SIM

6.2.4 LED Drivers

The PMU implements both LCD backlight and Keypad LED drivers.

There are 8 current sink sources provided for the LCD backlight use, which named as LED1-8.. Each of them can sink 30mA.

The Keypad LED drivers include 3 sink sources as KP_LEDR, KP_LEDG, and KP_LEDB. Each of them can provide 30mA sink current. BL_LED_OUT is voltage feedback current regulator which is suggested to connect to the anode of the keypad LEDs.

Both LCD backlight and Keypad LED drivers can be adjusted driver capability by software via changing current or PWM settings.

6.2.5 Charger Circuit

The RDA8810PL integrates the most of charger circuits, except for a few components such as one PMOS, one accurate resistor and one diode, which should be applied externally.

When the charger is on, this block controls the charging phase and turns on the appropriate LDOs according to the battery status. If the voltage is greater than 4.2~ 4.4 V, charging is stopped immediately to prevent permanent damage to the battery.

Battery charging states include No Charge mode, Constant Current (CC) charge mode (pre-charge, constant current), and Constant Voltage (CV) charge mode. No matter what state the phone is in, the PMIC charger handles the charging state transition. A typical charging process is described as what the Figure 5 shows.

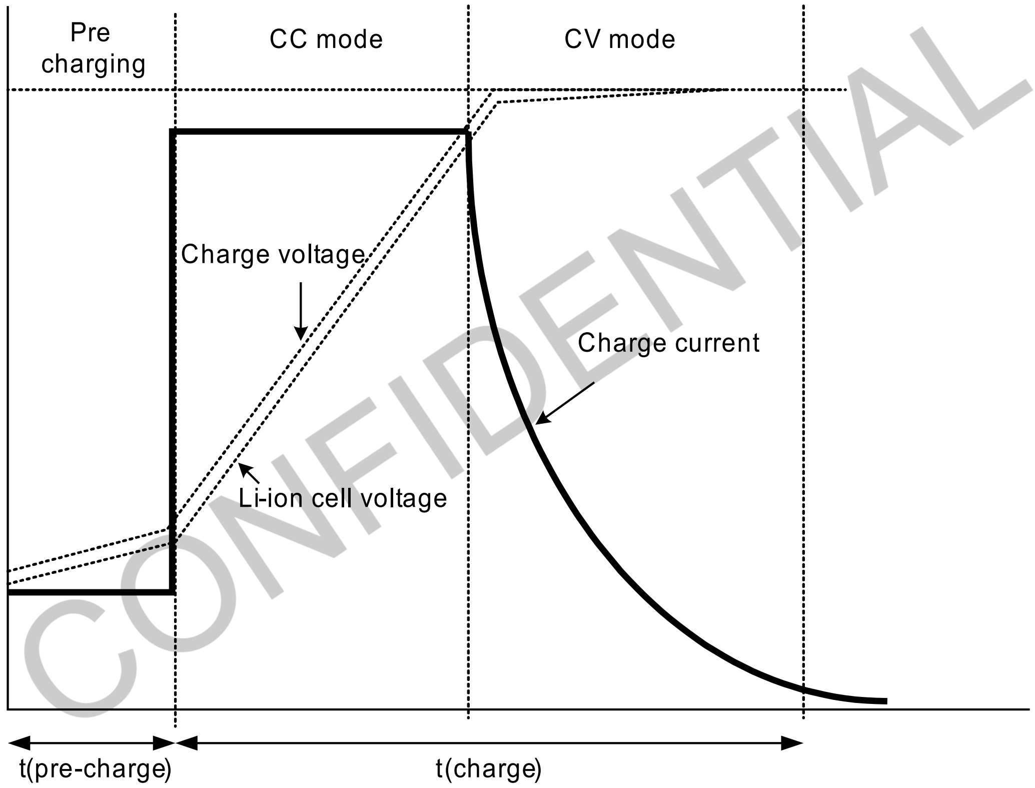


Figure 6.5: Charging I-V Curve

The charge current is calculated by the following expression,

I_charge = I_ref * (200/0.2)

I_ref is 0.05mA in pre_charge mode and 0.5mA in constant mode, and these two mode can be switched by internal charger circuit.

6.3 SIM Interface

This interface can provide power supply, clock, reset and data I/O lines to the SIM cards. While all of SIM cards can be powered simultaneously, only one of them can be accessed at a time by DBB.

The SIM cards interface can be either 1.8V or 3V hence there are independent internal LDOs respectively for each.

All of SIM cards can be enabled at the same time but only one can be selected at a time.

Due to special requirements of some SIM cards, SIM channel(s) are able to keep their CLK and RST lines to either high or low level when not selected.

6.4 Resistor Touchscreen Interface

The resistor touchscreen provide a mode where X and Y measures (and associated switching) are performed automatically, the results being stored in dedicated registers.

The measure will be trigged by an SPI command sent by DBB. The touchscreen interface will then:

- switch the X+, X-, Y+ and Y- lines to perform the X measurement,
- wait for a programmable time (accounting for settling time to the final steady-state value prior to the ADC sampling),
- then perform the X measure and store it (and also set the End Of Conversion for X).
- switch the X+, X-, Y+ and Y- lines to perform the Y measurement
- wait for a programmable time (same time as for X)
- then perform the Y measure and store it (and also set the End Of Conversion for Y)
- generate the End of Measurement Interrupt, EOMIRQ to DBB (see 6.5).

When the touchscreen interface is in IDLE mode (when it is not performing measurements), it will be able to detect that the touchscreen panel is pressed through the X+, X-, Y+ and Y- lines. The PENIRQ line reflects the detection result. When the touchscreen interface is performing measurements the PENIRQ line is not expected to be valid and can be set to '0'.

6.5 Interrupt management

The PMU has two lines of interruption to DBB:

- the PENIRQ line dedicated to touchscreen
- the PMU_INT line that handles several other causes of interruption.

6.6 Analog BaseBand (ABB)

6.6.1 PLL and clock squarer

ABB includes PLLs that generates digital clocks from a 26MHz input. It also includes a “clock squarer” that buffers the clock coming from the internal transceiver to provide a clean version (at VCORE levels) of this clock.

In order for the system to be synchronized to the GSM network, the 26 MHz must be the same as the one used by the transceiver.

The following drawing examples the PLL clock path:

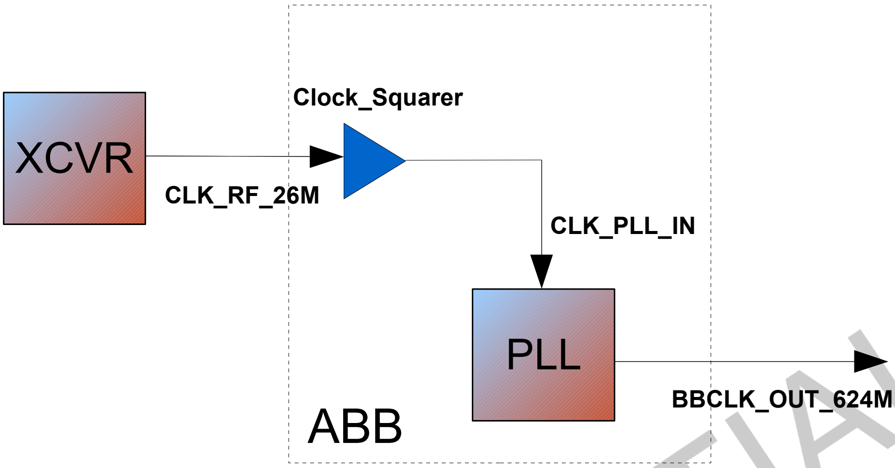
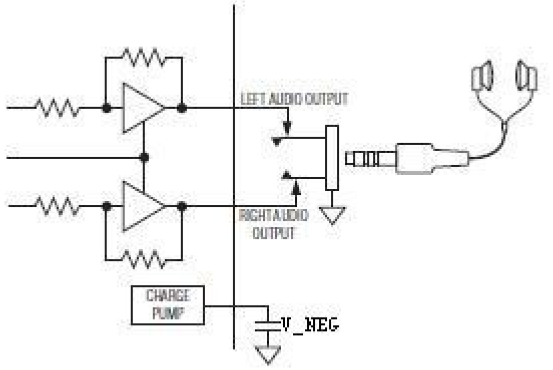


Figure 6.6: PLL Clock Path

6.6.2 AUDIO

The Audio part includes ADC, DAC, Audio Amplifiers and Audio Muxes. The following diagram provides a global view of audio blocks. Voice data can be input from microphone. Voice and audio data can be output to a stereo DAC and connected to peripherals such as receiver, headset and loudspeaker. It is able to support multiple sample rate audio data including 8 kHz, 11.025 kHz, 12kHz, 16 kHz, 22,05 kHz, 24kHz, 32kHz, 44,1 kHz, 48kHz.



6.7 RF Transceiver

6.7.1 General Description

RDA8810PL Integrates a complete RF front end for GSM/GPRS/EDGE wireless communication. The RF receiver employs a digital low-IF architecture, and supports DigRF 1.12 digital interface. The receive section interfaces between the RF band-select SAW filter and the digital baseband. The transmit section provides a direct modulation PLL transmitter path from the baseband subsystem to the power amplifier (PA). A fast settling fractional-N synthesizer is fully integrated, including RF VCO, loop filters, and varactors, etc. The RF transceiver includes a digitally-controlled crystal (DCXO) and completely integrates the reference oscillator and frequency tuning varactors.

