S805

Datasheet

Revision: 0.8

Release date: 1/26/2015

Amlogic, Inc.

TRADEMARKS
AMLOGIC is a trademark of Amlogic, Inc. All other trademarks and registered trademarks are property of their respective companies.

DISCLAIMER
Amlogic Inc. may make improvements and/or changes in this document or in the product described in this document at any time.

This product is not intended for use in medical, life saving, or life sustaining applications.

Circuit diagrams and other information relating to products of Amlogic Inc. are included as a means or illustrating typical applications. Consequently, complete information sufficient for production design is not necessarily given. Amlogic makes no representations or warranties with respect to the accuracy or completeness of the contents presented in this document.

REVISION HISTORY

Revision Number	Revision Date	Changes
0.8	2015/1/26	Initial version release

CONTACT INFORMATION
Amlogic, Inc.
2518 Mission College Blvd, Ste 120
Santa Clara, CA 95054
U.S.A.
www.amlogic.com

Contents

Section I System

This part describes the S805 system architecture from the following aspects:

- GENERAL DESCRIPTIONS
- FEATURES SUMMARY
- CPU and GPU SUBSYSTEM
- MEMORY MAP
- POWER DOMAIN
- CLOCK AND RESET UNIT
- SYSTEM BOOT
- GENERAL PURPOSE INPUT/OUTPUT (GPIO)
- INTERRUPT CONTROLLER
- DIRECT MEMORY ACCESS CONTROLLER (DMAC)
- TIMER
- REAL TIME CLOCK

1. GENERAL DESCRIPTIONS

S805 is an advanced application processor designed for Set Top Box(STB) and high-end media player applications. It integrates a powerful CPU/GPU subsystem, and a secured FHD video CODEC engine with all major peripherals to form the ultimate low power multimedia AP.

The main system CPU is a quad-core ARM Cortex-A5 CPU with 32KB L1 instruction and 32KB data cache for each core and a large 512KB L2 unified cache to improve system performance. In addition, the Cortex-A5 CPU includes the NEON SIMD co-processor to improve software media processing capability. The quad core ARM Cortex-A5 CPU can run up to 1.5GHz and has a wide bus connecting to the memory sub-system.

The graphic subsystem consists of four graphic engines and a flexible video/graphic output pipeline. The four core ARM Mali-450 GPU including dual geometry processors (GP) and dual pixel processors (PP) handles all OpenGL ES 1.1/2.0 and OpenVG graphics programs, while the 2.5D graphics processor handles additional scaling, alpha, rotation and color space conversion operations. The video output pipeline can perform advanced image correction and enhancements. Together, the CPU and GPU handle all operating system, networking, user-interface and gaming related tasks.

One additional processor offloads the Cortex-A5 CPUs by handling all video CODEC processing. The Amlogic Video Engine (AVE) is a dedicated hardware video decoder and encoder capable of decoding 1080p resolution video with complete Trusted Video Path (TVP) for secure applications. The AVE supports full formats including MVC, MPEG-1/2/4, VC-1/WMV, AVS, RealVideo, MJPEG streams, H.264, H265 and also JPEG pictures with no size limitation. The independent encoder is able to encode in JPEG and H.264 up to 1080p at 30fps.

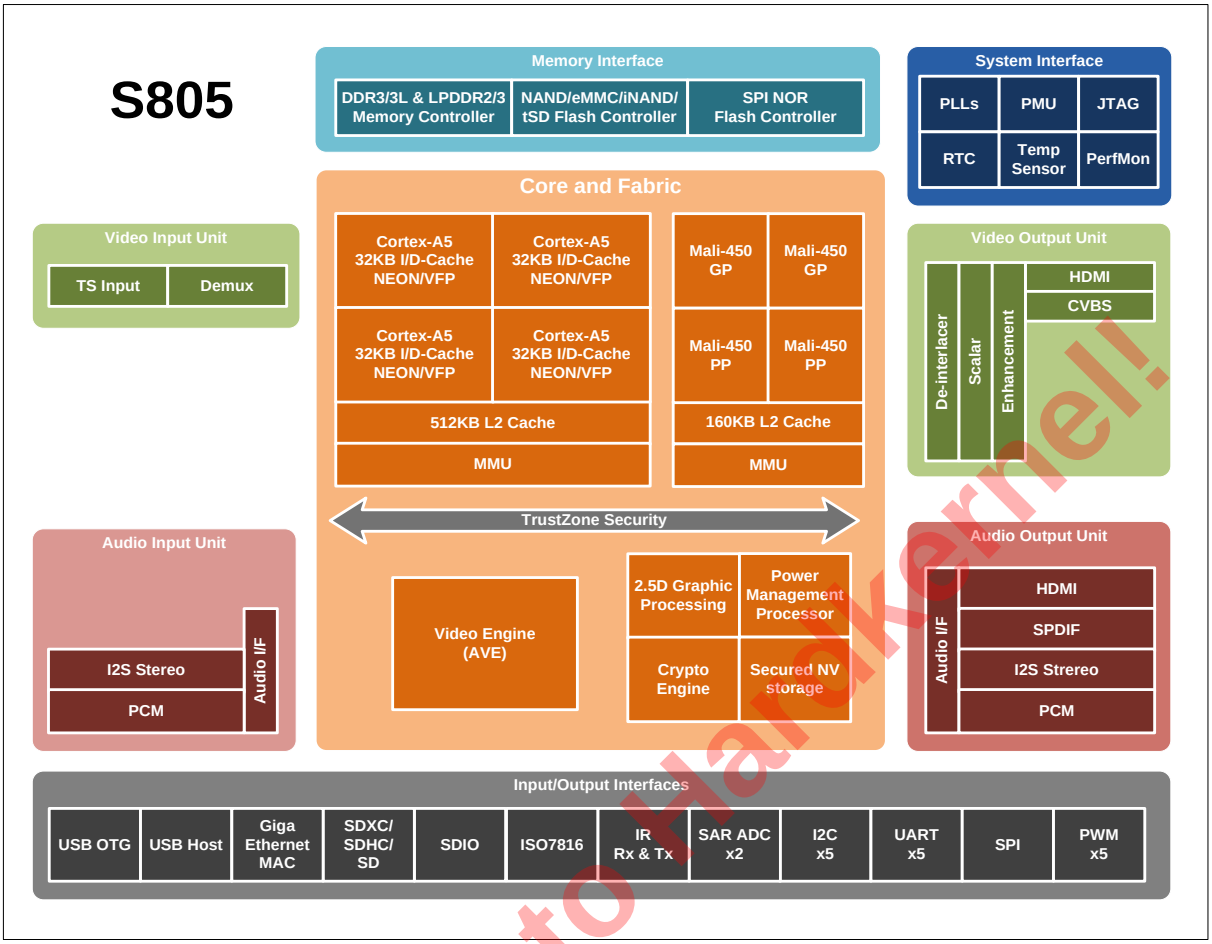
The S805 integrates all standard audio/video input/output interfaces including an HDMI1.4b transmitter with 3D support, CEC and HDCP, a CVBS output, I2S and SPDIF digital audio input/output interfaces and a PCM audio interface.

The S805 also integrates a set of functional blocks for digital TV broadcasting streams. The built-in three-way demux can process the TV streams from the serial transport stream input interface, which can connect to external tuner/demodulator. An ISO7816 smart card interface and a crypto-processor are built in to help handling encrypted traffic and media streams.

The processor has advanced network and peripheral interfaces, including a Gigabit Ethernet MAC with RMII/RGMII interface, dual USB 2.0 high-speed ports (one OTG and one HOST), three SDIOs with multi-standard memory card controller, five UART interfaces, five I2C interfaces, one high-speed SPI interface and five PWMs.

A standard development environment utilizing GNU/GCC Android tool chain is supported. Please contact your AMLOGIC sales representative for more information.

2. FEATURES SUMMARY



CPU Sub-system

- Quad core ARM Cortex-A5 CPU up to 1.5GHz (DVFS)
- ARMv7 instruction set, power efficient architecture
- 32KB instruction cache and 32KB data cache
- 512KB Unified L2 cache
- Advanced NEON and VFP co-processor
- Advanced TrustZone security system
- Application based traffic optimization using internal QoS-based switching fabrics

3D Graphics Processing Unit

- Quad-core ARM Mali-450 GPU up to 600MHz+ (DVFS)
- Dual Geometry Processors with 32KB L2 cache
- Dual Pixel Processors with 128KB L2 caches
- Concurrent multi-core processing
- 1200Mpix/sec and 132Mtri/sec
- Full scene over-sampled 4X anti-aliasing engine with no additional bandwidth usage
- OpenGL ES 1.1/2.0 and OpenVG 1.1 support

2.5D Graphics Processor

- Fast bitblt engine with dual inputs and single output
- Programmable raster operations (ROP)

- Programmable polyphase scaling filter
- Supports multiple video formats 4:2:0, 4:2:2 and 4:4:4 and multiple pixel formats (8/16/24/32 bits graphics layer)
- Fast color space conversion
- Advanced anti-flickering filter

Crypto Engine

- Supports AES block cipher with 128/192/256 bits keys, standard 16 bytes block size and streaming ECB, CBC and CTR modes
- Supports DES/3DES block cipher with ECB and CBC modes supporting 64 bits key for DES and 192 bits key for 3DES
- Built-in LSFR Random number generator

Video/Picture CODEC

- Amlogic Video Engine (AVE) with dedicated hardware decoders and encoders
- Hardware based trusted video path (TVP)
- Supports multiple "secured" video decoding sessions and simultaneous decoding and encoding
- Video/Picture Decoding
    - H.265 HEVC MP@L4.1 up to 1080P@60fps
    - H.264 AVC HP@L4.2 up to 1080P@60fps
    - H.264 MVC up to 1080P@60fps
    - MPEG-4 ASP@L5 up to 1080P@60fps (ISO-14496)
    - WMV/VC-1 SP/MP/AP up to 1080P@60fps
    - AVS JiZhun Profile up to 1080P@60fps
    - MPEG-2 MP/HL up to 1080P@60fps (ISO-13818)
    - MPEG-1 MP/HL up to 1080P@60fps (ISO-11172)
    - RealVideo 8/9/10 up to 1080P
    - WebM up to VGA
    - Multiple language and multiple format sub-title video support
    - MJPEG and JPEG unlimited pixel resolution decoding (ISO/IEC-10918)
    - Supports JPEG thumbnail, scaling, rotation and transition effects
    - Supports *.mkv,*.wmv,*.mpg, *.mpeg, *.dat, *.avi, *.mov, *.iso, *.mp4, *.rm and *.jpg file formats
- Video/Picture Encoding
    - Independent JPEG and H.264 encoder with configurable performance/bit-rate
    - JPEG image encoding
    - H.264 video encoding up to 1080P@30fps

Video Post-Processing Engine

- Motion adaptive 3D noise reduction filter
- Advanced motion adaptive edge enhancing de-interlacing engine
- 3:2 pull-down support
- Programmable poly-phase scalar for both horizontal and vertical dimension for zoom and windowing
- Programmable color management filter (to enhance blue, green, red, face and other colors)
- Dynamic Non-Linear Luma filter
- Programmable color matrix pipeline
- Video mixer: 2 video planes and 2 graphics planes per video output

Video Output

- Built-in HDMI 1.4b transmitter including both controller and PHY with CEC and HDCP, 1200p@60 max resolution output
- CVBS 480i/576i standard definition output

- Supports all standard SD/HD/FHD video output formats: 480i/p, 576i/p, 720p, and 1080i/p
- Supports 3D HDMI display

Audio Decoder and Input/Output

- Supports MP3, AAC, WMA, RM, FLAC, Ogg and programmable with 5.1 down-mixing
- I2S audio interface supporting 2-channel input/output
- Built-in serial digital audio SPDIF/IEC958 output and PCM input/output
- Supports concurrent dual audio stereo channel output with a combination of I2S+PCM

Memory and Storage Interface

- 16/32-bit SDRAM memory interface running up to DDR1600
- Supports up to 2GB DDR3, DDR3L with single rank and LPDDR2, LPDDR3 with dual ranks
- TrustZone protected DRAM memory region and internal SRAM
- Supports SLC/MLC/TLC NAND Flash with 60-bit ECC, compatible to ONFI 2.1 and Toggle 2.0 mode
- SDSC/SDHC/SDXC card and SDIO interface with 1-bit and 4-bit data bus width supporting spec version 2.x/3.x/4.x DS/HS modes up to UHS-I SDR50
- eMMC and MMC card interface with 1/4/8-bit data bus width supporting spec version 4.4x/4.5x HS200 (up to 100MHz clock), compatible with standard iNAND interface
- Supports serial 1, 2 or 4-bit NOR Flash via SPI interface
- Built-in 4k bits One-Time-Programming ROM for key storage

Network

- Integrated IEEE 802.3 10/100/1000 Gigabit Ethernet controller with RMII/RGMII interface
- Supports Energy Efficiency Ethernet (EEE) mode
- Optional 50MHz and 125MHz clock output to Ethernet PHY
- WiFi/IEEE802.11 & Bluetooth supporting via SDIO/USB/UART/PCM

Digital Television Interface

- Transport stream (TS) input interface with built-in demux processor for connecting to external digital TV tuner/demodulator
- Built-in PWM, I2C and SPI interfaces to control tuner and demodulator
- Integrated ISO 7816 smart card controller

Integrated I/O Controllers and Interfaces

- Dual USB 2.0 high-speed USB I/O, one USB Host and one USB OTG
- 5 UART, 5 I2C and 1 SPI interface with 3 slave selects
- Five PWMs
- Programmable IR remote input/output controllers
- Built-in 10bit SAR ADC with 2 input channels
- A set of General Purpose IO interfaces with built-in pull up and pull down

System, Peripherals and Misc. Interfaces

- Integrated general purpose timers, counters, DMA controllers
- Integrated RTC with battery backup option
- 24 MHz and 32 KHz crystal oscillator input
- Embedded debug interface using ICE/JTAG

Power Management

- Multiple external power domains controlled by PMIC
- Multiple internal power domains controlled by software
- Multiple sleep modes for CPU, system, DRAM, etc.
- Multiple internal PLLs for DVFS operation
- Multi-voltage I/O design for 1.8V and 3.3V
- Power management auxiliary processor in a dedicated always-on (AO) power domain that can communicate with external PMIC

Security

- Trustzone based Trusted Execution Environment (TEE)
- Secured boot, OTP, internal control buses and storage
- Protected memory regions and scrambled memory data interface
- Trusted Video Path and Secured (needs SecureOS software)

Package

- LFBGA, 12x12mm,306-ball, 0.65 ball pitch, RoHS compliant

3. CPU AND GPU SUBSYSTEM

3.1. Overview

The Cortex™-A5 MP subsystem of the chip is a high-performance, low-power, ARM macrocell with an L1 cache subsystem that provides full virtual memory capabilities. The Cortex-A5 processor implements the ARMv7 architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions. The developers can follow the ARM official reference documents for programming details.

The Cortex-A5 processor features are:

- in-order pipeline with dynamic branch prediction
- ARM, Thumb, and ThumbEE instruction set support
- TrustZone security extensions
- Harvard level 1 memory system with a Memory Management Unit (MMU)
- 64-bit AXI master interface
- ARM v7 debug architecture
- trace support through an Embedded Trace Macrocell (ETM) interface
- Intelligent Energy Manager (IEM) support with
  - asynchronous AXI wrappers
  - two voltage domains
- Media Processing Engine (MPE) with NEON technology

The Mali-450 MP GPU is a hardware accelerator for 2D and 3D graphics system which compatible with the following graphics standards: OpenGL ES 2.0, OpenGL ES 1.1, OpenVG 1.1, EGL 1.5. The developers can follow the ARM and Khronos official reference documents for programming details.

4. MEMORY MAP

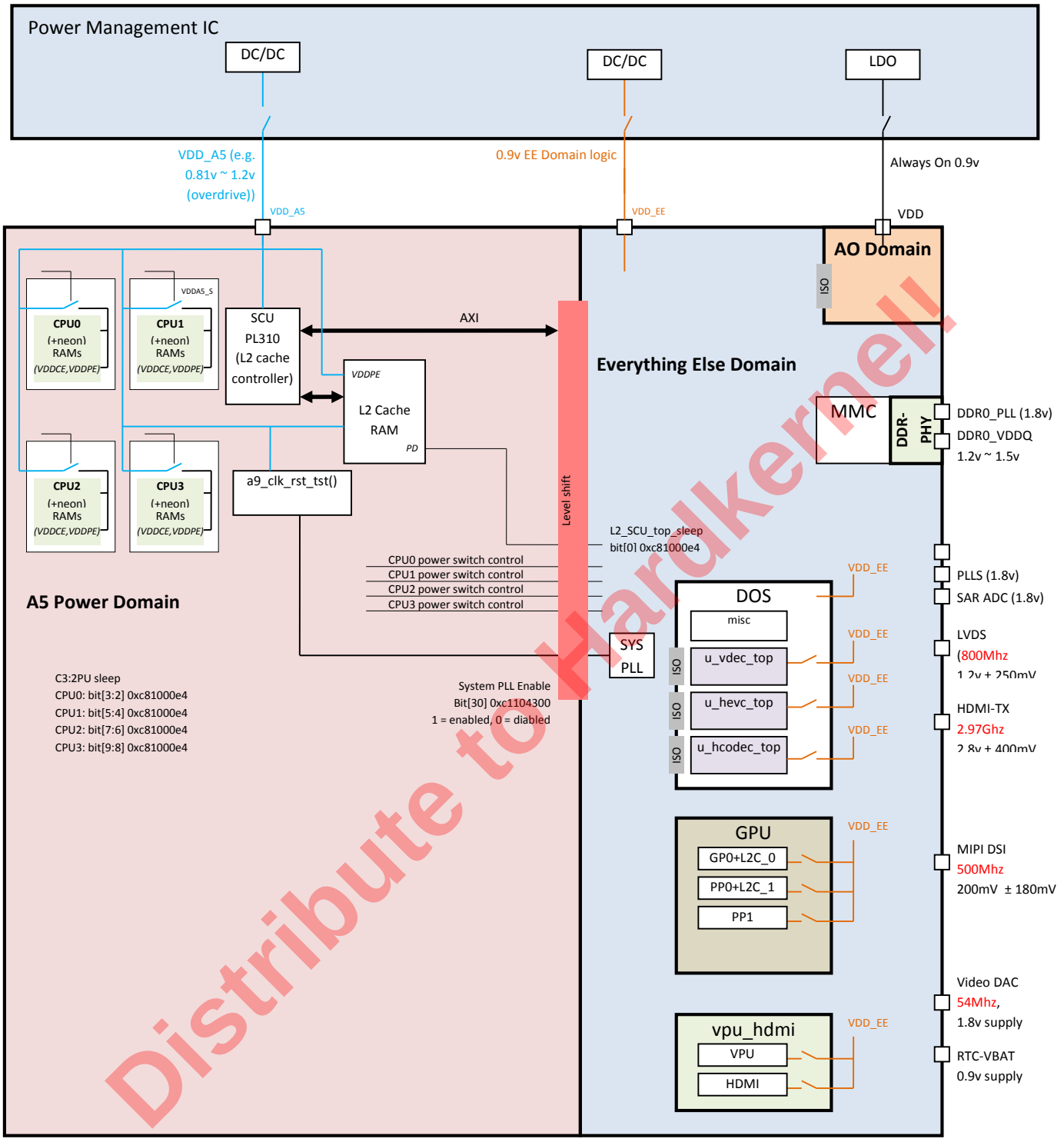
Start	End	Region (Boot)	Region (Normal)
0x00000000	0x00FFFFFF	Boot ROM	DDR
0x01000000	0xBFFFFFFF	DDR	DDR
0xC0000000	0xC12FFFFF	APB3 CBUS	APB3 CBUS
0xC1300000	0xC13FFFFF	AXI	AXI
0xC4200000	0xC4200FFF	L2 Cache regs	L2 Cache regs
0xC4300000	0xC430FFFF	A9 Periph base	A9 Periph base
0xC8000000	0xC8003FFF	DDR APB	DDR APB
0xC8006000	0xC8007FFF	DMC APB	DMC APB
0xC8100000	0xC81FFFFF	RTI	RTI
0xC9040000	0xC90BFFFF	USB0	USB0
0xC90C0000	0xC90FFFFF	USB1	USB1
0xC9410000	0xC941FFFF	Ethernet slave	Ethernet slave
0xCC000000	0xCFFFFFFF	SPI	SPI
0xD0000000	0xD003FFFF	A9_DAPLITE	A9_DAPLITE
0xD0042000	0xD0043FFF	HDMI TX	HDMI TX
0xD0048000	0xD004FFFF	NAND	NAND
0xD0050000	0xD005FFFF	DOS	DOS
0xD00C0000	0xD00FFFFF	Mali APB	Mali APB
0xD0100000	0xD013FFFF	VPU	VPU
0xD0150000	0xD015FFFF	MIPI	MIPI
0xD9000000	0xD903FFFF	AHB SRAM	AHB SRAM
0xD9040000	0xD904FFFF	Boot ROM	Boot ROM
0xDA000000	0xDA001FFF	Efuse	Efuse
0xDA002000	0xDA003FFF	SEC MMC regs	SEC MMC regs
0xDA004000	0xDA005FFF	SEC Cntl Regs	SEC Cntl Regs
0xDA006000	0xDA007FFF	BLKMV	BLKMV
0xE0000000	0xFFFFFFFF	DDR	DDR

5. POWER DOMAIN

5.1. Overview

The chip has three large power domains named: Always On (AO), Everything Else (EE) and A5. Within the EE domain are several sub-power domains:

- Always On: Single domain with no sub-domains. Powered externally
- Everything Else domain: Powered externally with internal sub-domains
- A5 Domain: Powered externally with level shifters for overvoltage driving. Contains 6 power domains
  - CPU0, CPU1, CPU2, CPU3: independently switched domains
  - SCU/L2 Cache controller: Powered / switched externally
  - L2 cache RAM: switched internally powered by the EE domain voltage (either through a pad or from the core)



5.2. Top Level Power Domains

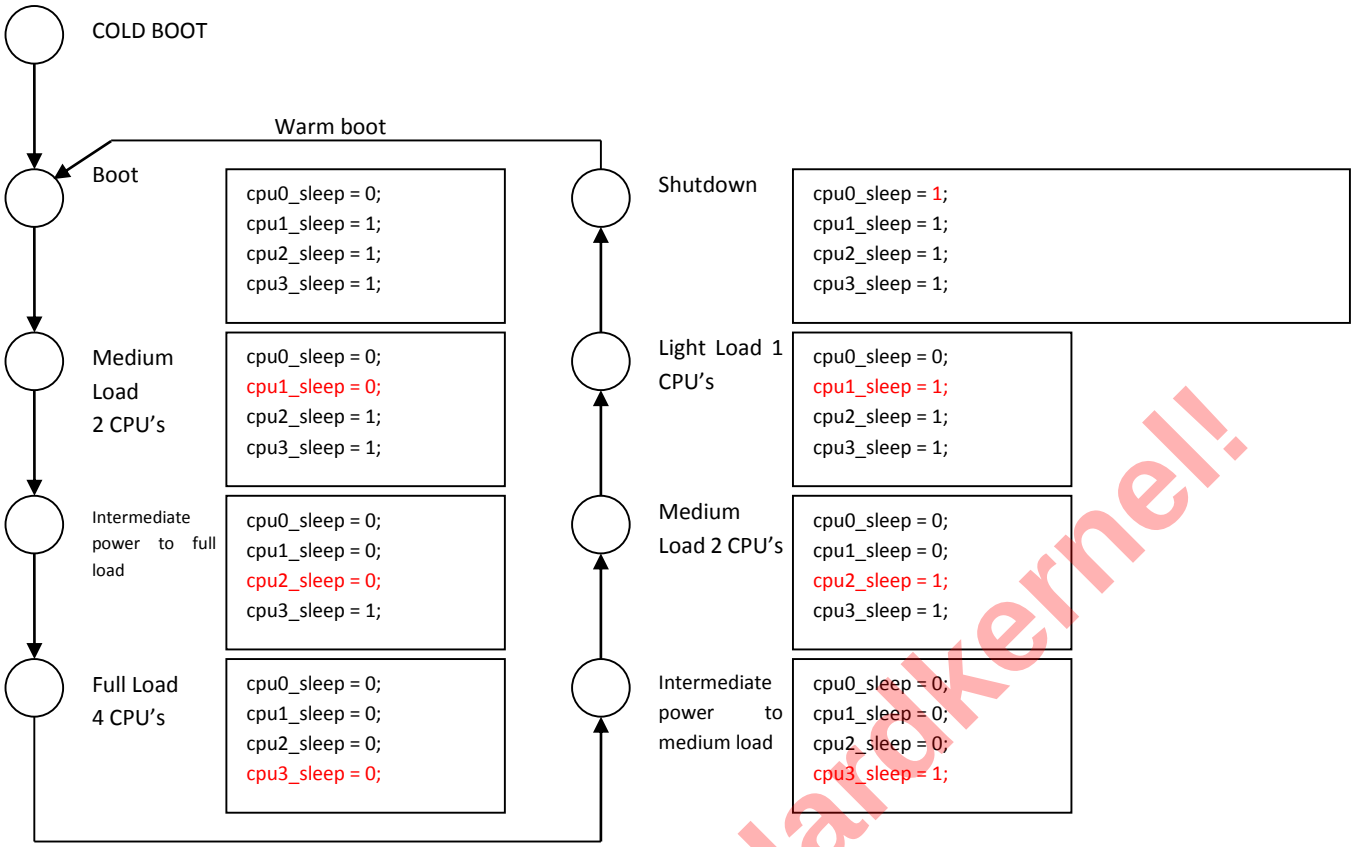
The power supplies for the different domains must follow a specific power supply order: The A5 can't be powered without the EE domain. The EE domain can't be powered up without the AO domain. If you read the table left to right then right to left, that's essentially the power up/down sequence for the entire chip.

	Always On	EE Domain		A5 domain				
	Logic	EE Logic	Mali and DOS and VPU	A5 (SCU/L2)	L2 Cache	CPU0	CPU1	CPU2/3
STATE 0 All off	Off	The EE domain must be OFF if the AO domain is off		The A5 domain must be off if the EE domain is OFF				
STATE 1 Hibernate: Only the Always on Domain is powered	ON	Off	The Mali must be OFF is if the EE domain is off	The A5 domain must be off if the EE domain is OFF				
STATE 2 Always On/EE only (for example audio applications or simple video applications that don't need the A5)	On	On	On or off as needed	Off	Off	Off	Off	Off
STATE 3 Single PU	On	On	On or off as needed	All three must be enabled			Off	Off
STATE 4 2 CPU's	On	On	On or off as needed	All three must be enabled			On	Off
STATE 5 4 CPUs	On	On	On or off as needed	All three must be enabled			On	On

5.3. A5 Power Modes

The A5 domain is the last to power up and the first to power down. The A5 domain itself consists of a quad (4) CPU, an L2 cache controller and an SCU. The A5 CPU boots with the SCU/L2 powered and CPU0 powered. After CPU0 boots, subsequent CPU's can be enabled and disabled independently of one another using the control bits described below. The most likely scenario would be for the CPU0 to be used for low load conditions, CPU0 and CPU1 to be used for medium load conditions and CPU0,1,2 and CPU3 to all be used for heavy load conditions. It's unlikely we will run a 3 CPU configuration (although it's not precluded by the hardware). The flow-diagram below illustrates the transition to each legal state.

	CPU0	CPU1	CPU2	CPU3
Domain Power Sleep bit (1 = power off)	Bit[2] of 0xC81000E4	Bit[4] of 0xC81000E4	Bit[6] of 0xC81000E4	Bit[8] of 0xC81000E4
Domain Power Acknowledge bit	Bit[16] of 0xC81000E4	Bit[17] of 0xC81000E4	Bit[18] of 0xC81000E4	Bit[19] of 0xC81000E4
Input Signal isolation bit (1 = isolated)	Bit[0] of 0xC81000E0	Bit[1] of 0xC81000E0	Bit[2] of 0xC81000E0	Bit[3] of 0xC81000E0
Output signal isolation bit (1 = isolated)	Bit[0] of 0xC81000E0	Bit[1] of 0xC81000E0	Bit[2] of 0xC81000E0	Bit[3] of 0xC81000E0



5.4. EE Top Level Power Modes

EE domain is powered off by DC-DC. VDD_EE also share the same VDD with VDDCE of A5 memory arrays. So if EE domains is shut off, A5 memory is also shut off. That does not matter. Before EE power domain is shut off, A5 should be shut off at first.

	Register/Bit
Ethernet Memory PD	Bits[3:2] of 0xC1104100 (CBUS HHI_MEM_PD_REG0) 0x3 = power off, 0x0 = normal operation
Audio DSP Memory PD	Bits[1:0] of 0xC1104100 (CBUS HHI_MEM_PD_REG0) 0x3 = power off, 0x0 = normal operation
USB0 SRAM memory PD	Bit[13:12] of 0xC1108800 0x3 = power off, 0x0 = normal operation
USB1 SRAM memory PD	Bit[13:12] of 0xC1108820 0x3 = power off, 0x0 = normal operation

5.5. Mali Power Modes

The Mali block sits within the EE domain and the Mali module itself has 4 distinct power domains:

- GP + L2C_0
- PP0 + L2C_1
- PP12
- PP456 + L2C_2

We will power these up in sequence so as to reduce the surge currents. The following combinations are possible.

Mali AXI Cores	1PP Power up/down	3PP Power up/down	4PP Power up/down	6PP Power up/down	3PP Power up/down
GP + L2C_0	1st / last	1st / last	1st / last	1st / last	1st / last
PP0 + L2C_1	2nd / 1st	2nd / 2nd	2nd/ 2nd	2nd/ 3rd	
PP12		3rd / 1st		3rd / 2nd	
PP456 + L2C_2			3rd / 1st	4th / 1st	2nd/ 1st

5.6. Power/Isolation/Memory Power Down Register Summary

5.6.1. A5 Isolation 0xc81000e0

Bit(s)	R/W	Default	Description
31~30	R	-	CPU3 CTRL_MODE set by the CPU
29~28	R	-	CPU3 CTRL_MODE set by the CPU
27~26	R	-	CPU3 CTRL_MODE set by the CPU
25~24	R	-	CPU3 CTRL_MODE set by the CPU
23~22	R/W	11	CPU3 CTRL_MODE
21~20	R/W	11	CPU2 CTRL_MODE
19~18	R/W	11	CPU1 CTRL_MODE
17~16	R/W	00	CPU0 CTRL_MODE (set to tell the CPU which mode to enter)
15~14	R/W	0	Reserved
13	R/W	0	A5 Pwr top level clamp
12	R/W	0	SCURAM Clamp
11~8	R/W	0xE	NEON[3:0] Clamp
7~4	R/W	0xE	CPURAM[3:0] clamp
3~0	R/W	0xE	CPU[3:0] clamp

5.6.2. A5 Power 0xc81000e4

Bit(s)	R/W	Default	Description
31~20	R	0	Reserved
19	R	-	CPU3 Sleep status: 1 = powered down
18	R	-	CPU2 Sleep status: 1 = powered down
17	R	-	CPU1 Sleep status: 1 = powered down
16	R	-	CPU0 Sleep status: 1 = powered down
15~10	R/W	00	Reserved
9~8	R/W	11	CPU3 sleep: 1 = powered down
7~6	R/W	11	CPU2 sleep: 1 = powered down
5~4	R/W	11	CPU1 sleep: 1 = powered down
3~2	R/W	00	CPU0 sleep: 1 = powered down
1-0	R/W	10	Reserved

5.6.3. A5 RAM Power Down Control 0xc81000f4

Bit(s)	R/W	Default	Description
31-28	R/W	0xF	CPU1 RAM power down (Each bit controls different RAMs): 1 = powered down
27-24	R/W	0xF	CPU2 RAM power down (Each bit controls different RAMs): 1 = powered down
23-20	R/W	0xF	CPU3 RAM power down (Each bit controls different RAMs): 1 = powered down
19-18	-	0	Reserved
17~0	R/W	0x00000	L2 RAM power down (Each bit controls different RAMs): 1 = powered down

5.6.4. A5 RAM Power Down Control (cont.) 0xc81000f8

Bit(s)	R/W	Default	Description
31-4	-	0	Reserved
3~0	R/W	0x0	CPU0 RAM power down (Each bit controls different RAMs): 1 = powered down

5.6.5. A5 SYS PLL 0xc1104300

PLL Enable = Bit[30]: 1 = pll enabled, 0 = pll disabled

5.6.6. Always On domain PWR_CNTL1 0xc810000c

Bit(s)	R/W	Default	Description
31~4	R/W	0	Reserved
3	R/W	0	DDR1PHYIO_RET_EN
2	R/W	0	DDR1PHYIO_REG_EN_N
1	R/W	0	DDR0PHYIO_RET_EN
0	R/W	0	DDR0PHYIO_REG_EN_N

5.6.7. Always On domain PWR_CNTL0 0xc8100010

Bit(s)	R/W	Default	Description
31~28	R/W	0	Reserved
27~22	R/W	0	reserved
21~20	R/W	0	AHB SRAM Memory power down: 00 = normal operation, 10 = power down. Other conditions reserved
19~16	R/W	0	Reserved
15~14	R/W	0	Reserved
13~12	R/W	0	32khz DS:
11~10	R/W	0	Alternate 32khz input clock select from GPIO pad
9	R/W	1	Reset to the EE domain. 0 = reset, 1 = normal operation
8	R/W	0	RTC oscillator input select: 1 = use RTC clock as clock. 0= used clk81 as the clock
7~5	R/W	0	reserved
4	R/W	0	EE domain isolation In
3	R/W	0	EE domain Isolation out
2	R/W	0	Reserved
1	R/W	0	Always on RESET isolation: 1 = isolated
0	R/W	0	Reserved

5.6.8. General Power gen_pwr_sleep_cntl0 0xc81000e8

Bit(s)	R/W	Default	Description
31~10	R/W	0	Reserved
9	R/W	0	VPU/HDMI Isolation
8	R/W	0	VPU/HDMI power: 1 = powered off
7~6	R/W	0	Reserved
5	R/W	1	Reserved
4	R/W	1	Reserved
3	R/W	1	Reserved
2	R/W	1	Reserved
1	R/W	1	Reserved
0	R/W	1	Reserved

5.6.9. General Isolation gen_pwr_iso_cntl0 0xc81000ec

Bit(s)	Description
31~10	Reserved
9	Reserved
8	Reserved
7	Reserved
6	Reserved
5	Reserved
4	Reserved
3	GPU ISO OUT
2	GPU ISO IN
1~0	reserved

5.6.10. General Acknowledge gen_pwr_sleep_ack0 0xc81000f0

Bit(s)	Description
31~9	Reserved
8	Reserved
7	MALI_PWRUP_ACK: PP456 acknowledge: 1 = powered off
6	MALI_PWRUP_ACK: PP12 acknowledge: 1 = powered off
5	MALI_PWRUP_ACK: PP0 acknowledge: 1 = powered off
4	MALI_PWRUP_ACK: GP acknowledge: 1 = powered off
3	Reserved
2	Reserved
1	Reserved
0	Reserved

5.6.11. Mali Power UP (domains_npwr_up) 0xd00c2000

Bit(s)	Description
31~3	Reserved
3	MALI PP456 power up: Write a 1 to power up
2	MALI PP12 power up
1	MALI PP0 power up
0	MALI GP power up

5.6.12. Mali Power Down (domains_npwr_up) 0xd00c2004

Bit(s)	Description
31~3	Reserved
3	MALI PP456 power down: Write a 1 to power down
2	MALI PP12 power down
1	MALI PP0 power down
0	MALI GP power down

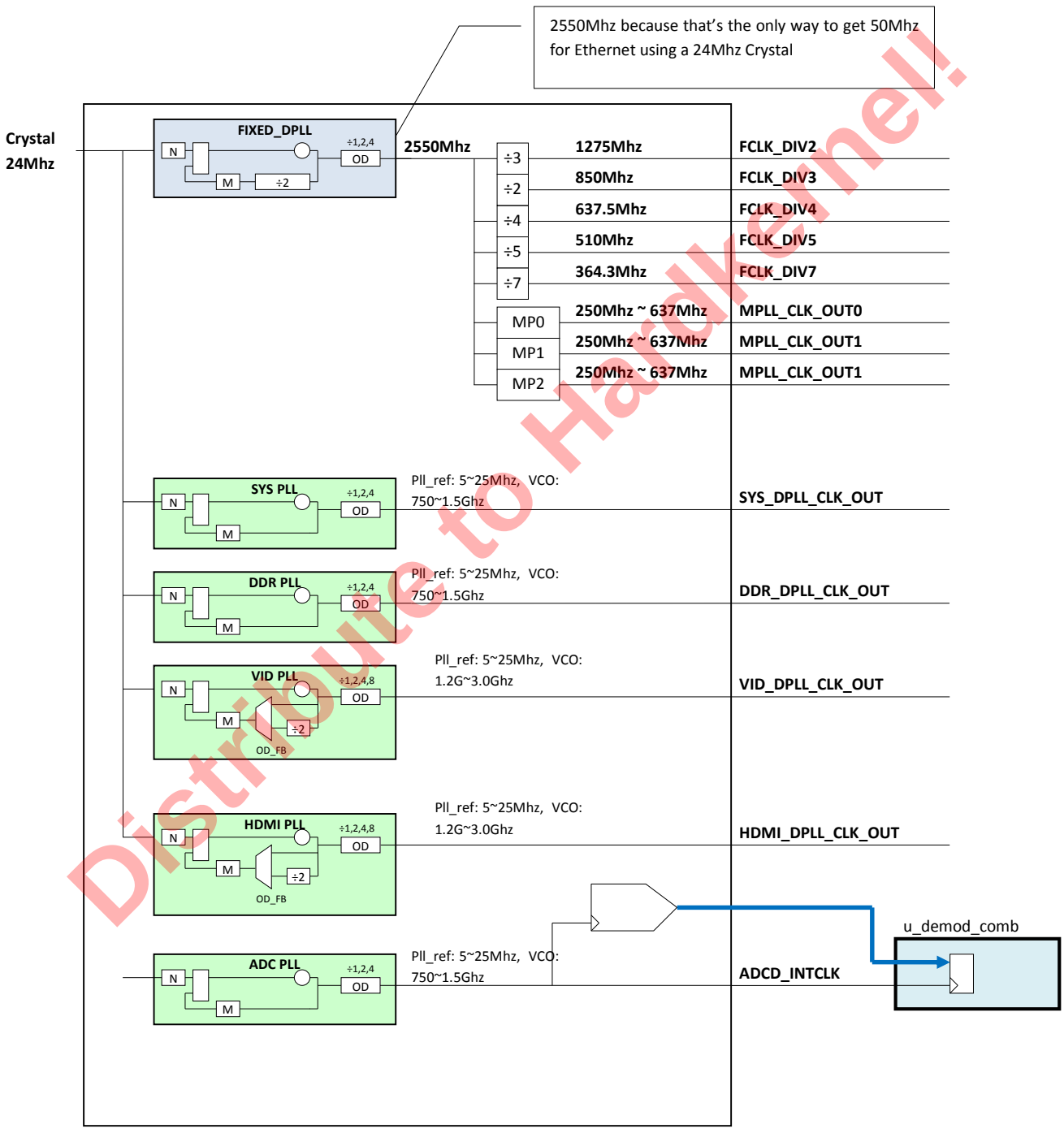
6. CLOCK AND RESET UNIT

6.1. Overview

The clock and reset unit is an APB slave module that is designed for generating all of the internal and system clocks, resets of chip.