6.7.2 Features

- Integrated transceiver including the following:
  - Digital low-IF receiver
  - Direct modulation transmitter
  - Frequency synthesizer
  - Integrated VCO, loop filters, etc.
- Support digital AFC
- DigRF digital interface for communication between digital baseband and transceiver
- RF FEM control

6.7.3 Block Description

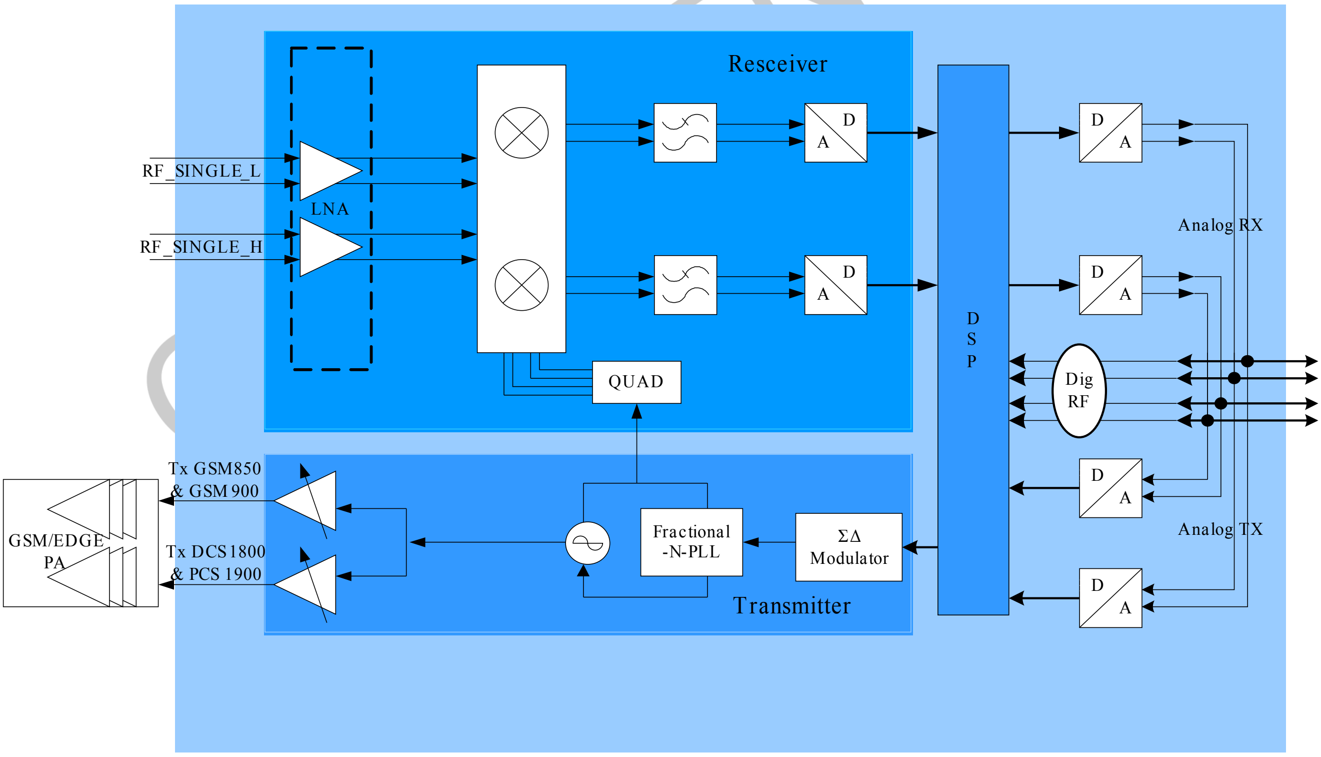


Figure 6.7: RF transceiver block digram

Receive Path

The receive path uses a digital low-IF architecture that avoids the difficulties associated with direct conversion while delivering lower solution cost and reduced complexity. It consists of two sections; the first section is the RF front-end, which integrates four differential input LNAs and a qudarature image-reject mixer, and supports GSM850, GSM900, DCS1800 and DCS1900 band. The RF front-end amplifies the signal from antenna, and down-converts it to a low IF of 100kHz or 200KHz (software configurable). The second section is the IF section, which downmixer the wanted signal to baseband and further amplifies the chosen channel and performs gain control to adjust the output level to the desired value. The IF gain range can vary over more than 100 dB.

Transmit Path

The transmit path consists of a direct modulation block, an SDM PLL and pre-amplifier block. Firstly the baseband signal is sampled to digital domain by ADC. Then SDM PLL direct modulation the signal to RF band and filters out the out-of-band noise to meet the GSM standard. The following block, pre-amplifier, further powers the transmit signal and send it to PA.

Synthesizer Section

The Local Oscillator (LO) signals are provided by an on-chip Voltage Controlled Oscillator (VCO) for operation of the RX and TX section. The frequency of the VCO is set by internal PLL circuits, which are programmable via the SPI bus.

7. Memory Map

AP Part Address Map:

Module name	Base address	End address	Description	Size	Note
SPIFLASH	F000_0000h	FFFF_FFFFh	SPINOR/SPINAND	1GB	
	E000_0000h	EFFF_FFFFh			
	D000_0000h	DFFF_FFFFh			
	C000_0000h	CFFF_FFFFh			
DRAMC	B000_0000h	BFFF_FFFFh	DDR SDRAM	1GB	
	A000_0000h	AFFF_FFFFh			
	9000_0000h	9FFF_FFFFh			
	8000_0000h	8FFF_FFFFh			
	7000_0000h	7FFF_FFFFh	(reserved)	256MB	
	6000_0000h	6FFF_FFFFh	(reserved)	256MB	
	5000_0000h	5FFF_FFFFh	(reserved)	256MB	
	4000_0000h	4FFF_FFFFh	(reserved)	256MB	
	3000_0000h	3FFF_FFFFh	(reserved)	256MB	

	2120_0000h	2FFF_FFFFh	(reserved)	238MB	
	2110_1000h	211F_FFFFh	(reserved)	1020kB	
L2CC	2110_0000h	2110_0FFFh	L2CC configuration regfile	4kB	
Connect	2100_0000h	210F_FFFFh		1MB	GPV
	20B0_0000h	20FF_FFFFh	(reserved)	5MB	
APB2 Peripherals	20A0_0000h	20AF_FFFFh		1MB	64kB*16
APB1 Peripherals	2090_0000h	209F_FFFFh		1MB	64kB*16
APB0 Peripherals	2080_0000h	208F_FFFFh		1MB	64kB*16
	2054_0000h	207F_FFFFh	(reserved in AHB1)	2M+ 768kB	
(reserved)	2050_0000h	2053_FFFFh		256kB	AHB1 Slave5
(reserved)	204C_0000h	204F_FFFFh		256kB	AHB1 Slave4
(reserved)	2048_0000h	204B_FFFFh		256kB	AHB1 Slave3
SPIFLASH	2044_0000h	2047_FFFFh		256kB	AHB1 Slave2
USBC	2040_0000h	2043_FFFFh		256kB	AHB1 Slave1
	2014_0000h	203F_FFFFh	(reserved in AHB0)	2M + 768kB	
(reserved)	2010_0000h	2013_FFFFh		256kB	AHB0 Slave5
VOC	200C_0000h	200F_FFFFh		256kB	AHB0 Slave4
GPU	2008_0000h	200B_FFFFh		256kB	AHB0 Slave3
GOUDA	2004_0000h	2007_FFFFh		256kB	AHB0 Slave2
CAMERA	2000_0000h	2003_FFFFh		256kB	AHB0 Slave1
Modem	1000_0000h	1FFF_FFFFh	Map to 0000_0000h~03FF_FFFFh in MD side.	256MB	
	0030_0000h	0FFF_FFFFh	(reserved)	253MB	
Internal Memory	0020_0000h	002F_FFFFh	DPSRAM mailbox Between MD and AP	1MB	
	0010_0000h	001F_FFFFh	Internal SRAM	1MB	
	0000_0000h	000F_FFFFh	BootROM	1MB	

APB0 Bus Address Map:

Module name	Base address	End address	Description	Size	Note
APB0 Peripherals:					
	2086_0000h	208F_FFFFh	(reserved)	640kB	
CAMERA (inner DMA)	2085_0000h	2085_FFFFh	CAMERA DMA configuration	64kB	Slave5
GOUDA	2084_0000h	2084_FFFFh	GOUDA configuration	64kB	Slave4
VPU	2083_0000h	2083_FFFFh	VPU ctrl/status Interface	64kB	Slave3
DMA	2082_0000h	2082_FFFFh	DMA configuration.	64kB	Slave2
IMEM	2081_0000h	2081_FFFFh	IMEM configuration.	64kB	Slave1
IRQ	2080_0000h	2080_FFFFh	Interrupt ctrl/status access	64kB	Slave0

IRQ:

```
constant AP_IRQ_UART3:          integer := 1;
constant AP_IRQ_NFSC:           integer := 2;
constant AP_IRQ_SDMMC1:         integer := 3;
constant AP_IRQ_SDMMC2:         integer := 4;
constant AP_IRQ_SDMMC3:         integer := 5;
constant AP_IRQ_SPI1:           integer := 6;
constant AP_IRQ_SPI2:           integer := 7;
constant AP_IRQ_SPI3:           integer := 8;
constant AP_IRQ_UART1:          integer := 9;
constant AP_IRQ_UART2:          integer := 10;
constant AP_IRQ_I2C1:           integer := 11;
constant AP_IRQ_I2C2:           integer := 12;
constant AP_IRQ_I2C3:           integer := 13;
constant AP_IRQ_GPIO:           integer := 14;
constant AP_IRQ_KEYPAD:         integer := 15;
constant AP_IRQ_TIMERS:             integer := 16;
constant AP_IRQ_OSTIMER:        integer := 17;
constant AP_IRQ_COM0:           integer := 18;
constant AP_IRQ_COM1:           integer := 19;
constant AP_IRQ_USBC:           integer := 20;
constant AP_IRQ_DMC:            integer := 21;
constant AP_IRQ_DMA:            integer := 22;
constant AP_IRQ_CAMERA:         integer := 23;
constant AP_IRQ_GOUDA:          integer := 24;
constant AP_IRQ_GPU:            integer := 25;
constant AP_IRQ_JPG_VPU:            integer := 26;
constant AP_IRQ_HOST_VPU:         integer := 27;
constant AP_IRQ_VOC:            integer := 28;
constant AP_IRQ_AUIFC0:         integer := 29;
constant AP_IRQ_AUIFC1:         integer := 30;
constant AP_IRQ_L2CC:           integer := 31;
```

APB1 Bus Address Map:

Module name	Base address	End address	Description	Size	Note
APB1 Peripherals:					
AUIFC	209F_0000h	209F_FFFFh	AUIFC configuration.	64kB	Slave15
AIF	209E_0000h	209E_FFFFh	Audio configuration and data access.	64kB	Slave14
(reserved)	209D_0000h	209D_FFFFh		64kB	Slave13
(reserved)	209C_0000h	209C_FFFFh		64kB	Slave12
Debug APB	209B_0000h	209B_FFFFh	System port to access debug apb bus.	64kB	Slave11
DDR_PHY	209A_0000h	209A_FFFFh	DDR_PHY configuration.	64kB	Slave10
DMC400	2099_0000h	2099_FFFFh	DMC400 configuration.	64kB	Slave9
COMREGS	2098_0000h	2098_FFFFh	COMREGS between MD and AP.	64kB	Slave8
I2C3	2097_0000h	2097_FFFFh	I2C master.	64kB	Slave7
I2C2	2096_0000h	2096_FFFFh	I2C master.	64kB	Slave6
I2C1	2095_0000h	2095_FFFFh	I2C master.	64kB	Slave5
PWM	2094_0000h	2094_FFFFh	Pulse width modulator.	64kB	Slave4
GPIO	2093_0000h	2093_FFFFh	GPIO.	64kB	Slave3
KEYPAD	2092_0000h	2092_FFFFh	Keypad.	64kB	Slave2
TIMER	2091_0000h	2091_FFFFh	OS timer, general purpose timer.	64kB	Slave1
SYS_CTRL	2090_0000h	2090_FFFFh	System control.	64kB	Slave0

DebugAPB Bus Address Map (CPU View):

Module name	Base address	End address	Description	Size	Note
Debug APB bus: (Map to 0000_0000h~0000_FFFFh in DAP side)					
	209B_4000h	209B_FFFFh	(reserved)	48kB	
A5 PMU	209B_3000h	209B_3FFFh	A5 core performance monitor unit	4kB	Slave3
A5 Debug	209B_2000h	209B_2FFFh	A5 core debug unit	4kB	Slave2
A5 DebugROM	209B_1000h	209B_1FFFh	A5 integration rom table	4kB	Slave1
DAPROM	209B_0000h	209B_0FFFh	Coresight debug rom table	4kB	Slave0

APB1 IFC:

```
constant AP_APB1_AUIFC_ID_RX_AIF:              integer := 0;
constant AP_APB1_AUIFC_ID_TX_AIF:              integer := 1;
```

APB2 Bus Address Map:

Module name	Base address	End address	Description	Size	Note
APB2 Peripherals:					
IFC	20AF_0000h	20AF_FFFFh	IFC configuration.	64kB	Slave15
(reserved)	20AE_0000h	20AE_FFFFh		64kB	Slave14
(reserved)	20AD_0000h	20AD_FFFFh		64kB	Slave13
(reserved)	20AC_0000h	20AC_FFFFh		64kB	Slave12
(reserved)	20AB_0000h	20AB_FFFFh		64kB	Slave11
(reserved)	20AA_0000h	20AA_FFFFh		64kB	Slave10
UART3	20A9_0000h	20A9_FFFFh	UART configuration and data access.	64kB	Slave9
NAND_FSC	20A8_0000h	20A8_FFFFh	NAND flash controller configuration and data access.	64kB	Slave8
SDMMC3	20A7_0000h	20A7_FFFFh	SDMMC configuration and data access.	64kB	Slave7
SDMMC2	20A6_0000h	20A6_FFFFh	SDMMC configuration and data access.	64kB	Slave6
SDMMC1	20A5_0000h	20A5_FFFFh	SDMMC configuration and data access.	64kB	Slave5
SPI3	20A4_0000h	20A4_FFFFh	SPI configuration and data access.	64kB	Slave4
SPI2	20A3_0000h	20A3_FFFFh	SPI configuration and data access.	64kB	Slave3
SPI1	20A2_0000h	20A2_FFFFh	SPI configuration and data access.	64kB	Slave2
UART2	20A1_0000h	20A1_FFFFh	UART configuration and data access.	64kB	Slave1
UART1	20A0_0000h	20A0_FFFFh	UART configuration and data access.	64kB	Slave0

APB2 IFC:

```
constant AP_APB2_IFC_ID_TX_UART1:                    integer := 0;
constant AP_APB2_IFC_ID_RX_UART1:                    integer := 1;
constant AP_APB2_IFC_ID_TX_UART2:                    integer := 2;
constant AP_APB2_IFC_ID_RX_UART2:                    integer := 3;
constant AP_APB2_IFC_ID_TX_SPI1:                         integer := 4;
constant AP_APB2_IFC_ID_RX_SPI1:                   integer := 5;
constant AP_APB2_IFC_ID_TX_SPI2:                   integer := 6;
constant AP_APB2_IFC_ID_RX_SPI2:                   integer := 7;
constant AP_APB2_IFC_ID_TX_SPI3:                   integer := 8;
constant AP_APB2_IFC_ID_RX_SPI3:                   integer := 9;
constant AP_APB2_IFC_ID_TX_SDMMC1:                 integer := 10;
constant AP_APB2_IFC_ID_RX_SDMMC1:                 integer := 11;
constant AP_APB2_IFC_ID_TX_SDMMC2:                 integer := 12;
constant AP_APB2_IFC_ID_RX_SDMMC2:                 integer := 13;
constant AP_APB2_IFC_ID_TX_SDMMC3:                 integer := 14;
constant AP_APB2_IFC_ID_RX_SDMMC3:                 integer := 15;
constant AP_APB2_IFC_ID_TX_NFSC:                   integer := 16;
constant AP_APB2_IFC_ID_RX_NFSC:                   integer := 17;
constant AP_APB2_IFC_ID_TX_UART3:                  integer := 18;
constant AP_APB2_IFC_ID_RX_UART3:                  integer := 19;
```

Modem Part Address Map:

Memory name	description	address	size
LPDDR	LPDDR	0x02800000 ~ 0x03FFFFFF	24MB
External CS0	PSRAM	0x02000000 ~ 0x027FFFFF	8MB
Xcpu rom	Xcpu/Bcpu rom	0x01e00000 ~ 0x01ffffff	2MB (128KB only)
Internal sram	Internal sram	0x01c00000 ~ 0x01DFFFFF	2MB (128KB only)
FIFO_regs	Internal register	0x018c0000 ~ 0x018fffff	256K
SYS APB	Sys apb slave	0x01a00000 ~ 0x01a7ffff	512K
BB SRAM	BB internal sram	0x01980000 ~ 0x0199ffff	128K (32KB only)
BB APB	BB apb slave	0x01900000 ~ 0x0193FFFF	256K
Mailbox AP	Dual port SRAM	0x018A0000 ~ 0x018BFFFF	128K (8KB only)
AP APB	AP apb ahb slave	0x00000000 ~ 0x0189FFFF	24.625MB

XCPU APB			
Sys_ctrl		0x01A00000	
Irq		0x01A01000	
Timer		0x01A02000	
Ebc		0x01A03000	
Dma		0x01A04000	
Ifc		0x01A05000	
Calendar		0x01A06000	
Comregs		0x01A07000	
Gpio		0x01A08000	
cfg	iomux config	0x01A09000	
Comregs_AP		0x01A0a000	
Page_spy		0x01A0b000	
Rf_spi		0x01A0c000	
Tcu		0x01A0d000	
Sys_ahb_mon		0x01A0e000	
Bb_ahb_mon		0x01A0f000	
Sci1		0x01A10000	
Sci2		0x01A11000	
Sci3		0x01A12000	
Spi1	TP	0x01A13000	
Spi2	pmu/abb	0x01A14000	
Debug_uart		0x01A15000	
Xcpu_reg		0x01A16000	
Xcpu_tag		0x01A17000	
Xcpu_idata		0x01A18000	
Xcpu_ddata		0x01A19000	
Debug_host		0x01A1f000	

BCPU APB			
BB_Xcor		0x01900000	
BB_ifc		0x01901000	

BB_irq		0x01902000	
BB_itlv		0x01903000	
BB_viterbi		0x01904000	
BB_A5		0x01905000	
BB_RF		0x01906000	
BB_cp2		0x01907000	
BB_bist		0x01908000	
BB_bcpu_reg		0x01909000	
BB_bcpu_tag		0x0190a000	
BB_bcpu_idata		0x0190b000	
BB_bcpu_ddata		0x0190c000	
BB_comregs		0x0190d000	
EXCOR		0x0190e000	
EVITAC		0x0190f000	
CORDIC		0x01910000	

BootMode:

Bit	Reg Bit	PIN	Function
15	19	PLUGIN	1: Force download
14	18	VOLUME_UP	&keyon: force download
13	17	VOLUME_DN	&keyon: drive test&keyon long: reset all
12	16	KEYON	
11	15	internal	1: Force download
10	14	internal	1: Drive test
9	13	GPIO_A_7	1: Disable mem self check
8	12	GPIO_A_6	1: JTAG enable
7	11	GPIO_A_5	1: 4K SLC / 8K MLC; 0: 2K SLC / 4K MLC/SPI Flash
6	10	GPIO_A_4	1: Nandflash HIGH 8-bit from CAM; 0: from LCD
5	9	GPIO_C_5	1: Spiflash from Nandflash pin; 0: Spiflash from CAM pin
4	8	GPIO_C_4	1: Nand Spiflash; 0: Nor Spiflash/emmc/Nandflash
3	7	GPIO_C_3	1: Boot from Spiflash; 0: Boot from Nand/emmc
2	6	GPIO_C_2	1: Boot eMMC/Spiflash; 0: Boot from Nandflash
1	5	GPIO_C_1	1: 8-bit Nandflash; 0: 16-bit Nandflash
0	4	GPIO_B_0	1: MLC; 0: SLC

	GPIO_B0	GPIO_C1	GPIO_C2	GPIO_C3	GPIO_C4	GPIO_C5	GPIO_A4	GPIO_A5
SPI Nor Flash from NF IO	x	x	1	1	0	1	x	0
SPI Nand Flash from NF IO	x	x	1	1	1	1	x	0
SLC 2K Nand 8-bit	0	1	0	0	0	x	x	0
SLC 2K Nand 16-bit from LCD	0	0	0	0	0	x	0	0
SLC 2K Nand 16-bit from CAM	0	0	0	0	0	x	1	0
SLC 4K Nand 8-bit	0	1	0	0	0	x	x	1