6.2. Clock Trees

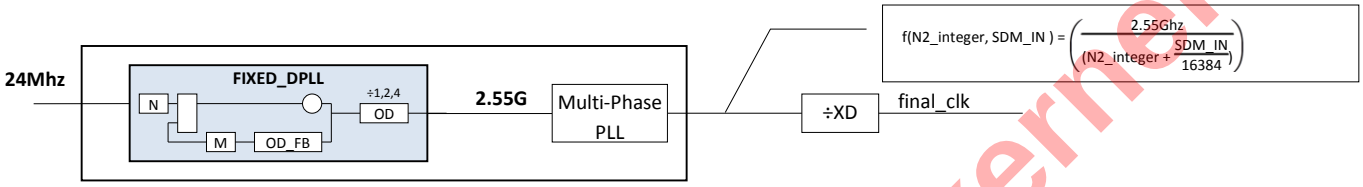
Figure 1. PLLs



Integer PLL Parameters

Integer PLL	F-Ref Min (XTAL/N)	F-Ref Max (XTAL/N)	N-bits	M-bits	OD Bits	VCO MIN	VCO MAX	OD divider
SYS_DPLL	5 Mhz	30 Mhz	5	9	2	1.25Ghz	2.5Ghz	2-bits: ÷1, ÷2, ÷4 50% duty cycle with OD = 0 (divide by 1)
DDR_DPLL	5 Mhz	30 Mhz	5	9	2	750Mhz	1.5Ghz	2-bits: ÷1, ÷2, ÷4
VID_DPLL	5 Mhz	30 Mhz	5	9	2	1.2Ghz	3.0Ghz	2-bits: ÷1, ÷2, ÷4, ÷8
HDMI_DPLL (VIID_DPLL)	5 Mhz	30 Mhz	5	9	2	1.2Ghz	3.0Ghz	OD: ÷1, ÷2, ÷4, ÷8 FB: ÷1, ÷2 LVDS: ÷1, ÷2, ÷4, ÷8
FIXED_DPLL	5 Mhz	30 Mhz	5	9	1	1Ghz	2Ghz	

Multi-Phase PLL Parameters



Multi-Phase	N2_integer [Min]	N2_integer [Max]	SDM_IN [Min]	SDM_IN [Max]	Fout [Min]	Fout [Max]
MP0_DPLL	4	127	1	16383	317.75 Mhz	637.5Mhz
MP1_DPLL	4	127	1	16383	317.75 Mhz	637.5Mhz
MP2_DPLL	4	127	1	16383	317.75 Mhz	637.5Mhz

Direct Divide Frequencies (assuming 2.550Ghz fixed source clock)

Target Frequency	Effective Divider from 2.550Ghz	PLL_DIV3 850Mhz	PLL_DIV4 637.5	PLL_DIV5 510	PLL_DIV7 364Mhz
850.00	3	÷1			
637.50	4		÷1		
510.00	5			÷1	
425.00	6	÷2			
364.29	7				÷1
318.75	8		÷2		
283.33	9	÷3			
255.00	10			÷2	
212.50	12	÷4	÷3		
182.14	14				÷2
170.00	15	÷5		÷3	
159.38	16		÷4		
141.67	18	÷6			
127.50	20		÷5	÷4	
121.43	21	÷7			÷3
106.25	24	÷8	÷6		
102.00	25			÷5	
94.44	27	÷9			
91.07	28		÷7		÷4
85.00	30	÷10		÷6	
79.69	32		÷8		
77.27	33	÷11			
72.86	35			÷7	÷5
70.83	36	÷12	÷9		
65.38	39	÷13			
63.75	40		÷10	÷8	÷6
60.71	42	÷14			

57.95	44		÷11		
56.67	45	÷15		÷9	
53.13	48	÷16	÷12		
52.04	49				÷7
51.00	50			÷10	
50.00	51	÷17			

Audio Frequencies (using Multi-phase PLL)

Sampling Frequency	Over Sampling	Target freq [Mhz]	N2_integer	SDM_IN	Divider in clk_rst_tst	final_clk [Mhz]	Error
7875 Hz	256	2.016	7	13296	127	2.016	0.00%
7875 Hz	384	3.024	8	9655	77	3.024	0.00%
8000 Hz	256	2.048	7	13312	125	2.048	0.00%
8000 Hz	384	3.072	7	9343	86	3.072	0.00%
11025 Hz	256	2.8224	8	2376	87	2.822	0.00%
11025 Hz	384	4.2336	7	12197	61	4.234	0.00%
12000 Hz	256	3.072	7	9343	86	3.072	0.00%
12000 Hz	384	4.608	6	7832	67	4.608	0.00%
16000 Hz	256	4.096	4	14464	100	4.096	0.00%
16000 Hz	384	6.144	7	9343	43	6.144	0.00%
22050 Hz	256	5.6448	4	1188	87	5.645	0.00%
22050 Hz	384	8.4672	4	1188	58	8.467	0.00%
24000 Hz	256	6.144	7	9343	43	6.144	0.00%
24000 Hz	384	9.216	4	1550	53	9.216	0.00%
32000 Hz	256	8.192	4	14464	50	8.192	0.00%
32000 Hz	384	12.288	7	1254	23	12.288	0.00%
44100 Hz	256	11.2896	4	3571	42	11.290	0.00%
44100 Hz	384	16.9344	4	1188	29	16.934	0.00%
48000 Hz	256	12.288	7	1254	23	12.288	0.00%
48000 Hz	384	18.432	4	2840	26	18.432	0.00%
96000 Hz	256	24.576	6	4260	13	24.576	0.00%
96000 Hz	384	36.864	4	2840	13	36.864	0.00%
192000 Hz	256	49.152	4	8538	9	49.152	0.00%
192000 Hz	384	73.728	4	8538	6	73.728	0.00%

Ethernet Frequencies:

- 10/100 Ethernet: 50Mhz
- 10/100/1000 Ethernet: 125Mhz and 25Mhz

USB Frequencies:

- 19.2, 20, 24, 50Mhz

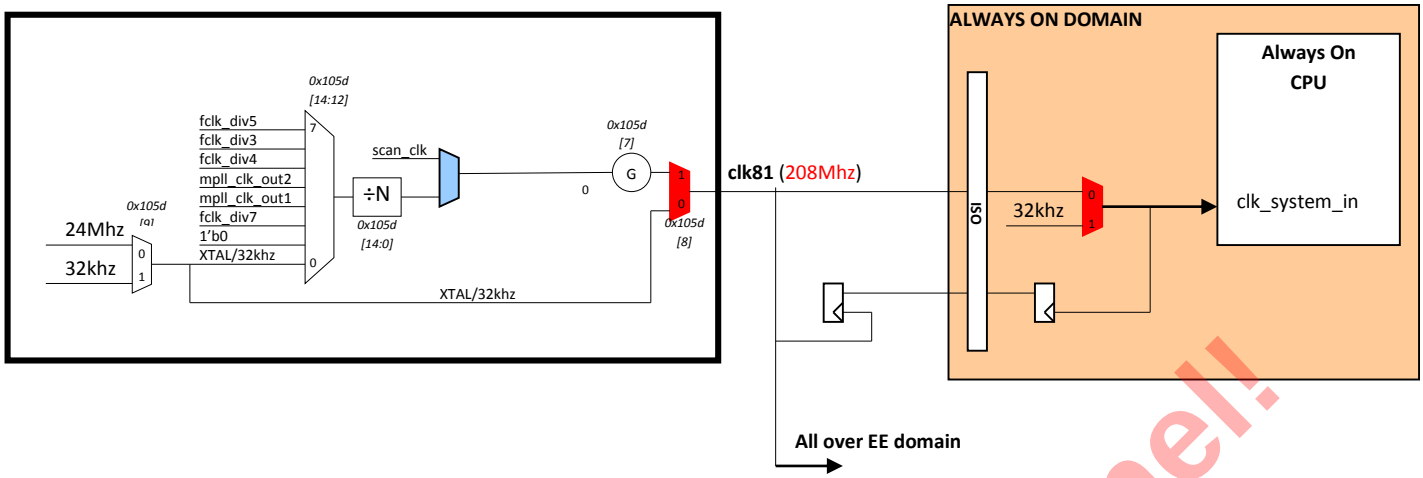
HDMI Frequencies:

- Min: 250Mhz (at the PHY) 25Mhz to the logic
- Max: 2.97Ghz at the PHY (4k2k)

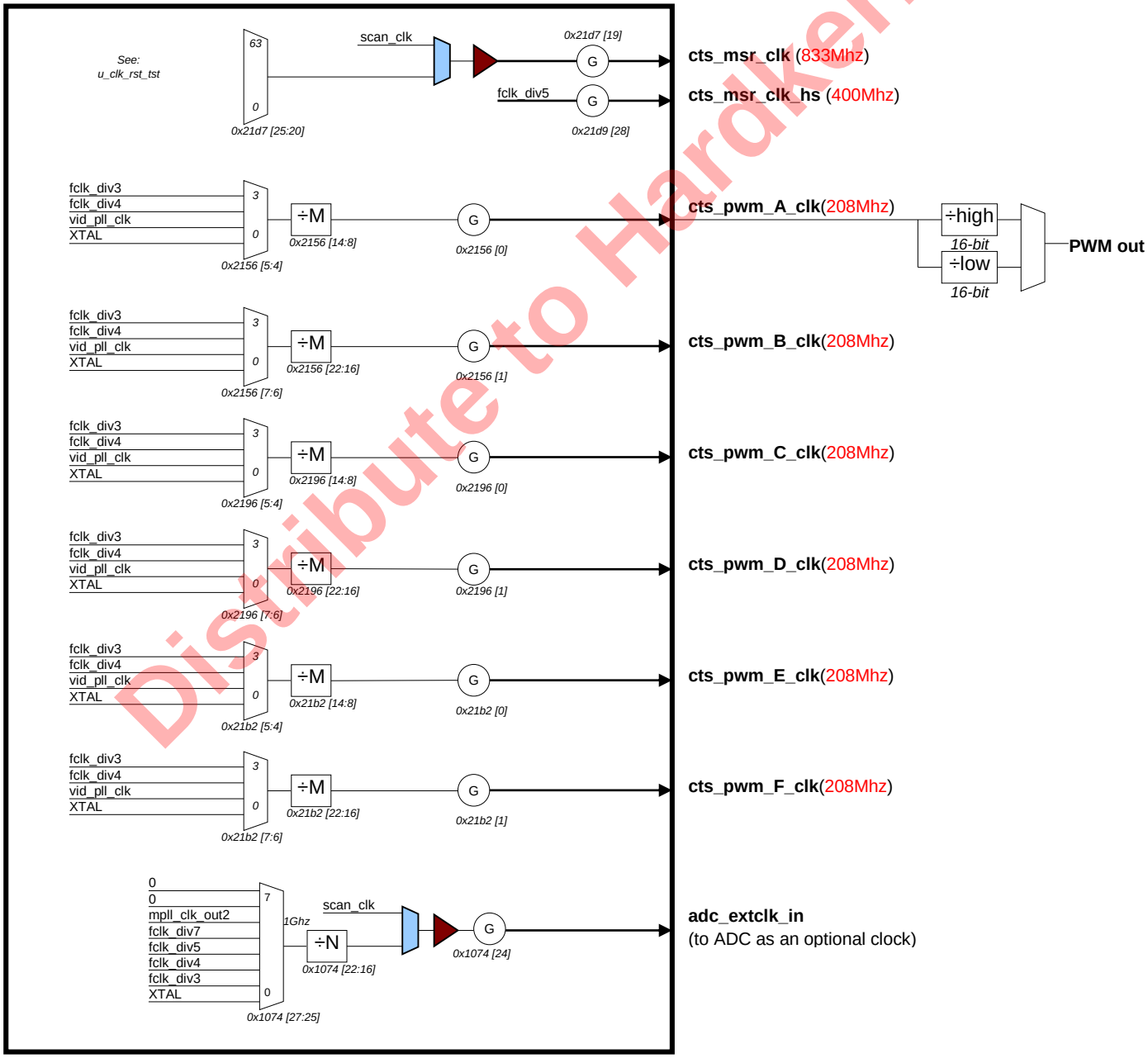
Demod ADC frequencies:

- 28.8 MHz (china cable, dvbt, isdbt)
- 25.175 MHz, (US atsc, cable, demod digital must be 75.375MHz)
- 20.833 (some tuners for dvbt, isdbt),
- 45 (some tuners for dvbt, isdbt).

clk_rst_tst()



clk_rst_tst()



6.3. Clock Gating

Modules and sub-modules within the chip can be disabled by shutting off the clock. The control for these clocks comes from six CBUS registers that collectively make up a 64-bit register that controls the MPEG_DOMAIN and a 32-bit register that controls the OTHER_ DOMAIN. The table below indicates the bits associated with either the MPEG_DOMAIN and OTHER_DOMAIN gated clock enables. The table is organized by function rather than by bit order because it makes it easier to determine how to turn on/off a particular function within the chip. If a bit is set high, the clock is enabled. If a bit is set low, the clock is turned off and the module is disabled.

6.3.1. AO Domain Clock Gating

CBUS 0x1055 control bits

Bit	Module Description
31-4	Unused
3	Always on Registers, timers,…
2	AHB bus in the always on domain
1	AHB SRAM
0	Media CPU

6.3.2. EE Domain Clock Gating

Module	CBUS 0x1050	CBUS 0x1051	CBUS 0x1052	CBUS 0x1054	Module Description
	31				
PERIPHS	30				SPI Interface
	29				
	28				
	27				
	26				
	25				
	24				
ASSIST	23				ASSIST_MISC
	22				unused
	21				
	20				
HIU	19				HIU Registers
ABUF	18				Audio Buffer u_abuf_top.HCLK
PERIPHS	17				SDIO
PERIPHS	16				ASYNC_FIFO
PERIPHS	15				STREAM Interface
PERIPHS	14				SDHC
PERIPHS	13				UART0
PERIPHS	12			21	Random Number generator
PERIPHS	11				Smart Card
PERIPHS	10				SAR ADC
PERIPHS	9				I2C Master / I2C SLAVE
PERIPHS	8				SPICC
PERIPHS	7				PERIPHS module top level (there are separate register bits for internal blocks)
DDR	6				PL301 (AXI Matrix) to CBUS
ISA	5				ISA module
	4				
	3				

	2				
DOS (MDEC)	1				u_dos_top()
DDR	0				DDR Interfaces and bridges
MISC		31			ROM BOOT ROM clock
MISC		30			EFUSE logic
MISC		29			AHB ARB0
VIU		28			VDIN1
		27			
USB		26			USB General
DOS		25			U_parser_top()
PERIPHS		24			NAND Interface
MISC		23			RESET
USB		22			USB 1
USB		21			USB 0
GE2D		20			General 2D Graphics Engine
		19			
		18			
		17			
PERIPHS		16			UART1
AIU		15			AIU Top level (there is internal gating shown below)
BLKMV		14			Block move core logic
AIU		13			ADC
AIU		12			Mixer Registers: u_ai_top.u_mixer_reg.clk
AIU		11			Mixer: u_ai_top.u_aud_mixer.clk
AIU		10			AIFIFO2: u_ai_top.u_aififo2.clk
AIU		9			AMCLK measurement circuit
AIU		8			I2S Out: This bit controls the clock to the logic between the DRAM control unit and the FIFO's that transfer data to the audio clock domain. (u_ai_top.i2s_fast.clk)
AIU		7			IEC958: iec958_fast()
AIU		6			AIU – ai_top_glue u_ai_top.ai_top_glue.clk u_ai_top.fifo_async_fast_i2s.clk u_ai_top.fifo_async_fast_958.clk
		5			
DEMUX		4			Set top box demux module u_stb_top.clk
ETHERNET		3			Ethernet core logic
AUDIN		2			I2S / SPDIF Input
		1			
		0			
			31		
			30		
MISC			29		CLK81 to the A5 domain
			28		
			27		
AHB			26		Secure AHB to APB3 Bridge
VPU			25		VPU Interrupt
			24		
			23		
PERIPHS			22		SANA
			21		
			20		
			19		
			18		
			17		
			16		
PERIPHS			15		UART 2
			13		

MISC			12		DVIN
DDR			11		MMC PCLK
			10		
USB			9		USB0 to DDR bridge
USB			8		USB1 to DDR bridge
			7		
			6		
			5		
HDMI			4		HDMI PCLK
HDMI			3		HDMI interrupt synchronization
MISC			2		AHB control bus
MISC			1		AHB data bus
			0		
EDP				31	Video clock enable
				30	
				29	
				28	
				27	
VENC				26	VCLK2_OTHER
VENC				25	VCLK2_VENCL
VENC				24	VCLK2_VENCL MMC Clock All
				23	
VENC				22	gclk_vencl_int
PERIPH				21	Random Number Generator
VENC				20	ENC480P
				19	
				18	
				17	
AIU				16	IEC958_GATE
				15	
AIU				14	AOCLK_GATE
				13	
				12	
				11	
VENC				10	DAC_CLK
VENC				9	gclk_vencp_int
VENC				8	gclk_venci_int
				7	
				6	
				5	
VENC				4	VCLK2_VENCP
VENC				3	VCLK2_VENCP
VENC				2	VCLK2_VENCI for venc_i_top other normal function
VENC				1	VCLK2_VENCI for venc_i_top BIST function
				0	

6.4. Register Description

6.4.1. RESET_REGISTER 0x1101

Bit(s)	R/W	Default	Description
15	R/W	0	VIFIFO
14	R/W	0	VLD_PART
13	R/W	0	MDEC
12	R/W	0	AIFIFO2
11	R/W	0	ASSIST
10	R/W	0	Video Encoder
9	R/W	0	PMUX
8	R/W	0	CCPU
7	R/W	0	MCPU
6	R/W	0	AIU
5	R/W	0	VIU
4	R/W	0	-
3	R/W	0	MC
2	R/W	0	IQIDCT
1	R/W	0	VLD
0	R/W	0	HIU

6.4.2. RESET1_REGISTER 0x1102

Bit(s)	R/W	Default	Description
15	R/W	0	ROM BOOT
14	R/W	0	AHB_CNTL
13	R/W	0	AHB_DATA
12	R/W	0	ABUF
11	R/W	0	Ethernet
10	R/W	0	ISA
9	R/W	0	BLKMV (NDMA)
8	R/W	0	PARSER
7	R/W	0	AHB_BRIDGE
6	R/W	0	AHB_SRAM
5	R/W	0	BT656
4	R/W	0	VDAC
3	R/W	0	DDR
2	R/W	0	USB_OTG
1	R/W	0	DEMUX
0	R/W	0	-

6.4.3. RESET2_REGISTER 0x1103

Bit(s)	R/W	Default	Description
15	R/W	0	Hdmi system reset
14	R/W	0	MALI
13	R/W	0	MEDIA CPU
12	R/W	0	Audio APB
11	R/W	0	HDMI APB
10	R/W	0	parser_top
9	R/W	0	parser_ctl
8	R/W	0	Paser_fetch
7	R/W	0	Parser_reg
6	R/W	0	GE2D
5	R/W	0	NAND
4	R/W	0	PSC
3	R/W	0	PIC_DC
2	R/W	0	DBLK
1	R/W	0	AUDIN

Bit(s)	R/W	Default	Description
0	R/W	0	VD_RMEM

6.4.4. RESET3_REGISTER 0x1104

Bit(s)	R/W	Default	Description
15	R/W	0	Demux reset 2
14	R/W	0	Demux reset 1
13	R/W	0	Demux reset 0
12	R/W	0	Demux S2P 1
11	R/W	0	Demux S2p 0
10	R/W	0	Demux DES
9	R/W	0	Demux top
8	R/W	0	Audio DAC
7	R/W	0	SYS CPU
6	R/W	0	AHB BRIDGE CNTL
5	R/W	0	Audio PLL modulator
4	R/W	0	AIFIFO
3	R/W	0	SYS CPU BVCI
2	R/W	0	EFUSE
1	R/W	0	SYS CPU
0	R/W	0	Ring oscillator

6.4.5. RESET4_REGISTER 0x1105

Bit(s)	R/W	Default	Description
15	R/W	0	
14	R/W	0	
13	R/W	0	VENCL
12	R/W	0	VDI6
11	R/W	0	A5 DEBUG
10	R/W	0	RTC
9	R/W	0	VDAC
8	R/W	0	VENCT
7	R/W	0	VENCP
6	R/W	0	VENCI
5	R/W	0	RDMA
4	R/W	0	DVIN
3	R/W	0	A5
2	R/W	0	A5 AXI
1	R/W	0	A5 APB
0	R/W	0	PL310

6.4.6. RESET5_REGISTER 0x1106

Bit(s)	R/W	Default	Description
15	R/W	0	
14	R/W	0	
13	R/W	0	
12	R/W	0	
11	R/W	0	
10	R/W	0	
9	R/W	0	
8	R/W	0	
7	R/W	0	
6	R/W	0	
5	R/W	0	VID2_PLL
4	R/W	0	AUDIO PLL
3	R/W	0	HPLL PLL
2	R/W	0	SYS PLL

Bit(s)	R/W	Default	Description
1	R/W	0	MISC PLL
0	R/W	0	DDR PLL

6.4.7. RESET6_REGISTER 0x1107

Bit(s)	R/W	Default	Description
15	R/W	0	PERIPHS: LED PWM
14	R/W	0	PERIPHS: SPI 1
13	R/W	0	PERIPHS: SPI 0
12	R/W	0	PERIPHS: Async 1
11	R/W	0	PERIPHS: Async 0
10	R/W	0	PERIPHS: UART 1
9	R/W	0	PERIPHS: UART 0
8	R/W	0	PERIPHS: SDIO
7	R/W	0	PERIPHS: Stream Interface
6	R/W	0	PERIPHS: I2C Slave
5	R/W	0	PERIPHS: I2C Master 1
4	R/W	0	PERIPHS: I2C Master 0
3	R/W	0	PERIPHS: SAR ADC
2	R/W	0	PERIPHS: Smart Card
1	R/W	0	PERIPHS: IR Remote
0	R/W	0	PERIPHS: General

6.4.8. RESET0_MASK 0x1110

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.9. RESET1_MASK 0x1111

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.10. RESET2_MASK 0x1112

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.11. RESET3_MASK 0x1113

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.12. RESET4_MASK 0x1114

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.13. RESET5_MASK 0x1115

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.14. RESET6_MASK 0x1116

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.15. RESET7_MASK 0x1117

The bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

6.4.16. CRT_MASK 0x1117

Bit(s)	R/W	Default	Description
31-2	R	0	Unused
1	R/W	0	**SYS CPU Mask**: If this bit is set to 1, then the SYS CPU clock dividers and logic will not be reset when the watchdog reset occurs
0	R/W	0	**CRT Reset Mask**: If this bit is set to 1, then internal clocks of clock reset test are not reset when the watchdog reset occurs

6.4.17. HHI_GCLK_MPEG0 0x1050

Bit(s)	R/W	Default	Description
31-0	R/W	0	Bits [31:0] of the composite MPEG clock gating register

6.4.18. HHI_GCLK_MPEG1 0x1051

Bit(s)	R/W	Default	Description
31-0	R/W	0	Bits [63:32] of the composite MPEG clock gating register

6.4.19. HHI_GCLK_MPEG2 0x1052

Bit(s)	R/W	Default	Description
31-0	R/W	0	Bits [63:32] of the composite MPEG clock gating register

6.4.20. HHI_GCLK_OTHER 0x1054

Bit(s)	R/W	Default	Description
31-0	R/W	0	Bits [31:0] of the composite Other clock gating register

6.4.21. HHI_GCLK_AO 0x1055

Bit(s)	R/W	Default	Description
31-8	R	0	Unused
7-4	R/W	0xF	unused
3	R/W	1	Always On Register logic
2	R/W	1	Always on general clock in the always on domain
1	R/W	1	AHB SRAM
0	R/W	1	Media CPU

6.4.22. HHI_SYS_CPU_CLK_CNTL1 0x1057

Bit(s)	R/W	Default	Description
31	R/W	1	MMC clock enable
30	R/W	0	AXI64to128 bridge (A5-to-MMC) soft reset (MMC interface)
29~20	R/W	0	SCALE_DIV: This value represents an N+1 divider of the input clock.
19	R/W	0	L2_DRAM_CLK_DIS: Set to 1 to manually disable the L2 DRAM clock when changing the mux selection. Typically this bit is set to 0 since the clock muxes can switch without glithces. This is a “just in case” bit
18	R/W	0	AXI_CLK_DIS: Set to 1 to manually disable the AXI clock when changing the mux selection. Typically this bit is set to 0 since the clock muxes can switch without glithces. This is a “just in case” bit
17	R/W	0	PERIPH_CLK_DIS: Set to 1 to manually disable the PERIPH clock when changing the mux selection. Typically this bit is set to 0 since the clock muxes can switch without glithces. This is a “just in case” bit
16	R/W	0	APB_CLK_DIS: Set to 1 to manually disable the APB clock when changing the mux selection. Typically this bit is set to 0 since the clock muxes can switch without glithces. This is a “just in case” bit
15	R/W	0	Reserved
14~12	R/W	0	L2_DRAM_MUX: 0 A5 clock divided by 2 1 A5 clock divided by 3 2 A5 clock divided by 4

Bit(s)	R/W	Default	Description
			3 A5 clock divided by 5 4 A5 clock divided by 6 5 A5 clock divided by 7 **6** A5 clock divided by 8
11~9	R/W	0	AXI_CLK _MUX: 1. A5 clock divided by 2 2. A5 clock divided by 3 3. A5 clock divided by 4 4. A5 clock divided by 5 5. A5 clock divided by 6 6. A5 clock divided by 7 **7.** A5 clock divided by 8
8~6	R/W	0	PERIPH_CLK _MUX: 0 A5 clock divided by 2 1 A5 clock divided by 3 2 A5 clock divided by 4 3 A5 clock divided by 5 4 A5 clock divided by 6 5 A5 clock divided by 7 **6** A5 clock divided by 8
5~3	R/W	0	APB_CLK _MUX: 1. A5 clock divided by 2 2. A5 clock divided by 3 3. A5 clock divided by 4 4. A5 clock divided by 5 5. A5 clock divided by 6 6. A5 clock divided by 7 **7.** A5 clock divided by 8
2-0	R/W	0	Reserved

6.4.23. HHI_VID_CLK_DIV 0x1059

Bit(s)	R/W	Default	Description
31-28	R/W	0	ENCI_CLK_SEL
27-24	R/W	0	ENCP_CLK_SEL
23-20	R/W	0	ENCT_CLK_SEL
19-18	R/W	0	UNUSED
17	R/W	0	CLK_DIV_RESET
16	R/W	0	CLK_DIV_EN
15-8	R/W	0	XD1
7-0	R/W	0	XD0

6.4.24. HHI_MPEG_CLK_CNTL 0x105d

Bit(s)	R/W	Default	Description
31	R/W	0	NEW_DIV_EN: If this bit is set to 1, then bits[30:16] make up the clk81 divider. If this bit is 0, then bits[6:0] dictate the divider value. This is a new feature that allows clk81 to be divided down to a very slow frequency.
30~16	R/W	0	NEW_DIV: New divider value if bit[31] = 1
15	R/W	0	Production clock enable
14-12	R.W	6	MPEG_CLK_SEL (See clock document)
11-10	R/W	0	unused
9	R.W	0	RTC Oscillator Enable: Set this bit to 1 to connect the RTC 32khz oscillator output as the XTAL input for the divider above
8	R/W	0	Divider Mux: 0 = the ARC clock and the MPEG system clock are connected to the 27Mhz crystal. 1 = the ARC clock and the MPEG system clock are connected to the MPEG PLL divider
7	R/W	1	PLL Mux: 0 = all circuits associated with the MPEG PLL are connected to 27Mhz. 1 = all circuits associated with the MPEG PLL are connected to the MPEG PLL
6-0	R/W	0	PLL Output divider. The MPEG System clock equals the video PLL clock frequency divided by (N+1). Note: N must be odd (1,3,5,….) so that the MPEG clock is divided by an even number to generate a 50% duty cycle.

6.4.25. HHI_AUD_CLK_CNTL 0x105e

Bit(s)	R/W	Default	Description
31-26	R/W	0	Unused
25-24	R/W	0	Audio DAC clock select (See clock document)
23	R/W	0	Audio DAC Clock enable
22-16	R/W	0	Audio DAC Clock divider
15-11	R/W	0	Unused
10-9	R/W	0	AMCLK_SRC_SEL: (see clock tree document)
8	R/W	0	PLL Clock Enable. Set this bit to 1 to enable the PLL to the rest of the logic. To avoid glitching state machines inside the chip please change the PLL using the following sequence: • Set bit[8] = 0 to block the logic from the PLL • set the PLL using register 0x15b and wait for the PLL to settle (1mS) • Set bit[8] = 1 to enable the PLL driving all of the logic
7-0	R/W	1	PLL Output divider. The Audio System clock equals the video PLL clock frequency divided by (N+1).

6.4.26. HHI_VID_CLK_CNTL 0x105f

Bit(s)	R/W	Default	Description
31-21	R/W	0	TCON_CLK0_CTRL
20	R/W	0	CLK_EN1
19	R/W	0	CLK_EN0
18-16	R/W	0	CLK_IN_SEL
15	R/W	0	SOFT_RESET
14	R/W	0	PH23_ENABLE
13	R/W	0	DIV12_PH23
12-5	R/W	0	UNUSED
4	R/W	0	DIV12_EN
3	R/W	0	DIV6_EN
2	R/W	0	DIV4_EN
1	R/W	0	DIV2_EN
0	R/W	0	DIV1_EN

6.4.27. HHI_AUD_CLK_CNTL2 0x1064

Bit(s)	R/W	Default	Description
31-28	R/W	0	Unused
27	R/W	0	IEC958_USE_CNTL: If this bit is set to 1, then bits[26;16] are used as the IEC958 clock divider
26-25	R/W	0	IEC958_CLK_SRC_SEL. See the clock tree document
23-16	R/W	0	IEC958_CLK_DIV. This is a new feature in M6TV. In the past, the IEC958 Clock was slaved to the AMCLK. In M6TV we added the ability to separately control the clock divider for IEC958. For these bits to take effect, you must set bit[27] above.
15-12	R/W	0	Unused
10-9	R/W	0	ADC_AMCLK_SRC_SEL
8	R/W	0	ADC AMCLK gate enable
7-0	R/W	0	DIVISOR_ADC_AMCLK

6.4.28. HHI_VID_CLK_CNTL2 0x1065

Bit(s)	R/W	Default	Description
31-8	R	0	Reserved
7	R/W	0	LCD_AN_CLK_PH2 gated clock: 1 = clock on, 0 = clock off. See the clock tree document
6	R/W	0	LCD_AN_CLK_PH3 gated clock: 1 = clock on, 0 = clock off
5	R/W	0	HDMI_TX pixel clk gated clock: 1 = clock on, 0 = clock off
4	R/W	0	VDAC clock gated clock: 1 = clock on, 0 = clock off
3	R/W	0	ENCL clock gated clock: 1 = clock on, 0 = clock off
2	R/W	0	ENCP clock gated clock: 1= clock on, 0 = clock off
1	R/W	0	ENCT clock gated clock: 1 = clock on, 0 = clock off
0	R/W	0	ENCI clock gated clock: 1 = clock on, 0 = clock off

6.4.29. HHI_VID_DIVIDER_CNTL 0x1066

Bit(s)	R/W	Default	Description
31-17	R/W	0	Unused
16	R/W	0	CLK_IN_EN
15	R/W	0	CLK_SEL
14-12	R/W	0	POST_TCNT
11	R/W	0	LVDS_CLK_EN
10	R/W	0	LVDS_DIV2
9-8	R/W	0	POST_SEL
7	R/W	0	SOFT_RESET_POST
6-4	R/W	0	PRE_SEL
3	R/W	0	SOFT_RESET_PRE
2	R/W	0	M_TEST
1	R/W	0	RESET_N_POST
0	R/W	0	RESET_N_PRE

6.4.30. HHI_SYS_CPU_CLK_CNTL0 0x1067

Bit(s)	R/W	Default	Description
31	R/W	0	L2_TRAM_CLKEN_SEL: Set to 0 to set l2_tram_clken high all of the time. Set to 1, to connect l2_tram_clken to divide by 2 (for a 2:1 A5 clock ratio)
30	R/W	0	L2 cache soft reset
29	R/W	0	AXI64to128 bridge (A5-to-MMC) soft reset (A5 interface)
28	R/W	0	SCU soft reset
27	R/W	0	CPU3 soft reset
26	R/W	0	CPU2 soft reset
25	R/W	0	CPU1 soft reset
24	R/W	0	CPU0 soft reset
23-19	R/W	0	reserved.
18	R/W	0	A5 Global Reset
17	R/W	0	A5 AXI Soft Reset
16	R/W	0	A5 APB Soft Reset
15	R/W	0	GEN_DIV_SOFT_RESET
14	R/W	0	SOFT_RESET
13-8	R/W	0	Reserved
7	R/W	0	Clock select: 1 = the output of the Mux / divider (bits[3;2] above). 0 = the system CPU clock comes from the XTAL. **NOTE: This bit should be set to 0 before changing the mux (bits[3:2])**
6	R/W	0	Reserved
5	R/W	1	A5 clock divided by 16 enable. This bit should only be set to 1 during debug. This bit enables a clock gate that drives a divided down version of the A5 clock back to the EE domain so that it can be measured or sent out on a GPIO.
4	R/W	1	APB_CLKEN
3-2	R/W	0	SCALE_OUT_SEL (see clock trees document): 0 = divide by 1 1 = divide by 2 2 = divide by 3 3 = divide by 2x(divide by N)
1-0	R/W	0	Reserved

6.4.31. HHI_MALI_CLK_CNTL 0x106c

Bit(s)	R/W	Default	Description
31~28	R/W	0	Reserved
27~25	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_mali _clk
24	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_ mali _clk
23	R/W	0	Reserved
22-16	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_ mali _clk
15-12	R/W	0	Reserved

Bit(s)	R/W	Default	Description
11~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_mali _clk
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_ mali _clk
7	R/W	0	Reserved
6-0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_ mali _clk

6.4.32. HHI_USB_CLK_CNTL 0x1089

This register creates an alternate 32khz clock for the USB module if the RTC clock is not available.