SLC 4K Nand 16-bit from LCD	0	0	0	0	0	x	0	1
SLC 4K Nand 16-bit from CAM	0	0	0	0	0	x	1	1
MLC 4K Nand 8-bit	1	1	0	0	0	x	x	0
MLC 4K Nand 16-bit from LCD	1	0	0	0	0	x	0	0
MLC 4K Nand 16-bit from CAM	1	0	0	0	0	x	1	0
MLC 8K Nand 8-bit	1	1	0	0	0	x	x	1
MLC 8K Nand 16-bit from LCD	1	0	0	0	0	x	0	1
MLC 8K Nand 16-bit from CAM	1	0	0	0	0	x	1	1
eMMC	x	x	1	0	0	x	x	x
T-Card (1)	x	x	x	0	1	x	x	x
T-Card (2)	x	x	x	1	x	x	x	1

8. Pins Description

8.1 Pin-out

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
LCD interface										
LCD_DATA_0	B	LCD	I	-	LCD_DATA_0	LCD_DATA_0	DSI_CLKP			B11
LCD_DATA_1	B	LCD	I	-	LCD_DATA_1	LCD_DATA_1	DSI_CLKN			B14
LCD_DATA_2	B	LCD	I	-	LCD_DATA_2	LCD_DATA_2	DSI_D0_P			F8
LCD_DATA_3	B	LCD	I	-	LCD_DATA_3	LCD_DATA_3	DSI_D0_N			F9
LCD_DATA_4	B	LCD	I	-	LCD_DATA_4	LCD_DATA_4	DSI_D1_P			E9
LCD_DATA_5	B	LCD	I	-	LCD_DATA_5	LCD_DATA_5	DSI_D1_N			E10
LCD_DATA_6	B	LCD	I	-	GPIO_A18	LCD_DATA_6	DSI_D2_P			D11
LCD_DATA_7	B	LCD	I	-	GPIO_A19	LCD_DATA_7	DSI_D2_N			D12
LCD_WR	B	LCD	I	-	GPIO_A20	LCD_WR	DSI_D3_P	lcd_rgb_de		B10
LCD_RS	B	LCD	I	-	GPIO_A21	LCD_RS	DSI_D3_N	lcd_rgb_hsync		C10
LCD_RD	B	LCD	I	-	GPIO_A22	LCD_RD	SPI_LCD_SDC	lcd_rgb_vsync		B13
LCD_FMARK	B	LCD	I	-	GPIO_A23	LCD_FMARK	SPI_LCD_CLK	lcd_rgb_clk		C9

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
LCD_CS_0	B	LCD	O	1	LCD CS_0	LCD CS_0	SPI_LCD_DIO	GPO_4		B12
LCD_CS_1	O	LCD	O	1	LCD CS_1	LCD CS_1	SPI_LCD_CS	GPO_3		D8
LCD_DATA_8	B	LCD/NF	I	-	GPIO_A24	LCD_DATA_8			NFD_8	E7
LCD_DATA_9	B	LCD/NF	I	-	GPIO_A25	LCD_DATA_9			NFD_9	B8
LCD_DATA_10	B	LCD/NF	I	-	GPIO_A26	LCD_DATA_10			NFD_10	A8
LCD_DATA_11	B	LCD/NF	I	-	GPIO_A27	LCD_DATA_11			NFD_11	A9
LCD_DATA_12	B	LCD/NF	I	-	GPIO_A28	LCD_DATA_12			NFD_12	A10
LCD_DATA_13	B	LCD/NF	I	-	GPIO_A29	LCD_DATA_13			NFD_13	A11
LCD_DATA_14	B	LCD/NF	I	-	GPIO_A30	LCD_DATA_14			NFD_14	B9
LCD_DATA_15	B	LCD/NF	I	-	GPIO_A31	LCD_DATA_15			NFD_15	E8
Nand Flash interface										
NFD_0	B	NF	I	-	NFD_0	NFD_0	SDAT3_0	LCD_DATA_16		A7
NFD_1	B	NF	I	-	NFD_1	NFD_1	SDAT3_1	LCD_DATA_17		D6
NFD_2	B	NF	I	-	NFD_2	NFD_2	SDAT3_2	LCD_DATA_18		A6
NFD_3	B	NF	I	-	NFD_3	NFD_3	SDAT3_3	LCD_DATA_19		J6
NFD_4	B	NF	I	-	NFD_4	NFD_4	SDAT3_4	LCD_DATA_20		A5
NFD_5	B	NF	I	-	NFD_5	NFD_5	SDAT3_5	LCD_DATA_21		E6
NFD_6	B	NF	I	-	NFD_6	NFD_6	SDAT3_6	LCD_DATA_22		F6
NFD_7	B	NF	I	-	NFD_7	NFD_7	SDAT3_7	LCD_DATA_23		E5
NFCEN	O	NF	O	1	NFCEN	NFCEN	SSD3_CLK			G6
NFALE	O	NF	O	0	NFALE	NFALE	SSD3_CMD			D5
NFCLE	B	NF	O	0	NFCLE	NFCLE	GPIO_B25	M_SPI_CLK		A4
NFWEN	B	NF	O	1	NFWEN	NFWEN	GPIO_B26	M_SPI_D3		B6
NFWPN	B	NF	O	1	NFWPN	NFWPN	GPIO_B27	M_SPI_D2		C4
NFREN	B	NF	O	1	NFREN	NFREN	GPIO_B28	M_SPI_D1		B5
NFRB	B	NF	I	-	NFRB	NFRB	GPIO_B29	M_SPI_D0		B4
NFDQS	B	NF	I	-	NFDQS	NFDQS	GPIO_D4 INT	M_SPI_CS_0		B7
Camera interface										
CAM_RST	B	CAM	I	-	GPIO_B10	CAM_RST	I2C2_SCL		NFD_14/LCD_DATA_22	AA20
CAM_PDN	B	CAM	I	-	GPIO_B11	CAM_PDN	I2C2_SDA		NFD_15LCD_DATA_23	T18
CAM_CLK	B	CAM	I	-	GPIO_B12	CAM_CLK				Y18
CAM_VSYNC	B	CAM	I	-	GPIO_B13	CAM_VSYNC	CSI1_CLK_P	M_SPI_CLK	NFD_13/LCD_DATA_21	V18
CAM_HREF	B	CAM	I	-	GPIO_B14	CAM_HREF	CSI2_CLK_P			U18
CAM_PCLK	B	CAM	I	-	GPIO_B15	CAM_PCLK	CSI2_CLK_N	SPI_CAM_S		AB19

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
								CK		
CAM_DATA_0	B	CAM	I	-	GPIO_B16	CAM_Data 0	CSI2_D0_P	SPI_CAM_DI/d0		AC19
CAM_DATA_1	B	CAM	I	-	GPIO_B17	CAM_Data 1	CSI2_D0_N	SPI_CAM_OF/d1		W18
CAM_DATA_2	B	CAM	I	-	GPIO_B18	CAM_Data 2	CSI2_D1_P	SPI_CAM_RD/d2		AB20
CAM_DATA_3	B	CAM	I	-	GPIO_B19	CAM_Data 3	CSI2_D1_N	SPI_CAM_SSN/d3		Y20
CAM_DATA_4	B	CAM	I	-	GPIO_B20	CAM_Data 4	CSI1_D0_P	M_SPI_DO	NFD_8/LCD_DATA_16	AC21
CAM_DATA_5	B	CAM	I	-	GPIO_B21	CAM_Data 5	CSI1_D0_N	M_SPI_D1	NFD_9/LCD_DATA_17	Y17
CAM_DATA_6	B	CAM	I	-	GPIO_B22	CAM_Data 6	CSI1_D1_P	M_SPI_D2	NFD_10LCD_DATA_18	AC20
CAM_DATA_7	B	CAM	I	-	GPIO_B23	CAM_Data 7	CSI1_D1_N	M_SPI_D3	NFD_11/LCD_DATA_19	AA19
M_SPI_CS_0	B	CAM	I	-	GPIO_B24	SPI Flash CS0	CSI1_CLK_N		NFD_12/LCD_DATA_20	AB17
SD card interface										
SSD1_CLK	B	MMC	I	-	GPIO_C9	SD CLK				P18
SSD1_CMD	B	MMC	I	-	GPIO_C10	SD CMD				P17
SDAT1_0	B	MMC	I	-	GPIO_C11	SD data0				AA16
SDAT1_1	B	MMC	I	-	GPIO_C12	SD data1				R16
SDAT1_2	B	MMC	I	-	GPIO_C13	SD data2				AB16
SDAT1_3	B	MMC	I	-	GPIO_C14	SD data3				W16
SSD2_CLK	B	STD	I	-	GPIO_C15	SD CLK				R18
SSD2_CMD	B	STD	I	-	GPIO_C16	SD CMD				R17
SDAT2_0	B	STD	I	-	GPIO_C17	SD data0				V16
SDAT2_1	B	STD	I	-	GPIO_C18	SD data1				U16
SDAT2_2	B	STD	I	-	GPIO_C19	SD data2				P16
SDAT2_3	B	STD	I	-	GPIO_C20	SD data3				U17
SPI interface										
SPI1_CLK	B	STD	I	-	GPIO_C21	SPI1_CLK	SPI_BB_CLK			Y14
SPI1_CS_0	B	STD	I	-	GPIO_C22	SPI1_CS_0	SPI_BB_CS_0			V10
SPI1_CS_1	B	STD	I	-	GPIO_A17	SPI1_CS_1	SPI_BB_CS_1			W15
SPI1_DIO	B	STD	I	-	GPIO_C23	SPI1_DIO	SPI_BB_DIO			U13
SPI1_DI	B	STD	I	-	GPIO_C24	SPI1_DI	SPI_BB_DI			AB10
UART interface										
UART1_RXD	B	STD	I	-	GPIO_C6	UART1 Receive Data				AB9
UART1_TXD	B	STD	I	-	GPIO_A14	UART1 Transmit Data				AA11
UART1_CTS	B	STD	I	-	GPIO_A15	UART1 Clear To Send	KEYIN_7			W11
UART1_RTS	B	STD	I	-	GPIO_A16	UART1 Request To Send	KEYOUT_7			Y10
HST_RXD	B	STD	I	-	HST_RXD	HST_RXD	GPIO_C7	UART2_RXD	UART1_DTR	W10

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
HST_TXD	B	STD	O	1	HST_TXD	HST_TXD	GPIO_C8	UART2_TXD	UART1 Data Carrier Detect	V11
UART2_CTS	B	STD	I	-	GPIO_B8	UART2_CTS	KEYIN_6	UART1 Data Set Ready	JTAG_TDI	U10
UART2_RTS	B	STD	I	-	GPIO_B9	UART2_RTS	KEYOUT_6	UART1 Ring Indicator	JTAG_NTRST	AC11
UART3_RXD	B	STD	I	-	UART3_RXD	UART3_RXD	GPIO_D0 INT			T15
UART3_TXD	B	STD	O	1	UART3_TXD	UART3_TXD	GPIO_D1 INT			AC18
UART3_CTS	B	STD	I	-	UART3_CTS	UART3_CTS	GPIO_D2 INT			AB18
UART3_RTS	B	STD	O	1	UART3_RTS	UART3_RTS	GPIO_D3 INT			R15
I2S interface										
I2S_BCK	B	STD	I	-	GPIO_A9	Audio BCK				V14
I2S_LRCK	B	STD	I	-	GPIO_A10	Audio LRCK	DAI_CLK	DAI_SIMPLE_CLK		AC17
I2S_DI_0	B	STD	I	-	GPIO_A11	Audio Serial Data In 0	DAI_DI	DAI_SIMPLE_DI		AC16
I2S_DI_1	B	STD	I	-	GPIO_A12	Audio Serial Data In 1	DAI_RST	DAI_SIMPLE_RST		V13
I2S_DO	B	STD	I	-	GPIO_A13	Audio Serial Data Out	DAI_DO	DAI_SIMPLE_DO		U14
I2C interface										
I2C1_SCL	B	STD	I	-	GPIO_B30	I2C1_SCL				P14
I2C1_SDA	B	STD	I	-	GPIO_B31	I2C1_SDA				AB14
GPIOs										
GPIO_A_0	B	STD	I	-	GPIO_A0	GPIO_A0 INT	I2C2_SCL			Y8
GPIO_A_1	B	STD	I	-	GPIO_A1	GPIO_A1 INT	I2C2_SDA			AA9
GPIO_A_2	B	STD	I	-	GPIO_A2	GPIO_A2 INT	SPI2_CLK			AC9
GPIO_A_3	B	STD	I	-	GPIO_A3	GPIO_A3 INT	SPI2_DIO			W9
GPIO_A_4	B	STD	I	-	GPIO_A4	GPIO_A4 INT	SPI2_DI			AC8
GPIO_A_5	B	STD	I	-	GPIO_A5	GPIO_A5 INT	SPI2_CS_0			V8
GPIO_A_6	B	STD	I	-	GPIO_A6	GPIO_A6 INT	SPI2_CS_1	KEYIN_3		AB8
GPIO_A_7	B	STD	I	-	GPIO_A7	GPIO_A7 INT	KEYIN_4	LPSCO_1		W8
GPIO_B_0	B	STD	I	-	GPIO_B0	GPIO_B0 INT	KEYIN_0			AC14
GPIO_B_1	B	STD	I	-	GPIO_B1	GPIO_B1 INT	KEYIN_1	SPI1_CS_2		Y13
GPIO_B_2	B	STD	I	-	GPIO_B2	GPIO_B2 INT	KEYIN_2	I2S_DI_2		AA14
GPIO_B_3	B	STD	I	-	GPIO_B3	GPIO_B3 INT	KEYOUT_0	TCO_0		W14
GPIO_B_4	B	STD	I	-	GPIO_B4	GPIO_B4 INT	KEYOUT_1	TCO_1		Y15
GPIO_B_5	B	STD	I	-	GPIO_B5	GPIO_B5 INT	KEYOUT_2	TCO_2		AB15
GPIO_B_6	B	STD	I	-	GPIO_B6	GPIO_B6 INT	I2C3_SCL	KEYOUT_3		AC15
GPIO_B_7	B	STD	I	-	GPIO_B7	GPIO_B7 INT	I2C3_SDA	KEYOUT_4		AB13
GPO_0	B	STD	O	0	GPO 0	GPO 0	PWT	KEYIN_5	JTAG_TMS	W13
GPO_1	O	STD	O	1	GPO 1	GPO 1	LPG	KEYOUT_5	JTAG_TDO	M14
GPO_2	O	STD	O	0	GPO 2	GPO 2	PWL_1	CLK_32K	JTAG_TCK	N14
GPIO_C_1	B	STD	I	-	GPIO_C1	GPIO_C1 INT	Selects NAND Flash 8bit/16bit ('0':16bit, '1':8bit)			J16
GPIO_C_2	B	STD	I	-	GPIO_C2	GPIO_C2 INT				K16

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
GPIO_C_3	B	STD	I	-	GPIO_C3	GPIO_C3 INT				J17
GPIO_C_4	B	STD	I	-	GPIO_C4	GPIO_C4 INT				L17
GPIO_C_5	B	STD	I	-	GPIO_C5	GPIO_C5 INT				K17
Miscellaneous Pins										
CLK_OUT	B	STD	I	-	HST_CLK	GPIO_A8	Clock Output (32kHz/26MHz/156Mhz/n)	HST_CLK		AC13
VOLUME_UP	I	STD	I	-	VOLUME_UP	VOLUME_UP				T11
VOLUME_DN	I	STD	I	-	VOLUME_DN	VOLUME_DN				AB11
PLUGIN	I	STD	I	-	PLUGIN	PLUGIN				Y11
TST_H	I	STD	I	-	TST_H	Test Mode=0				T13
ANA_TEST_EN	I	STD	I	-	ANA_TEST_EN					R13
RESETB_EXT	I	STD	I	-	RESETB_EXT					R14
CLK26M_REQUEST	I	STD	I	-	CLK26M_REQUEST					AC10
ZQ	I	STD	I	-	LPDDR2 External Impedance					A17
ZQ1	I	STD	I	-	LPDDR2 External Impedance					B16
PROG_EFUSE	I	PMU				High voltage input for OTP/EFUSE programming				Y6
ANA_TEST_EN	I	PMU				XVR test mode				R13
POWERKEY	I	PMU				Power-on switch enable signal. Active High.				V2
VSPIM_VSEL	I	PMU				Selects V_SPIMEM power supply ('0':2.8V, '1':1.8V).				U11
SIM card interface										
SIM1_RST	O	SIM1	O	0	SIM1_RST					Y7
SIM1_CLK	O	SIM1	O	0	SIM1_CLK					W5
SIM1_DIO	B	SIM1	I	-	SIM1_DIO					U7
SIM2_RST	B	SIM2	I	-	GPIO_C25	GPIO_C25	SIM2_RST			AA6
SIM2_CLK	B	SIM2	I	-	GPIO_C26	GPIO_C26	SIM2_CLK			AC7
SIM2_DIO	B	SIM2	I	-	GPIO_C27	GPIO_C27	SIM2_DIO			AB7
SIM3_RST	B	SIM3	I	-	GPIO_C28	GPIO_C28	SIM3_RST			U8
SIM3_CLK	B	SIM3	I	-	GPIO_C29	GPIO_C29	SIM3_CLK			T10
SIM3_DIO	B	SIM3	I	-	GPIO_C30	GPIO_C30	SIM3_DIO			AA8
Boost power supply										
VBOOST1	O	PMU				DCDC power supply				R1
VBOOST2	O	PMU								R2

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
BOOST_CPP1	O	PMU				DC/DC control				T1
BOOST_CPP2	O	PMU								T2
BOOST_CPP3	O	PMU								T3
BOOST_CPN1	O	PMU								U1
BOOST_CPN2	O	PMU								U2
BOOST_CPN3	O	PMU								U3
Charger interface										
GDRV	O	PMU				Charger drive				R4
TS	I	PMU				Battery connection detect				N4
AC_R	I	PMU				Input from the AC charger or USB inlet				P5
VBAT_SENSE	I	PMU				Battery voltage ADC detection				P3
IS_CHG	I	PMU				Current Sens for Charger Control				R5
Power supplies										
V_CORE1	O	PMU				Supplies BB Core				K6
V_CORE2	O	PMU								K7
V_CORE3	O	PMU								L6
V_CORE4	O	PMU								L7
V_CORE5	O	PMU								L8
V_CORE6	O	PMU								M6
V_CORE7	O	PMU								M7
V_CORE8	O	PMU								M8
V_CORE9	O	PMU								N8
V_CORE10	O	PMU								P8
SW_BUCK1_1	O	PMU				DC/DC control				M1
SW_BUCK1_2	O	PMU								M2
AVDD_2V4_1	O	PMU				Supplies XCVER				K21
AVDD_2V4_2	O	PMU				Supplies XCVER				K1
SW_BUCK2	O	PMU				DC/DC control				L1
AVDD_DDR_1V2_1	O	PMU				Suplies external DDR				A20
AVDD_DDR_1V2_2	O	PMU								A21
AVDD_DDR_1V2_3	O	PMU								B17
AVDD_DDR_1V2_4	O	PMU				Suplies external DDR				C16
AVDD_DDR_1V2_5	O	PMU								C17
AVDD_DDR_1V2_6	O	PMU								D1
AVDD_DDR_1V2_7	O	PMU								E1

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
SW_BUCK3	O	PMU				DC/DC control				F1
VBUCK4_3V3	O	PMU				Boost voltage output				J1
SW_BUCK4	O	PMU				DC/DC control				G1
V_IO	O	PMU				Supplies Standard PADs I/O ring				A3
V_SPIMEM	O	PMU				Supplies external SPI flash				G5
V_MEM	O	PMU				Supplies Memory PADs I/O ring and V_MEM output PAD				G21
V_CAM	O	PMU				Supplies Camera PADs I/O ring				B1
V_MMC	O	PMU				Supplies MMC PADs I/O ring				A2
V_LCD	O	PMU				Supplies LCD PADs I/O ring				A12
V_RTC	O	PMU				Supplies RTC domain				L2
V_VIB	O	PMU				Supplies Vibrator				N3
V_ASW	O	PMU				Supplies external devices				A13
V_USB	O	PMU				Supplies USB PHY				V1
BAT_BACKUP	O	PMU				Charges Backup CAP				K4
VRF28_1	O	PMU				Supplies RF Transceiver				L20
VRF28_2	O	PMU				Supplies RF Transceiver				L21
VTT_DDR	O	PMU				Supplies external DDR VTT				A16
VREF_DDR_1	O	PMU				Supplies external DDR VREF				A14
VREF_DDR_2	O	PMU				Supplies external DDR VREF				A15
BL_LED_OUT	O	PMU				LED driver				K2
VBAT_PMU1	I	PMU				PMU Battery Power Supply				N1
VBAT_PMU2	I	PMU								N2
VBAT_PMU3	I	PMU								P1
VBAT_PMU4	I	PMU								P2
VSIM1	O	PMU				SIM Card 1 Power Supply				AA5
VSIM2	O	PMU				SIM Card 2 Power Supply				Y5