Bit(s)	R/W	Default	Description
31-11	R/W	0	unused
10	R/W	0	Clock enable
9-0	R/W	0	Clock divider

6.4.33. HHI_GEN_CLK_CNTL 0x108a

Bit(s)	R/W	Default	Description
31-16	R/W	0	Reserved
15~12	R/W	0	CLK_SEL: See the Clock Tree document for information related to gen_clk_out
11	R/W	0	CLK_EN: See the Clock Tree document for information related to gen_clk_out
10~0	R/W	0	CLK_DIV: See the Clock Tree document for information related to gen_clk_out

6.4.34. HHI_VDIN_MEAS_CLK_CNTL 0x1094

Bit(s)	R/W	Default	Description
31-24	R/W	0	Reserved
23~21	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_dsi_meas_clk
20	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_dsi_meas_clk
19	R/W	0	Reserved
18~12	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_dsi_meas_clk
11~0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_vdin_meas_clk

6.4.35. HHI_PCM_CLK_CNTL 0x1096

Clock control for cts_pcm_mclk and cts_pcm_sclk

Bit(s)	R/W	Default	Description
31-23	R/W	0	unused
22	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_pcm_sclk
21-16	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_pcm_sclk
15-13	R/W	0	Unused
12-10	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_pcm_mclk
9	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_pcm_mclk
8-0	R/W	48	CLK_DIV: See the Clock Tree document for information related to cts_pcm_mclk

6.4.36. HHI_NAND_CLK_CNTL 0x1097

Clock control for cts_pcm_mclk and cts_pcm_sclk

Bit(s)	R/W	Default	Description
31-12	R/W	0	unused
11~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_nand_core_clk
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_nand_core_clk
7	R/W	0	Reserved
6~0	R/W	48	CLK_DIV: See the Clock Tree document for information related to cts_nand_core_clk

6.4.37. HHI_HDMI_PHY_CNTL0 0x10e8

Bit(s)	R/W	Default	Description
31~16	R/W	0	HDMI_CTL1
15~0	R/W	0	HDMI_CTL0

6.4.38. HHI_HDMI_PHY_CNTL1 0x10e9

Bit(s)	R/W	Default	Description
31~16	R/W	0	HDMI_CTL5
13	R/W	0	HDMI_TX_PRBS_EN: Set to 1 to enable the PRBS engine
12	R/W	0	HDMI_TX_PRBS_ERR_EN: Set to 1 to enable the error flag detector. Set to 0 to reset the error detection logic
11~8	R/W	0	HDMI_TX_SET_HIGH: Set each bit to 1 to set the HDMI pin high
7~4	R/W	0	HDMI_TX_SET_LOW: Set each bit to 0 to set the HDMI Pins low
3~2	R/W	0	RESERVED
1	R/W	0	HDMI_TX_PHY_CLK_EN: Set to 1 to enable the HDMI TX PHY
0	R/W	0	HDMI_TX_PHY_SOFT_RESET: Set to 1 to reset the HDMI TX PHY

6.4.39. HHI_HDMI_PHY_CNTL2 0x10ea

Bit(s)	R/W	Default	Description
31~8	R	0	Reserved
7~0	R	0	HDMI_REGRD

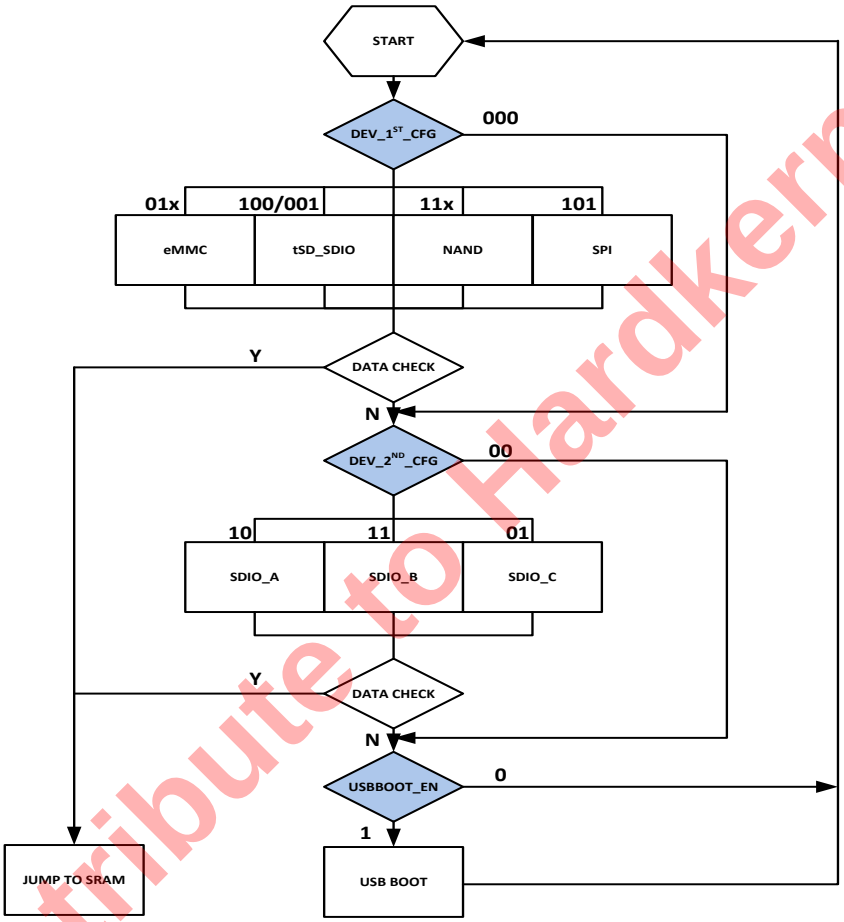
7. SYSTEM BOOT

7.1. Overview

The part describes the power–on mode configuration of S805, which can be configured to boot from NAND, eMMC, tSD, SPI NOR and SD card.

7.2. Power-on Flow Chart

Figure 2. Power-on Flow Chart



7.3. Power-on Configuration

As shown in the table below, user can simply pull the corresponding pads to high or low to set up the booting modes.

Table 1. Power-on Configuration

Pad	Function	Bit	Default	Comments
BOOT_14	DEV_1ST_CFG2	9	1	****DEV_1ST_CFG[2:0]:** 111:NAND(Default) 110:NAND_TOGGLE 101:SPI 100:tSD_SD_A 011:eMMC_SD_C 010:eMMC_SD_A 001:tSD_SD_C 000:SKIP_1ST
BOOT_13	POC_RESERVED	8	1	
BOOT_12	DEV_1ST_CFG1	7	1	
BOOT_11	DEV_1ST_CFG0	6	1	
BOOT_7	POC_RESERVED	5	1	
BOOT_6	POC_RESERVED	4	1	
BOOT_5	DEV_2ND_CFG1	3	1	****DEV_2ND_CFG[1:0]:** 11:SD_B (default) 10:SD_A 01: SD_C 00:SKIP_2ND
BOOT_4	DEV_2ND_CFG0	2	1	
BOOT_3	POC_RESERVED	1	1	
BOOT_2	USBOOT_EN	0	1	****USBOOT_EN:** 1:USBOOT(default) 0:SKIP_USBOOT

Here are typical applications:

a) NAND w/o toggle(boot pins) + SDIO_B(card pins), ---- **no external pull-down**
b) tSD/eMMC(boot pins) + SDIO_B(card pins), ---- **only one pull-down on boot_14/nand_ren_wr**.
c) SPI(boot pins, boot device) + NAND w/o toggle(boot pins, companion device) + SDIO_B(card pins), ---- **only one pull-down on boot_12/nand_cle/nor_q**.
d) NAND w/ toggle(boot pins) + SDIO_B(card pins), -- **only one pull-down on boot_11/nand_ale/nor_d**.

A typical initialization routine is:

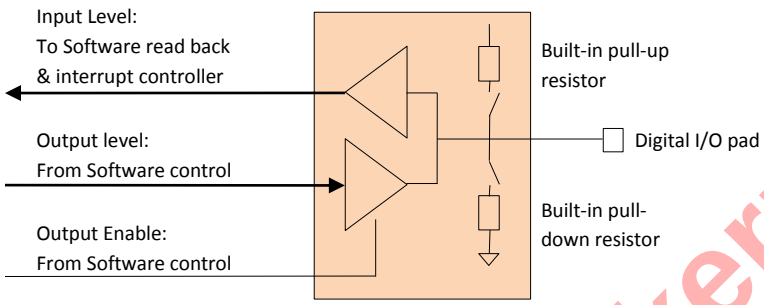
a) Disable Instruction and Data Cache;
b) Disable all pad's pull-up except TEST_N;
c) Initialize stack pointer;
d) Call **RomCode** ;
e) Jump to memory location where boot_loader is moved in;

8. GENERAL PURPOSE INPUT/OUTPUT (GPIO)

8.1. Overview

The SOC has a number of multi-function digital I/O pads that can be multiplexed to a number of internal resources (e.g. PWM generators, SDIO controllers …). When a digital I/O is not being used for any specific purpose, it is converted to a general purpose GPIO pin. A GPIO pin can be statically set to high/low logical levels. The structure of a GPIO is given below.

Figure 3. GPIO Structure



Each GPIO pin has an individual control bit that can be used to set the output level, output enable of the I/O pad (0 = output, 1 = input). Additionally all digital I/O pads have built-in pull-up and pull-down resistors. The input from the digital I/O pad can be read back in software and is also connected to an interrupt controller. The interrupt controller allows up to 8 GPIO pins to be used as active high, active low, rising edge or falling edge interrupts.

Finally, the GPIO's are grouped into voltage domains. Some GPIOs (depending on the PCB design) can be configured to operate between 0 and 1.8v while others may operate between 0.0v and 3.3v.

8.2. GPIO Multiplex Function

The GPIO multiplex functions are shown in the sections below, where the RegNN[MM] corresponds to CBUS registers defined in Table 2.

Table 2. Pin Mux Registers

Pin Mux Registers	Abbreviation	Offset	Default Value after power-on/Reset
PERIPHS_PIN_MUX_0	REG0	0x202C	0x0000
PERIPHS_PIN_MUX_1	REG1	0x202D	0x0000
PERIPHS_PIN_MUX_2	REG2	0x202E	0x0000
PERIPHS_PIN_MUX_3	REG3	0x202F	0x0000
PERIPHS_PIN_MUX_4	REG4	0x2030	0x0000
PERIPHS_PIN_MUX_5	REG5	0x2031	0x0000
PERIPHS_PIN_MUX_6	REG6	0x2032	0x0000
PERIPHS_PIN_MUX_7	REG7	0x2033	0x0000
PERIPHS_PIN_MUX_8	REG8	0x2034	0x0000
PERIPHS_PIN_MUX_9	REG9	0x2035	0x0000
AO_RTI_PIN_MUX_REG	REG	0xc8100014	0x0000

8.2.1. GPIO Bank X

Table 3. GPIO Bank X Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4	Func5
GPIOX_0	SD_D0_A Reg8[5]	SDXC_D0_A Reg5[14]			
GPIOX_1	SD_D1_A Reg8[4]	SDXC_D1_A Reg5[13]			
GPIOX_2	SD_D2_A Reg8[3]	SDXC_D2_A Reg5[13]			
GPIOX_3	SD_D3_A Reg8[2]	SDXC_D3_A Reg5[13]			
GPIOX_4	SDXC_D0_A Reg5[29] SDXC_D4_A Reg5[12]	PCM_OUT_A Reg3[30]	UART_TX_A Reg4[17]		
GPIOX_5	SDXC_D1_A Reg5[28] SDXC_D5_A Reg5[12]	PCM_IN_A Reg3[29]	UART_RX_A Reg4[16]		
GPIOX_6	SDXC_D2_A Reg5[28] SDXC_D6_A Reg5[12]	PCM_FS_A Reg3[28]	UART_CTS_A Reg4[15]	ISO7816_CLK Reg5[9]	
GPIOX_7	SDXC_D3_A Reg5[28] SDXC_D7_A Reg5[12]	PCM_CLK_A Reg3[27]	UART_RTS_A Reg4[14]	ISO7816_DATA Reg5[8]	
GPIOX_8	SD_CLK_A Reg8[1]	SDXC_CLK_A Reg5[11]	UART_TX_B Reg6[19]	SPI_SCLK reg4[22]	Tsin_CLK_B reg3[6]
GPIOX_9	SD_CMD_A Reg8[0]	SDXC_CMD_A Reg5[10]	UART_RX_B Reg6[18]	SPI_MISO reg4[24]	Tsin_SOP_B reg3[7]
GPIOX_10	XTAL_32K_OUT Reg3[22]	PWM_VS Reg7[31] PWM_E Reg9[19]	UART_CTS_B Reg6[17]	SPI_MOSI reg4[23]	Tsin_D0_B reg3[8]
GPIOX_11	XTAL_24M_OUT , XTAL/2 Reg3[20]	PWM_VS Reg7[30]			PWM_B Reg2[3]
GPIOX_16	UART_TX_B Reg4[9]			ISO7816_DET Reg4[21]	I2C_SDA_D Reg4[5]
GPIOX_17	UART_RX_B Reg4[8]			ISO7816_RESET Reg4[20]	I2C_SCK_D Reg4[4]
GPIOX_18	UART_CTS_B Reg4[7]			ISO7816_CLK Reg4[19]	
GPIOX_19	UART_RTS_B Reg4[6]			ISO7816_DATA Reg4[18]	
GPIOX_20			UART_RTS_B Reg6[16]	SPI_SS0 reg4[25]	Tsin_D_VALID_B reg3[9]
GPIOX_21					

8.2.2. GPIO Bank Y

Table 4. GPIO Bank Y Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4	Func5
GPIOY_0	TSin_D_VALID_A Reg3[2]				

Package Name	Func1	Func2	Func3	Func4	Func5
GPIOY_1	TSin_SOP_A Reg3[1]				
GPIOY_3		SPDIF_OUT Reg1[7]	XTAL_24Mhz Reg3[18]		
GPIOY_6	TSin_D6_A Reg3[5]				
GPIOY_7	TSin_D7_A Reg3[5]				
GPIOY_8	Tsin_CLK_A Reg3[0]				
GPIOY_9	TSin_D0_A Reg3[4]				
GPIOY_10	TSin_D1_A Reg3[5]				
GPIOY_11	TSin_D2_A Reg3[5]				
GPIOY_12	TSin_D3_A Reg3[5]				
GPIOY_13	TSin_D4_A Reg3[5]		ISO7816_CLK Reg5[7]		
GPIOY_14	TSin_D5_A Reg3[5]		ISO7816_DATA Reg5[6]		

8.2.3. GPIO Bank DV

Table 5. GPIO Bank DV Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4	Func5
GPIODV_9		PWM_VS Reg7[28]			PWM_C Reg3[24]
GPIODV_24				UART_TX_C Reg6[23]	I2C_SDA_A Reg9[31]
GPIODV_25				UART_RX_C Reg6[22]	I2C_SCK_A Reg9[30]
GPIODV_26				UART_CTS_C Reg6[21]	I2C_SDA_B Reg9[29]
GPIODV_27				UART_RTS_C Reg6[20]	I2C_SCK_B Reg9[28]
GPIODV_28	PWM_D Reg3[26]	PWM_VS Reg7[27]			I2C_SDA_C Reg9[27]
GPIODV_29	PWM_C Reg3[25]	PWM_VS Reg7[26]	XTAL24_output Reg7[25]		I2C_SCK_C Reg9[26]

8.2.4. GPIO Bank H

Table 7. GPIO Bank H Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4	Func5
GPIOH_0	HDMI_HPD Reg1[26]	SPI_SS1 Reg8[11]			
GPIOH_1	HDMI_SDA Reg1[25]	SPI_SS2 Reg8[12]			
GPIOH_2	HDMI_SCL Reg1[24]				
GPIOH_3	HDMI_CEC Reg1[23]	SPI_SS0 Reg9[13]		I2C_SDA_B Reg5[27]	
GPIOH_4		SPI_MISO Reg9[12]		I2C_SCK_B Reg5[26]	
GPIOH_5	ETH_TXD1 Reg7[21]	SPI_MOSI Reg9[11]		I2C_SDA_C Reg5[25]	

Package Name	Func1	Func2	Func3	Func4	Func5
GPIOH_6	ETH_TXD0 Reg7[20]	SPI_SCLK Reg9[10]		I2C_SCK_C Reg5[24]	
GPIOH_7			ETH_TXD3 Reg6[13]	I2C_SDA_D Reg4[3]	
GPIOH_8			ETH_TXD2 Reg6[12]	I2C_SCL_D Reg4[2]	
GPIOH_9	CLK_24M_OUT, CLK_12M_OUT Reg4[1]		ETH_TX_CLK Reg6[11]		

8.2.5. GPIO Bank AO

Table 10. GPIO Bank AO Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4
GPIOAO_0	UART_TX_AO_A Reg[12]		UART_TX_AO_B Reg[26]	
GPIOAO_1	UART_RX_AO_A Reg[11]		UART_RX_AO_B Reg[25]	
GPIOAO_2	UART_CTS_AO_A Reg[10]		UART_CTS_AO_B Reg[8]	
GPIOAO_3	UART_RTS_AO_A Reg[9]	PWM_C Reg[22]	UART_RTS_AO_B Reg[7]	
GPIOAO_4	I2C_MST_SCK_AO Reg[6]	I2C_SLAVE_SCK_AO Reg[2]	UART_TX_AO_B Reg[24]	
GPIOAO_5	I2C_MST_SDA_AO Reg[5]	I2C_SLAVE_SDA_AO Reg[1]	UART_RX_AO_B Reg[23]	
GPIOAO_6	CLK_32K_IN CLK_32K_OUT Reg[18]		SPDIF_OUT Reg[16]	I2S_IN_CH01 Reg1[13]
GPIOAO_7	IR_REMOTE_INPUT(IR _BLASTER) Reg[0]	IR_REMOTE_OUTPUT(IR_DEC) Reg[21]		
GPIOAO_8	JTAG_TCK	I2S_AM_CLK_OUT Reg[30]		
GPIOAO_9	JTAG_TMS	I2S_AO_CLK_OUT Reg[29]		I2S_AO_CLK_IN Reg1[15]
GPIOAO_10	JTAG_TDI	I2S_LR_CLK_OUT Reg[28]		I2S_LR_CLK_IN Reg1[14]
GPIOAO_11	JTAG_TDO	I2S_OUT_01 Reg[27]		
GPIOAO_12	HDMI_CEC Reg[17]			
GPIOAO_13	IR_BLASTER Reg[31]			
TEST_N	PWM_F Reg[19]			

8.2.6. GPIO Bank BOOT

Table 6. GPIO Bank BOOT Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4	Func5
BOOT_0	NAND_IO_0 Reg2[26]	SDXC_D0_C Reg4[30]	SD_D0_C Reg6[29]	BSD_TDO	
BOOT_1	NAND_IO_1 Reg2[26]	SDXC_D1_C Reg4[29]	SD_D1_C Reg6[28]	BSD_TDI	
BOOT_2	NAND_IO_2 Reg2[26]	SDXC_D2_C Reg4[29]	SD_D2_C Reg6[27]	BSD_TMS	
BOOT_3	NAND_IO_3 Reg2[26]	SDXC_D3_C Reg4[29]	SD_D3_C Reg6[26]	BSD_TCK	
BOOT_4	NAND_IO_4 Reg2[26]	SDXC_D4_C Reg4[28]			
BOOT_5	NAND_IO_5	SDXC_D5_C			

Package Name	Func1	Func2	Func3	Func4	Func5
	Reg2[26]	Reg4[28]			
BOOT_6	NAND_IO_6 Reg2[26]	SDXC_D6_C Reg4[28]			
BOOT_7	NAND_IO_7 Reg2[26]	SDXC_D7_C Reg4[28]			
BOOT_8	NAND_IO_CE0 Reg2[25]	SDXC_CLK_C Reg7[19]	SD_CLK_C Reg6[31]		
BOOT_9	NAND_IO_CE1 Reg2[24]				
BOOT_10	NAND_IO_RB0 Reg2[17]	SDXC_CMD_C Reg7[18]	SD_CMD_C Reg6[30]		
BOOT_11	NAND_ALE Reg2[21]	NOR_D Reg5[1]			
BOOT_12	NAND_CLE Reg2[20	NOR_Q Reg5[3]			
BOOT_13	NAND_WEN_CLK Reg2[19]	NOR_C Reg5[2]			
BOOT_14	NAND_REN_CLK Reg2[18]				
BOOT_15	NAND_DQS Reg2[27]				
BOOT_18	NAND_DQS Reg2[28]	NOR_CS Reg5[0]			

8.2.7. GPIO Bank CARD

Table 7. GPIO Bank CARD Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4	Func5
CARD_0	SD_D1_B Reg2[14]	SDXC_D1_B Reg2[6]	JTAG_TDI		
CARD_1	SD_D0_B Reg2[15]	SDXC_D0_B Reg2[7]	JTAG_TDO		
CARD_2	SD_CLK_B Reg2[11]	SDXC_CLK_B Reg2[5]	JTAG_CLK		
CARD_3	SD_CMD_B Reg2[10]	SDXC_CMD_B Reg2[4]	JTAG_TMS		
CARD_4	SD_D3_B Reg2[12]	SDXC_D3_B Reg2[6]		UART_TX_AO_A Reg8[10]	UART_RX_AO_A Reg8[8]
CARD_5	SD_D2_B Reg2[13]	SDXC_D2_B Reg2[6]		UART_RX_AO_A Reg8[9]	UART_TX_AO_A Reg8[7]
CARD_6					

8.2.8. GPIO Bank DIF

Table 8. GPIO Bank DIF Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4	Func5
DIF_0_P	ETH_RXD1 Reg6[0]				
DIF_0_N	ETH_RXD0 Reg6[1]				
DIF_1_P	ETH_RX_DV Reg6[2]				
DIF_1_N	ETH_RX_CLK Reg6[3]				
DIF_2_P	ETH_TXD0 Reg6[4]				
DIF_2_N	ETH_TXD1 Reg6[5]				

Package Name	Func1	Func2	Func3	Func4	Func5
DIF_3_P	ETH_TX_EN Reg6[6]				
DIF_3_N	ETH_PHY_REF_CLK Reg6[8]				
DIF_4_P	ETH_MDC Reg6[9]				
DIF_4_N	ETH_MDIO Reg6[10]				

8.3. GPIO Interrupt

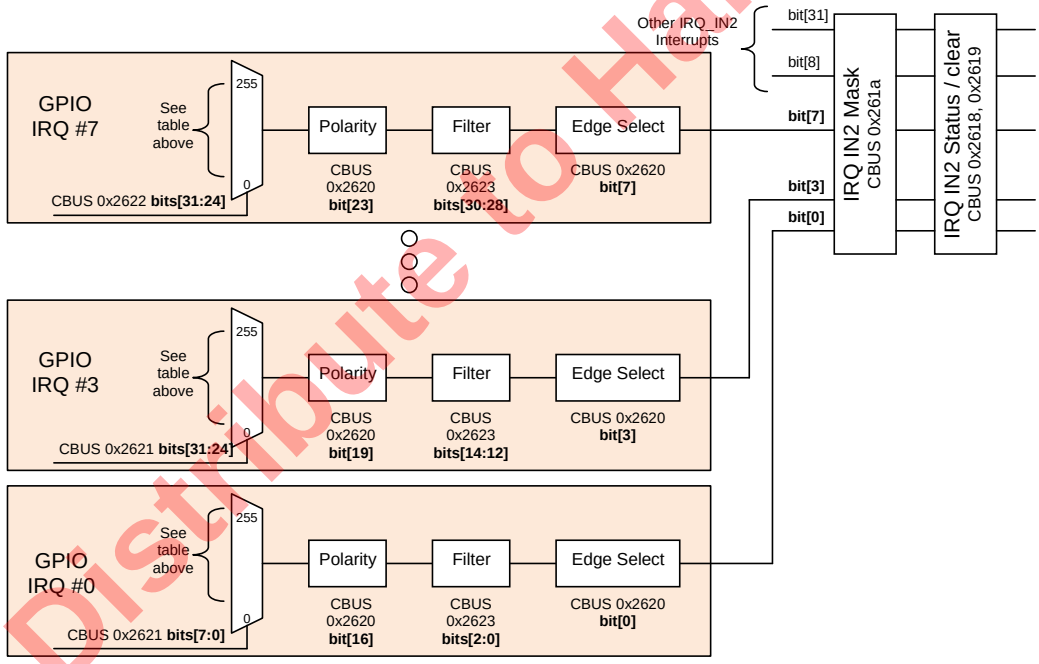
There are 8 independent filtered GPIO interrupt modules that can be programmed to use any of the GPIOs in the chip as an interrupt source (listed in the table below). For example, to select boot_3 as the source for GPIO IRQ #0, then CBUS 0x2621 bits[7:0] = 27 (according to the table below).

Table 9. GPIO Interrupt Sources

Input Mux Location CBUS registers 0x2621 and 0x2622	Description
255~119	Undefined (no interrupt)
118~97	gpioX[21:0]
96~80	gpioY[16:0]
79~50	gpioDV[29:0]
49~43	Card[6:0]
42~24	Boot[18:0]
23~14	gpioH[9:0]
13 ~ 0	gpioAO[13:0]

The diagram below illustrates the path a GPIO takes to become an interrupt. The eight GPIO interrupts respond to the MASK, STATUS and STATUS/CLEAR registers just like any other interrupt in the chip. The difference for the GPIO interrupts is that they can be filtered and conditioned. Filtering can either be bypassed or enabled so that a GPIO must be high or low for [3n] * 125nS, where n = 1, 2, …, 7.

Figure 4. GPIO Interrupt Path



NOTE: The input for the GPIO interrupt module (the input into the 256:1 mux) comes directly from the I/O pad of the chip. Therefore if a pad (say card_6) is configured as a UART TX pin, then in theory, the UART TX pin can be a GPIO interrupt since the TX pin will drive card_6 which in turn can drive the GPIO interrupt module.

8.4. Register Description

Package Name	OEN (Read Write)	OUT (Write Only)	IN (Read Only)	PU/PD-EN (Read Write)	PU/PD (Read Write)	IO Type	Default State after reset
GPIO X							
GPIOX_0	0x200C bit[0]	0x200D bit[0]	0x200E bit[0]	0x204c Bit[0]	0x203e bit[0]	Tri-State	pull up
GPIOX_1	0x200C bit[1]	0x200D bit[1]	0x200E bit[1]	0x204c bit[1]	0x203e bit[1]	Tri-State	pull up
GPIOX_2	0x200C bit[2]	0x200D bit[2]	0x200E bit[2]	0x204c bit[2]	0x203e bit[2]	Tri-State	pull up
GPIOX_3	0x200C bit[3]	0x200D bit[3]	0x200E bit[3]	0x204c bit[3]	0x203e bit[3]	Tri-State	pull up
GPIOX_4	0x200C bit[4]	0x200D bit[4]	0x200E bit[4]	0x204c bit[4]	0x203e bit[4]	Tri-State	pull up
GPIOX_5	0x200C bit[5]	0x200D bit[5]	0x200E bit[5]	0x204c bit[5]	0x203e bit[5]	Tri-State	pull up
GPIOX_6	0x200C bit[6]	0x200D bit[6]	0x200E bit[6]	0x204c bit[6]	0x203e bit[6]	Tri-State	pull up
GPIOX_7	0x200C bit[7]	0x200D bit[7]	0x200E bit[7]	0x204c bit[7]	0x203e bit[7]	Tri-State	pull up
GPIOX_8	0x200C bit[8]	0x200D bit[8]	0x200E bit[8]	0x204c bit[8]	0x203e bit[8]	Tri-State	pull up
GPIOX_9	0x200C bit[9]	0x200D bit[9]	0x200E bit[9]	0x204c bit[9]	0x203e bit[9]	Tri-State	pull up
GPIOX_10	0x200C bit[10]	0x200D bit[10]	0x200E bit[10]	0x204c bit[10]	0x203e bit[10]	Tri-State	pull up
GPIOX_11	0x200C bit[11]	0x200D bit[11]	0x200E bit[11]	0x204c bit[11]	0x203e bit[11]	Tri-State	Pull down
GPIOX_16	0x200C bit[16]	0x200D bit[16]	0x200E bit[16]	0x204c bit[16]	0x203e bit[16]	Tri-State	pull up
GPIOX_17	0x200C bit[17]	0x200D bit[17]	0x200E bit[17]	0x204c bit[17]	0x203e bit[17]	Tri-State	pull up
GPIOX_18	0x200C bit[18]	0x200D bit[18]	0x200E bit[18]	0x204c bit[18]	0x203e bit[18]	Tri-State	pull down
GPIOX_19	0x200C bit[19]	0x200D bit[19]	0x200E bit[19]	0x204c bit[19]	0x203e bit[19]	Tri-State	pull up
GPIOX_20	0x200C bit[20]	0x200D bit[20]	0x200E bit[20]	0x204c bit[20]	0x203e bit[20]	Tri-State	pull down
GPIOX_21	0x200C bit[21]	0x200D bit[21]	0x200E bit[21]	0x204c Bit[21]	0x203e bit[21]	Tri-State	pull up
GPIO Y							
GPIOY_0	0x200F bit[0]	0x2010 bit[0]	0x2011 bit[0]	0x204b Bit[0]	0x203d bit[0]	Tri-State	pull up
GPIOY_1	0x200F bit[1]	0x2010 bit[1]	0x2011 bit[1]	0x204b bit[1]	0x203d bit[1]	Tri-State	pull up
GPIOY_3	0x200F bit[3]	0x2010 bit[3]	0x2011 bit[3]	0x204b bit[3]	0x203d bit[3]	Tri-State	pull up
GPIOY_6	0x200F bit[6]	0x2010 bit[6]	0x2011 bit[6]	0x204b bit[6]	0x203d bit[6]	Tri-State	pull up
GPIOY_7	0x200F bit[7]	0x2010 bit[7]	0x2011 bit[7]	0x204b bit[7]	0x203d bit[7]	Tri-State	pull up
GPIOY_8	0x200F bit[8]	0x2010 bit[8]	0x2011 bit[8]	0x204b bit[8]	0x203d bit[8]	Tri-State	pull up
GPIOY_9	0x200F bit[9]	0x2010 bit[9]	0x2011 bit[9]	0x204b bit[9]	0x203d bit[9]	Tri-State	pull up
GPIOY_10	0x200F bit[10]	0x2010 bit[10]	0x2011 bit[10]	0x204b bit[10]	0x203d bit[10]	Tri-State	pull up
GPIOY_11	0x200F bit[11]	0x2010 bit[11]	0x2011 bit[11]	0x204b bit[11]	0x203d bit[11]	Tri-State	pull up

Package Name	OEN (Read Write)	OUT (Write Only)	IN (Read Only)	PU/PD-EN (Read Write)	PU/PD (Read Write)	IO Type	Default State after reset
GPIOY_12	0x200F bit[12]	0x2010 bit[12]	0x2011 bit[12]	0x204b bit[12]	0x203d bit[12]	Tri-State	pull up
GPIOY_13	0x200F bit[13]	0x2010 bit[13]	0x2011 bit[13]	0x204b bit[13]	0x203d bit[13]	Tri-State	pull up
GPIOY_14	0x200F bit[14]	0x2010 bit[14]	0x2011 bit[14]	0x204b bit[14]	0x203d bit[14]	Tri-State	pull up
GPIODV_9	0x2012 bit[9]	0x2013 bit[9]	0x2014 bit[9]	0x2048 bit[9]	0x203a bit[9]	Tri-State	pull down
GPIODV_24	0x2012 bit[24]	0x2013 bit[24]	0x2014 bit[24]	0x2048 bit[24]	0x203a bit[24]	Tri-State	no pull up/down
GPIODV_25	0x2012 bit[25]	0x2013 bit[25]	0x2014 bit[25]	0x2048 bit[25]	0x203a bit[25]	Tri-State	no pull up/down
GPIODV_26	0x2012 bit[26]	0x2013 bit[26]	0x2014 bit[26]	0x2048 bit[26]	0x203a bit[26]	Tri-State	no pull up/down
GPIODV_27	0x2012 bit[27]	0x2013 bit[27]	0x2014 bit[27]	0x204s8 bit[27]	0x203a bit[27]	Tri-State	no pull up/down
GPIODV_28	0x2012 bit[28]	0x2013 bit[28]	0x2014 bit[28]	0x2048 bit[28]	0x203a bit[28]	Tri-State	pull down
GPIODV_29	0x2012 bit[29]	0x2013 bit[29]	0x2014 bit[29]	0x2048 bit[29]	0x203a bit[29]	Tri-State	pull up
GPIO H							
GPIOH_0	0x2015 bit[19]	0x2016 bit[19]	0x2017 bit[19]	0x2049 bit[16]	0x203b bit[16]	Tri-State	no pull up/down
GPIOH_1	0x2015 bit[20	0x2016 bit[20]	0x2017 bit[20]	0x2049 bit[17]	0x203b bit[17]	Tri-State	no pull up/down
GPIOH_2	0x2015 bit[21]	0x2016 bit[21]	0x2017 bit[21]	0x2049 bit[18]	0x203b bit[18]	Tri-State	no pull up/down
GPIOH_3	0x2015 bit[22]	0x2016 bit[22]	0x2017 bit[22]	0x2049 bit[19]	0x203b bit[19]	Tri-State	pull up
GPIOH_4	0x2015 bit[23]	0x2016 bit[23]	0x2017 bit[23]	0x2049 bit[20]	0x203b bit[20]	Tri-State	pull down
GPIOH_5	0x2015 bit[24]	0x2016 bit[24]	0x2017 bit[24]	0x2049 bit[21]	0x203b bit[21]	Tri-State	pull down
GPIOH_6	0x2015 bit[25]	0x2016 bit[25]	0x2017 bit[25]	0x2049 bit[22]	0x203b bit[22]	Tri-State	pull down
GPIOH_7	0x2015 bit[26]	0x2016 bit[26]	0x2017 bit[26]	0x2049 bit[23]	0x203b bit[23]	Tri-State	no pull up/down
GPIOH_8	0x2015 bit[27]	0x2016 bit[27]	0x2017 bit[27]	0x2049 bit[24]	0x203b bit[24]	Tri-State	no pull up/down
GPIOH_9	0x2015 bit[28]	0x2016 bit[28]	0x2017 bit[28]	0x2049 bit[25]	0x203b bit[25]	Tri-State	no pull up/down
GPIO AO							
GPIOAO_0	0xc8100024 bit[0]	0xc8100024 bit[16]	0xc8100028 bit[0]	0xc810002c bit[0]	0xc810002c bit[16]	Tri-State	High-z
GPIOAO_1	0xc8100024 bit[1]	0xc8100024 bit[17]	0xc8100028 bit[1]	0xc810002c bit[1]	0xc810002c bit[17]	Tri-State	High-z
GPIOAO_2	0xc8100024 bit[2]	0xc8100024 bit[18]	0xc8100028 bit[2]	0xc810002c bit[2]	0xc810002c bit[18]	Tri-State	High-z
GPIOAO_3	0xc8100024 bit[3]	0xc8100024 bit[19]	0xc8100028 bit[3]	0xc810002c bit[3]	0xc810002c bit[19]	Tri-State	High-z
GPIOAO_4	0xc8100024 bit[4]	0xc8100024 bit[20]	0xc8100028 bit[4]	0xc810002c bit[4]	0xc810002c bit[20]	Tri-State	High-z
GPIOAO_5	0xc8100024 bit[5]	0xc8100024 bit[21]	0xc8100028 bit[5]	0xc810002c bit[5]	0xc810002c bit[21]	Tri-State	High-z
GPIOAO_6	0xc8100024 bit[6]	0xc8100024 bit[22]	0xc8100028 bit[6]	0xc810002c bit[6]	0xc810002c bit[22]	Tri-State	High-z
GPIOAO_7	0xc8100024 bit[7]	0xc8100024 bit[23]	0xc8100028 bit[7]	0xc810002c bit[7]	0xc810002c bit[23]	Tri-State	High-z

Package Name	OEN (Read Write)	OUT (Write Only)	IN (Read Only)	PU/PD-EN (Read Write)	PU/PD (Read Write)	IO Type	Default State after reset
GPIOAO_8	0xc8100024 bit[8]	0xc8100024 bit[24]	0xc8100028 bit[8]	0xc810002c bit[]	0xc810002c bit[24]	Tri-State	High-z
GPIOAO_9	0xc8100024 bit[9]	0xc8100024 bit[25]	0xc8100028 bit[9]	0xc810002c bit[9]	0xc810002c bit[25]	Tri-State	High-z
GPIOAO_10	0xc8100024 bit[10]	0xc8100024 bit[26]	0xc8100028 bit[10]	0xc810002c bit[10]	0xc810002c bit[26]	Tri-State	High-z
GPIOAO_11	0xc8100024 bit[11]	0xc8100024 bit[27]	0xc8100028 bit[11]	0xc810002c bit[11]	0xc810002c bit[27]	Tri-State	High-z
GPIOAO_12	0xc8100024 bit[12]	0xc8100024 bit[28]	0xc8100028 bit[12]	0xc810002c bit[12]	0xc810002c bit[28]	Tri-State	High-z
GPIOAO_13	0xc8100024 bit[13]	0xc8100024 bit[29]	0xc8100028 bit[13]	0xc810002c Bit[13]	0xc810002c Bit[29]	Tri-State	High-z
GPIO BOOT							
BOOT_0	0x2015 bit[0]	0x2016 bit[0]	0x2017 bit[0]	0x204a bit[0]	0x203c bit[0]	Tri-State	no pull up/down
BOOT_1	0x2015 bit[1]	0x2016 bit[1]	0x2017 bit[1]	0x204a bit[1]	0x203c bit[1]	Tri-State	no pull up/down
BOOT_2	0x2015 bit[2]	0x2016 bit[2]	0x2017 bit[2]	0x204a bit[2]	0x203c bit[2]	Tri-State	pull up
BOOT_3	0x2015 bit[3]	0x2016 bit[3]	0x2017 bit[3]	0x204a bit[3]	0x203c bit[3]	Tri-State	pull up
BOOT_4	0x2015 bit[4]	0x2016 bit[4]	0x2017 bit[4]	0x204a bit[4]	0x203c bit[4]	Tri-State	pull up
BOOT_5	0x2015 bit[5]	0x2016 bit[5]	0x2017 bit[5]	0x204a bit[5]	0x203c bit[5]	Tri-State	pull up
BOOT_6	0x2015 bit[6]	0x2016 bit[6]	0x2017 bit[6]	0x204a bit[6]	0x203c bit[6]	Tri-State	pull up
BOOT_7	0x2015 bit[7]	0x2016 bit[7]	0x2017 bit[7]	0x204a bit[7]	0x203c bit[7]	Tri-State	pull up
BOOT_8	0x2015 bit[8]	0x2016 bit[8]	0x2017 bit[8]	0x204a bit[8]	0x203c bit[8]	Tri-State	pull up
BOOT_9	0x2015 bit[9]	0x2016 bit[9]	0x2017 bit[9]	0x204a bit[9]	0x203c bit[9]	Tri-State	pull up
BOOT_10	0x2015 bit[10]	0x2016 bit[10]	0x2017 bit[10]	0x204a bit[10]	0x203c bit[10]	Tri-State	pull up
BOOT_11	0x2015 bit[11]	0x2016 bit[11]	0x2017 bit[11]	0x204a bit[11]	0x203c bit[11]	Tri-State	pull up
BOOT_12	0x2015 bit[12]	0x2016 bit[12]	0x2017 bit[12]	0x204a bit[12]	0x203c bit[12]	Tri-State	pull up
BOOT_13	0x2015 bit[13]	0x2016 bit[13]	0x2017 bit[13]	0x204a bit[13]	0x203c bit[13]	Tri-State	pull up
BOOT_14	0x2015 bit[14]	0x2016 bit[14]	0x2017 bit[14]	0x204a bit[14]	0x203c bit[14]	Tri-State	pull up
BOOT_15	0x2015 bit[15]	0x2016 bit[15]	0x2017 bit[15]	0x204a bit[15]	0x203c bit[15]	Tri-State	pull down
BOOT_18	0x2015 bit[18]	0x2016 bit[18]	0x2017 bit[18]	0x204a bit[18]	0x203c bit[18]	Tri-State	pull up
GPIO CARD							
CARD_0	0x200C bit[22]	0x200D bit[22]	0x200E bit[22]	0x204a bit[20]	0x203c bit[20]	Tri-State	pull up
CARD_1	0x200C bit[23]	0x200D bit[23]	0x200E bit[23]	0x204a bit[21]	0x203c bit[21]	Tri-State	pull up
CARD_2	0x200C bit[24]	0x200D bit[24]	0x200E bit[24]	0x204a bit[22]	0x203c bit[22]	Tri-State	pull up
CARD_3	0x200C bit[25]	0x200D bit[25]	0x200E bit[25]	0x204a bit[23]	0x203c bit[23]	Tri-State	pull up
CARD_4	0x200C bit[26]	0x200D bit[26]	0x200E bit[26]	0x204a bit[24]	0x203c bit[24]	Tri-State	pull up

Package Name	OEN (Read Write)	OUT (Write Only)	IN (Read Only)	PU/PD-EN (Read Write)	PU/PD (Read Write)	IO Type	Default State after reset
CARD_5	0x200C bit[27]	0x200D bit[27]	0x200E bit[27]	0x204a bit[25]	0x203c bit[25]	Tri-State	pull up
CARD_6	0x200C bit[28]	0x200D bit[28]	0x200E bit[28]	0x204a bit[26]	0x203c bit[26]	Tri-State	pull up
GPIO DIF							
DIF_0_P	0xc1108060 Bit[12]	0xc1108064 Bit[12]	0xc1108068 Bit[12]	0x204d Bit[8]	0x203f Bit[8]	Tri-State	no pull up/down
DIF_0_N	0xc1108060 Bit[13]	0xc1108064 Bit[13]	0xc1108068 Bit[13]	0x204d Bit[9]	0x203f Bit[9]	Tri-State	no pull up/down
DIF_1_P	0xc1108060 Bit[14]	0xc1108064 Bit[14]	0xc1108068 Bit[14]	0x204d Bit[10]	0x203f Bit[10]	Tri-State	no pull up/down
DIF_1_N	0xc1108060 Bit[15]	0xc1108064 Bit[15]	0xc1108068 Bit[15]	0x204d Bit[11]	0x203f Bit[11]	Tri-State	no pull up/down
DIF_2_P	0xc1108060 Bit[16]	0xc1108064 Bit[16]	0xc1108068 Bit[16]	0x204d Bit[12]	0x203f Bit[12]	Tri-State	no pull up/down
DIF_2_N	0xc1108060 Bit[17]	0xc1108064 Bit[17]	0xc1108068 Bit[17]	0x204d Bit[13]	0x203f Bit[13]	Tri-State	no pull up/down
DIF_3_P	0xc1108060 Bit[18]	0xc1108064 Bit[18]	0xc1108068 Bit[18]	0x204d Bit[14]	0x203f Bit[14]	Tri-State	no pull up/down
DIF_3_N	0xc1108060 Bit[19]	0xc1108064 Bit[19]	0xc1108068 Bit[19]	0x204d Bit[15]	0x203f Bit[15]	Tri-State	no pull up/down
DIF_4_P	0xc1108060 Bit[20]	0xc1108064 Bit[20]	0xc1108068 Bit[20]	0x204d Bit[16]	0x203f Bit[16]	Tri-State	no pull up/down
DIF_4_N	0xc1108060 Bit[21]	0xc1108064 Bit[21]	0xc1108068 Bit[21]	0x204d Bit[17]	0x203f Bit[17]	Tri-State	no pull up/down

9. INTERRUPT CONTROLLER

9.1. Overview

Generic Interrupt Controller (GIC) is a centralized resource that supports and manages interrupts in a system. For more details about GIC, please refer to the ARM GIC Architecture Specification V2.0.

9.2. Interrupt Source

There 224 interrupt sources in the chip. All of the interrupts are connected to the integrated GIC in Cortex-A5 while the AO-CPU see a sub-set of the interrupts. The control bits of AO-CPU interrupt are listed in the following table. The user can refer to section 0 for more details

Table 10. Interrupt Source

A5 GIC Bit	Interrupt sources	Description	EE Domain to AO-CPU STATUS=0x2610 MASK=0x2612	AO Domain to AO-CPU STATUS=0x2628 MASK=0x262A
255	1'b0	unused		
254	1'b0	unused		
253	1'b0	unused		
252	1'b0	unused		
251	1'b0	unused		
250	1'b0	unused		
249	1'b0	unused		
248	1'b0	unused		
247	1'b0	unused		
246	1'b0	unused		
245	1'b0	unused		
244	1'b0	unused		
243	1'b0	unused		
242	1'b0	unused		
241	1'b0	unused		
240	1'b0	unused		
239	1'b0	unused		
238	1'b0	unused		
237	1'b0	unused		
236	1'b0	unused		
235	1'b0	unused		
234	1'b0	unused		
233	1'b0	unused		
232	1'b0	unused		
231	1'b0	unused		
230	1'b0	unused		
229	1'b0	unused		
228	1'b0	unused		
227	1'b0	unused		
226	1'b0	unused		
225	1'b0	unused		
224	1'b0	unused		
223	1'b0	unused		
222	1'b0	unused		

A5 GIC Bit	Interrupt sources	Description	EE Domain to AO-CPU STATUS=0x2610 MASK=0x2612	AO Domain to AO-CPU STATUS=0x2628 MASK=0x262A
221	1'b0	unused		
220	1'b0	unused		
219	1'b0	unused		
218	1'b0	unused		
217	isp_int	unused		
216	edptx_int	unused		
215	1'b0	unused		
214	1'b0	unused		
213	1'b0	unused		
212	1'b0	unused		
211	mali_irq_ppmmu7			
210	mali_irq_pp7			
209	mali_irq_ppmmu6			
208	mali_irq_pp6			
207	mali_irq_ppmmu5			
206	mali_irq_pp5			
205	mali_irq_ppmmu4			
204	mali_irq_pp4			
203	mali_irq_ppmmu3			
202	mali_irq_pp3			
201	mali_irq_ppmmu2			
200	mali_irq_pp2			
199	mali_irq_ppmmu1			
198	mali_irq_pp1			
197	mali_irq_ppmmu0			
196	mali_irq_pp0			
195	mali_irq_pmu			
194	mali_irq_pp			
193	mali_irq_gpmmu			
192	mali_irq_gp			
191	viu2_line_n_int			
190	A5_dgb_comrx_irq[3]	A5 Debug RX interrupt CPU 3		
189	A5_dgb_comrx_irq[2]	A5 Debug RX interrupt CPU 3		
188	A5_dgb_comtx_irq[3]	A5 Debug TX interrupt CPU 3		
187	A5_dgb_comtx_irq[2]	A5 Debug TX interrupt CPU 3		
186	A5_pmu_irq[3]	A5 Performance CPU3		
185	A5_pmu_irq[2]	A5 Performance CPU2		
184	viu1_line_n_int			
183	ao_cec_irq	Always On CEC Interrupt		
182	ge2d_int	GE2D Interrupt		
181	cusad_interrupt	CUSAD		
180	assist_mbox_irq_ee[3]	Mail Box	bit[16]	bit[25]
179	assist_mbox_irq_ee[2]	Mail Box	bit[15]	bit[24]
178	assist_mbox_irq_ee[1]	Mail Box	bit[14]	bit[23]
177	assist_mbox_irq_ee[0]	Mail Box	bit[13]	bit[22]
176	det3d_int	unused		
175	l2ccintr_irq	L2Cachecontrollercombinedinterrupt		

A5 GIC Bit	Interrupt sources	Description	EE Domain to AO-CPU STATUS=0x2610 MASK=0x2612	AO Domain to AO-CPU STATUS=0x2628 MASK=0x262A
174	A5_dgb_comrx_irq[1]	A5 Debug RX interrupt CPU 1		
173	A5_dgb_comrx_irq[0]	A5 Debug RX interrupt CPU 0		
172	A5_dgb_comtx_irq[1]	A5 Debug TX interrupt CPU 1		
171	A5_dgb_comtx_irq[0]	A5 Debug TX interrupt CPU 0		
170	A5_pmu_irq[1]	A5 Performance CPU1		
169	A5_pmu_irq[0]	A5 Performance CPU0		
168	1'b0	Reserved Demodulator		
167	ir_blaster_irq	Always On IR Blaster		
166	mipi_dsi_te_intr			
165	mipi_dsi_err_intr			
164	1'b0	Reserved USB 3		
163	1'b0	Reserved USB 4		
162	1'b0	MMC Interrupt		
161	m_i2c_2_irq	I2C Master #2		
160	m_i2c_1_irq	I2C Master #1		
127	csi2_adapt_int	MIPI CSI adapter		
126	uart3_irq	UART 3		
125	uart2_irq	UART 2		
124	ao_m_i2c_irq	Always On Master I2C		
123	ao_s_i2c_irq	Always On Slave I2C		
122	ao_uart_irq	Always On UART		
121	rdma_done_int	RDMA		
120	i2s_cbus_ddr_irq	Audio I2S CBUS IRQ		bit[21]
119		unused		
118	vid1_wr_irq			
117	vdin1_vsync_int			
116	vdin1_hsync_int			
115	vdin0_vsync_int			
114	vdin0_hsync_int			
113	spi2_int			
112	spi_int			
111	vid0_wr_irq			
110	sdhc_irq			
109		unused		
108	1'b0	unused		
107	uart1_irq	unused		
106		unused		
105	sar_adc_irq	SAR ADC	bit[12]	bit[20]
104	ao_rtc_irq	Always On RTC		
103	gpio_irq[7]	GPIO Interrupt		
102	gpio_irq[6]	GPIO Interrupt		
101	gpio_irq[5]	GPIO Interrupt		
100	gpio_irq[4]	GPIO Interrupt		
99	gpio_irq[3]	GPIO Interrupt	bit[11]	bit[19]
98	gpio_irq[2]	GPIO Interrupt	bit[10]	bit[18]
97	gpio_irq[1]	GPIO Interrupt	bit[9]	bit[17]
96	gpio_irq[0]	GPIO Interrupt	bit[8]	bit[16]

A5 GIC Bit	Interrupt sources	Description	EE Domain to AO-CPU STATUS=0x2610 MASK=0x2612	AO Domain to AO-CPU STATUS=0x2628 MASK=0x262A
95	TimerI	TimerI	bit[7]	
94	TimerH	TimerH	bit[6]	
93	TimerG	TimerG	bit[5]	
92	TimerF	TimerF	bit[4]	
91	1'b0			
90	1'b0			
89	hdmi_tx_interrupt			
88	1'b0			
87	hdmi_cec_interrupt			
86	1'b0			
85	demux_int_2			
84	dmc_irq			
83	dmc_sec_irq			
82	ai_iec958_int	IEC958 interrupt		bit[15]
81	iec958_ddr_irq	IEC958 DDR interrupt		bit[14]
80	i2s_irq	I2S DDR Interrupt		bit[13]
79	crc_done	From AIU CRC done		bit[27]
78	deint_irq	Reserved for Deinterlacer		
77	dos_mbox_slow_irq[2]	DOS Mailbox 2		
76	dos_mbox_slow_irq[1]	DOS Mailbox 1		
75	dos_mbox_slow_irq[0]	DOS Mailbox 0		
74	1'b0			
73	1'b0			
72	1'b0			
71	m_i2c_3_irq	I2C Master #3		
70	ao_uart2_irq	Always On UART 2		
69	smartcard_irq			
68	ndma_irq	Block Move		bit[12]
67	spdif_irq			bit[11]
66	pnand_irq	NAND		
65	viff_empty_int_cpu			
64	parser_int_cpu			bit[10]
63	usb1_irq			
62	usb0_irq			
61	Timer D	Timer D	bit[3]	bit[9]
60	arc_sdio_irq			
59	mipi_dsi_phy_irq			
58	uart0_irq			
57	async_fifo2_flush_irq			bit[8]
56	async_fifo2_fill_irq			bit[7]
55	demux_int			bit[6]
54	encif_irq			
53	m_i2c_0_irq			
52	bt656_irq			
51	async_fifo_flush_irq			
50	async_fifo_fill_irq			
49	abuf_rd_interrupt			bit[5]

A5 GIC Bit	Interrupt sources	Description	EE Domain to AO-CPU STATUS=0x2610 MASK=0x2612	AO Domain to AO-CPU STATUS=0x2628 MASK=0x262A
48	abuf_wr_interrupt			bit[4]
47	ao_ir_dec_irq	Always On IR Remote		
46	eth_lpi_intr_o		bit[20]	
45	viu2_vsync_int			
44	viu2_hsync_int			
43	Timer B	Timer B	bit[2]	bit[3]
42	Timer A	Timer A	bit[1]	bit[2]
41	1'b0	unused		
40	eth_gmac_int		bit[18]	
39	audin_irq			bit[1]
38	Timer C	Timer C	bit[0]	bit[0]
37	demux_int_1			
36	eth_pmt_intr_o		bit[19]	
35	viu1_vsync_int	VSYNC		
34	viu1_hsync_int	HSYNC		
33	1'b0	HIU Mailbox		
32	1'b0	Watchdog Timer		

9.3. Register Description

9.3.1. AO_CPU_IRQ_IN0_STATUS 0x2610

Writing a 1 to a bit of this register clears the respective interrupt.

9.3.2. AO_CPU_IRQ_IN0_STATUS_CLEAR 0x2611

Writing a 1 to a bit of this register clears the respective interrupt.

9.3.3. AO_CPU_IRQ_IN0_MASK 0x2612

Writing a bit to 1, enables that interrupt.

9.3.4. AO_CPU_IRQ_IN0_FIQ_SEL 0x2613

Writing a bit to 1, muxes that interrupt to the FAST interrupt of the CPU

9.3.5. GPIO Interrupt EDGE and Polarity: 0x2620

This register controls the polarity of the GPIO interrupts and whether or not the interrupts are level or edge triggered. There are 8 GPIO interrupts. These 8 GPIO interrupts can be assigned to any one of up to 256 pins on the chip.

Bit(s)	R/W	Default	Description
31-24	R	0	unused
23			GPIO_POLARITY_PATH_7: If a bit in this field is 1, then the GPIO signal for GPIO interrupt path 7 is inverted.
22			GPIO_POLARITY_PATH_6:
21			GPIO_POLARITY_PATH_5:
20			GPIO_POLARITY_PATH_4:
19			GPIO_POLARITY_PATH_3:
18			GPIO_POLARITY_PATH_2:
17			GPIO_POLARITY_PATH_1:
16	R/W	0	GPIO_POLARITY_PATH_0:
15-8	R	0	Unused
7	R/W		GPIO_EDGE_SEL_PATH_7: If a bit is set to 1, then the GPIO interrupt for GPIO path 7 is configured to be an edge generated interrupt. If the polarity (above) is 0, then the interrupt is generated on the rising edge. If the polarity is 1, then the interrupt is generated on the falling edge of the GPIO. If a bit is this field is 0, then the GPIO is a level interrupt.
6	R/W		GPIO_EDGE_SEL_PATH_6
5	R/W		GPIO_EDGE_SEL_PATH_5
4	R/W		GPIO_EDGE_SEL_PATH_4
3	R/W		GPIO_EDGE_SEL_PATH_3
2	R/W		GPIO_EDGE_SEL_PATH_2
1	R/W		GPIO_EDGE_SEL_PATH_1
0	R/W	0	GPIO_EDGE_SEL_PATH_0

9.3.6. GPIO 0 ~ 3 Pin Select: 0x2621

Each GPIO interrupt can select from any number of up to 256 GPIO pins on the chip. The bits below control the pin selection for GPIO interrupts 0 ~3.

Bit(s)	R/W	Default	Description
31-24	R/W	0	GPIO_PIN_SEL3: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 3
23-16	R/W	0	GPIO_PIN_SEL2: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 2
15-8	R/W	0	GPIO_PIN_SEL1: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 1
7-0	R/W	0	GPIO_PIN_SEL0: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 0

9.3.7. GPIO 4 ~ 7 Pin Select: 0x2622

Bit(s)	R/W	Default	Description
31-24	R/W	0	GPIO_PIN_SEL7: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 7
23-16	R/W	0	GPIO_PIN_SEL6: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 6
15-8	R/W	0	GPIO_PIN_SEL5: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 5
7-0	R/W	0	GPIO_PIN_SEL4: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 4

9.3.8. GPIO Filter Select (interrupts 0~7): 0x2623

Bit(s)	R/W	Default	Description
31	R/W	0	unused
30-28	R/W	0	FILTER_SEL7: (see FILTER_SEL0)
27	R/W	0	Unused
26-24	R/W	0	FILTER_SEL6: (see FILTER_SEL0)
23	R/W	0	Unused
22-20	R/W	0	FILTER_SEL5: (see FILTER_SEL0)
19	R/W	0	Unused
18-16	R/W	0	FILTER_SEL4: (see FILTER_SEL0)
15	R/W	0	Unused
14-12	R/W	0	FILTER_SEL3: (see FILTER_SEL0)
11	R/W	0	Unused
10-8	R/W	0	FILTER_SEL2: (see FILTER_SEL0)
7	R/W	0	Unused
6-4	R/W	0	FILTER_SEL1: (see FILTER_SEL0)
3	R/W	0	unused
2-0	R/W	0	FILTER_SEL0: This value sets the filter selection for GPIO interrupt 0. A value of 0 = no filtering. A value of 7 corresponds to 7 x 3 x (111nS) of filtering.

10. DIRECT MEMORY ACCESS CONTROLLER (DMAC)

10.1. Overview

The DMA controller is an engine connected to the DDR controller for the purposes of moving data to/from DDR memory. The DMA controller supports 4 independent threads. Each thread is driven by DMA table descriptors placed in DDR memory. Each DMA descriptor consists of 8 entries that describe the source/destination location as well as the stride and burst. Inline processing is also supported to allow data from DDR to be processed using an AES, Triple-DES or CRC module before being placed back into DDR memory.

10.2. Descriptor Table

Entry	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Owner ID		Pre Endian			Source Hold	Dest Hold	INLINE Type			IRQ	-	-	-	-	-	-	-	-	-	-	-	-	NO Break	Thread Slice Count (0 = use default slice) Number of 256-byte blocks to allocate to this thread							
1	SP: Source Pointer (AHB address)																															
2	DP: Destination Pointer (AHB address) (0xFFFFFFFF = send data to the parser)																															
3	-	-	-	-	-	-	-	Byte transfer count																								
4	Source Skip																Source Burst Count															
5	Destination Skip																Destination Burst Count															
6-TDES																										restart		Current key		Post Endian		
6-AES													CTR Endian				CTR Limit		Mode		Reset IV	Encrypt		Type	Post Endian			Pre Endian				
6-Divx																														Post Endian		
6-CRC						CRC Count																						now rite	restart	Post Endian		
7	UNUSED Fill with 0. Filler to ensure each descriptor is 32-bytes so that the descriptors land on a cache line boundary																															

10.3. DMA Hardware Algorithm

```
If( ~2D ) {
    While( byte_transfer_count-- ) {
        *DP = *SP;                    // Move data
        SP += (move size);
        DP += (move size);
    }
} else {
    SBBC_tmp = SBBC;  // initialize the Source Burst Byte Count
    DBBC_tmp = SBBC;  // initialize the Destination Burst Byte Count
    While( byte_transfer_count-- ) {
        *DP = *SP;                    // Move data

        if( SBS == 0 ) {              // If NOT skipping the source pointer
            SP += (move size);
        } else {                      // 2D Move
            SBBC_tmp -= (move size); // decrement the source burst byte count
            if( !SBBC_tmp == 0 ) {    // if burst is NOT done
               SP += (move size);
            } else {                  // If the burst is done, then skip
               SBBC_tmp = SBBC;       // reset the burst byte count
               SP += SBS;             // skip the source pointer
            }
        }

        if( DBS == 0 ) {              // If NOT skipping the destination pointer
            DP += (move size);
        } else {                      // 2D Move
            DBBC_tmp -= (move size); // decrement the destination burst byte count
            if( DBBC_tmp == 0 ) {     // if burst is NOT done
               DP += (move size);
            } else {                  // If  the burst is done, then skip
               DBBC_tmp = SBBC;       // reset the burst byte count
               DP += DBS;             // skip the destination pointer
            }
        }
    }
}
```

CONTROL: Descriptor Entry 0	
31-30	Thread ID: Each DMA table entry is assigned an ID. The DMA engine rotates between threads based on the ID
29-27	Pre-Endian: DATA Endian [63:0] = <b0,b1,b2,b3,b4,b5,b6,b7}; where b0 = byte 0, b1 = byte1,... Endian Byte order --- 000 7 6 5 4 3 2 1 0 // no change 001 6 7 4 5 2 3 0 1 // swap bytes 010 5 4 7 6 1 0 3 2 // swap words 011 4 5 6 7 0 1 2 3 // reverse 100 3 2 1 0 7 6 5 4 101 2 3 0 1 6 7 4 5 110 1 0 3 2 5 4 7 6 111 0 1 2 3 4 5 6 7 NOTE: AES has its own endian control as well
26	Source Hold: Normally the source pointer is incremented as data is transferred. Set this bit to force the source pointer to keep the same position. This is useful if you want to DMA from an internal module at a fixed address.
25	Destination Hold: Normally the destination pointer is incremented as data is transferred. Set this bit to force the destination pointer to keep the same position. This is useful if you want to DMA to an internal module at a fixed address.
24-22	INLINE_type: These bits dictate the format of the last entry of the descriptor 000: Normal 1D or 2D move. The last descriptor entry is ignored 001: The last descriptor entry is for Triple-DES processing 010: The last descriptor entry is for Rijndael (DIVX) processing 011: The last descriptor entry is for CRC processing 100: The last descriptor entry is for AES processing
21	IRQ: Set this bit if you want the DMA engine to generate an interrupt after this table entry has been processed
20-9	unused
8	NO Break: The DMA engine will move from thread to thread based on the thread slice count (bits[7:0] below). If you want a DMA process to complete without interrupt set this bit to 1. If this bit is set to 0, then the DMA will be operation will be interrupted after THREAD_SLICE*256 bytes have been transferred
7-0	THREAD_SLICE: This value represents the number of 256 byte transfers to allocate to this thread before the DMA engine suspends this DMA transfer and moves to the next thread.
SOURCE POINTER: Descriptor Entry 1	
31-0	Source Pointer. This value corresponds to the AHB address of the source data.

DESTINATION POINTER: Descriptor Entry 2	
31-0	Destination Pointer. This value corresponds to the destination AHB address. If this value is set to 0xFFFFFFFF then data is send directly to the Parser
Bytes to Transfer: Descriptor Entry 3	
31-25	Unused
24-0	Number o bytes to transfer
SOURCE 2D MOVE: Descriptor Entry 4	
31-16	2D: Source Byte Skip [0 to 65535 bytes]. This value indicates how many bytes should be skipped between read bursts. The length of the burst is described by bits [11:0] below. Set this value to zero if no skipping is involved for the source pointer. This is often the case for linear transfers.
15-0	2D: Source Burst Byte Count [0 to 65535 bytes]. This value indicates the number of bytes to read before applying the skip value described in bits [31:16] above. For example, to read 24 bytes, skip 2 bytes, read 24 bytes, skip 2,…. Set descriptor 3 to 0x00020018 *NOTE: If the skip size above is set to zero, then this value is ignored*.
DESTINATION 2D MOVE: Descriptor Entry 5	
31-16	2D: Destination Byte Skip [0 to 65535 bytes]. This value indicates how many bytes should be skipped between write bursts. The length of the burst is described by bits [11:0] below.
15-0	2D: Destination Burst Byte Count [0 to 65535 bytes]. This value indicates the number of bytes to write before applying the skip value described in bits [31:16] above. For example, to write 24 bytes, skip 2 bytes, read 24 bytes, skip 2,…. Set descriptor 3 to 0x00020018 *NOTE: If the skip size above is set to zero, then this value is ignored*.
INLINE Specific Descriptor Entry 6 - TDES	
31-7	Unused
6	Restart: Set this bit to 1 to reset the CBC chaining pipeline
5	Unused
4-3	TDES_INDEX: The Triple DES engine has 4 stored keys for processing. These two bits are used to index an internal RAM (structure) this DMA transfer.
2-0	POST_ENDIAN: After processing Triple DES endian processing can be applied before data is written back to SDRAM.
INLINE Specific Descriptor Entry 6 - AES	
31-12	Unused
13-12	MODE: 00 = ECB mode, 01 = CBC mode, 10 = CTR mode, 11 = reserved
11	Restart: Set this bit to 1 to reset the CBC chaining pipeline
10	Encryption: 0 = decrypt, 1 = encrypt
9-8	TYPE: 00 = 128, 01 = 192, 10 = 256, 11 = reserved
7-4	POST_ENDIAN: Endian control of the outgoing message
3-0	PRE_ENDIAN: Endian control of the incoming message
INLINE Specific Descriptor Entry 6 - DivX	
31-3	Unused
2-0	POST_ENDIAN: After processing DivX, endian processing can be applied before data is written back to SDRAM.
INLINE Specific Descriptor Entry 6 - CRC	
31-3	Unused
4	No Write: Set this bit to 1 to disable writing of DMA data after CRC processing
3	CRC_RESET: Set this bit to 1 to reset the CRC engine
2-0	POST_ENDIAN: After processing AES, endian processing can be applied before data is written back to SDRAM.

10.4. Register Description

10.4.1. NDMA_CONTROL_REG 0x2270

Bit(s)	R/W	Default	Description
31	R/W	1	IN-LINE processing clock gating mode: If this bit is set to 0, then the clocks to the DMA engine are always on. If this bit is set to 1, then the DMA engine operates in auto-power down mode. That is, the internal clocks are shut down dynamically to save power.
30	R/W	1	CRC_GATED_CLK_MODE: If this bit is set to 0, then the clocks to the CRC engine are always enabled. If this bit is set to 1, then the CRC block operates in auto-power down mode which will save power when the CRC module is not being used.
29	R/W	1	AES_GATED_CLK_MODE: If this bit is set to 0, then the clocks to the AES engine are always enabled. If this bit is set to 1, then the AES block operates in auto-power down mode which will save power when the AES module is not being used.
28	R/W	1	TDES_GATED_CLK_MODE: If this bit is set to 0, then the clocks to the TDES engine are always enabled. If this bit is set to 1, then the TDES block operates in auto-power down mode which will save power when the TDES module is not being used.
27	R/W	1	DMA_GATED_CLK_MODE: If this bit is set to 0, then the clocks to the general DMA engine are always enabled. If this bit is set to 1, then the general DMA block operates in auto-power down mode which will save power when the general DMA module is not being used.
26	R	-	Status: '1' indicates the DMA engine is processing a descriptor. '0' indicates that all descriptors have been processed
25-16	R/W	0x63	Periodic Interrupt Delay: This value dictates the rate at which periodic interrupts are generated if bit 15 is set below. Note: If you change this value, then disable the periodic interrupt (bit 15 below) and then re-enable the period interrupt to reset the internal periodic counter.
15	R/W	0	Periodic Interrupt Enable: If this bit is set to '1', then the DMA engine will generate an interrupt every N+1 uS where N is described above as the Periodic Interrupt Delay.
14	R/W	0	DMA Engine Enable: Set this bit to '1' to enable the DMA Engine. Note, the DMA engine will not do anything until a descriptor has been placed into memory and the NEW_DMA_ADD_DESCRIPTOR register has been written. Set this bit to '0' to abort the current DMA and return the DMA engine to an idle state. Note: If you abort a transfer you should also clear the internal descriptor count by writing 0x2271 with 0x00000001.
12	R	-	AES STATUS: 1 = AES decryption/encryption engine is busy.
11-0	R/W	0x00	Reserved

10.4.2. NDMA Table Descriptor Add 0x2272

This register should be written with a count value indicating how much to increment an internal descriptor count. This register should be written whenever a descriptor has been added to the DMA descriptor list to tell the DMA engine that another descriptor needs to be processed. This register can be written before or after the DMA engine has been enabled (bit 14 of 0x2270). If you want to reset the internal table count value, write this register with 0x00000000.

Bit(s)	R/W	Default	Description
31-8	R	0	unused
9-8	R/W	0	THREAD_ID: Whenever a descriptor has been added to a particular thread, this register is written with the number of descriptors added. Bits[9:8] indicate the ID associated with the descriptor added.
7-0	R/W	0	Write this register with 1,2,…,128 to increment the internal descriptor count value by 1,2,…128. Write this register with 0x00000000 to clear the internal descriptor table counter. For example, if 3 descriptors were added to the descriptor table list, then this register would be written with 3 to tell the DMA engine that there are 3 more descriptors to process. Reading this register returns the current internal descriptor count value.

10.4.3. NDMA_TDES_IV_KEY 0x2273

The Triple DES engine maintains up to 4 IV keys used for CBC processing. The following three registers are used to setup and write the IV keys to an internal memory. Because each IV key is 64 bits, the key must be written to an internal register before a 3rd register write is used to push the stored IV key into the internal RAM.