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
VSIM3	O	PMU				SIM Card 3 Power Supply				W7
LED driver										
KP_LED_R	I	PMU				LED driver current sink				W4
KP_LED_G	I	PMU				LED driver current sink				U4
KP_LED_B	I	PMU				LED driver current sink				V4
LED1	I	PMU				LED driver current sink				A1
LED2	I	PMU								F2
LED3	I	PMU								E2
LED4	I	PMU								D2
LED5	I	PMU								C2
LED6	I	PMU								C1
LED7	I	PMU								B2
LED8	I	PMU								G2
Touch screen interface										
TSC_XN	I	PMU				Horizontal Measure Input				N7
TSC_XP	I	PMU				Horizontal Measure Input				P7
TSC_YN	I	PMU				Vertical Measure Input				T7
TSC_YP	I	PMU				Vertical Measure Input				N6
AU_LSL_P	O	PMU				Loudspeaker Left Out +				Y2
AU_LSL_N	O	PMU				Loudspeaker Left Out -				W2
AU_LSR_N	O	PMU				Loudspeaker Right Out +				W1
AU_LSR_P	O	PMU				Loudspeaker Right Out -				Y1
AU_RCV_P	O	PMU				Receiver Out +				AA1
AU_RCV_N	O	PMU				Receiver Out -				AA2
AU_HPR	O	PMU				Headset Right Out				AB1
AU_HPL	O	PMU				Headset Left Out				AC1
USB interface										
USB_DM	B	PMU				USB D-				AC12
USB_DP	B	PMU				USB D-				AB12
USB_ID	I	PMU				USB ID				AA13
USB_VBUS	I	PMU				USB Power Supply				AA12
GPADC interface										

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
AC_DIV_IN	I	PMU				AC charger voltage detection				P4
GPADC_IN_0	I	PMU				GPADC input channel 0				J2
GPADC_IN_1	I	PMU				GPADC input channel 1				J3
GPADC_IN_2	I	PMU				GPADC input channel 2				H3
GPADC_IN_3	I	PMU				GPADC input channel 3				J4
Audio interface										
V_ANA	O	ABB				Supplies analog BB				AC2
V_MIC	O	ABB				Microphone Biasing				Y3
V_NEG	O	ABB				Negative Voltage generator				AB2
LINE_IN_R	I	ABB				Stereo Line input right				AC3
LINE_IN_L	I	ABB				Stereo Line input left				AB3
AU_AUXMIC_N	I	ABB				Audio Aux In-				AB4
AU_AUXMIC_P	I	ABB				Audio Aux In+				AC4
AU_MICN_R	I	ABB				Audio In right -				AB5
AU_MICP_R	I	ABB				Audio In right +				AC5
AU_MICN_L	I	ABB				Audio In left -				AB6
AU_MICP_L	I	ABB				Audio In left +				AC6
RF interface										
AVDD_2V4_1	O	XCVER				Supplies XCVER				K1
AVDD_2V4_2	O	XCVER				Supplies XCVER				K21
AUXCLK_OUT	O	XCVER				Clock Output (32kHz/26MHz /156Mhz/n)				AB21
XVR_XTAL1	I	XCVER				26MHz Crystal port Input				AA21
XVR_XTAL2	O	XCVER				26MHz Crystal port Output				Y21
PAON	O	XCVER				Power Amplifier On				H21
XVR_BS0	O	XCVER				FEM Band select signal 0				J21
XVR_BS1	O	XCVER				FEM Band select signal 1				J20
XVR_BS2	O	XCVER				FEM Band select signal 2				K20
RF_SINGLE_L	I	XCVER				GSM 900/GSM850 band single input				U21

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
RF_SINGLE_H	I	XCVER				PCS 1900/DCS1800 band single input				V21
RFOL	O	XCVER				GSM 900 and GSM 850 band transmit Output				T21
RFOH	O	XCVER				PCS 1900 and DCS 1800 band transmit Output				R21
RAMPOUT	O	XCVER				Ramp DAC Output				M20
IP	O	XCVER				BB Analog IP for Test				N17
IN_STROBE	O	XCVER				BB Analog IN for Test	DigRf strobe			M16
QP_RXTXEN	O	XCVER				BB Analog QP for Test	DigRF's RxTxEn			N16
QN_RXTXDATA	O	XCVER				BB Analog QN for Test	DigRF's RxTxData			M17
Ground										
SW_GND1						DC/DC Switch GND				H1
SW_GND2										H2
AVSS_XCV1						XVR_GND				M21
AVSS_XCV2										N21
AVSS_XCV3										P20
AVSS_XCV4										P21
AVSS_XCV5										R20
AVSS_XCV6										T20
AVSS_XCV7										U20
AVSS_XCV8										V20
AVSS_XCV9										W20
AVSS_XCV10										W21
CORE_GND1						CORE_GND				C12
CORE_GND2										C13
CORE_GND3										C14
CORE_GND4										D13
CORE_GND5										D16
CORE_GND6										D17
CORE_GND7										E13
CORE_GND8										E17
CORE_GND9										F16
CORE_GND10										F17
CORE_GND11										G16
CORE_GND12										G17
CORE_GND13										H17
CORE_GND14										J11
CORE_GND15										J12

Pin Name	I/O	Power Domain	Default Direction	Default Value	Default Function	First usage	Secondary Usage	Third Usage	Fourth Usage	Ball Name
CORE_GND16										J13
CORE_GND17										K11
CORE_GND18										K12
CORE_GND19										K13
CORE_GND20										L11
CORE_GND21										L12
CORE_GND22										L13
CORE_GND23										M11
CORE_GND24										M12
CORE_GND25										M13
CORE_GND26										N10
CORE_GND27										N11
CORE_GND28										N12
CORE_GND29										N13
CORE_GND30										P12
CORE_GND31										R12

9. Electrical Characteristics

9.0.1 Absolute Maximum Rating

Name	Min	Typical	Max	Usage (power domain)
VBAT			6V	For chip
DC Charger			7V	For charger circuit

9.0.2 Operating Rating

Name	Min	Typical	Max	Usage (power domain)
VBAT	3.4V	3.8V	4.2V	For chip

9.0.3 Temperature Characteristics

Parameter	Min	Typ	Max	Unit
Work Temperature	-20	25	65	°C
Storage Temperature	-40	/	125	°C

9.0.4 Relative Humidity [%] <85%

9.0.5 RF Characteristics

TX

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
Fo	output frequency	GSM850 Band	824		849	MHz
		EGSM Band	880		915	MHz
		DCS Band	1710		1785	MHz
		PCS Band	1850		1910	MHz
Po	output power	GSM850 Band	4	6	8	dBm
		EGSM Band	4	6	8	dBm
		DCS Band	4	6	8	dBm
		PCS Band	4	6	8	dBm
PE_rms	RMS phase error	GSM850 Band		1.5	3	deg
		EGSM Band		1.5	3	deg
		DCS Band		2.2	4	deg
		PCS Band		2.2	4	deg
PE_peak	peak phase error	GSM850 Band			10	deg
		EGSM Band			10	deg
		DCS Band			15	deg
		PCS Band			15	deg
M400K	output modulation @ +-400KHz offset	GSM850 Band		-65	-62	dBc
		EGSM Band		-65	-62	dBc
		DCS Band		-64	-62	dBc
		PCS Band		-64	-62	dBc
M1.8M	output modulation @ +-1.8MHz offset	GSM850 Band		-74	-68	dBc
		EGSM Band		-74	-68	dBc
		DCS Band		-71	-65	dBc
		PCS Band		-71	-65	dBc
HS2	2nd harmonic suppression	GSM850 Band			-30	dBc
		EGSM Band			-30	dBc
		DCS Band			-25	dBc
		PCS Band			-25	dBc
HS3	3rd harmonic suppression	GSM850 Band			-8	dBc
		EGSM Band			-8	dBc
		DCS Band			-12	dBc
		PCS Band			-12	dBc

RX

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit	
Fi	input frequency	GSM850 Band	869		894	MHz	
		EGSM Band	925		960	MHz	
		DCS Band	1805		1880	MHz	
		PCS Band	1930		1990	MHz	
Srx	RX sensitivity	GSM850 Band		-111.5	-109.5	dBm	note1
		EGSM Band		-111.5	-109.5	dBm	note1
		DCS Band		-111.5	-109.5	dBm	note1
		PCS Band		-110.5	-108.5	dBm	note1
Block3	the blocking characteristic @+-3MHz offset	GSM850 Band		-23		dBm	note2
		EGSM Band		-24		dBm	note2
		DCS Band		-23		dBm	note2
		PCS Band		-23		dBm	note2
Block20	the blocking characteristic @+-20MHz offset	GSM850 Band		-18		dBm	note2

		EGSM Band		-20		dBm	note2
		DCS Band		-19		dBm	note2
		PCS Band		-20		dBm	note2
SUPam	AM suppression	GSM850 Band		-22		dBm	note2
		EGSM Band		-22		dBm	note2
		DCS Band		-22		dBm	note2
		PCS Band		-22		dBm	note2
ACS200K	adjacent channel selectivity @+-200KHz	GSM850 Band	10	16		dBc	note3
		EGSM Band	10	16		dBc	note3
		DCS Band	11	17		dBc	note3
		PCS Band	11	17		dBc	note3
ACS400K	adjacent channel selectivity @+-400KHz	GSM850 Band	45	57		dBc	note3
		EGSM Band	45	57		dBc	note3
		DCS Band	44	56		dBc	note3
		PCS Band	44	56		dBc	note3
Pi_max	maximum input power	GSM850 Band	-15			dBm	
		EGSM Band	-15			dBm	
		DCS Band	-24			dBm	
		PCS Band	-24			dBm	

Note1. The insertion loss of the antenna switch and the PCB have been calibrated
Note2. Pdesired=-99dBm
Note3. Pdesired=-85dBm