Writing this register sets the IV key bits [31:0] of the register to be written to the RAM. Reading this register returns the current IV value for processing. This is useful if Cipher Block Chaining must be interrupted. Software can read the current IV key and restore it later to continue processing.

Bit(s)	R/W	Default	Description
31-0	W	0	IV Key bits[31:0]: Write Only

10.4.4. NDMA_TDES_IV_KEY 0x2274

The register represents the 2nd 32-bit component of the 64-bit IV key. Writing this register sets the IV key bits [63:32] of the register to be written to the RAM. Reading this register returns the current IV value for processing. This is useful if Cipher Block Chaining must be interrupted. Software can read the current IV key and restore it later to continue processing.

Bit(s)	R/W	Default	Description
31-0	W	0	IV Key bits[63:32] Write Only

10.4.5. NDMA_TDES_CONTROLS 0x2275

In addition to the 4 IV keys, the Triple DES engine also maintains 4 structures that dictate the individual DES modes, CBC enable, and whether or not we're encrypting or decrypting.

Once the 64-bit IV Key has been established and the MODE, CBC_EN and DECRYPT_EN bits have been set below, write this register again setting bits 30 and 31. This pushes the 64-bit IV key and MODE, CBC_EN and DECRYPT_EN bits into internal RAM. The processing engine will look up these values later using the 2-bit index (TDES_INDEX) supplied in the DMA table entry.

Bit(s)	R/W	Default	Description
31	W	0	This bit is a write only bit. Writing this bit pushes the IV key stored in registers (0x2273 & 0x2274) into the IV key internal RAM. The RAM index is supplied by bits [3:0] below.
30	W	0	This bit is a write only bit. Writing this bit pushes the DES_MODES, CBC_EN and DECRYPT_EN bits into an internal RAM. The RAM index is supplied by bits [3:0] below.
29-9	R	0	unused
8-6	R/W	0	MODE: Set to 0x5 for Triple DES decryption and 0x2 for Triple DES Encryption. Note this value is simply shadowed in this register. This bit should be written first before setting bits 30/31 above to push this value into the internal MODE RAM.
5	R/W	0	CBC_EN: 0 = ECB mode, 1 = CBC mode. Note this bit is simply shadowed in this register. This bit should be written first before setting bits 30/31 above to push this value into the internal MODE RAM.
4	R/W	0	DECRYPT_EN: Set this bit to 1 to decrypt, 0 to encrypt using triple DES. Note this bit is simply shadowed in this register. This bit should be written first before setting bits 30/31 above to push this value into the internal MODE RAM.
3-2	R/W	0	IV KEY and Modes Address. These 2 bits are used to index the 4 locations in the IV KEY RAM and the MODE RAM. These 2 bits should be set before writing bits 30 and 31 above.
1-0	R/W	0	unused

10.4.6. NDMA_RIJNDAEL_CONTROL 0x2276

This register directs DIVX processing.

Bit(s)	R/W	Default	Description
31	W	0	This bit is a write only bit. If this bit is set when writing, the RK fifo read pointer is reset.
30-4	R	0	unused
3-0	R/W	0	NR Value: 10,12 or 14

10.4.7. NDMA_RIJNDAEL_RK_FIFO 0x2277

Bit(s)	R/W	Default	Description
31-0	R/W	0	Writing this register writes the RK FIFO for DIVX Decryption

10.4.8. NDMA_CRC_CONTROL 0x2278

Bit(s)	R/W	Default	Description
31-0	R/W	0	Writing this register writes the CRC Value into the CRC module. Reading this register returns the current CRC value.

10.4.9. NDMA_THREAD_REG 0x2279

Bit(s)	R/W	Default	Description
31-28	R/W	0	Unused
27-24	R/W	0	THREAD_INIT: Each bit is used to initialize a particular thread. The initialization of a particular thread is described below (or in subsequent e-mails from chris@amlogic.com)
23-16	R/W	0	Unused
15	R	0	CORE_BUSY: This bit indicates that the thread state-machine is busy
14	R/W	0	unused
13-12	R	-	CURR_THREAD_NUM: This value represents the current thread ID being processed by the DMA state-machine
11-8	R/W	0	THREAD_ENABLE: There are 4 threads that can be enabled/disabled corresponding to bits[11:8]
7-0	R/W	1	DEFAULT_SLICE_CNT: The DMA engine will move from one thread to another after [n] * 256 byte transfers. If a slice count is not provided in the descriptor table, the default value is applied to that thread's descriptor

11. TIMER

11.1. Overview

The SOC contains 11 general purpose timers and 2 watchdog timers.

11.1.1. General-Purpose Timer

The SOC contains a number of general-purpose timers that can be used as general counters or interrupt generators. Each counter (except TIMER E) can be configured as a periodic counter (for generating periodic interrupts) or a simple count-down and stop counter. Additionally, the timers have a programmable count rate ranging from 1uS to 1mS. The table below outlines the general-purpose timers available in the chip.

Table 11. General-Purpose Timer

Timer	Timebase Options	Counter size	Comment
Timer A	1uS, 10uS, 100uS, 1mS	16-bits	The 16-bit counter allows the timer to generate interrupts as infrequent as every 65.535 Seconds
Timer B	1uS, 10uS, 100uS, 1mS	16-bits	
Timer C	1uS, 10uS, 100uS, 1mS	16-bits	
Timer D	1uS, 10uS, 100uS, 1mS	16-bits	
Timer E	System clock, 1uS, 10uS, 100uS, 1mS	32-bits	Doesn't generate an interrupt. This is a count up counter that counts from 0 to 0xFFFFFFFF. The counter can be written at any time to reset the value to 0.
Timer F	1uS, 10uS, 100uS, 1mS	16-bits	
Timer G	1uS, 10uS, 100uS, 1mS	16-bits	
Timer H	1uS, 10uS, 100uS, 1mS	16-bits	
Timer I	1uS, 10uS, 100uS, 1mS	16-bits	
Timer A-AO	System clock, 1uS, 10uS, 100uS	16-bits	Used in the Always On domain to generate interrupts for the AO-CPU
Timer E-AO	System clock	32-bits	This Always On counter doesn't generate an interrupt. Instead it simply counts up from 0 to 0xFFFFFFFF. The counter can be written at any time to reset the value to 0.

11.1.2. Watchdog Timer

There are also two watchdog timers in both AO and EE domain.

The AO Domain watchdog timer is driven from the system clock (typically 157Mhz). It is a 16-bit counter that is periodically reset by either the AO CPU or the System CPU (A5). This AO-watchdog timer can be used to generate an interrupt of the AO domain. Additionally, the AO-watchdog timer can be used to "enable" a delay generator that can toggle a GPIO pin (currently the TEST_n I/O pad). The "delay generator" allows an interrupt to first be acknowledged by the AO-CPU before the TEST_N pad is toggled. The "delay generator" is programmable from 1 to 65535 system clocks (typically 417uS).

It should be noted that the AO watchdog timer can also be used to reset the AO domain but this feature is only used when operating in a suspend mode (only the AO-domain is powered). As long as the system periodically resets the AO-watchdog timer the WD_GPIO_CNT (delay generator) will not be enabled and the I/O pad will not toggle.

NOTE: The maximum delay between two AO-watchdog periodic resets is about 100mS (assuming a 157Mhz system clock).

The EE Domain watchdog timer is driven by the 24Mhz crystal clock and can be used to generate an interrupt to the system CPU (the A5) or optionally, the watchdog timer can completely reset the chip (causing a cold boot). There are a few registers that are not affected by watchdog timer. These registers are only reset by the external RESET_n I/O pad and can be used to store information related to a possible watchdog event. As long as the system CPU periodically resets the EE-watchdog timer, it will never timeout and cause an interrupt or system reset.

NOTE: The maximum delay between two EE-watchdog periodic resets is about 8.3 Seconds. This time is independent of the system clocks and is driven by the external 24Mhz crystal.

11.2. Register Definitions

11.2.1. ISA_TIMER_MUX CBUS: 0x2650

Bit(s)	R/W	Default	Description
31-20	R	0	unused
19	R/W	1	TIMERD_ EN: Set to 1 to enable Timer D
18	R/W	1	TIMERC_ EN: Set to 1 to enable Timer C
17	R/W	1	TIMERB_ EN: Set to 1 to enable Timer B
16	R/W	1	TIMERA_EN: Set to 1 to enable Timer A
15	R/W	0	TIMERD_MODE: If this bit is set to 1, then timerD is a periodic . 0 = one-shot timer
14	R/W	0	TIMERC_MODE: If this bit is set to 1, then timerC is a periodic . 0 = one-shot timer
13	R/W	1	TIMERB_MODE: If this bit is set to 1, then timerB is a periodic . 0 = one-shot timer
12	R/W	1	TIMERA_MODE: If this bit is set to 1, then timerA is a periodic . 0 = one-shot timer
11	R	0	unused
10-8	R/W	0x1	TIMER E input clock selection: 000: System clock 001: 1uS Timebase resolution 010: 10uS Timebase resolution 011: 100uS Timebase resolution 100: 1mS timebase NOTE: The mux selection for Timer E is different from timer A, B, C and D
7-6	R/W	0x0	TIMER D input clock selection: See TIMER A below
5-4	R/W	0x0	TIMER C input clock selection: See TIMER A below
2-3	R/W	0x0	TIMER B input clock selection: See TIMER A below
1-0	R/W	0x0	TIMER A Input clock selection: These bits select the input timebase for the counters for TimerA 00: 1uS Timebase resolution 01: 10uS Timebase resolution 10: 100uS Timebase resolution 11: 1mS Timebase resolution

11.2.2. ISA_TIMERA CBUS: 0x2651

Timer A is a 16 bit count DOWN counter driven by the clock selected in register 0x01000530. TIMER A will count down from some value to zero, generate an interrupt and then re-load the original start count value. This timer can be used to generate a periodic interrupt (e.g. interrupt every 22 uS).

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER A.

11.2.3. ISA_TIMERB CBUS: 0x2652

Timer B is just like Timer A.

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER B

11.2.4. ISA_TIMERC CBUS: 0x2653

Timer C is just like Timer A.

Bit(s)	R/W	Default	Description
31-16	R	0	unused
15-0	R/W	0x0	Starting count value. Write this value to start TIMER C

11.2.5. ISA_TIMERD CBUS: 0x2654

Timer D is identical to Timer A.

Bit(s)	R/W	Default	Description
31-16	R	0	unused
15-0	R/W	0x0	Starting count value. Write this value to start TIMER D

11.2.6. ISA_TIMERE CBUS: 0x2655

Timer E is simply a 32-bit counter that increments at a rate set by register 0x2650. To reset the counter to zero, simply write this register with any value. The value below is a read-only value that reflects the current count of the internal counter. This register can be used by software to simply provide a polling delay loop based on a programmable timebase.

Bit(s)	R/W	Default	Description
31-0	R	-	Current value of Timer E. Write this register with any value to clear the counter.

11.2.7. ISA_TIMER_MUX1 CBUS: 0x2664

Bit(s)	R/W	Default	Description
31-20	R	0	unused
19	R/W	1	TIMERD_ EN: Set to 1 to enable Timer D
18	R/W	1	TIMERC_ EN: Set to 1 to enable Timer C
17	R/W	1	TIMERB_ EN: Set to 1 to enable Timer B
16	R/W	1	TIMERA_EN: Set to 1 to enable Timer A
15	R/W	0	TIMERD_MODE: If this bit is set to 1, then timerD is a periodic . 0 = one-shot timer
14	R/W	0	TIMERC_MODE: If this bit is set to 1, then timerC is a periodic . 0 = one-shot timer
13	R/W	1	TIMERB_MODE: If this bit is set to 1, then timerB is a periodic. 0 = one-shot timer
12	R/W	1	TIMERA_MODE: If this bit is set to 1, then timerA is a periodic. 0 = one-shot timer
11	R	0	unused
10-8	R/W	0x1	TIMER E input clock selection: 000: System clock 001: 1uS Timebase resolution 010: 10uS Timebase resolution 011: 100uS Timebase resolution 100: 1mS timebase NOTE: The mux selection for Timer E is different from timer A, B, C and D
7-6	R/W	0x0	TIMER D input clock selection: See TIMER A below
5-4	R/W	0x0	TIMER C input clock selection: See TIMER A below
2-3	R/W	0x0	TIMER B input clock selection: See TIMER A below
1-0	R/W	0x0	TIMER A Input clock selection: These bits select the input timebase for the counters for TimerA 00: 1uS Timebase resolution 01: 10uS Timebase resolution 10: 100uS Timebase resolution 11: 1mS Timebase resolution

11.2.8. ISA_TIMERF CBUS: 0x2665

Timer F is a 16 bit count DOWN counter driven by the clock selected in register 0x01000530. TIMER A will count down from some value to zero, generate an interrupt and then re-load the original start count value. This timer can be used to generate a periodic interrupt (e.g. interrupt every 22 uS).

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER A.

11.2.9. ISA_TIMERG CBUS: 0x2666

Timer G is just like Timer F.

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER B

11.2.10. ISA_TIMER_H CBUS: 0x2667

Timer H is just like Timer F.

Bit(s)	R/W	Default	Description
31-16	R	0	unused
15-0	R/W	0x0	Starting count value. Write this value to start TIMER C

11.2.11. AO_TIMER_REG 0xc810004c

Timer controls.

Bit(s)	R/W	Default	Description
31-5	R/W	0	Unused

4	R/W	0	TIMER_E_EN
3	R/W	0	TIMER_A_EN
2	R/W	0	TIMER_A_MODE: 1 = periodic, 0 = one-shot
1-0	R/W	0	TIMER_CLK_MUX: 00 = TimerA clock = Media CPU clock 01 = Timer A clock = 1uS ticks 10 = Timer A clock = 10uS ticks 11 = Timer A clock = 100uS ticks

11.2.12. AO_TIMERA_REG 0xc8100050

Timer A starts at a non-zero value and decrements to 0. When timer A reaches a count of 0 it will re-load with the TIMER_A_TCNT value.

Bit(s)	R/W	Default	Description
31-16	R	0	TIMER A current count.
15-0	R/W	0	TIMER_A_TCNT: Timer A Terminal count

11.2.13. AO_TIMERE_REG0 0xc8100054

If this register is written (with any value), then Timer E is reset to 0. Immediately after being cleared, timer E will start incrementing at a clock rate equal to the clock used for the Media CPU..

Bit(s)	R/W	Default	Description
31-0	R/W	0	TIMER E current Count

11.2.14. WATCHDOG_TC: 0x2640

EE domain watchdog

Bit(s)	R/W	Default	Description
31-28	R	0	Unused
27-24	R/W	0	There are 4 bits corresponding to the 4 A9 CPU's. for Dual A9 CPU's only bits[25:24] are valid. There are two WDRESETREQ bits can come from the A9 Dual core. If bit[24] is set and WDRESETREQ[0] goes high, then a general chip level watchdog reset is executed. The same applies to WDRESETREQ[1] and bit[25].
19	R/W	0	**ENABLE**: Set to '1' to enable the watchdog reset
18	R/W	0	**IRQ_ONLY**: If this bit is set to 1, then the watchdog timer will not reset the chip. Instead it will generate an interrupt. This is useful for debugging watchdog timer problems.
17	R/W	0	**SIM:** Reserved…keep at 0
15-0	R/W	0	**TCNT**: This value corresponds to the time before the watchdog timer issues a chip reset. An internal counter counts up from zero in increments of 128uS until it reaches the TERMINAL_COUNT value. If the internal counter reaches a value of TERMINAL_COUNT, the chip will be reset. . 1 second = (7812 * 128uS)

11.2.15. WATCHDOG_RESET: 0x2641

Write this register to reset the watchdog timer

Bit(s)	R/W	Default	Description
31-22	R/W	0	Unused
15-0	R	0	**WATCHDOG_COUNT**: Reading this register returns the watchdog timer count. Writing this register (with any value) resets the watchdog timer (and this count). The watchdog timer counts in increments of 10uS.

11.2.16. AO_WATCHDOG_REG 0xc8100044

AO domain watchdog

Bit(s)	R/W	Default	Description
31-26	R/W	0	Unused
25	R/W	0	IRQ Only. If this bit is set to 1, then an interrupt will be generated instead of a reset that resets the Always On domain.
24	R/W	0	Watchdog Enable
23-0	R/W	0x186a0	This value corresponds to the time before the watchdog timer issues a chip reset. The counter counts the clocks associated with the Media CPU.

11.2.17. AO_WATCHDOG_RESET 0xc8100048

Writing this register resets the watchdog timer.

Bit(s)	R/W	Default	Description
31-24	R	0	Unused
23-0	R	0	Watchdog timer count

12. Real Time Clock

12.1. Overview

The S805 integrates a Real Time Clock (RTC) that can operate using the backup battery while the system power is off. With an external 32.768kHz crystal oscillator, it provides a low frequency reference clock and second-based real time clock to the whole system.

12.2. Features

The RTC provides the following features:

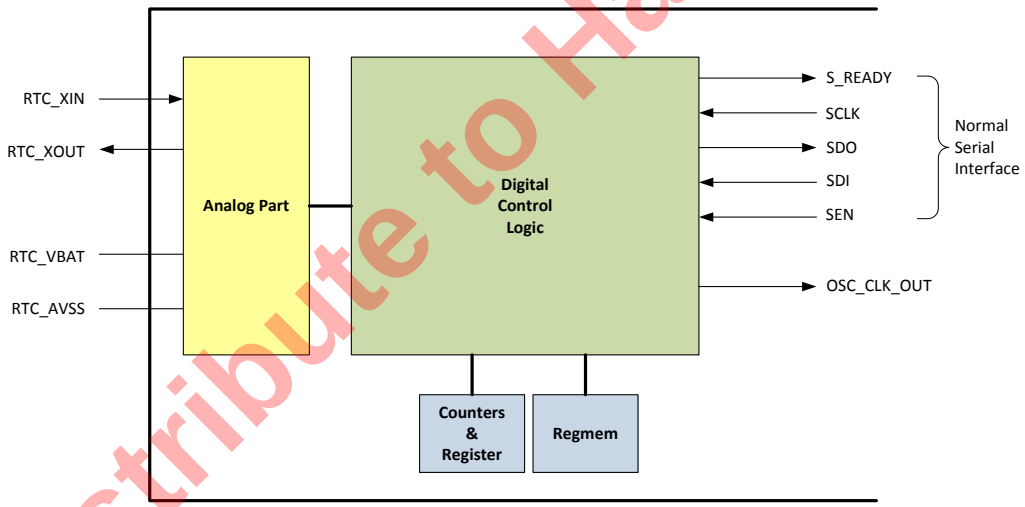
- Serial interface between RTC and chip core.
- A real-time counter that the core can read/write
- Adjustable RTC counter to fine-tune the crystal oscillator accuracy up to 2 ppm.
- Programmable register to the crystal oscillator pad for current control
- Counter-based general purpose signal output for alarm and other timer based applications externally and internally
- Four 32-bit register based memories for system power-off data storage

12.3. Functional Description

12.3.1. Functional Block Diagram

There are several sub-modules in the RTC module.

Figure 5. RTC Functional Block Diagram



The digital control logic: it implements a ripple counter to divide the 32.768KHz oscillator clock input down to 1 Hz, and a counter which increments by one per second. The processor in the chip core can initialize the counter by writing a new value. And it can read the counter value, for example, after a power outage, the real-time information can be retrieved by the processor via reading the counter value from RTC. The accuracy of the oscillator clock can be fine-tuned by CPU in way of dropping or inserting a second to the real-time counter.

The analog part: The crystal oscillator driver and voltage regulator are implemented in the analog part. The regulator is powered by RTC_VBAT pin and provides power supply to all sub-blocks of RTC Unit. The analog part also outputs the crystal oscillator clock as the reference clock to all digital parts.

In the **counters and registers part**, the control registers are used by RTC Unit and their operation requires reference clock.

- The RTC counter: The counter is loaded with a new initial value when its “load” pin is high, otherwise it increments once per second on “clk”.

The **regmem part** is register-based memory for system CPU usage only.

The chip core and RTC operates in different power domain – while core domain may be out of power, RTC domain can take power from batteries, and therefore is assumed always powered. The signals crossing the two domains are gated with isolation cells for power-down protection.

12.4. RTC Peripheral Register Description

The following registers are physically located in chip core's Periphs module, not in RTC. However, by accessing these registers the processor is able to initiate serial I/F transfer and read/write the RTC counter, and memory inside RTC.

12.4.1. RTC_ADDR0 0xc8100740

Bit(s)	R/W	Default	Description
31-24	R/W	0	**STATIC_REG: this value represents bits[7:0] of the static value written inside the RTC analog block. Bits[15:8] are held in register 4 below.** The RTC block has an internal 8-bit register that saves state when the power to the chip is removed. This 8-bit register is used to control the RTC oscillator pad. These 8-bits correspond to the 8-bits to be written to the static register inside the RTC block
23	R/W	0	Unused
22	R	0	SERIALIZER_BUSY: The static register can be written manually or using an automatich state-machine that writes the appropriate code to the static register in the RTC block. 1 = automatic serializer is busy
21	R/W	0	Unused. Set to 0
20	R/W	0	MANUAL_CLOCK
19	R/W	0	Unused. Set to 0
18	R/W	0	Unused. Set to 0
17	R/W	0	AUTO_SERIALIZER_START: Write this bit with a '1' to start the auto-serializer. This bit doesn't need to be set to '0'.
16	R/W	0	ONE_SHOT_POLARITY: This bit sets the polarity of the ONE_SHOT level from the RTC block to the digital logic
15-12	R/W	0	Reserved
11	R/W	0	Reserved
10-8	R/W	0	Unused
7-6	R/W	0	Reserved
5	R/W	0	TEST_MODE: Set to 0 for normal operation (clock for the RTC logic comes from the normal RTC oscillator pad). Set to 1 to bypass the external oscillator pad and mux in a clock generated by TEST_CLK bit below.
4	R/W	0	TEST_CLK: Set to 0 for normal operation (it is really ignored, but 0 is safe). You can use this as a software-generated clock for the RTC block in AM_ANALOG_TOP, if TEST_MODE above is set to 1.
3	R/W	0	TEST_BYPASS: Set to 0
2	R/W	0	SDI
1	R/W	0	SEN
0	R/W	0	SCLK

12.4.2. RTC_ADDR1 0xc8100744

Bit(s)	R/W	Default	Description
31-16	R	0	Unused
15-12	R	-	Reserved
11-4	R	0	Unused
3	R	-	ONE_SHOT_LEVEL: Current level of the ONE_SHOT signal from the RTC block
2	R	-	Reserved
1	R/W	-	S_READY
0	R	-	SDO

12.4.3. RTC_ADDR2 0xc8100748

Bit(s)	R/W	Default	Description
31-0	R	0	OSC_CLK_COUNT: The 32khz oscillator can be measured using this register. The value represents the number of rising edges detected for a specified time (MSR_GATE_TIME…see below). Typically this value will increment at a 32khz rate.

12.4.4. RTC_ADDR3 0xc810074c

Bit(s)	R/W	Default	Description
31-30	R/W	0	Unused
29	R/W	0	USE_CLK_TB: This bit can be set to use the system clock as the measure clock for faster oscillator clock measurements.
28	R/W	0	Use Nike-D RTC: set to 0

Bit(s)	R/W	Default	Description
27-26	R/W	0	AUTO_TB_SEL: The automatic programming logic can be run at a fast or slow rate depending on these bits: 0 = 111nS, 1 = 1uS, 2 = 10uS, 3 = 100uS timebase.
25-23	R/W	0	FILTER_SEL: The GPI input signal from the RTC block can be filtered. A value of 0 indicates no filtering. A value of 7 indicates the more filtering
22-21	R/W	0	FILTER_TB: These bits set the filtering timebase: 0 = 1uS, 1 = 10uS, 2 = 100uS, 3 = 1mS
20	R	0	MSR_BUSY: If this bit is high, then the oscillator counter is busy measuring. When this bit is low, then the OSC_CLK_COUNT value above represents the number of rising edges of the oscillator clock for a given MSR_GATE_TIME.
19	R/W	0	Unused
18	R/W	0	FAST_CLK_MODE:
17	R/W	0	COUNT_ALWAYS: If this bit is set, then the OSC_CLK_COUNT is incremented for every rising edge of the 32khz oscillator. This mode can be used to simply detect if the 32khz RTC oscillator is working. That is, read the OSC_CLK_COUNT value, set this bit high, wait for 40uS, then read the OSC_CLK_COUNT value. If it didn't change, then the 32khz oscillator is not oscillating. The COUNT_ALWAYS bit is also useful for measuring the 32khz over hours to get a very accurate measurement of the 32khz (assuming the system clock oscillator is more accurate).
16	R/W	0	MSR_EN: Set this bit high to measure the 32khz oscillator for a specified number of milliseconds. Keep this bit high until the MSR_BUSY bit above goes low indicating that the measurement is done. When the measurement is done, set this bit low
15-0	R/W	0	MSR_GATE_TIME: This value sets the number of millisecond that should be used to measure the 32khz oscillator. If the MSR_GATE_TIME is set to 10mS, then we should expect to see a value of 327 in the OSC_CLK_COUNT register.

12.4.5. RTC_REG4 0xc8100750

Bit(s)	R/W	Default	Description
31-8	R	0	Unused
7-0	R/W	0	STATIC_VALUE: This value represents **bits[15:8]** of the static value written to the RTC block when the auto-serializer state-machine is used. **See bits[31:24] of register 0 for the lower 8 bits.**

12.5. RTC Register Description

The following registers are physically located in RTC. These registers are programmed via normal serial interface.

Table 12. RTC Registers

Addr	Name	RW	Function
0	RTC_COUNTER	RW	Program RTC counter value
1	UNUSED	R	Reserved
2	RTC_SEC_ADJUST	RW	Control digital time adjustment
3	UNUSED	R	Reserved
4	RTC_REGMEM_0	RW	Register-based memory 0
5	RTC_REGMEM_1	RW	Register-based memory 1
6	RTC_REGMEM_2	RW	Register-based memory 2
7	RTC_REGMEM_3	RW	Register-based memory 3

12.5.1. RTC_COUNTER

Bit(s)	R/W	Default	Description
31:0	RW	0	Write this to set current time in the unit of second.

12.5.2. RTC_SEC_adjust

Bit(s)	R/W	Default	Description
25	RW	0	secdiv128_adj_dsr. One-off sec_div128 cycle adjustment valid flag. After setting this bit: If read back this bit is 1, then the adjustment is still pending; If read back this bit is 0, then the adjustment is completed.
24	RW	0	secdiv128_adj_val. Define how to adjust time by a single sec_div128 cycle, must be qualified by secdiv128_adj_dsr.

Bit(s)	R/W	Default	Description
			0 = One-time remove a 1/128 second; 1 = One-time insert a 1/128 second.
23	W	0	adj_valid. Set to 1 to make sec_adjust_ctrl and match_counter effective. This is useful when we only want to switch between adjusted and un-adjusted second pulses for observation, but do not want to change time adjustment control.
22	RW	0	monitor_use_adjusted. For debugging purpose. 0 = sec_pulse_out outputs un-adjusted second pulse; 1 = sec_pulse_out outputs adjusted second pulse.
21	RW	0	Reserved
20:19	RW	0	sec_adjust_ctrl. CPU command to control time adjustment. 0x = No adjustment; 10 = Swallow a second once every match_counter+1 seconds; 11 = Insert a second once every match_counter+1 seconds.
18:0	RW	0	match_counter. In the unit of second. Once every match_counter+1 seconds, a second pulse is to be swallowed/inserted.

12.5.3. RTC_REGMEM_ADDR_0/1/2/3

Bit(s)	R/W	Default	Description
31:0	RW	0	For CPU to stored data.

Section II INTERFACE

This part describes the S805 peripheral interfaces from the following aspects:

- MMC/SD/SDIO CONTROLLER
- INTER-INTEGRATED CIRCUIT (I2C)
- SERIAL PERIPHERAL INTERFACE COMMUNICATION CONTROLLER
- SERIAL PERIPHERAL INTERFACE FLASH CONTROLLER
- UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER
- INFRARED REMOTE
- UNIVERSAL SERIAL BUS
- PULSE-WIDTH MODULATION
- SAR ADC
- ETHERNET MAC

13. MMC/SD/SDIO CONTROLLER

13.1. Overview

MMC/SD/SDIO host controller is designed for connecting varied SD/MMC Card, SDIO device (e.g. Wi-Fi module), or eMMC protocol compatible memory with high throughput.

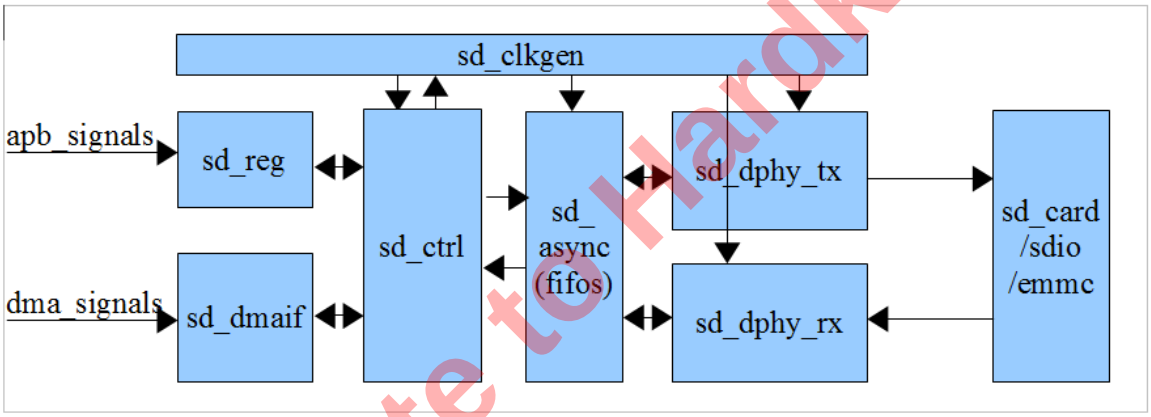
13.2. Features

- Support SD/SDIO Spec 3.x, eMMC Spec 4.5x, support SDSC, SDHC, SDXC memory card.
- Provide UHS-I both 3.3v and 1.8v signaling, and support DS, HS, SDR12, SDR25, SDR50, DDR50, and SDR104 speed modes.
- Support both PIO and DMA mode.
- Back compatible to SD Card Spec 2.0 and 1.0.
- 1 bit, 4 bits, 8 bits date lines supported (8 bits only for MMC).

13.3. Functional Description

There are several sub-modules in MMC/SD/SDIO host controller:

Figure 6. MMC/SD/SDIO Controller Block Diagram



- sdhc_reg is APB slave for register configurations of whole SDHC module.
- sdhc_ctrl is the main controller, which is responsible for coordinating other modules. This module runs in system clock domain.
- sdhc_dmaif is the DMA engine, which is responsible for issuing Wr/Rd DDR request if thresholds get satisfied. This module runs in system clock domain.
- sdhc_async is responsible for synchronizing the signals from different clock domain. Asynchronous FIFO is also here. This module runs in both system clock domain and tx/rx clock domain.
- sdhc_dphy_tx is responsible for transmitting the command and data. This module runs in tx clock domain.
- sdhc_dphy_rx is responsible for receiving the response, crc busy and data. This module runs in rx clock domain.
- crt_sdhc is responsible for creating or enabling the system/tx/rx clock. This module is placed in dedicated clock module.

13.4. Timing Specification

Figure 7. MMC/SD/SDIO Timing Diagram

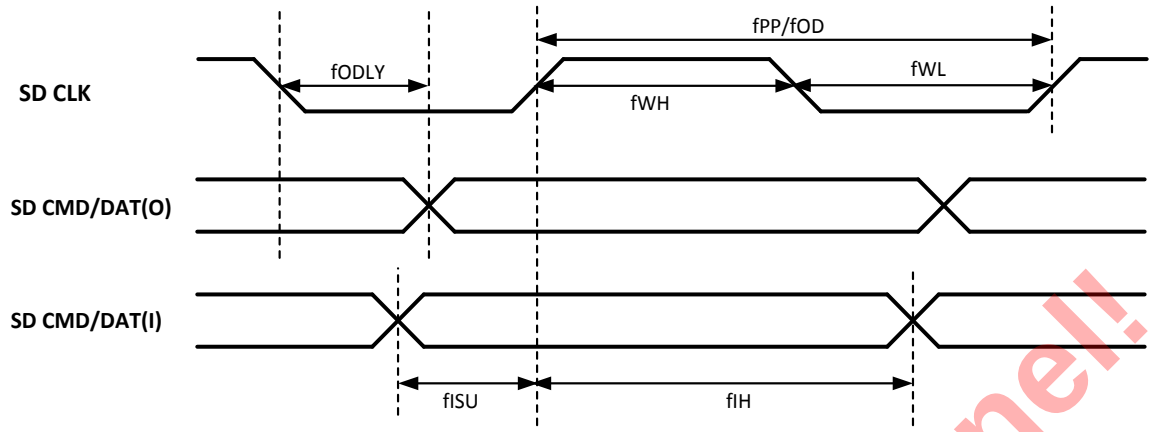


Table 13. SDHC Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
fPP	Clock Frequency Low Speed	0	400	KHz	
	Clock Frequency Full/High Speed	0	25/80	MHz	
	Clock Frequency UHS-I Speed	0	100	MHz	
fOD	Clock Frequency Identification Mode	100	400	KHz	
tWL	Clock Low Time	5		ns	
tWH	Clock High Time	5		ns	
tODLY	Output Delay	-2.5	2.5	ns	
tISU	Input Setup Time	4		ns	
tIH	Input Hold Time	4		ns	

Table 14. SDIO Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
fPP	Clock Frequency Low Speed	0	400	KHz	
	Clock Frequency Full/High Speed	0	25/50	MHz	
fOD	Clock Frequency Identification Mode	100	400	KHz	
tWL	Clock Low Time	8		ns	
tWH	Clock High Time	8		ns	
tODLY	Output Delay	-3.5	3.5	ns	
tISU	Input Setup Time	6		ns	
tIH	Input Hold Time	6		ns	

13.5. Register Description

13.5.1. SD_REG0_ARGU 0xc1108e00

Bit(s)	R/W	Default	Description
31:0	RW	0	command_argu (Command argument)

Note1: Response can be read from this register combined with SD_REG6_PDMA[3:1].

13.5.2. SD_REG1_SEND 0xc1108e04

Bit(s)	R/W	Default	Description
31:12	RW	0	total_pack (Total package number for writing or reading)
11	RW	0	data_stop(0:rx or tx, 1:data stop, ATTN:will give rx a softreset)
10	RW	0	data_direction(0:data rx,1:data tx)
9	RW	0	response_no_crc(0:check crc7 1:don't check crc7)
8	RW	0	response_length(0:48bit 1:136bit)
7	RW	0	command_has_data(0:no data 1:has data)
6	RW	0	command_has_resp(0:no resp 1:has resp)
5:0	RW	0	command_index(Command index)

Note1: data_stop need to be set when sending CMD12.

13.5.3. SD_REG2_CNTL 0xc1108e08

Bit(s)	R/W	Default	Description
31:29	RW	0	Tx Endian Control
28	RW	0	Dat0 Interrupt selection, 0:Busy check after response, 1:any rising-edge of dat0
27	RW	0	SDIO IRQ Mode, 0:Normal Mode, 1:Support Data Block Gap(need turn off clock gating)
26:24	RW	0	Rx Endian Control
23:20	RW	0x8	rc_period(Period between response/cmd and next cmd,default 8)
19:13	RW	0x40	rx_timeout(cmd or wcrc Receiving Timeout, default 64)
12:4	RW	0	pack_len (0:512Bytes,1:1,2:2,...,511:511Bytes, ATTN:setting limmit:must be a multiple of 8bytes)
3	RW	0	(0:Check sd write crc result, 1:Disable Tx crc check)
2	RW	0	ddr_mode(0:SDR mode,1:DDR mode)
1:0	RW	0	dat_type(0:1 bit,1:4 bits,2:8 bits,3:reserved)

Note1: DDR mode will be supported in late version.

13.5.4. SD_REG3_STAT 0xc1108e0c

Bit(s)	R/W	Default	Description
31:24	R	0	Reserved
23:20	R	0	(DAT[7:4])
19:14	R	0	(TxFIFO count)
13:8	R	0	(RxFIFO count)
7	R	0	Reserved
6	R	0	Reserved
5	R	0	(CMD)
4:1	R	0	(DAT[3:0])
0	R	0	(0:Ready for command,1:busy)

13.5.5. SD_REG4_CLKC 0xc1108e10

Bit(s)	R/W	Default	Description
31:25	RW	0	Reserved
24	RW	0	Clock JIC for clock gating control(1'b1 will turn off clock gating)
23	RW	0	clk_ctl_enable. Every time parameters are set, need turn it off and then turn it on.
22	RW	0	Rx clock feedback enable
21:20	RW	0	Rx clock phase_sel(0:0,1:90,2:180,3:270) related to Tx clock
19	RW	0	clk_en
18:16	RW	0	clk_in_sel(0:osc 1:fclk_div2(1GHz) 2:fclk_div3(666.67MHz) 3:fclk_div5(400MHz) 4..7:reserved)
15:0	RW	0	clk_div(0: dont set it, 1:div2, 2:div3, 3:div4...)

13.5.6. SD_REG5_ADDR 0xc1108e14

Bit(s)	R/W	Default	Description
31:0	RW	0	DMA Address

13.5.7. SD_REG6_PDMA 0xc1108e18

Bit(s)	R/W	Default	Description
31:28			Reserved (Don't write)
27	RW	0x0	RxFIFO manual flush(self clear)
26:21	RW	0x3	TxFIFO threshold(<=txth, will request read)
20:15	RW	0x3	RxFIFO threshold(>=rxth, will request write)
14:10	RW	0x3	Number in one Read request burst(0:1,1:2...)
9:5	RW	0x3	Number in one Write request burst(0:1,1:2...)
4	RW	0	dma_urgent(0:Not Urgent,1:Urgent)
3:1	RW	0	pio_rdresp(0:[39:8] 1:1st 32bits,2:2nd...,6 or 7:command argument)
0	RW	0	dma_mode(0:PIO mode,1:DMA mode)

Note1: dma_urgent is just set when bandwidth is very tight

Note2: pio_rdresp need to be combined with REG0_ARGU; For R0, when 0, reading REG0 will get the normal 32bit response; For R2, when 1, reading REG0 will get CID[31:0], when 2, get CID[63:32], and so on; 6 or 7, will get original command argument.

13.5.8. SD_REG7_MISC 0xc1108e1c

Bit(s)	R/W	Default	Description
31:29	RW	0	Reserved
28	RW	0	(0:auto stop mode 1:manual stop mode)
27:22	RW	0	(Thread ID)
21:16	RW	0	(Burst Number)
15:10	RW	0	(TXFIFO Empty Threshold,default 0)
9:4	RW	0x1e	(RXFIFO Full Threshold,default 30)
3:2	RW	0	(rx dat line delay control)
1:0	RW	0	(rx cmd line delay control)

13.5.9. SD_REG8_DATA 0xc1108e20

Bit(s)	R/W	Default	Description
31:0	RW	0	DATA for pio write and read

13.5.10. SD_REG9_ICTL 0xc1108e24

Bit(s)	R/W	Default	Description
31:18	RW	0	Reserved
17:16	RW	0	sdio dat1 interrupt mask windows clear delay control,0:2cycle 1:1cycles
15	RW	0	Reserved
14	RW	0	IRQ EN for Additional SDIO DAT1 Interrupt
13	RW	0	IRQ EN for TxFIFO Empty
12	RW	0	IRQ EN for RxFIFO Full
11	RW	0	IRQ EN for DMA Done
10	RW	0	IRQ EN for SDIO DAT1 Interrupt
9	RW	0	IRQ EN for TxFIFO count < threshold
8	RW	0	IRQ EN for RxFIFO count > threshold
7	RW	0	IRQ EN for Data Transfer Completed ok
6	RW	0	IRQ EN for One Package Data Failed (CRC Error)
5	RW	0	IRQ EN for One Package Data Failed (Timeout Error)
4	RW	0	IRQ EN for One Package Data Completed ok
3	RW	0	IRQ EN for Data bit0 change to not busy from busy
2	RW	0	IRQ EN for Response is received Failed (CRC Error)
1	RW	0	IRQ EN for Response is received Failed (Timeout Error)
0	RW	0	IRQ EN for Response is received OK

13.5.11. SD_REGA_ISTA 0xc1108e28

Bit(s)	R/W	Default	Description
31:13	RW	0	Reserved
14	RO	0	IRQ STA for Additional SDIO DAT1 Interrupt
13	RW	0	IRQ STA for TxFIFO Empty(W1C)
12	RW	0	IRQ STA for RxFIFO Full(W1C)
11	RW	0	IRQ STA for DMA Done (W1C)
10	RW	0	IRQ STA for SDIO DAT1 Interrupt (W1C)
9	RW	0	IRQ STA for TxFIFO count < threshold (W1C)
8	RW	0	IRQ STA for RxFIFO count > threshold (W1C)
7	RW	0	IRQ STA for Data Transfer Completed ok (W1C)
6	RW	0	IRQ STA for One Package Data Failed (CRC Error) (W1C)
5	RW	0	IRQ STA for One Package Data Failed (Timeout Error) (W1C)
4	RW	0	IRQ STA for One Package Data Completed ok (W1C)
3	RW	0	IRQ STA for Data bit0 change to not busy from busy (W1C)
2	RW	0	IRQ STA for Response is received Failed (CRC Error) (W1C)
1	RW	0	IRQ STA for Response is received Failed (Timeout Error) (W1C)
0	RW	0	IRQ STA for Response is received OK (W1C)

Note1: W1C is write one clear.

13.5.12. SD_REGB_SRST 0xc1108e2c

Bit(s)	R/W	Default	Description
31:6	RW	0	Reserved
5	RW	0	Soft reset for DMA IF(self clear)
4	RW	0	Soft reset for DPHY TX
3	RW	0	Soft reset for DPHY RX
2	RW	0	Soft reset for TX FIFO(self clear)
1	RW	0	Soft reset for RX FIFO(self clear)
0	RW	0	Soft reset for MAIN CTRL(self clear)

Note1: Soft reset for DPHY TX/RX needs programmer to set it and then clear it manually.

14. INTER-INTEGRATED CIRCUIT (I2C)

14.1. Overview

Inter-Integrated Circuit (IIC or I2C) is a multi-slave serial communication bus between ICs. S805 integrates the I2C interface and signals allowing communications with other I2C peripheral devices.

14.2. Features

The I^{2}C Master Module has the following features:

- Support for 7-bit and 10-bit addressable devices
- Programmable bus speed including standard speed (100kbits/s) and fast speed (400kbits/sec)
- Error transfer detection
- "Transfer complete" indication by polling or interrupt (Interrupts handled by the ISA module. See the ISA module for details).
- Internal buffer holding up to 8 bytes for transfer (in either direction)
- Flexible architecture allowing the software to dictate the format of the I^{2}C bit streams
- Manual setting of the I^{2}C bus to accommodate a software only mode

14.3. Timing Specification

There are two modes to the I2C master interface: Standard (100khz) and fast (400khz).

Figure 8. I2C Interface Timing Diagram

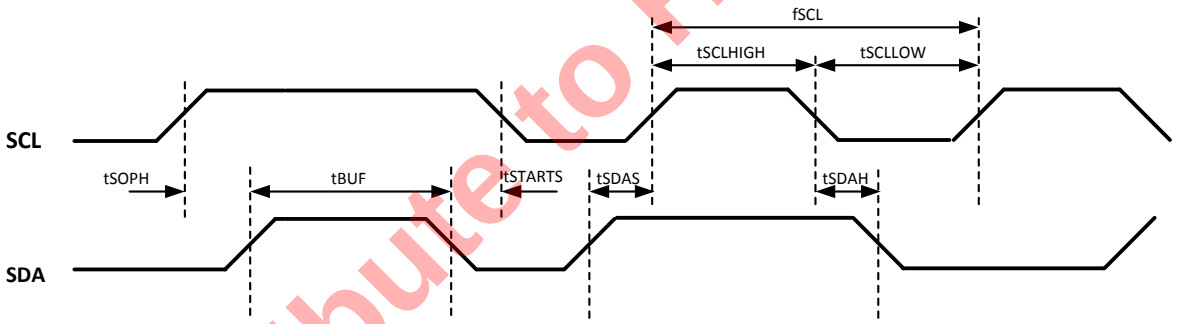


Table 15. I2C Interface Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
fSCL	SCL clock frequency		Std. 100 Fast 400	kHz	
tSDAS	Data setup before the rising edge of clock	Std. 4.0 Fast 0.6		µs	
tSDAH	Data hold after the falling edge of clock	Std. 4.0 Fast 0.6	-	µs	
tSTARTS	Clock hold time after the falling edge of SDA when a START command is issued	Std. 4.0 Fast 0.6	-	µs	
tSTOPH	Clock setup time before the rising edge of SDA when a STOP command is issued	Std. 4.0 Fast 0.6		µs	
tBUF	Delay between start and stop	Software Controlled		µs	
tSCLLOW	Clock LOW time	Std. 8.0 Fast 1.3		µs	
tSCLHIGH	Clock HIGH time	Std. 8.0 Fast 1.3		µs	

14.4. Register Description

14.4.1. I2C_M_0_CONTROL_REG 0x2140

Bit(s)	R/W	Default	Description
31	R/W	0	CNTL_JIC: There is internal logic to dynamically enable the gated clocks. If this gated clock logic doesn't work, you can set this bit to always enable the clock. Setting this bit wastes power.
30	R	0	Unused
29-28	R/W	0	QTR_CLK_EXT: These two bits extend the clock divider to 12 bits: `QTR_CLK = {[29:28],[21:12]}`
27	R	0	unused
26	R	0	Read back level of the SDA line
25	R	0	Read back level of the SCL line
24	R/W	0	Sets the level of the SDA line if manual mode is enabled. If this bit is '0', then the SDA line is pulled low. If this bit is '1' then the SDA line is tri-stated.
23	R/W	0	Sets the level of the SCL line if manual mode is enabled. If this bit is '0', then the SCL line is pulled low. If this bit is '1' then the SCL line is tri-stated.
22	R/W	0	This bit is used to enable manual mode. Manual I²C mode is controlled by bits 12,13,14 and 15 above.
21:12	R/W	0x142	**QTR_CLK_DLY**. This value corresponds to period of the SCL clock divided by 4 Quarter Clock Delay = * System Clock Frequency For example, if the system clock is 133Mhz, and the I²C clock period is 10uS (100khz), then Quarter Clock Delay = * 133 Mhz = 332
11:8	R	-	READ_DATA_COUNT: This value corresponds to the number of bytes READ over the I²C bus. If this value is zero, then no data has been read. If this value is 1, then bits [7:0] in TOKEN_RDATA_REG0 contains valid data. The software can read this register after an I²C transaction to get the number of bytes to read from the I²C device.
7:4	R	-	CURRENT_TOKEN: This value reflects the current token being processed. In the event of an error, the software can use this value to determine the error location.
3	R	-	ERROR: This read only bit is set if the I²C device generates a NACK during writing. This bit is cleared at on the clock cycle after the START bit is set to 1 indicating the start of list processing. Errors can be ignored by setting the ACK_IGNORE bit below. Errors will be generated on Writes to devices that return NACK instead of ACK. A NACK is returned by a device if it is unable to accept any more data (for example because it is processing some other real-time function). In the event of an ERROR, the I²C module will automatically generate a STOP condition on the bus.
2	R	-	STATUS: This bit reflects the status of the List processor: 0: IDLE 1: Running. The list processor will enter this state on the clock cycle after the START bit is set. The software can poll the status register to determine when processing is complete.
1	R/W	0	ACK_IGNORE: Set to 1 to disable I²C ACK detection. The I²C bus uses an ACK signal after every byte transfer to detect problems during the transfer. Current Software implementations of the I²C bus ignore this ACK. This bit is for compatibility with the current Amlogic software. This bit should be set to 0 to allow NACK operations to abort I²C bus transactions. If a NACK occurs, the ERROR bit above will be set.
0	R/W	0	START: Set to 1 to start list processing. Setting this bit to 0 while the list processor is operating causes the list processor to abort the current I²C operation and generate an I²C STOP command on the I²C bus. Normally this bit is set to 1 and left high until processing is complete. To re-start the list processor with a new list (after a previous list has been exhausted), simply set this bit to zero then to one.

14.4.2. I2C_M_0_SLAVE ADDRESS 0x2141

Bit(s)	R/W	Default	Description
31:29	R	0	Reserved
28	R/W	0	USE_CNTL_SCL_LOW: If this bit is set to 1, then bits[27:16] control the SCL low time.
27:16	R/W	0	SCL Low delay. This is a new feature in M8baby. In the previous M8baby design, the SCL low time was controlled by bits[21:12] of the register above. In this design, the SCL delay is controlled independently by these bits.
15:14	R	0	Unused
13-11	R/W	0	SCL_FILTER: A filter was added in the SCL input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
10:8	R/W	0	SDA FILTER: A filter was added in the SDA input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
7:0	R/W	0x00	SLAVE_ADDRESS. This is a 7-bit value for a 7-bit I²C device, or (0xF0 | {A9,A8}) for a 10-bit I²C device. By convention, the slave address is typically stored in by first left shifting it so that it's MSB is D7 (The I²C bus assumes

Bit(s)	R/W	Default	Description
			the 7-bit address is left shifted one). Additionally, since the SLAVE address is always an 7-bit value, D0 is always 0. NOTE: The I²C always transfers 8-bits even for address. The I²C hardware will use D0 to dictate the direction of the bus. Therefore, D0 should always be '0' when this register is set.

14.4.3. I2C_M_0_TOKEN_LIST_REG0 0x2142

The register below describes the first 8 tokens in the token list.

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	8th token in the list to process
27:24	R/W	0x00	7th token in the list to process
23:20	R/W	0x00	6th token in the list to process
19:16	R/W	0x00	5th token in the list to process
15:12	R/W	0x00	4th token in the list to process
11:8	R/W	0x00	3rd token in the list to process
7:4	R/W	0x00	2nd token in the list to process
3:0	R/W	0x00	1st token in the list to process (See the table below for token definitions)

Table 16. Token Definitions

Command Token	Value	Data	Description
END	0x0	N/A	Used to tell the I²C module that this is the end of the Token list. This token is not associated with the I²C bus, but rather with the state-machine that drives the token list processor.
START	0x1	N/A	The START Token is used to tell an I²C device that this is the beginning of an I²C transfer
SLAVE_ADDR-WRITE	0x2	7-bits	This bit-sequence is used to address a device and tell the device it is being WRITTEN
SLAVE_ADDR-READ	0x3	7-bits	This bit sequence is used to address a device and tell the device it is being READ.
DATA	0x4	8-bits	This 8-bit byte sequence is a byte transfer (READ or WRITE). The DATA token corresponds to a WRITE if it follows a SLAVE_ADDR-WRITE token. The DATA token corresponds to a READ if it follows a SLAVE_ADDR-READ token.
DATA-LAST	0x5	8-bits	Used to indicate the last 8-bit byte transfer is a byte transfer of a READ.
STOP	0x6	N/A	This tells the I²C device it is no longer being addressed

Write data associated with the DATA token should be placed into the I2C_TOKEN_WDATA_REG0 or I2C_TOKEN_WDATA_REG1 registers. Read data associated with the DATA or DATA-LAST token can be read from the I2C_TOKEN_RDATA_REG0 or I2C_TOKEN_RDATA_REG1 registers.

14.4.4. I2C_M_0_TOKEN_LIST_REG1 0x2143

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	16th token in the list to process
27:24	R/W	0x00	15th token in the list to process
23:20	R/W	0x00	14th token in the list to process
19:16	R/W	0x00	13th token in the list to process
15:12	R/W	0x00	12th token in the list to process
11:8	R/W	0x00	11th token in the list to process
7:4	R/W	0x00	10th token in the list to process
3:0	R/W	0x00	9th token in the list to process

14.4.5. I2C_M_0_TOKEN_WDATA_REG0 0x2144

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte written for a DATA (write) token.
23:16	R/W	0x00	3rd data byte written for a DATA (write) token.
15:8	R/W	0x00	2nd data byte written for a DATA (write) token.
7:0	R/W	0x00	1st data byte written for a DATA (write) token.

14.4.6. I2C_M_0_TOKEN_WDATA_REG1 0x2145

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte written for a DATA (write) token.
23:16	R/W	0x00	7th data byte written for a DATA (write) token.
15:8	R/W	0x00	6th data byte written for a DATA (write) token.
7:0	R/W	0x00	5th data byte written for a DATA (write) token.

14.4.7. I2C_M_0_TOKEN_RDATA_REG00x2146

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	3rd data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	2nd data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	1st data byte read for a DATA or DATA-LAST (READ) token.

14.4.8. I2C_M_0_TOKEN_RDATA_REG10x2146

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	7th data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	6th data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	5th data byte read for a DATA or DATA-LAST (READ) token.

15. SERIAL PERIPHERAL INTERFACE COMMUNICATION CONTROLLER

15.1. Overview

SPI Communication Controller is designed for connecting general SPI protocol compatible module. This controller allows rapid data communication with less software interrupts than conventional serial communications.

15.2. Features

- Full-duplex synchronous serial interface
- Master/Slave configurable
- Four chip selects to support multiple peripherals
- Transfer continuation function allows unlimited length data transfers
- 64-bit wide by 16-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SS) and SPI Clock (SCLK) are configurable
- Both PIO(Programming In/Out interface) and DMA(Direct Memory Access interface) supported

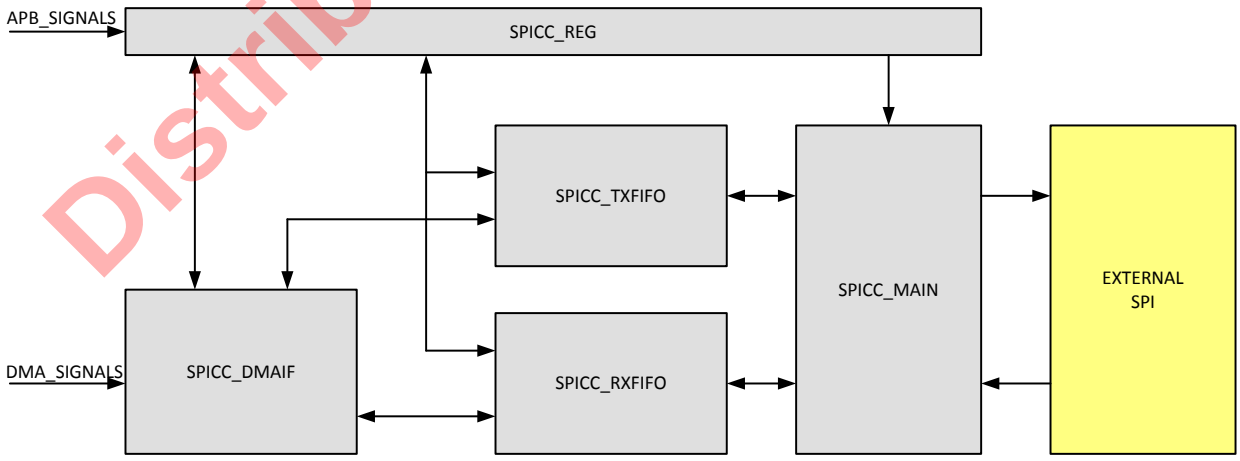
15.3. Functional Description

The following are two SPICC modes of operation:

- Master Mode—When the SPICC module is configured as a master, it uses a serial link to transfer data between the SPICC and an external device. A chip-enable signal and a clock signal are used to transfer data between these two devices. If the external device is a transmit-only device, the SPICC master's output port can be ignored and used for other purposes. To use the internal TXFIFO and RXFIFO, two auxiliary output signals, SS and SPI_RDY, are used for data transfer rate control. The user can also program the sample period control register to a fixed data transfer rate.
- Slave Mode–When the SPICC module is configured as a slave, the user can configure the SPICC Control register to match the external SPI master's timing. In this configuration, SS becomes an input signal, and is used to control data transfers through the Shift register, as well as to load/store the data FIFO.

There are 5 sub-modules in sd host controller, i.e. sd_ctrl, sd_async, sd_dphy_tx, sd_dphy_rx, and sd_clkgen. Transmitting and receiving are using different channel, that means they have different buffer.

Figure 9. SPICC Block Diagram



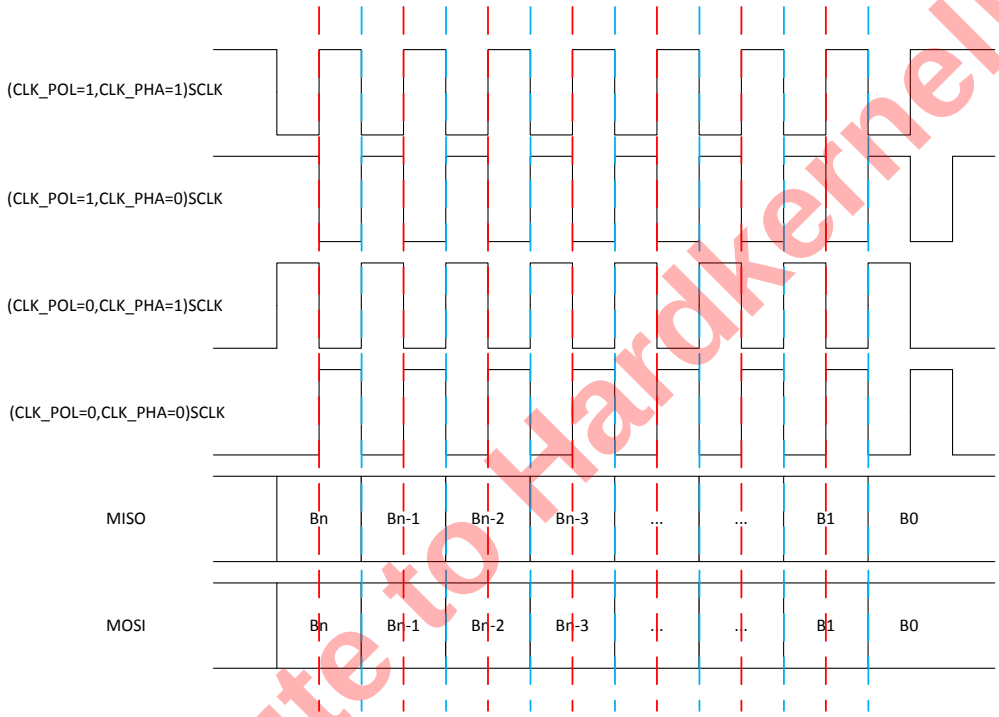
- spicc_reg is driven by host cpu, and spicc_reg is responsible for configuring other modules.
- spicc_dmaif is responsible for dealing with DMA operatoins.
- spicc_txfifo contains a transmission FIFO.
- spicc_rxfifo contains a receiving FIFO.
- spicc_main is responsible for main control of basic spi operation.

Here are SPI External Signals:

Signal Name	I/O	Description
spicc_sclk	IO	SCLK, SPI Clock
spicc_miso	IO	MISO, Master Input Slave Output
spicc_mosi	IO	MOSI, Master Output Slave In
spicc_ss[3:0]	IO	SS, SPI chip Select, Supports up to 4 slaves.
spicc_rdy_i	I	RDY input, Data Ready Input

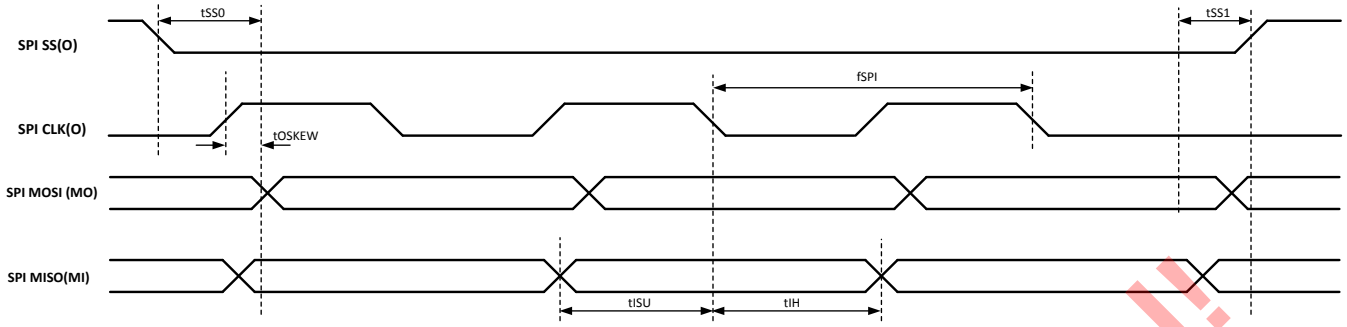
And here is SPI Generic Timing:

Figure 10. SPI Generic Timing



15.4. Timing Specification

Figure 11. SPICC Timing Diagram



Master Mode, CPOL=0, CPHA=1

Table 17. SPICC Master Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
fSPI	Clock Frequency	0	30	MHz	
tOSKEW	Output SKEW	-3.5	3.5	ns	
tISU	Input Setup Time	8		ns	
tIH	Input Hold Time	8		ns	
tSS0	SS active time before transition	8		ns	
tSS1	SS inactive time after transition	8		ns	

Slave Mode, CPOL=0, CPHA=1

Table 18. SPICC Slave Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
fSPI	Clock Frequency	0	30	MHz	
tOSKEW	Output SKEW	-3.5	3.5	ns	
tISU	Input Setup Time	8		ns	
tIH	Input Hold Time	8		ns	
tSS0	SS active time before transition	5		ns	
tSS1	SS inactive time after transition	5		ns	

15.5. Register Description

15.5.1. RXDATA 0xc1108d80

Bit(s)	R/W	Default	Description
31:0	R	0	Rx Data

Note1: when PIO mode, programmer can get data from this register.

15.5.2. TXDATA 0xc1108d84

Bit(s)	R/W	Default	Description
31:0	W	0	Tx Data

Note1: when PIO mode, programmer need send data to this register.

15.5.3. CONREG 0xc1108d88

Bit(s)	R/W	Default	Description
31:19	RW	0	[13]burst_length ([5:0]bit number of one word/package, [12:6]burst length-1)
18:16	RW	0	[3]data_rate (sclk will be divided by system clock with equation: 2^(data_rate+2), Example: if system clock = 128MHz and data_rate=2, sclk's frequency equals 8MHz)
15:14			Reserved
13:12	RW	0	[2]chip_select (00:select ss_0, 01:select ss_1, 10:select ss_2, 11:select ss_3,)
11:10			Reserved
9:8	RW	0	[2]drctl (0:ignore RDY input, 1:Data ready using pin rdy_i's failing edge, 2:Data ready using pin rdy_i's low level, 3:reserved)
7	RW	0	[1]sspol (0:SS polarity Low active,1:High active)
6	RW	0	[1]ssctl (see details in Note1)
5	RW	0	[1]pha (clock/data phase control, see section 2.2)
4	RW	0	[1]pol (clock polarity control, see section 2.2)
3	RW	0	[1]smc (start mode control, see Note2)
2	RW	0	[1]xch(exchange bit, ATTN:will automatically cleared when burst finished, see Note3)
1	RW	0	[1]mode (0:slave,1:master)
0	RW	0	[1]en (0:spicc disable,1:enable)

Note1: In one burst of master mode, if ssctl ==0, ss will output 0 between each spi transition. And if ssctl ==1, ss will output 1.
Note2: smc is for start mode control. If smc ==0, burst will start when xch is set to 1'b1; if smc==1, burst will start when txfifo is not empty.
Note3: setting xch will issue a burst when smc==0, and this bit will be self-cleared after burst is finished.

15.5.4. INTREG 0xc1108d8c

Bit(s)	R/W	Default	Description
31:8			Reserved
7	RW	0	[1]tcen(transfer completed interrrupt enable)
6	RW	0	[1]roen(rxfifo overflow interrrupt enable)
5	RW	0	[1]rfen(rxfifo full interrrupt enable)
4	RW	0	[1]rhen(rxfifo half full interrrupt enable)
3	RW	0	[1]rren(rxfifo ready interrrupt enable)
2	RW	0	[1]tfen(txfifo full interrrupt enable)
1	RW	0	[1]then(txfifo half full interrrupt enable)
0	RW	0	[1]teen(txfifo empty interrrupt enable)

Note1: Interrupt Status presents in STATREG.

15.5.5. DMAREG 0xc1108d90

Bit(s)	R/W	Default	Description
31:26	RW	0	[6]DMA Burst Number
25:20	RW	0	[6]DMA Thread ID
19	RW	0	[1]DMA Urgent
18:15	RW	0x7	[4]Number in one Write request burst(0:1,1:2...)
14:11	RW	0x7	[4]Number in one Read request burst(0:1,1:2...)
10:6	RW	0x8	[5]RxFIFO threshold(RxFIFO's count>=thres, will request write)
5:1	RW	0	[5]TxFIFO threshold(TxFIFO's count<=thres, will request read)
0	RW	0	[1]DMA Enable

15.5.6. STATREG 0xc1108d94

Bit(s)	R/W	Default	Description
31:8			Reserved
7	RW	0	[1]tc(transfer completed, w1c, see Note1)
6	R	0	[1]ro(rxfifo overflow)
5	R	0	[1]rf(rxfifo full)
4	R	0	[1]rh(rxfifo half full)
3	R	0	[1]rr(rxfifo ready)
2	R	0	[1]tf(txfifo full)

Note1: tc is the status bit which indicates a burst transfer is completed. And a burst transfer should be started by writing xch 1'b1. This bit supports w1c(Write 1 clear).

15.5.7. PERIODREG 0xc1108d98

Bit(s)	R/W	Default	Description
31:15			Reserved
14:0	RW	0	[15]period(wait cycles, see Note1)

Note1: Programmer can add wait cycles through this register if transmission rate need to be controlled.

15.5.8. TESTREG 0xc1108d9c

Bit(s)	R/W	Default	Description
31:23	RW	0	Reserved
Read Only (Need pay attention)			
22:21	R	0	[2]fiforst(fifo soft reset)
20:15	R	0x15	[6]dlyctl(delay control)
14	R	0	[1]swap(data swap for reading rxfifo)
13	R	0	[1]lbc(loop back control)
Write Only (Need pay attention)			
23:22	W	0	[2]fiforst(fifo soft reset)
21:16	W	0x15	[6]dlyctl(delay control)
15	W	0	[1]swap(data swap for reading rxfifo)
14	W	0	[1]lbc(loop back control)

Bit(s)	R/W	Default	Description
12:10	R	0	[3]smstatus(internal state machine status)
9:5	R	0	[5]rxcnt(internal RxFIFO counter)

Note1: Programmer can only use the TESTREG[9:0], rxcnt(internal RxFIFO counter) and txcnt(internal TxFIFO counter) , and other bits just for test.

15.5.9. DRADDR 0xc1108da0

Bit(s)	R/W	Default	Description
31:0	RW	0	Read Address of DMA

15.5.10. DWADDR 0xc1108da4

Bit(s)	R/W	Default	Description
31:0	RW	0	Write Address of DMA

16. SERIAL PERIPHERAL INTERFACE FLASH CONTROLLER

16.1. Overview

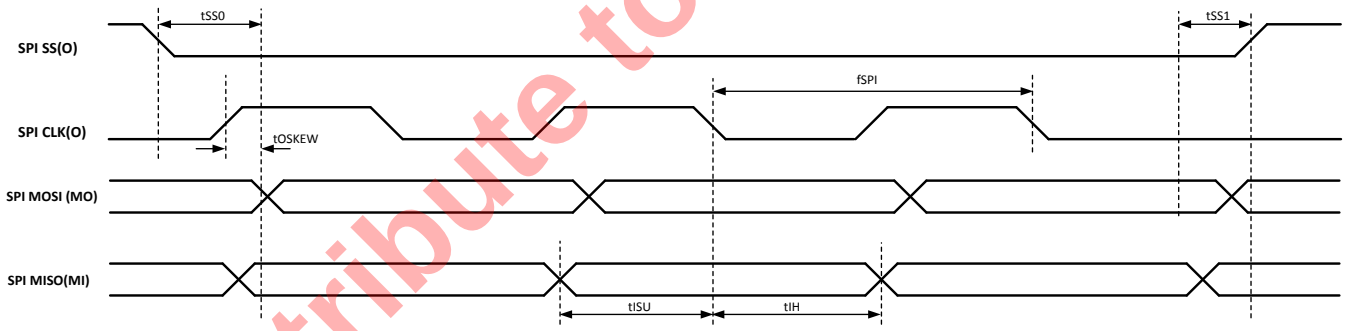
SPI Flash Controller is designed for connecting varied SPI Flash memory.

16.2. Features

- Support three operation modes, NOR Flash mode, Master mode, and Slave mode.
- Support read/write buffer up to 64bytes.
- Support no clock toggling during DUMMY state.
- Support hold by an external pin during a transition.
- AHB read support byte and halfword.
- Support bit-number rather than byte-number for each stage.
- Support 2/4 wire writing like fast reading
- Support both rising-edge and falling-edge for SPI slave sampling and SPI master sampling.
- Support 1 wire for SPI_D and SPI_Q.
- Support SPI_CK setup and hold time by cycles
- Support 8bit clock divider, so SPI_CK can be low as 1/256 HCLK
- Support byte-order in a word
- Support no command state, so the command is sent/received in address state by 2/4 wires.
- Support both data input and data output in a transition. SPI_DOUT->(SPI_DUMMY)->SPI_DIN

16.3. Timing Specification

Figure 12. SPIFC Timing Diagram



Master Mode, CPOL=0, CPHA=1

Table 19. SPIFC Master Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
fSPI	Clock Frequency	0	20	MHz	
tWL	Clock Low Time	20		ns	
tWH	Clock High Time	20		ns	
tOSKEW	Output SKEW	-5	5	ns	
tISU	Input Setup Time	10		ns	
tIH	Input Hold Time	10		ns	
tSS0	SS active time before transition	0		ns	
tSS1	SS inactive time after transition	0		ns	
tWP0	WP active time before transition	Software Controlled		ns	
tWP1	WP inactive time after transition	Software Controlled		ns	

16.4. Register Description

16.4.1. SPIFC FLASH Command register 0xc1108c80

Bit(s)	R/W	Default	Description
31	R/W	0	READ command. 1 = read. When it becames 0, the read command is finished. The READ command could be (0xEB, 0x6B, 0xBB, 0x3B, 0x0B, 0x03). By default is 0x0B. One read command will read continous 32x8bits data. And saved in data cache.
30	R/W	0	WREN command. (0x06)
29	R/W	0	WRDI command. (0x04).
28	R/W	0	RDID command. (0x9f).
27	R/W	0	RDSR command. (0x05).
26	R/W	0	WRSR command. (0x01).
25	R/W	0	Page program command. (0xAD or 0x02).
24	R/W	0	SE command (0x20).
23	R/W	0	BE command (0xD8).
22	R/W	0	CE command.(0xC7).
21	R/W	0	Deep Power Down command(0xB9).
20	R/W	0	RES command. (0xAB).
19	R/W	0	HPM command.(0xA3). (Just For winbond SPI flash).
18	R/W	0	USER defined command.
17:0	R/W	0	Reserved for future.