9.0.6 Audio Characteristics

Class K						
OUT RMS ON	THD	SINAD	OUT RMS OFF	OUT PWR	SNR	Battery Voltage
(v)	%	(dB)	(v)			
0.407738	0.028088	60.39795	0.000038	0.02	80.61	4
1.61769	0.089419	58.53993	0.000106	0.33	83.67	4
3.102641	1.108774	38.38455	0.000198	1.2	83.9	4
3.541256	13.22943	17.52715	0.000277	1.57	82.13	4
3.691639	19.08801	14.3687	0.000346	1.7	80.56	4
3.788224	22.92822	12.76289	0.000415	1.79	79.21	4
3.842082	25.25349	11.92124	0.000468	1.85	78.29	4
3.887335	27.4225	11.20234	0.000533	1.89	77.26	4
Class D						
OUT RMS ON	THD	SINAD	OUT RMS OFF	OUT PWR	SNR	Battery Voltage
(v)	%	(dB)	(v)			
0.40716	0.02335	61.2679	0.000034	0.02	81.57	4
1.62075	0.09883	58.1445	0.000093	0.33	84.82	4
2.4675	15.0481	16.4175	0.000181	0.76	82.69	4
2.60929	22.7619	12.8336	0.000245	0.85	80.55	4
2.6702	26.2587	11.5887	0.000306	0.89	78.82	4

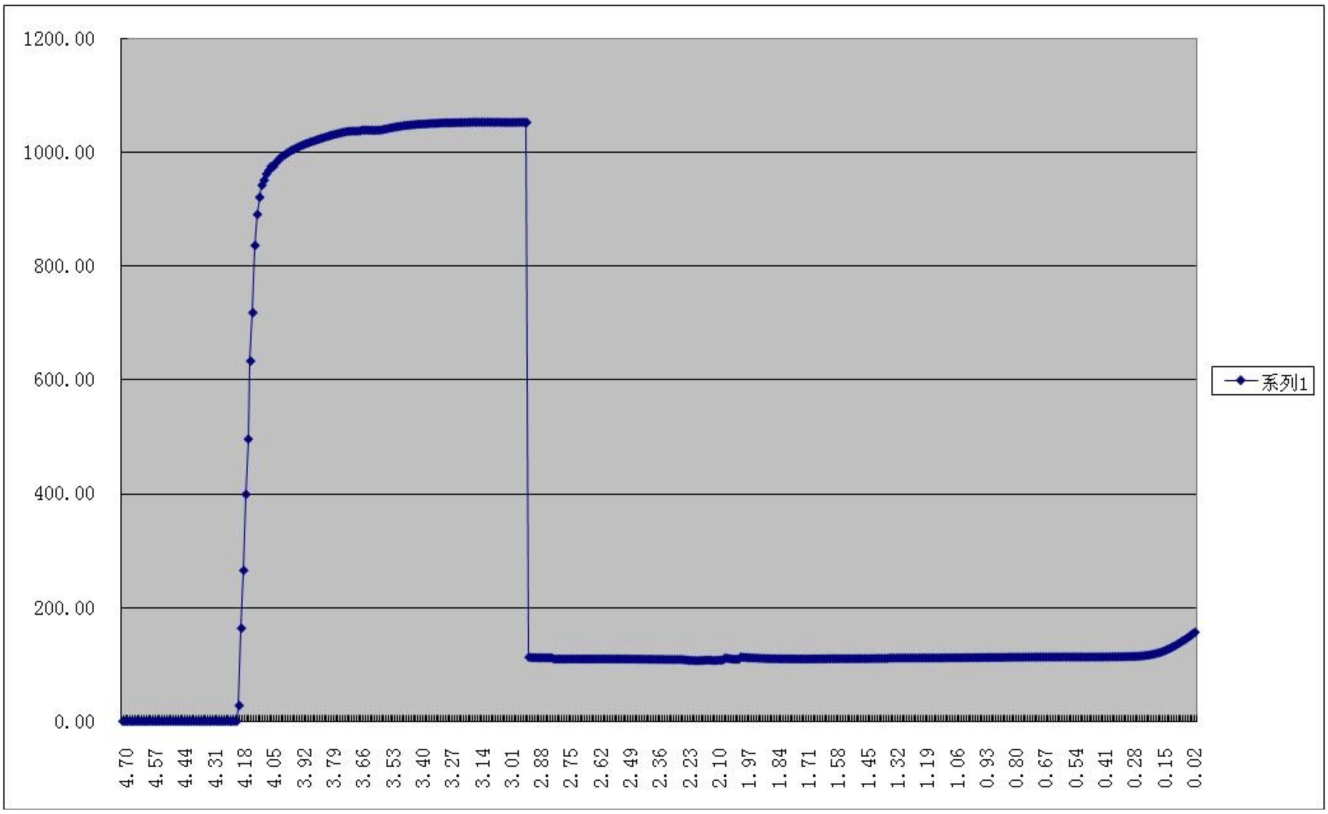
2.70852	28.5461	10.8718	0.000366	0.92	77.39	4
2.73082	29.8979	10.4747	0.000414	0.93	76.39	4
2.7511	31.1126	10.1232	0.00048	0.95	75.17	4
Headphone						
OUT RMS ON	THD	SINAD	OUT RMS OFF	OUT PWR	SNR	Battery Voltage
(v)	%	(dB)	(v)			
0.23615	0.00844	46.353	0.000024	0.001743	79.86	4
0.32909	0.01013	46.3297	0.000036	0.003384	79.22	4
0.46948	0.01152	46.3691	0.000046	0.006888	80.18	4
0.6456	2.45938	31.9825	0.000066	0.013025	79.81	4
0.82236	13.9234	17.1464	0.000091	0.021134	79.12	4
1.02057	21.5071	13.3848	0.000127	0.032549	78.1	4
1.11581	26.4702	11.5432	0.000193	0.038907	75.24	4
1.16341	30.8232	10.1984	0.000326	0.042298	71.05	4
Receiver						
OUT RMS ON	THD	SINAD	OUT RMS OFF	OUT PWR	SNR	Battery Voltage
(v)	%	(dB)	(v)			
0.131718	0.02263	46.29659	0.000019	0.002169	76.82	4
0.186607	0.032254	46.29754	0.000023	0.004353	78.18	4
0.261563	0.046055	46.2518	0.000029	0.008552	79.1	4
0.370479	0.066329	46.17642	0.00004	0.017157	79.33	4
0.519742	0.096372	46.15026	0.000053	0.033766	79.83	4
0.731471	0.162468	45.74333	0.000074	0.066881	79.9	4
0.978627	4.08141	27.70084	0.000106	0.119714	79.31	4
1.121201	14.80495	16.579	0.000143	0.157136	77.89	4

9.0.7 DC Characteristics

Name	Min.(V)	Typical(V)	Max.(V)	Description
VBAT	3.4	3.8	4.2	Normal work
VCORE	0.9	1.3	1.8	Normal work
V_BUCK_2V4	1.4	2.4	2.8	Normal work
AVDD_2V4	2.4	2.4	2.7	Normal work
AVDD_DDR1V2	1.1	1.2	1.5	Normal work
VBUCK4_3V3	3.0	3.3	3.5	Normal work
V_ANA	2.6	2.8	3.0	Normal work

Name	Min.(V)	Typical(V)	Max.(V)	Description
V_RTC	1.1	1.35	1.5	Normal work
V_MIC	1.9	2.1	2.3	Normal work
V_USB	2.9	3.25	3.6	Normal work
V_SIM	1.6	1.8	2.0	When 1.8V mode
	2.6	2.8	3.0	When 3.0V mode
V_PAD,V_MEM, V_SPIMEM, V_ASW,V_CAM V_LCD V_MMC V_VIB	1.6	1.8	2.0	When 1.8V mode
	2.6	2.8	3.0	When 3.0V mode