16.4.2. SPIFC address register 0xc1108c84

Bit(s)	R/W	Default	Description
31:0	R/W	0	The address[31:0] of the user command

16.4.3. SPIFC control register 0xc1108c88

Bit(s)	R/W	Default	Description
31:27	R/W	0	Reserved.
26	R/W	0	Write bit order. 1 = 0, 1, 2, 3, 4, 5, 6, 7. 0 = 7, 6, 5, 4, 3, 2, 1, 0.
25	R/W	0	Read bit order. . 1 = 0, 1, 2, 3, 4, 5, 6, 7. 0 = 7, 6, 5, 4, 3, 2, 1, 0.
24	R/W	0	Fast read QIO mode.
23	R/W	0	Fast read DIO mode.
22	R/W	0	Write 2 bytes status mode. For some of winbond SPI flash, the status register is 16bits.
21	R/W	1	SPI flash WP pin value if use SPI flash WP pin as write protection.
20	R/W	0	Fast read QOUT mode.
19	R/W	1	1 = SPI share pins with SDRAM. 0 = doesn't share.
18	R/W	0	SPI hold mode. 1=SPI controller would use SPI hold function. 0 = SPI controller won't use hold function. The SPI flash hold pin can be tie high on the board. Or SPI controller can use hold pin as QIO/QOUT mode.
17	R/W	1	1 = enable AHB request. 0 = disable AHB request when you reconfigure SPI controller or running APB bus commands.
16	R/W	0	1 =enable SST SPI Flash aai command. The APB bus PP command will send AAI command.
15	R/W	1	1 = release from Deep Power-Down command is with read electronic signature.
14	R/W	0	Fast read DOUT mode.
13	R/W	1	Fast read mode. AHB bus read requirement and APB bus read command use the command 0x0Bh.
12:0	R/W	0	Reserved for future.

16.4.4. SPIFC control register 0xc1108c8c

Bit(s)	R/W	Default	Description
31:28	R/W	5	SPI Clock cycles for SPI flash timing requirement tCSH.
27:16	R/W	0xfff	SPI Clock cycles for SPI flash timing requirement tRES.
15:0	R/W	0x0120	System clock cycles for SPI bus timer. In SPI share bus and SPI hold function mode. SPI bus timer used , if SPI use the bus for a limit time, SPI controller will diassert SPI hold pin to halt the SPI Flash, and give the bus control to SDRAM.

16.4.5. SPIFC status register 0xc1108c90

Bit(s)	R/W	Default	Description
31:24	R/W	0	Reserved.
23:16	R/W	0	For winbond SPI flash, this 8 bits used for DIOmode M7~M0,
15:0	R/W	0	SPI status register value. WRSR command will write this value to SPI flash status. RDSR or RES command will save the read result to this register.

When SPI controller in the slave mode, this register are the status for the SPI master to read out.

Bit(s)	R/W	Default	Description
31:0	R/W	0	In SPI Slave mode, the read status of the user command

16.4.6. SPIFC control register 2 0xc1108c904

Bit(s)	R/W	Default	Description
31:28	R/W	0	Delay cycle number of SPI_CS input in SPI slave mode or SPI_CS output in SPI master mode 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
27:26	R/W	0	delay mode of SPI_CS input in SPI slave mode or SPI_CS output in SPI master mode 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
25:23	R/W	0	Delay cycle number of SPI Data from SPI Master to SPI Slave In SPI master mode, it is for data outputs; in SPI slave mode, it is for data inputs. 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
22:21	R/W	0	Delay mode of SPI Data from SPI Master to SPI Slave In SPI master mode, it is for data outputs; in SPI slave mode, it is for data inputs. 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
20:18	R/W	0	Delay cycle number of SPI Data from SPI Slave to SPI Master. In SPI master mode, it is for data inputs; in SPI slave mode, it is for data outputs. 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
17:16	R/W	0	Delay mode of SPI Data from SPI Slave to SPI Master. In SPI master mode, it is for data inputs; in SPI slave mode, it is for data outputs. 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
15:12	R/W	0	In SPI master mode, SPI_CK rising edge mode 4'b1000 = later by 1/4 cycle of SPI_CK 4'b1001 = later by 1/8 cycle of SPI_CK 4'b1010 = later by 1/16 cycle of SPI_CK 4'b1011 = later by 1/32 cycle of SPI_CK 4'b1100 = earlier by 1/4 cycle of SPI_CK 4'b1101 = earlier by 1/8 cycle of SPI_CK 4'b1110 = earlier by 1/16 cycle of SPI_CK 4'b1111 = earlier by 1/32 cycle of SPI_CK Others = Normal
11:8	R/W	0	In SPI master mode, SPI_CK falling edge mode 4'b1000 = later by 1/4 cycle of SPI_CK 4'b1001 = later by 1/8 cycle of SPI_CK 4'b1010 = later by 1/16 cycle of SPI_CK 4'b1011 = later by 1/32 cycle of SPI_CK 4'b1100 = earlier by 1/4 cycle of SPI_CK 4'b1101 = earlier by 1/8 cycle of SPI_CK 4'b1110 = earlier by 1/16 cycle of SPI_CK 4'b1111 = earlier by 1/32 cycle of SPI_CK Others = Normal
7:4	R/W	1	In master mode, SPI clock cycles for SPI hold timing.
3:0	R/W	1	In master mode, SPI clock sycles for SPI setup timing. SPI setup time and SPI hold time is used to configure how soon the controller can enable spi_cs_n after the controller get the bus and how long the controller still keep the bus after the spi_cs_n become to be high.

Bit(s)	R/W	Default	Description
31:28	R/W	0	Delay cycle number of SPI_CS input in SPI slave mode or SPI_CS output in SPI master mode 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
27:26	R/W	0	delay mode of SPI_CS input in SPI slave mode or SPI_CS output in SPI master mode 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
25:23	R/W	0	Delay cycle number of SPI Data from SPI Master to SPI Slave In SPI master mode, it is for data outputs; in SPI slave mode, it is for data inputs. 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
22:21	R/W	0	Delay mode of SPI Data from SPI Master to SPI Slave In SPI master mode, it is for data outputs; in SPI slave mode, it is for data inputs. 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
20:18	R/W	0	Delay cycle number of SPI Data from SPI Slave to SPI Master. In SPI master mode, it is for data inputs; in SPI slave mode, it is for data outputs. 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
17:16	R/W	0	Delay mode of SPI Data from SPI Slave to SPI Master. In SPI master mode, it is for data inputs; in SPI slave mode, it is for data outputs. 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
15:12	R/W	0	In SPI master mode, SPI_CK rising edge mode 4'b1000 = later by 1/4 cycle of SPI_CK 4'b1001 = later by 1/8 cycle of SPI_CK 4'b1010 = later by 1/16 cycle of SPI_CK 4'b1011 = later by 1/32 cycle of SPI_CK 4'b1100 = earlier by 1/4 cycle of SPI_CK 4'b1101 = earlier by 1/8 cycle of SPI_CK 4'b1110 = earlier by 1/16 cycle of SPI_CK 4'b1111 = earlier by 1/32 cycle of SPI_CK Others = Normal
11:8	R/W	0	In SPI master mode, SPI_CK falling edge mode 4'b1000 = later by 1/4 cycle of SPI_CK 4'b1001 = later by 1/8 cycle of SPI_CK 4'b1010 = later by 1/16 cycle of SPI_CK 4'b1011 = later by 1/32 cycle of SPI_CK 4'b1100 = earlier by 1/4 cycle of SPI_CK 4'b1101 = earlier by 1/8 cycle of SPI_CK 4'b1110 = earlier by 1/16 cycle of SPI_CK 4'b1111 = earlier by 1/32 cycle of SPI_CK Others = Normal
7:4	R/W	1	In master mode, SPI clock cycles for SPI hold timing.
3:0	R/W	1	In master mode, SPI clock sycles for SPI setup timing. SPI setup time and SPI hold time is used to configure how soon the controller can enable spi_cs_n after the controller get the bus and how long the controller still keep the bus after the spi_cs_n become to be high.

16.4.7. SPIFC Clock register 0xc1108c98

Bit(s)	R/W	Default	Description
31	R/W	1	1=SPI clock frequency is same as system clock. 0 = SPI clock frequency will use clock divider.
30:18	R/W	0	Clock counter for Pre-scale divider: 0= not pre-scale divider, 1= pre-scale divided by 2, 2= pre-scale divided by 3, ….,
17:12	R/W	0	Clock counter for clock divider.
11:6	R/W	0	Clock high counter in SPI master mode. In SPI slave mode, it is for the delay counter for the rising edges of spi_ck_i
5:0	R/W	0	Clock low counter, in SPI master mode. In SPI slave mode, it is for the delay counter for the falling edges of spi_ck_i If the SPI clock frequcncy = sys_clock_frequency / n. Then the clock divider counter = n – 1; the clock high counter = n /2 – 1; the clock low counter = n– 1;

Bit(s)	R/W	Default	Description
			For example, if you want to SPI clock freqency is divided by 2 of the system clock. The clock divider counter = 1, clock high counter = 0, clock low counter = 1. For SPI clock frequency = system clock / 4. The clock divider counter = 3, clock high counter = 1, clock low counter = 3.
31	R/W	1	1=SPI clock frequency is same as system clock. 0 = SPI clock frequency will use clock divider.
30:18	R/W	0	Clock counter for Pre-scale divider: 0= not pre-scale divider, 1= pre-scale divided by 2, 2= pre-scale divided by 3, ….,
17:12	R/W	0	Clock counter for clock divider.
11:6	R/W	0	Clock high counter in SPI master mode. In SPI slave mode, it is for the delay counter for the rising edges of spi_ck_i
5:0	R/W	0	Clock low counter, in SPI master mode. In SPI slave mode, it is for the delay counter for the falling edges of spi_ck_i If the SPI clock frequcncy = sys_clock_frequency / n. Then the clock divider counter = n – 1; the clock high counter = n /2 – 1; the clock low counter = n– 1; For example, if you want to SPI clock freqency is divided by 2 of the system clock. The clock divider counter = 1, clock high counter = 0, clock low counter = 1. For SPI clock frequency = system clock / 4. The clock divider counter = 3, clock high counter = 1, clock low counter = 3.

16.4.8. SPIFC User register 0xc1108c9c

Bit(s)	R/W	Default	Description
31	R/W	1	USER command COMMAND bit. 1 = user command includes command. 0 = no command. If some SPI slaves may support 2/4 IO at the first cycle, clear this bit.
30	R/W	0	USER command ADDRESS bit. 1 = user command includes address. 0 = no address.
29	R/W	0	USER command DUMMY bit. 1= user command includes Dummy bytes.
28	R/W	0	USER command DIN bit. 1 = user command includes data in. 0 = no data in.
27	R/W	0	USER command DO bit. 1 = user command includes data output. 0 = no data output. If both DIN and DO are valid, SPI master is firstly in data output state and then in data input state. If all of DUMMY, DO and DIN are valid, SPI master is firstly in data output state and then in dummy state, finally in data input state.
26	R/W	0	USER command dummy idle bit. 1= no SPI clock toggling in dummy state. 0= normal
25	R/W	0	USER command highpart bit for SPI_DOUT stage. It is for data-output in spi master mode and for data-input in spi slave mode. 1 = only high half part of buffer are used. 0 = low half part or the whole 64bytes are used.
24	R/W	0	USER command highpart bit for SPI_DIN stage. It is for data-input in spi master mode and for data-output in spi slave mode. 1 = only high half part of buffer are used. 0 = low half part or the whole 64bytes are used.
23	R/W	0	User command external hold bit for prep. 1 = in prep state, SPI master controller can be hold by the external pin SPI_HOLD
22	R/W	0	User command external hold bit for command. 1 = in command state, SPI master controller can be hold by the external pin SPI_HOLD
21	R/W	0	User command external hold bit for address. 1 = in address state, SPI master controller can be hold by the external pin SPI_HOLD
20	R/W	0	User command external hold bit for dummy. 1 = in dummy state, SPI master controller can be hold by the external pin SPI_HOLD
19	R/W	0	User command external hold bit for data input. 1 = in data input state, SPI master controller can be hold by the external pin SPI_HOLD
18	R/W	0	User command external hold bit for data output. 1 = in data output state, SPI master controller can be hold by the external pin SPI_HOLD
17	R/W	1	User command external hold polarity bit. 1 = high is valid for hold, 0 = low is valid for hold.
16	R/W	0	Single DIO mode: Data output and input apply only 1 wire.
15	R/W	0	Fast write QIO mode.
14	R/W	0	Fast write DIO mode.
13	R/W	0	Fast write QOUT mode.
12	R/W	0	Fast write DOUT mode.
11	R/W	0	Write byte order. 0 = d[7:0], d[15:8], d[23:16], d[31:24]. 1 = d[31:24], d[23:16], d[15:8], d[7:0]
10	R/W	0	Read byte order. . 0 = d[7:0], d[15:8], d[23:16], d[31:24]. 1 = d[31:24], d[23:16], d[15:8], d[7:0]
9:8	R/W	0	AHB endian mode: 0= little-endian; 1= big-endian; 2~3 reserved

Bit(s)	R/W	Default	Description
7	R/W	0	In SPI master mode, the clock output edge bit: 0 = SPI_CK is inverted, 1 = SPI_CK is not inverted
6	R/W	1	In SPI slave mode, the clock input edge bit: 0 = SPI_CK_I is inverted, 1 = SPI_CK_I is not inverted
5	R/W	0	SPI CS setup bit: 1 = valid in prep state
4	R/W	0	SPI CS hold bit: 1 = valid in done state
3	R/W	0	AHB-read apply the configurations of user-command, such as command value, bit-length,…
2	R/W	1	Backward Compatible: 1 = compatible to Apollo SPI This bit affect the three registers: “SPI Flash Commmand Register”, “SPI Address Register” and “SPI Control Register”
1	R/W	0	AHB-read support 4byte address, when AHB-read apply the configurations of user-command. 1 = 4byte address, 0 = 3byte address
0	R/W	0	In SPI master mode, Enable bit for Data input during SPI_DOUT stage. 1 = enable; 0 = disable This bit shall not be used in 2/4wire or SIO. When this bit is 1, during SPI_DOUT stage, data input are stored into cacheline/buffer from address 0, i.e., Bit24 is not controlling the start address. The data output can be specified by bit25

16.4.9. SPIFC User register 1 0xc1108ca0

Bit(s)	R/W	Default	Description
31:26	R/W	0	USER command bit number for address state 0 = 1 bit, 1= 2 bits, …
25:17	R/W	0	USER command bit number for data output state 0 = 1 bit, 1= 2 bits, …
16:8	R/W	0	USER command bit number for data input state 0 = 1 bit, 1= 2 bits, …
7:0	R/W	0	USER command cycle number for dummy state 0 = 1 cycle, 1= 2 cycles, …

16.4.10. SPIFC User register 2 0xc1108ca4

Bit(s)	R/W	Default	Description
31:28	R/W	0	USER command bit number for command state 0 = 1 bit, 1= 2 bits, …
27:16	R/W	0	Reserved
15:0	R/W	0	The command content of the user command

16.4.11. SPIFC User register 3 0xc1108ca8

Bit(s)	R/W	Default	Description
31:0	R/W	0	In SPI Master mode, the address[63:32] of the user command In SPI Slave mode, the write status of the user command

16.4.12. SPIFC PIN register 0xc1108cac

Bit(s)	R/W	Default	Description
31	R/W	0	Pin swap when it is in DIN stage and SPI data input are 4wire. This feature is for Ubec Zigbee chips. 1 = swap between {spi_q, spi_d_i} and {spi_hold_i, spi_wp_i} 0 = normal
30	R/W	0	In SPI Master mode, CS keep active after a transition. 1 = enable; 0 = disable
29	R/W	0	Idle edge of SPI_CK 0 = low when it is idle 1 = high when it is idle
28:24	R/W	0	Reserved
23	R/W	0	In the SPI slave mode, spi_cs_i polarity: 1= high voltage is active 0= low voltage is active
22:21	R/W	0	In the SPI slave mode, spi_ck_i and spi_cs_i source pins 0=SPI_CK and SPI_CS pins, respectively 1=SPI_CS2 and SPI_CS1 pins, respectively 2=SPI_HOLD and SPI_WP pins, respectively

Bit(s)	R/W	Default	Description
20	R/W	0	SPI_CS2 and SPI_CS1 pin function MUX 0= spi_ck and spi_cs in the SPI master mode 1= input pads for spi_ck_i and spi_cs_i, respectively, in the SPI slave mode
19	R/W	0	SPI_CK and SPI_CS pin function MUX 0= spi_ck and spi_cs in the SPI master mode 1= input pads for spi_ck_i and spi_cs_i, respectively, in the SPI slave mode
18:17	R/W	0	SPI_HOLD and SPI_WP pin function MUX 0= normal 1= spi_q and spi_d, respectively 2= spi_cs3 and spi_cs2, repectively 3= input pads for spi_ck_i and spi_cs_i, respectively, in the SPI slave mode
16	R/W	0	SPI_D and SPI_Q switch 1= SPI_D and SPI_Q pin-functions are swapped 0= normal
15:11	R/W	0	In Master mode, these are spi_ck MUX bit[4:0] for spi_cs4 (SPI_HOLD pin), spi_cs3 (SPI_WP pin), spi_cs2, spi_cs1 and spi_cs, respectively 1= this pin is spi_ck, if this pin is not idle 0= this pin is spi_cs, if this pin is not idle
10:6	R/W	0	In Master mode, these are polarity bit[4:0] for spi_cs4 (SPI_HOLD pin), spi_cs3 (SPI_WP pin), spi_cs2, spi_cs1 and spi_cs, respectively 1= high voltage is active 0= low voltage is active
5:0	R/W	0x1E	In Master mode, these are idle bit[5:0] for SPI_CK, for spi_cs4 (SPI_HOLD pin), spi_cs3 (SPI_WP pin), spi_cs2, spi_cs1 and spi_cs, respectively 1= idle, i.e., the spi_ck signal is 0 or the spi_cs is at the inactive level 0= active if SPI controller is working

16.4.13. SPIFC Slave register 0xc1108cb0

Bit(s)	R/W	Default	Description
31	R/W	0	SPI controller SW reset: 1 = reset, 0 = none
30	R/W	0	SPI slave mode: 1 = slave, 0 = master
29	R/W	0	In SPI slave mode, the enable bit for the command of write-buffer-and-read-buffer 0= disable, 1=enable
28	R/W	0	In SPI slave mode, the enable bit for the command of write-status-and-read-status 0= disable, 1=enable
27	R/W	0	SPI slave command define enable 0=Apply the last 3bits of Flash commands and two extra command 1=Apply the user defined command in SPI Slave register 3
26:4	R/W	0	Reserved
11:10	R/W	0	spi_cs_i recovery mode: 0= and 1= or 2= normal 3= delayed
9:5	R/W	0	Interrupt enable for bit4:0 1= enable, 0=disable
4	R/W	0	A SPI transition is done. (whatever it is in SPI master mode or SPI slave mode)
3	R/W	0	In SPI slave mode, a status write is done
2	R/W	0	In SPI slave mode, a status read is done
1	R/W	0	In SPI slave mode, a buffer write is done
0	R/W	0	In SPI slave mode, a buffer read is done

16.4.14. SPIFC Slave register 1 0xc1108cb4

Bit(s)	R/W	Default	Description
31:27	R/W	0	In SPI slave mode, status bit number 0 = 1 bit, 1= 2 bits, …
26	R/W	0	In SPI slave mode, status fast read/write enable bit: 1 = enable, 0 = disable

Bit(s)	R/W	Default	Description
25	R/W	0	In SPI slave mode, status read back enable bit: 1 = reading status is written status in SPI User register 3, 0 = reading status is in SPI Status register.
24:16	R/W	0	In SPI slave mode, buffer bit number 0 = 1 bit, 1= 2 bits, …
15:10	R/W	0	In SPI slave mode, address bit number for reading buffer 0 = 1 bit, 1= 2 bits, …
9:4	R/W	0	In SPI slave mode, address bit number for writing buffer 0 = 1 bit, 1= 2 bits, …
3	R/W	0	In SPI slave mode, dummy enable bit for writing status 1=enable, 0=disable
2	R/W	0	In SPI slave mode, Dummy enable bit for reading status 1=enable, 0=disable
1	R/W	0	In SPI slave mode, Dummy enable bit for writing buffer 1=enable, 0=disable
0	R/W	0	In SPI slave mode, Dummy enable bit for reading buffer 1=enable, 0=disable

16.4.15. SPIFC Slave register 2 0xc1108cb8

Bit(s)	R/W	Default	Description
31:24	R/W	0	In SPI slave mode, Dummy cycle number for writing buffer 0 = 1 cycle, 1= 2 cycles, …
23:16	R/W	0	In SPI slave mode, Dummy cycle number for reading buffer 0 = 1 cycle, 1= 2 cycles, …
15:8	R/W	0	In SPI slave mode, Dummy cycle number for writing status 0 = 1 cycle, 1= 2 cycles, …
7:0	R/W	0	In SPI slave mode, Dummy cycle number for reading status 0 = 1 cycle, 1= 2 cycles, …

16.4.16. SPIFC Slave register 3 0xc1108cbC

Bit(s)	R/W	Default	Description
31:24	R/W	0	In SPI slave mode, Command value for writing status, when bit27 “SPI slave command define enable” in SPI Slave register is 1
23:16	R/W	0	In SPI slave mode, Command value for reading status, when bit27 “SPI slave command define enable” in SPI Slave register is 1
15:8	R/W	0	In SPI slave mode, Command value for writing buffer, when bit27 “SPI slave command define enable” in SPI Slave register is 1
7:0	R/W	0	In SPI slave mode, Command value for reading buffer, when bit27 “SPI slave command define enable” in SPI Slave register is 1

16.4.17. SPIFC controller cache 0~7 0xc1108cc0~0xc1108cdc

Bit(s)	R/W	Default	Description
31:0	R/W	0	Cache line Word 0~7. Cache is used to read data both for AHB or APB read command. Cache is also used for APB page programming etc.

16.4.18. SPIFC controller buffer 8~15 0xc1108ce0~0xc1108cfc

Bit(s)	R/W	Default	Description
31:0	R/W	0	Buffer Word 8. Buffer is used to read/write data only for APB read/write user commands.

17. UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER

17.1. Overview

There are a number of UART's in the chip that offer 2-wire (RX/TX) and 4-wire (RX/TX, CTS/RTS) connections at the digital I/O pins. Each UART contains one transmit FIFO and a receive FIFO (see depths below). The FIFO's are filled by the CPU and read by the CPU. In some cases, the receive FIFO can be configured to be pushed directly to DDR memory without CPU intervention.

UART	RX/TX FIFO depths	RX FIFO DMA to DDR	Comment
UART0	128 bytes	Yes	Located in the EE domain
UART1	64 bytes	Yes	Located in the EE domain
UART2	64 bytes	Yes	Located in the EE domain
UART0-AO	64 bytes	No	Located in the Always On domain
UART2-AO	64 bytes	No	Located in the Always On domain

17.2. Features

Input filters: The CTS (clear to send) and RX (receive) input paths have input filters to deal with slow rise times. The filters are configurable to use a 125nS or 1uS sampling mechanism. There is an implied 3 system clock cycle delay (15nS for a typical system clock of 200Mhz) that is used to synchronize and detect the rising/falling edge of the RXD signal. The RXD signal may be passed through an optional filter to deglitch the external signal in noisy conditions. The deglitch filter has two settings which add to the ¡§detection delay¡¨ of the RXD signal by the internal logic:

- Filter setting 1 (125nS strobe): 375nS ~ 2.6uS
- Filter setting 2 (1uS strobe): 3uS ~ 21uS

The filter is described in the register specification. If the filter is disabled, the shortest RXD low time and high time is 12 system clock cycles (60nS for a system clock of 200Mhz).

Clear to Send: CTS is a signal sent from the receiver UART back to the transmitting UART to tell the transmitting UART to stop sending data. The CTS signal must be received before the next START symbol is sent. The transmitting UART is allowed to send one more byte after the CTS signal is recognized. The CTS signal coming into the chip goes through some synchronization and detection which adds an additional 5 system clocks (typically 25nS for a 200Mhz system clock). This setup time for CTS detection is called CTS_{stop}. The CTS input also has an optional filter can be used to deglitch the incoming CTS signal. If the filter is disabled, the CTS signal must be de-asserted 5 system clock cycles before the start of the next BYTE transfer. If the CTS filter is enabled, then additional time must be added to the 25nS requirement. There are two programmable filter settings that effectively delay CTS being seen by the internal logic:

- Filter setting 1 (125nS strobe): 375nS ~ 2.6uS
- Filter setting 2 (1uS strobe): 3uS ~ 21uS

Interrupts: The UARTs can generate interrupts if the receive FIFO exceeds a pre-programmed threshold. An interrupt can also be generated if there is a frame or parity error.

Clock independent operation: Because the system clock can be altered to accommodate dynamic frequency scaling, the UARTs have an option in which they use the 24Mhz crystal clock as the source for the UART.

17.3. Functional Description

The UART requires that a Baud Rate be established. The UART supports rates as slow as 1Hz up to rates as high as 8 Mbits/Sec. Once the baud rate has been established, bytes are transmitted as they are written to the transmit-FIFO by the CPU. A large transmit-FIFO exists to allow the CPU to pre-load a transmit package because the CPU can often write faster than the UART can transmit the data.

Data this automatically received by the UART is placed into the receive FIFO one byte at a time. The receive-FIFO decouples the UART from the CPU allowing the CPU to read the UART byte data at a rate not dictated by the UART.

Figure 13. UART Timing Diagram

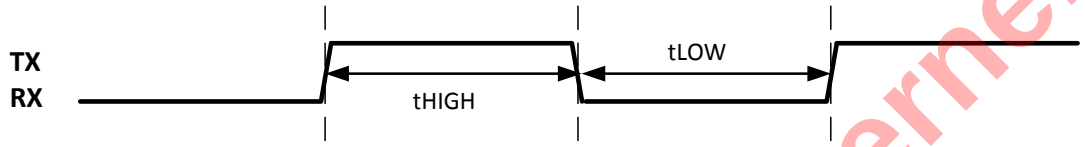


Table 20. UART Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
tHIGH	TX/RX high time	60ns	167ms		
tLOW	TX/RX low time	60ns	167ms		

17.4. Register Description

The following register definition is uniformly applied to all UART instantiations in the chip.

Instantiations	Base Address
UART0	0x2130 ~ 0x2135
UART1	0x2137 ~ 0x213c
UART2	0x21C0 ~ 0x21c5
UART0-AO	0xc81004c0 ~ 0xc81004d4
UART2-AO	0xc81004e0 ~ 0xc81004f4

17.4.1. UARTx_WFIFO: Write data

Bit(s)	R/W	Default	Description
31-8	R	0	unused
7-0	R/W	-	Write FIFO data. The Write FIFO holds 64 bytes. The Write FIFO can be written as long as it is not full.

17.4.2. UARTx_RFIFO: Read Data

Bit(s)	R/W	Default	Description
31-8	R	0	unused
7-0	R/W	-	Read FIFO data. The Read FIFO holds 64 bytes. The empty flag can be used to determine if data is available

17.4.3. UARTx_CONTROL: UART Mode

Bit(s)	R/W	Default	Description
31	R/W	0	Invert the RTS signal
30	R/W	0	Mask Error: Set to 1 to mask errors
29	R/W	0	Invert the CTS signal
28	R/W	0	Transmit byte Interrupt: Set to 1 to enable the generation an interrupt whenever a byte is read from the transmit FIFO
27	R/W	0	Receive byte Interrupt: Set to 1 to enable the generation an interrupt whenever a byte is written to the receive FIFO
26	R/W	0	Set to 1 to invert the TX pin
25	R/W	0	Set to 1 to invert the RX pin
24	R/W	0	Clear Error
23	R/W	0	Reset the receive state machine
22	R/W	0	Reset the transmit state machine
21-20	R/W	0	Character length: 00 = 8 bits, 01 = 7 bits, 10 = 6 bits, 11 = 5 bits
19	R/W	1	Parity Enable: Set to 1 to enable parity
18	R/W	0	Parity type: 0 = even, 1 = odd
17-16	R/W	0	Stop bit length: 00 = 1 bit, 01 = 2 bits
15	R/W	0	Two Wire mode:
14	R/W	0	Unused
13	R/W	0	Receive Enable. Set to 1 to enable the UART receive function
12	R/W	0	Transmit Enable. Set to 1 to enable the UART transmit function
11-0	R/W	0x120	Baud rate: This value sets the baud rate by dividing the MPEG system clock. The baud rate is set according to the following equation: Baud rate = . For example: 9600 =

17.4.4. UARTx_STATUS: UART Status

Bit(s)	R/W	Default	Description
31-27	R	0	Unused
26	R	0	UART_RECV_BUSY: This bit will be 1 if the uart receive state machine is busy
25	R	0	UART_XMIT_BUSY: This bit will be 1 if the uart transmit state machine is busy
24	R	0	RECV_FIFO_OVERFLOW:
23	R	0	CTS Level
22	R	0	Transmit FIFO Empty
21	R	0	Transmit FIFO Full
20	R	0	Receive FIFO empty
19	R	0	Receive FIFO full
18	R	0	This bit is set if the FIFO is written when it is full. To clear this bit, write bit 24 of register 0x2132
17	R	0	Frame error. To clear this bit, write bit 24 of register 0x2132
16	R	0	Parity error. To clear this bit, write bit 24 of register 0x2132
15	R	0	Unused

14-8	R	0	Transmit FIFO count. Number of bytes in the transmit FIFO
7	R	0	Unused
8-0	R	0	Receive FIFO count. Number of bytes in the receive FIFO

17.4.5. UARTx_MISC: UART IRQ CONTROL

Bit(s)	R/W	Default	Description
31	R/W	0	Added a "just in case" bit that can be set to 1 to enable clocks always. The default is 0 meaning the auto-clock gating logic is enabled.
30	R/W	0	USE old Rx Baud: There was a bug in the RX baud rate generator. The Rx baud rate generator was re-designed to compute a baud rate correctly. If you want to use the old (stupid) logic, you can set this bit to 1.
29	R/W	0	ASYNC_FIFO_PURGE: This bit can be set to 1 after all UART bytes have been received in order to purge the data into the Async FIFO. This bit is needed because the UART receives 8-bit data, but the ASYNC FIFO can only be written with 16-bit data. In this case there might be a residual byte if the UART is not receiving an even number of bytes.
28	R/W	0	ASYNC_FIFO_EN: If this bit is set to 1, then the UART received data is automatically sent to the Async FIFO module which will in turn automatically send the data to DDR memory
27	R/W	0	CTS: Filter Timebase select: 1 = 1uS, 0 = 111nS timebase. A filter was added to the CTS input to allow for a little digital filtering. The amount of filtering is controlled by this timebase (longer = more filtering) and the value in bits FILTER_SEL below.
26-24	R/W	0	CTS: FILTER_SEL: 0 = no filter, 7 = max filtering
23-20	R/W	0	BAUD_RATE_EXT: These 4 bits extend the baud rate divider to 16-bits: `Baud Rate = {Reg4[23:20],Reg2[11:0]}`
19	R/W	0	RX: Filter Timebase select: 1 = 1uS, 0 = 111nS timebase. A filter was added to the RX input to allow for a little digital filtering. The amount of filtering is controlled by this timebase (longer = more filtering) and the value in bits FILTER_SEL below.
18-16	R/W	0	RX: FILTER_SEL: 0 = no filter, 7 = max filtering
15-8	R/W	32	XMIT_IRQ_CNT: The UART can be configured to generate an interrupt if the number of bytes in the transmit FIFO drops below this value.
7:0	R/W	15	RECV_IRQ_CNT: The UART can be configured to generate an interrupt after a certain number of bytes have been received by the UART.

17.4.6. UARTx_REG5

Bit(s)	R/W	Default	Description
31-24	R/W	0	unused
24	R/W	0	USE_XTAL_CLK: If this bit is set, then the clock for generating the UART Baud rate comes from the crystal pad. This allows the UART to operate independent of clk81.
23	R/W	0	USE New Baud rate. Over the years, the baud rate has been extended by concatenating bits from different registers. To take advantage of the full 23-bit baud rate generate (extended to 23 bits to accommodate very low baud rates), you must set this bit. If this bit is set, then the baud rate is configured using bits [22:0] below
22:0	R/W	15	NEW_BAUD_RATE: If bit[23] = 1 above, then the baud rate for the UART is computed using these bits. This was added in M6 to accommodate lower baud rates.

18. INFRARED REMOTE

18.1. Overview

An IR signal based Remote Control (RC) signal decoder and blaster are built in S805 to provide software a low-cost, convenient way to implement remote control function in applications. This module is located in the AO power domain.

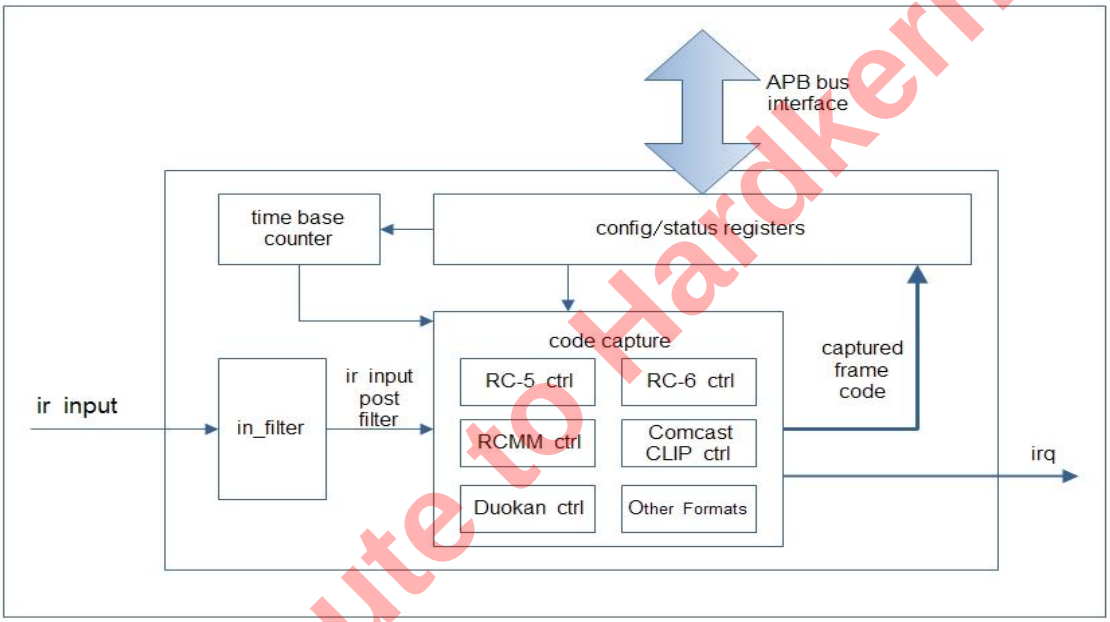
18.2. Decoder Functional Description

The decoder mainly consisted of two blocks:

- Decoder with input filter
- A set of registers including control & clock, data and tuning

The function diagram of IR decoder is illustrated in the figure below.

Figure 14. IR Decoder Function Block



IR Decoder decodes the IR remote control input signal. 13 operation modes are supported:

- Hardware Decode IR transmission protocol compatible frame decoder mode (NEC MITSUBISHI Thomson Toshiba Sony SIRC RC5 RC6 RCMM Duokan Comcast Sanyo Modes)
- General programmable time measurement frame decoder mode (General Mode)

In Hardware Decode Mode, the Decoder uses signal pattern search mechanism to decode data frame. It can detect logical '0' ,'1' ,"00","01","10" and "11", as well as data frame start and end. Whenever Decoder detects and decodes the data frame, the data are kept in data register.