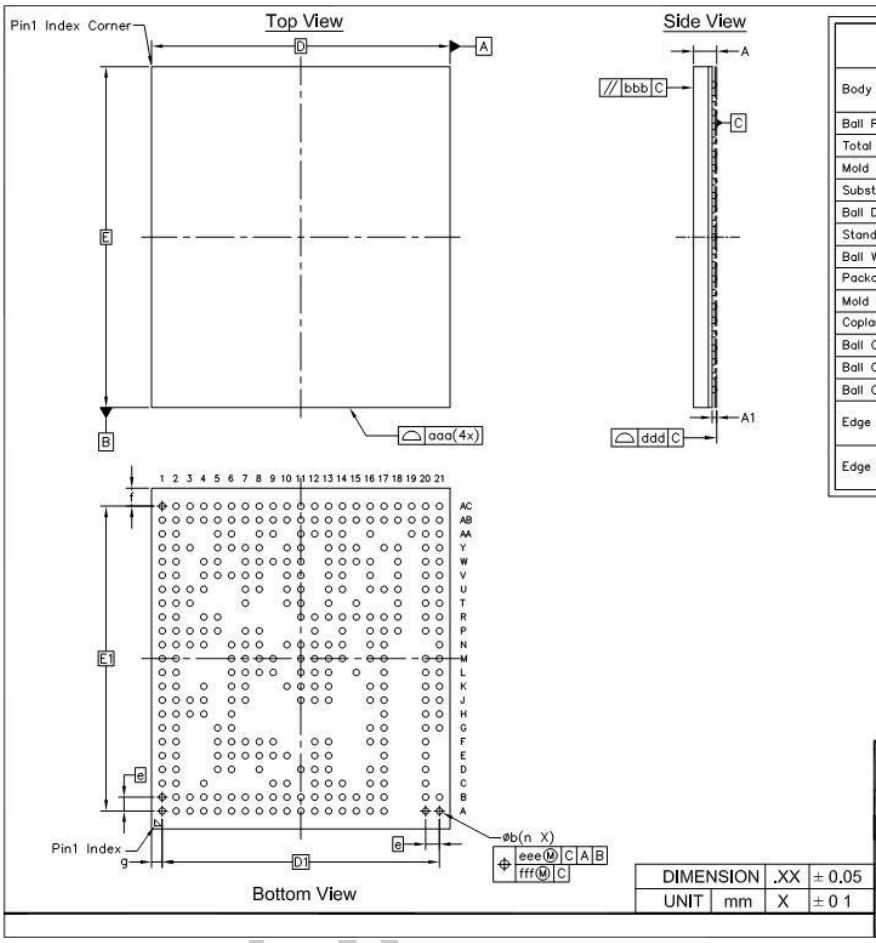
9.0.8 Charge Characteristics



10. BALL OUT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A	LED1	V_MMC	V_IO	NFCLE	NFD<4>	NFD<2>	NFD<0>	LCD_DATA<10>	LCD_DATA<11>	LCD_DATA<12>	LCD_DATA<13>	V_LCD	V_ASW	VREF_DDR	VREF_DDR	VTT_DDR	ZQ			AVDD_DDR_1V2	AVDD_DDR_1V2
B	V_CAM	LED7	NC	NFRB	NFREN	NFWEN	NFDQS	LCD_DATA<9>	LCD_DATA<14>	LCD_WR	LCD_DATA<0>	LCD_CS0	LCD_RD	LCD_DATA<1>	NC	ZQ1	AVDD_DDR_1V2			NC	NC
C	LED6	LED5		NFWPN					LCD_FMARK	LCD_RS		CORE_GND	CORE_GND	CORE_GND		AVDD_DDR_1V2	AVDD_DDR_1V2			NC	
D	AVDD_DDR_1V2	LED4			NFALE	NFD<1>		LCD_CS1			LCD_DATA<6>	LCD_DATA<7>	CORE_GND			CORE_GND	CORE_GND			NC	
E	AVDD_DDR_1V2	LED3			NFD<7>	NFD<5>	LCD_DATA<8>	LCD_DATA<15>	LCD_DATA<4>	LCD_DATA<5>		NC	CORE_GND				CORE_GND			NC	
F	SW_BUCK3	LED2			NC	NFD<6>	NC	LCD_DATA<2>	LCD_DATA<3>			NC	NC			CORE_GND	CORE_GND			NC	
G	SW_BUCK4	LED8			V_SPIMEM	NFCEN										CORE_GND	CORE_GND			NC	V_MEM
H	SW_GND	SW_GND	GPADC_IN<2>	NC		NC											CORE_GND			NC	PAON
J	VBUCK4_3V3	GPADC_IN<0>	GPADC_IN<1>	GPADC_IN<3>		NFD<3>	NC				CORE_GND	CORE_GND	CORE_GND			GPIO_C<1>	GPIO_C<3>			XVR_BS1	XVR_BS0
K	AVDD_2V4	BL_LED_OUT		BAT_BACKUP		V_CORE	V_CORE			NC	CORE_GND	CORE_GND	CORE_GND			GPIO_C<2>	GPIO_C<5>			XVR_BS2	AVDD_2V4
L	SW_BUCK2	V_RTC				V_CORE	V_CORE	V_CORE	NC		CORE_GND	CORE_GND	CORE_GND		NC		GPIO_C<4>			VRF28	VRF28
M	SW_BUCK1	SW_BUCK1				V_CORE	V_CORE	V_CORE	NC		CORE_GND	CORE_GND	CORE_GND	GPO<1>		IN_STROBE	QN_RXTXDATA			RAMPOUT	AVSS_XCV
N	VBAT_PMU	VBAT_PMU	V_VIB	TS		TSC_YP	TSC_XN	V_CORE		CORE_GND	CORE_GND	CORE_GND	CORE_GND	GPO<2>		QP_RXTXEN	IP				AVSS_XCV
P	VBAT_PMU	VBAT_PMU	VBAT_SENSE	AC_DIV_IN	AC_R		TSC_XP	V_CORE				CORE_GND		I2C1_SCL		SDAT2<2>	SSD1_CMD	SSD1_CLK		AVSS_XCV	AVSS_XCV
R	VBOOST	VBOOST		GDRV	IS_CHG						BBPLL_TEST	CORE_GND	ANA_TEST_EN	RESETB_EXT	UART3_RTS	SDAT1<1>	SSD2_CMD	SSD2_CLK		AVSS_XCV	RFOH
T	BOOST_CPP	BOOST_CPP	BOOST_CPP				TSC_YN			SIM_CLK<3>	VOLUME_UP		TST_H		UART3_RXD			CAM_PDN		AVSS_XCV	RFOL
U	BOOST_CPN	BOOST_CPN	BOOST_CPN	KP_LED_G			SIM_DIO<1>	SIM_RST<3>		UART2_CTS	VSPIM_VSEL		SPI1_DIO	I2S_DO		SDAT2<1>	SDAT2<3>	CAM_HREF		AVSS_XCV	RF_SINGLE_L
V	V_USB	POWERKEY		KP_LED_B	CP_GND	CP_GND	CP_GND	GPIO_A<5>		SPI1_CS_0	HST_TXD		I2S_DI<1>	I2S_BCK		SDAT2<0>		CAM_VSYNC		AVSS_XCV	RF_SINGLE_H
W	AU_LSR_N	AU_LSL_N		KP_LED_R	SIM_CLK<1>		VSIM3	GPIO_A<7>	GPIO_A<3>	HST_RXD	UART1_CTS		GPO<0>	GPIO_B<3>	SPI1_CS_1	SDAT1<3>		CAM_DATA<1>		AVSS_XCV	AVSS_XCV
Y	AU_LSR_P	AU_LSL_P	V_MIC		VSIM2	PROG_EFUSE	SIM_RST<1>	GPIO_A<0>		UART1_RTS	PLUGIN		GPIO_B<1>	SPI1_CLK	GPIO_B<4>		CAM_DATA<5>	CAM_CLK		CAM_DATA<3>	XVR_XTAL2
AA	AU_RCV_P	AU_RCV_N			VSIM1	SIM_RST<2>		SIM_DIO<3>	GPIO_A<1>		UART1_TXD	USB_VBUS	USB_ID	GPIO_B<2>		SDAT1<0>			CAM_DATA<7>	CAM_RST	XVR_XTAL1
AB	AU_HPR	V_NEG	LINE_IN_L	AU_AUXMIC_N	AU_MICN_R	AU_MICN_L	SIM_DIO<2>	GPIO_A<6>	UART1_RXD	SPI1_DI	VOLUME_DN	USB_DP	GPIO_B<7>	I2C1_SDA	GPIO_B<5>	SDAT1<2>	M_SPI_CS0	UART3_CTS	CAM_PCLK	CAM_DATA<2>	AUXCLK_OUT
AC	AU_HPL	V_ANA	LINE_IN_R	AU_AUXMIC_P	AU_MICP_R	AU_MICP_L	SIM_CLK<2>	GPIO_A<4>	GPIO_A<2>	CLK26M_REQUEST	UART2_RTS	USB_DM	CLK_OUT	GPIO_B<0>	GPIO_B<6>	I2S_DI<0>	I2S_LRCK	UART3_TXD	CAM_DATA<0>	CAM_DATA<6>	CAM_DATA<4>

11. Package



		Symbol	Common Dimensions		
			MIN.	NOM.	MAX.
Body Size:	X	D	13.90	14	14.10
	Y	E	15.90	16	16.10
Ball Pitch:		e	0.65		
Total Thickness:		A	--	--	1.3
Mold Thickness:		--	0.675	0.70	0.725
Substrate Thickness:		--	0.15	0.18	0.21
Ball Diameter:		--	0.30		
Stand Off:		A1	0.16	0.21	0.26
Ball Width:		b	0.25	0.30	0.35
Package Edge Tolerance:		aaa	0.10		
Mold Flatness:		bbb	0.10		
Coplanarity:		ddd	0.08		
Ball Offset (Package):		eee	0.15		
Ball Offset (Ball):		fff	0.08		
Ball Count:		n	325		
Edge Ball Center to Center:	X	D1	13.00		
	Y	E1	14.30		
Edge Ball Center to Package Edge:	X	g	0.40	0.50	0.60
	Y	f	0.75	0.85	0.95

DIMENSION	.XX	± 0.05
UNIT	mm	X	± 0 1

TITLE	RDA TFBGA 14x16 mm² 325L Package Outline Drawing

12. GLOSSARY

Name	Description
AHB	AMBA Advanced High-performance Bus.
AHB2AHB	Bridge allowing Sys AHB to communicate with BB AHB, and vice versa.
AIF	RDA8810PL's Audio Interface, including PCM, DAI and Audio Analog Interface.
APB	AMBA Advanced Peripheral Bus, slow bandwidth bus.
ATR	Answer To Reset, see SCI module for details.
BB	Base Band part of RDA8810PL, dedicated to GSM and GPRS modem.
Comfort Tone	Tone generated by the AIF, used to indicate ringing, occupied...
CS	Chip Select.
DAI	Digital Audio Interface, part of the AIF used for test purpose.
DMA	Direct Memory Access, can copy a part of the memory at an other address.
DTMF	Dual Tone Multi Frequency generated by the AIF when a number is dialed.
EBC	External Bus Controller, to connect external RAM, polyphonic chips, MMC...
ETU	Elementary Time Unit, see SCI module for details.
GP ADC	General Purpose Analog to Digital Converter.
IFC	Base Band Intelligent Flow Controller, a bridge between AHB and APB buses.
IrDA	Infra Red communication protocol used by RDA8810PL's UART infra red mode.
Mailbox	Communication module between the Sys AHB bus and the BB AHB bus.
MMC	Multi Media Chips.
MMI	Man-Machine Interface, a part of the software that handle the user interface.
PA DAC	Power Amplifier Digital to Analog Converter.
PCM	Pulse Code Modulation, the part of the AIF that outputs samples in serial.
PLL	Phase-Locked Loop.
RF IF	RF Interface, the part of RDA8810PL connected to the RF chip.
SCI	SIM Card Interface.
SIM	Subscriber Identity Module, used to read SIM Cards.
SPI	Serial Peripheral Interface.
Sys	System side of RDA8810PL, dedicated to the application, MMI and GSM stack.
TCO	Time-Controlled Output. Pins controlled by the TCU.
TCU	Timing Control Unit, handle the scheduling of individual events.
UART	Universal Asynchronous Receiver Transmitter, asynchronous serial port.
VCO	Voltage Controlled Oscillator.
XCPU	System Coolsand's 16/32-bit RISC CPU.

Document History

The evolution of this document is described in the following table:

Version	Date	Description
1.0	2013-12-11	Initial version.
1.03	2014-07-11	Change the pixel of the Camera Sensor to 2M, change the package total thickness to 1.3mm
1.04	2014-08-06	Change the Max work temperature to 65°C