In General Mode, the Decoder uses edge detection mechanism to decode data frame. It can detect each input signal edge and record the time between two edges. The time measurement result is kept in control register.

The user should set proper operation mode corresponding to the selection of remote controller.

There is a simple time-based signal Filter between the signal input and the Decoder. The Filter is programmable and helps to improve signal integrity.

18.3. NEC Infrared Transmission Protocol Example

18.3.1. Message Bit

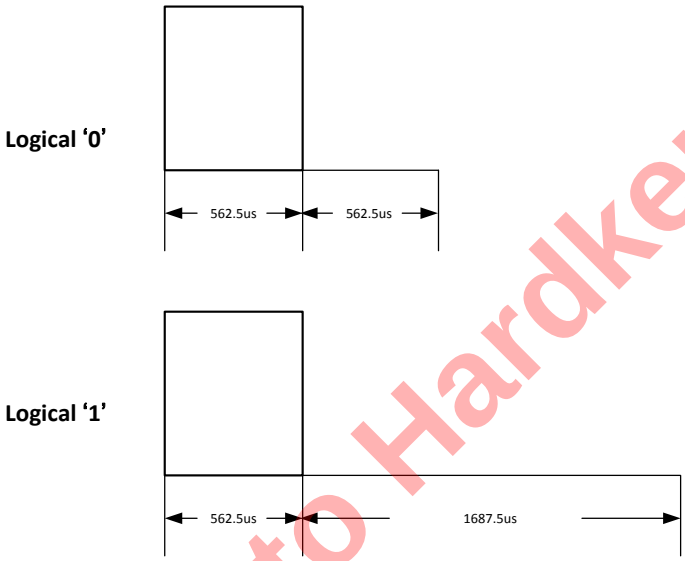
Standard NEC IR transmission protocol uses pulse distance encoding of the message bits. The basic clock circle time is 562.5us.

The logical bits are defined and transmitted as follow:

- Logical '0' – a 562.5us pulse burst (1 clock) followed by a 562.5us space (1 clock). The total transmit time is 1.125ms.
- Logical '1' – a 562.5us pulse burst (1 clock) followed by a 1.6875ms space (3 clocks). The total transmit time is 2.25ms.

The signals are illustrated in the picture below.

Figure 15. Message Bit Signal



18.3.2. Data Frame

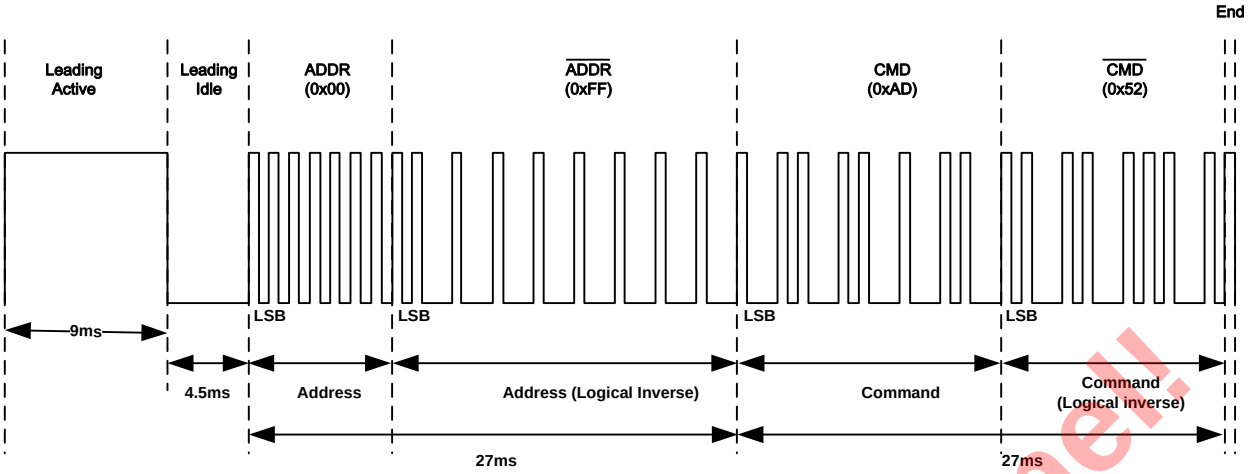
A data transmit frame is exchanged via IR to represent a message when a key is pressed on the remote control. Typically, the data frame consists of, in order:

- A Leading Active – 9ms pulse burst (16 clocks)
- A Leading Idle – 4.5ms space (8 clocks)
- An 8-bit receiver address (ADDR), LSB first.
- The complementary 8-bit receiver address (), LSB first
- An 8-bit command (CMD), LSB first
- The complementary 8-bit command (), LST first
- A final 562.5us pulse burst to indicate the end of message transmission.

The byte ADDR/ and CMD/ are sent with least significant bit (LSB) first.

The example of data frame is illustrated below with ADDR=0x00 and CMD=0xAD.

Figure 16. Data Frame



18.3.3. Repeat Code(s)

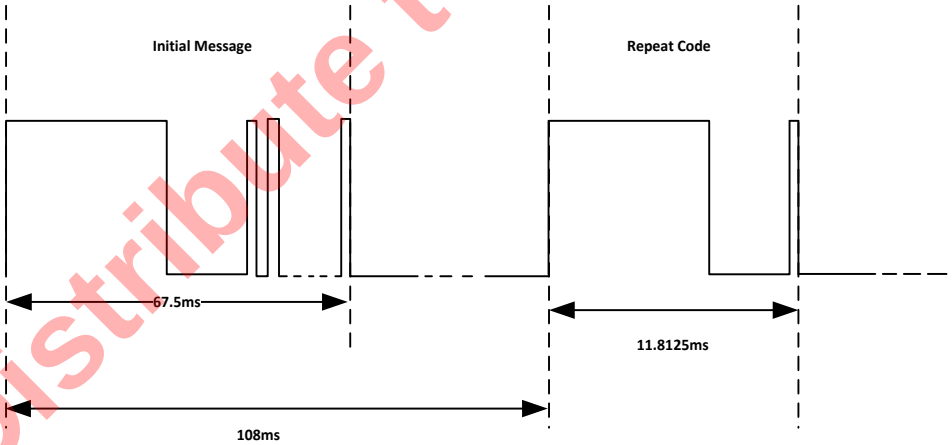
A repeat code will be issued if the key on the remote controller is kept pressed. A repeat code will continue to be sent out at 108ms intervals until the key is released.

Typically, the repeat code consists of, in order:

- A Leading Active – 9ms pulse burst (16 clocks)
- A Leading Idle – 2.25ms space (4 clocks)
- A final 562.5us pulse burst to indicate the end of message transmission.

The example of repeat code is illustrated below

Figure 17. Repeat Code



18.4. Register Description

18.4.1. AO_MF_IR_DEC_LDR_ACTIVE: Leader Active control 0xc8100580

This register controls the min/max Leader Active time window. For example, for NEC format, the Leader Active time is about 9mS. To identify a Leader Active time between 8.60 mS and 9.40 mS (assuming base resolution = 20uS), user can set Max duration = 0x1d6 (‘d470) to represent 9.40 mS, and set Min duration = 0x1ae (‘d430) to represent 8.60 mS.

Bit(s)	R/W	Default	Description
31-29	R	0	Unused
28-16	R/W	0	Max duration of Leader’s active part
15-13	R	0	Unused
12-0	R/W	0	Min duration of Leader’s active part

18.4.2. AO_MF_IR_DEC_LDR_IDLE: Leader Idle control 0xc8100584

Bit(s)	R/W	Default	Description
31-29	R	0	Unused
28-16	R/W	0	Max duration of Leader’s idle part
15-13	R	0	Unused
12-0	R/W	0	Min duration of Leader’s idle part

18.4.3. AO_MF_IR_DEC_LDR_REPEAT: Repeat Leader Idle Time 0xc8100588

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Repeat Code’s Leader. In NECformat, it defines for the repeat leader’s idle part. In Toshiba format, it defines for the repeat leader’s second idle part (In Toshiba format, the repeat leader’s first idle part has the same duration time as the normal leader idle part.)
15-10	R	0	Unused
9-0	R/W	0	Min duration of Repeat Code’s Leader

18.4.4. AO_MF_IR_DEC_BIT_0: 0xc810058C

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Duration Setting Regiser 0. It defines max tining duration for: Logic”0” for NEC/Toshiba/Sony/Thomas format or Half trailer bit for RC6 format (RC6’s half trailer bit typically 888.89us) or time of Duokan/RCMM/4ppm format’s Logic “00”
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 0.

18.4.5. AO_MF_IR_DEC_REG0:0xc8100590

Bit(s)	R/W	Default	Description
31	R/W	0	Clock gating control just in case. Set 1 can force clock gating disabled.
30-28	R/W	0	Filter ctrl. Set the monitor timing for input filter, bigger value means longer monitor time. Value 0 = no filtering.
27-25	R	0	Unused
24-12	R/W	0	Max frame time. Max duration of one whole frame.
11-0	R/W	0	Base time parameter. Used to generate the timing resolution. Resolution = (base_time_paramter + 1) * (1/ Freq_sys_clk). For example, if Frequency of sys_clk is 1Mhz, and base_time_parameter=19,

Bit(s)	R/W	Default	Description
			Then resolution = (19+1)*(1uS) = 20uS.

18.4.6. AO_MF_IR_DEC_STATUS: 0xc8100598

Bit(s)	R/W	Default	Description
31	R/W	0	Frame data valid 1. (This bit is set to 1 when a captured frame is updated/stored into "FrameBody_1" register. A read of "**FrameBody_1**" register will clear this bit. "FrameBody_1" register is used to store the over 32bit MSBs of the formats whose length is more than 32 bit)
30	R/W	0	bit_1_match_en. Set to 1 to enable the check of whether logic"1" bit matches timing configure during the frame input process.
29-20	R/W	0	Max Duration 1. Max duration of Duration Setting Regiser 1. It defines max duration for: Logic"1" for NEC/Toshiba/Sony format or Whole trailer bit for RC6 format (RC6's whole trailer bit typically 1777.78us) or time of Duokan/RCMM/4ppm format's Logic "01"
19-10	R/W	0	Min Duration 1. Min duration of Duration Setting Regiser 1.
9	R	0	irq_status. Appear as 1 if there is an interrupt.
8	R	0	ir_i_sync. IR remote serial input after synchronization. This is the level of the digital signal coming into the IR module for decoding. This is the same as reading the I/O pad level.
7	R	0	Busy. When =1, means state machine is active.
6-4	R	0	Decoder_status (for debug only). 000: OK 001: last frame timed out 010: leader time error (invalid IR signal) 011: repeat error (repeat leader, but other IR transitions found). 100: Invalid bit
3-0	R	0	Frame status. Bit 3: Frame data valid (This bit is set to 1 when a captured frame is updated/stored into "FrameBody" register. A read of "**FrameBody**" register will clear this bit. If store and read occurs at the same time, this bit is set to 1 in common, But if "Hold first" is set to true and this valid bit is already 1, a read clear takes precedance and this bit is clear to 0.) Bit 2: data code error (data != ~data in IR bit stream) Bit 1: custom code error (custom_code != ~custom_code in IR bit stream) Bit 0: 1 = received frame is repeat key, 0 = received frame is normal key

18.4.7. AO_MF_IR_DEC_REG1: 0xc810059C

Bit(s)	R/W	Default	Description
31	R/W	0	Set to 1 to use faster timebase. -
30	R/W	0	cntl_1us_eq_clk. Just use sys_clk to relace 1uS tick.
29	R/W	0	cntl_xtal3_eq_clk. Just use sys_clk to relace 111ns tick.
28-16	R	0	Pulse Width Counter. It stores the internal counter of pulse width duration. Commonly used as time measurement when **decode_mode** is set to measure width mode (software decode). Time measurement starts at the last time the internal time counter was reset by the rising and/or falling edge of the IR signal. The selection of reset on rising and/or falling edge is determined by the **IRQ Selection** field (Bits 3-2 below)
15	R/W	0	Enable. 1 = enable the state machine of IR decoder. 0 = disable the state machine of IR decoder.
14	R/W	0	cntl_use_sys_clk. Use sys_clk for the timebase. It's useful when sys_clk at low frequency (such as 32Khz) and cannot create 1uS timebase tick. 1 = use the system clock as timebase. 0 = use the 1uS timebase tick as timebase.

Bit(s)	R/W	Default	Description
13-8	R/W	0	Bit_length minus 1. (N-1). Used to set the value of frame body's bit length (frame body commonly includes address and data code part). If a format has 24 bit frame body, this value shall be set to 23.
7	R/W	0	Record_at_error. 1= record the frame body and status forcibly, even if data/custom code error check enabled by frame_mask and relative error occurs. 0 = if data/custom code error check enabled by frame_mask and relative error occurs, not record the frame body and status forcibly
6	R/W	0	Hold_first Used to hold the first captured frame data. If this bit is set to 1, then the "**FrameBody/FrameBody_1**" register will only be updated if hasn't already been updated. Once updated, the "**FrameBody/FrameBody_1**" register will not be updated again until it has been read. This bit can be used to guarantee the first TV remote code captured will not be overwritten by subsequent transmissions from a TV remote. **NOTE: Read the"FrameBody" register can clear the internal "Frame data valid" flag, and read the "FrameBody_1" register can clear the "Frame data valid 1" flag.**
5-4	R/W	0	Frame_mask. Some formats' body include bit-inversed data or custom/address code for error check. 00 = ignore error check from either data or custom/address code 01= check if data code matches its inverse values, ignore error check from custom/address code 10= check if custom/address code matches its inverse values, ignore error check from data code 11= check if data and custom codes match their inverse values
3-2	R/W	0	Irq_sel. IRQ Selection and width measurement reset: 00: IR Decoder done 01: IR input rising or falling edge detected 10: IR input falling edge detected 11: IR rising edge detected
1	R/W	0	IR input polarity selection. Used to adjust/invert the polarity of IR input waveform.
0	R/W	0	Decoder Reset. Set to 1 to reset the IR decoder. This is useful because the IR remote state machine thinks in terms of milliseconds and may take tens of milliseconds to return to idle by itself.

18.4.8. AO_MF_IR_DEC_REG2 0xc81005A0

Bit(s)	R/W	Default	Description
31-27	R	0	Unused
26	R/W	0	Width_low_enable. Enable counter record of low pulse width duration. 0 = do not force enable of width low counter record 1 = force enable of width low counter record Some IR formats' decoding need to use internal width low counter record. By default, the width low counter record is enabled automatically for related formats. This bit is used for enable forcibly just in case. Besides, if "leader plus stop bit" method is enabled for repeat detection, this bit is also need to be enabled.
25	R/W	0	Width_high_enable. Enable counter record of high pulse width duration. 0 = do not force enable of width high counter record 1 = force enable of width high counter record Some IR formats' decoding need to use internal width high counter record. By default, the width high counter record is enabled automatically for related formats. This bit is used for enable forcibly just in case.
24	R/W	0	Enable "leader plus stop bit" method for repeat detection. 0 = "leader plus stop bit" method disabled 1 = "leader plus stop bit" method enabled Some IR formats use one normal frame's leader followed by a stop bit to represet repeat. There is no frame data in this kind repeat frame. To use this method, **width_low_enable** (bit26 of 0x20 offset register) shall be set to 1, and **max_duration_3** and **min_duration_3** in 0x28 offset register shall be set to appropriate value for stop bit's timing duration.
23-22	R	0	Unused
21-16	R/W	0	Repeat_bit_index. These bits are used for compare bit method to set the index of the bit that is used as repeat flag. The index value can be 0 to 63.

Bit(s)	R/W	Default	Description
			Compare bit method is one of the methods for repeat detection . Some IR formats use one bit in frame to represet whether the frame is repeat.
15	R/W	0	Running_count_tick_mode. This bit is only valid when **use_clock_to_counter** bit is 0. 0 = use 100uS as increasing time unit of frame-to-frame counter 1 = use 10uS as increasing time unit of frame-to-frame counter
14	R/W	0	Use_clock_to_counter. If this bit is set to 1, the **running_count_tick_mode** bit is ignored. 0 = do not use system clock as increasing time unit of frame-to-frame counter 1 = use system clock as increasing time unit of frame-to-frame counter
13	R/W	0	Enable frame-to-frame time counter (running-counter). 0 = frame-to-frame time counter disabled 1 = frame-to-frame time counter enabled If enabled, the frame-to-frame counter increases every 100uS or 10uS until it reaches its max value(all bits are 1) or it is reset. When it reaches its max value, it keeps the value until it is reset. When it is reset, it becomes zero and then begin increasing again. The counter can be reset even when it has not reached its max value. The increasing time unit can be 100uS or 10uS or system clock frequency which is set by **running_count_tick_mode** and **use_clock_to_counter** settings. When a frame's data are capured and stored into FrameBody/FrameBody_1 register, frame-to-frame counter is reset to zero. After reset to zero, the frame-to-frame counter will begin increasing again, until it reaches its max value or it is reset. For repeat frame detection, users can use hardware detection by enabling compare frame or compare bit method, or users can read frame-to-frame counter to let software to make the decision.
12	R/W	0	Enable repeat time check for repeat detection. This bit is valid only when compare frame method or compare bit method is enabled. 0 = repeat time check disabled 1 = repeat time check enabled When repeat frame detection is enabled by enabling compare frame or compare bit method, the frame time interval may need to be checked in order to decide whether the frames are repeat (key pressed without release) or not. You can configure the **repeat_time_max** value by setting **0x38 offset** register. If frame interval is smaller than the "repeat time max", it may considered as repeat. If frame interval is bigger than the "repeat time max", it is considered as not repeat.
11	R/W	0	Enable compare frame method for repeat detection. 0 = compare frame method disabled 1 = compare frame method enabled Some IR formats transer the same data frame as repeat frame when the key is kept pressed without release. For repeat detection, compare frame method can be used. If a new frame and the old received frame are the same and the repeat time is under the limit(frame-to-frame time counter value is smaller than the repeat_time_max), the status register's **frame_status0** is set to 1 automatically as repeat detected flag. You can configure the **repeat_time_max** value by setting **0x38 offset** register.
10	R/W	0	Enable compare bit method for repeat detection. 0 = compare bit method disabled 1 = compare bit method enabled Some IR formats use only one bit to represet whether the frame is repeat. You can compare only one bit instead of compare the whole frame for repeat detection. If compare frame method is enabled, then this bit is ignored.
9	R/W	0	Disable read-clear of FrameBody/FrameBody_1. 0 = read-clear enabled 1 = read-clear disabled FrameBody/FrameBody_1 registers are read-cleared in default. When these register are read, they are cleared to zero. This bit is used to disable this read-clear feature. (FrameBody/FrameBody_1 registers are used to store captured frame data).
8	R/W	0	input stream bit order. 0 = LSB first mode (first bit in input stream is considered as LSB) 1= MSB first mode (first bit in input stream is considered as MSB) Note: Commonly the following formats shall set 1 to enable MSB first mode (unless you insist on LSBfirst mode for your specified use): RC5, RC5 extend, RC6, RCMM, Duokan, Comcast

Bit(s)	R/W	Default	Description
7:4	R	0	Unused
3:0	R/W	0	Decode_mode.(format selection) 0x0 =NEC 0x1= skip leader (just bits, without leader) 0x2=General time measurement (measure width, software decode) 0x3=MITSUBISHI 0x4=Thomson 0x5=Toshiba 0x6=Sony SIRC 0x7=RC5 0x8=Reserved 0x9=RC6 0xA=RCMM 0xB=Duokan 0xC=Reserved 0xD=Reserved 0xE=Comcast 0xF=Sanyo

18.4.9. AO_MF_IR_DEC_DURATN2 0xc81005A4

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Duration Setting Regiser 2. It defines max duration for: Half bit for RC5/6 format (RC5 typically 888.89us for half bit, RC6 typically 444.44us) or time of Duokan/RCMM/4ppm format's Logic "10" or time of Comcast/16ppm's base duration
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 2.

18.4.10. AO_MF_IR_DEC_DURATN3 0xc81005A8

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Duration Setting Regiser 3. It defines max duration for: Whole bit for RC5/6 format (RC5 typically 1777.78us for whole bit, RC6 typically 888.89us) or time of Duokan/RCMM/4ppm format's Logic "11" or time of Comcast/16ppm's offset duration
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 3.

18.4.11. AO_MF_IR_DEC_REG3 0xc81005B8

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19-0	R/W	0	Repeat time max. Used to set the maximum time between two repeat frames for repeat frame detection. Time unit is 100uS or 10uS according to the running_count_tick_mode. When repeat frame detection is enabled by enabling compare frame or compare bit method, the frame time interval may need to be checked in order to decide whether the frames are repeat (key pressed without release) or not. If frame interval is smaller than the "repeat time max", it may considered as repeat. If frame interval is bigger than the "repeat time max", it is considered as not repeat.

18.4.12. AO_MF_IR_DEC_FRAME: Frame Body (Frame Data, LSB 32bit) 0xc8100594

Note: New keys will be ignored until ***FrameBody*** register is read if the *hold first key* bit is set in the decode control register. Reading this register resets an internal frame data valid flag.

Bit(s)	R/W	Default	Description
31-0	R	0	32 bit Read-Only register stores frame body (LSB 32bit) captured from IR remote data flow, commonly includes custom/address code and data code.

18.4.13. AO_MF_IR_DEC_FRAME: Frame Body 1 (Frame Data, MSB 32bit) 0xc81005AC

Note: New keys will be ignored until ***FrameBody*** register is read if the *hold first key* bit is set in the decode control register. Reading this register resets an internal frame data valid flag.

Bit(s)	R/W	Default	Description
31-0	R	0	Stores frame body excess 32 bit range. (MSB 32bit)

18.4.14. AO_MF_IR_DEC_STATUS_1 0xc81005B0

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19-0	R	0	Stores the last frame-to-frame counter value before the last counter reset caused by the last frame data record/update.

18.4.15. AO_MF_IR_DEC_STATUS_2 0xc81005B4

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19-0	R	0	Stores the value of the frame-to-frame counter which is running currently.

18.4.16. IR_BLASTER_CNTL0 0xc81000c0

Bit(s)	R/W	Default	Description
31-27	R	0	unused
26	R	-	BUSY: If this bit is 1, then the IR Blaster module is busy.
25	R	-	This output is 1 when the FIFO is Full
24	R	-	This output is 1 when the FIFO is Empty
23-16	R	-	FIFO Level
15-14	R/W	0	Unused
13-12	R/W	0	MODULATOR_TB: This input controls the clock used to create the modulator output. The modulator is typically run between 32khz and 56khz. The modulator output will equal a divided value of the following: 00: system clock “clk” 01: mpeg_xtal3_tick 10: mpeg_1uS_tick 11: mpeg_10uS_tick
11-4	R/W	0	SLOW_CLOCK_DIV: This is a divider value used to divide down the input “clk”. The divider is N+1 so a value of 0 equals divide by 1.
3	R/W	0	SLOW_CLOCK_MODE: Set this signal high to use a special mode in which the “clk” input is driven by a slow clock less than 1Mhz. This is used for low power cases where we want to run the IR Blaster between 32khz and 1Mhz
2	R/W	0	INIT_LOW: Setting this bit to 1 initializes the output to be high. Please set this bit back to 0 when done
1	R/W	0	INIT_LOW: Setting this bit to 1 initializes the output to be low. Please set this bit back to 0 when done
0	R/W	0	ENABLE: 1 = Enable. If this bit is set to 0, then the IR blaster module is reset and put into an IDLE state.

18.4.17. IR_BLASTER_CNTL1 0xc81000c4

Bit(s)	R/W	Default	Description
31-28	R/W	0	unused
27-16	R/W	0	This value is used with “modulator_tb[1:0]” above to create a low pulse. The time is computed as (mod_lo_count+1) x modulator_tb. The purpose for having a low/high count is the modulator output might not be 50% duty cycle. Hi/Lo counters allow us to modulate using a non-50% duty cycle waveform.
15-12	R/W	0	Unused

11-0	R/W	0	This value is used with “modulator_tb[1:0]” above to create a high pulse. The time is computed as (mod_hi_count+1) x modulator_tb

18.4.18. IR_BLASTER_CNTL2 0xc81000c8

Bit(s)	R/W	Default	Description
31-17	R	0	unused
16	W	0	Set this bit to 1 to write the data below to the FIFO
15-12	R	0	Unused
11-0	R/W	0	FIFO data to be written: Bit[12] output level (or modulation enable/disable: 1 = enable) Bit[11:10] Timebase: 00 = 1uS 01 = 10uS 10 = 100uS 11 = Modulator clock Bit[9:0] Count of timebase units to delay

19. UNIVERSAL SERIAL BUS

19.1. Overview

The chips integrates in one USB OTG (On-the-GO) controller and one USB Host controller.

The USB OTG controller is a Dual-Role-Device (DRD) controller that supports both device and host functions and complies fully with the On-The-Go Supplement to the USB 2.0 Specification, Revision 1.3a and Revision 2.0. It can also be configured as a host-only or device-only controller, fully compliant with the USB 2.0 Specification. The USB host controller supports host functions and is fully compliant with USB2.0 specification,

19.2. Features

The OTG controller features:

- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
- Multiple DMA/non DMA mode access support on the application side
- Supports up to 16 bidirectional endpoints, including control endpoint 0.
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports up to 16 host channels.

The Host controller features:

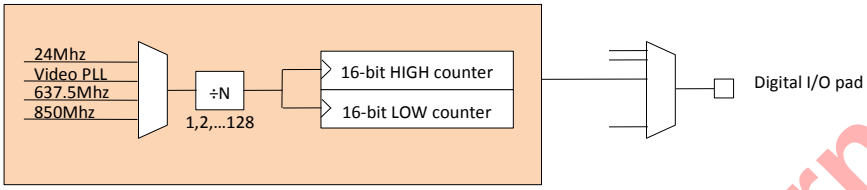
- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
- Multiple DMA/non DMA mode access support on the application side
- Supports up to 16 host channels.

20. PULSE-WIDTH MODULATION

20.1. Overview

The chip has a number of PWM generators that can be connected to various digital I/O pins. Each PWM is driven by a programmable divider driven by a 4:1 clock selector. The PWM signal is generated using two 16-bit counters. The High and Low counters are individually programmable with values between 1 and 65535. Using a combination of the divided clock (divide by N) and the HIGH and LOW counters, a wide number of PWM configurations are possible.

Figure 18. PWM Block Diagram



20.2. Register Description

Each PWM module contains two PWM generators call A and B.

20.2.1. PWM_PWM_A: PWM_A_DUTY_CYCLE

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A_HIGH: This sets the high time (in clock counts) for the PWM_A generator output
15-0	R/W	0	PWM_A_LOW: This sets the high time (in clock counts) for the PWM_A generator output

20.2.2. PWM_PWM_B: PWM_B_DUTY_CYCLE

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B_HIGH: This sets the high time (in clock counts) for the PWM_A generator output
15-0	R/W	0	PWM_B_LOW: This sets the high time (in clock counts) for the PWM_A generator output

20.2.3. PWM_MISC_REG_AB:

Bit(s)	R/W	Default	Description
31-24	R	0	Unused
23	R/W	0	PWM_B_CLK_EN: Set this bit to 1 to enable PWM A clock
22-16	R/W	0	PWM_B_CLK_DIV: Selects the divider (N+1) for the PWM A clock. See the clock tress document
15	R/W	0	PWM_A_CLK_EN: Set this bit to 1 to enable PWM A clock
14-8	R/W	0	PWM_A_CLK_DIV: Selects the divider (N+1) for the PWM A clock. See the clock tress document
7-6	R/W	0	PWM_B_CLK_SEL: Select the clock for the PWM B. See the clock tress document
5-4	R/W	0	PWM_A_CLK_SEL: Select the clock for the PWM A. See the clock tress document
3	R/W	0	DS_B_EN: This bit is only valid if PWM_B_EN is 0: if this bit is set to 1, then the PWM_B output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_B output is set low.
2	R/W	0	DS_A_EN: This bit is only valid if PWM_A_EN is 0: if this bit is set to 1, then the PWM_A output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_A output is set low.
1	R/W	0	PWM_B_EN: If this bit is set to 1, then the PWM_B output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_B output is controlled by DS_B_EN above.
0	R/W	0	PWM_A_EN: If this bit is set to 1, then the PWM_A output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_A output is controlled by DS_A_EN above.

20.2.4. DS_A_B

Bit(s)	R/W	Default	Description
31-15	R/W	0	DS_B_VAL: This value represents the delta sigma setting for channel B (PWM_B)
15-0	R/W	0	DS_A_VAL: This value represents the delta sigma setting for channel A (PWM_A)

21. SAR ADC

21.1. Overview

This SAR ADC is a general purpose ADC for measuring analog signals. The module can make RAW ADC measurements or average a number of measurements to introduce filtering. The SAR ADC is a single block so an analog mux is placed in front of the mux to allow multiple different measurements to be made sequentially. Timing of the samples, and delays between muxing are all programmable as is the averaging to be applied to the SAR ADC.

21.2. Register Description

21.2.1. SAR_ADC_REG0: Control Register #0 0x21a0

Bit(s)	R/W	Default	Description
31	R	0	PANEL DETECT level.
30	R/W	0	DELTA_BUSY: If this bit is 1, then it indicates the delta processing engine is busy
29	R/W	0	AVG_BUSY: If this bit is 1, then it indicates the averaging engine is busy
28	R/W	0	SAMPLE_BUSY: If this bit is 1, then it indicates the sampling engine is busy
27	R/W	0	FIFO_FULL:
26	R/W	0	FIFO_EMPTY:
25-21	R/W	4	FIFO_COUNT: Current count of samples in the acquisition FIFO
20-19	R/W	0	ADC_BIAS_CTRL
18-16	R/W	0	CURR_CHAN_ID: These bits represent the current channel (0..7) that is being sampled.
15	R/W	0	ADC_TEMP_SEN_SEL
14	R/W	0	SAMPLING_STOP: This bit can be used to cleanly stop the sampling process in the event that continuous sampling is enabled. To stop sampling, simply set this bit and wait for all processing modules to no longer indicate that they are busy.
13-12	R/W	0	CHAN_DELTA_EN: There are two bits corresponding to Channels 0 and 1. Channel 0 and channel 1 can be individually enabled to take advantage of the delta processing module.
11	R/W	0	Unused
10	R/W	0	DETECT_IRQ_POL: This bit sets the polarity of the detect signal. The detect signal is used during X/Y panel applications to detect if the panel is touched
9	R/W	0	DETECT_IRQ_EN: If this bit is set to 1, then an interrupt will be generated if the DETECT signal is low/high. The polarity is set in the bit above.
8-4	R/W	0	FIFO_CNT_IRQ: When the FIFO contains N samples, then generate an interrupt (if bit 3 is set below).
3	R/W	0	FIFO_IRQ_EN: Set this bit to 1 to enable an IRQ when the acquisition FIFO reaches a certain level.
2	W	0	SAMPLE_START: This bit should be written to 1 to start sampling.
1	R/W	0	CONTINUOUS_EN: If this bit is set to 1, then the channel list will be continually processed
0	R/W	0	SAMPLING_ENABLE: Setting this bit to '1' enables the touch panel controller sampling engine, averaging module, XY processing engine and the FIFO.

21.2.2. SAR_ADC_CHAN_LIST:Channel List 0x21a1

Bit(s)	R/W	Default	Description
31-27	R/W	0	unused
26-24	R/W	2	Length of the list of channels to process. If this value is 2, then only channels in bits [8:0] below are processed.
23-21	R/W	7	8th channel
20-18	R/W	6	7th channel
17-15	R/W	5	6th channel
14-12	R/W	4	5th channel
11-9	R/W	3	4th channel
8-6	R/W	2	3rd channel
5-3	R/W	1	2nd channel
2-0	R/W	0	First channel in the list of channels to process

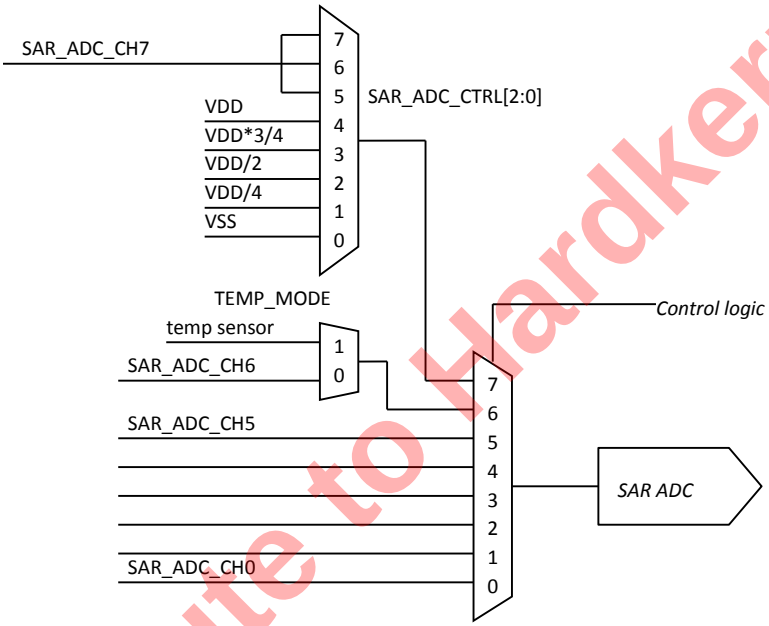
21.2.3. SAR_ADC_AVG_CNTL:Sampling/Averaging Modes 0x21a2

Each channel listed in the CHANNEL_LIST is given independent control of the number of samples to acquire and averaging mode

Bit(s)	R/W	Default	Description
31-30	R/W	0	Channel 7: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
29-28	R/W	0	Channel 6: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
27-26	R/W	0	Channel 5: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
25-24	R/W	0	Channel 4: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
23-22	R/W	0	Channel 3: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
21-20	R/W	0	Channel 2: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
19-18	R/W	0	Channel 1: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.

Bit(s)	R/W	Default	Description
17-16	R/W	0	Channel 0: Averaging mode: 0 = no averaging. 1 = simple averaging of the number of samples acquired (1,2,4 or 8). 2 = median averaging. NOTE: If these bits are set to 2, then you must set the number of samples to acquire below to 8.
15-13	R/W	0	Channel 7: Number of samples to acquire 2^N:
13-12	R/W	0	Channel 6: Number of samples to acquire 2^N:
11-10	R/W	0	Channel 5: Number of samples to acquire 2^N:
9-8	R/W	0	Channel 4: Number of samples to acquire 2^N:
7-6	R/W	0	Channel 3: Number of samples to acquire 2^N:
5-4	R/W	0	Channel 2: Number of samples to acquire 2^N:
3-2	R/W	0	Channel 1: Number of samples to acquire 2^N:
1-0	R/W	0	Channel 0: Number of samples to acquire 2^N: 0 = 1, 1 = 2, 2 = 4, 4 = 8.

21.2.4. SAR_ADC_REG3: Control Register #3 0x21a3



Bit(s)	R/W	Default	Description
31	R/W	0	CNTL_USE_SC_DLY: hold time delay was added to the start conversion clock. Unfortunately, it appears that the analog ADC design requires that we use the inverted clock so this bit is meaningless.
30	R/W	0	SAR ADC_CLK_EN: 1 = enable the SAR ADC clock
29	R/W	0	reserved
28	R/W	0	reserved
27	R/W	0	SARADC_CTRL[4]: is used to control the internal ring counter. 1 = enable the continuous ring counter. 0 = disable
26	R/W	0	SARADC_CTRL[3]: used to select the internal sampling clock phase
25~23	R/W	0	SARADC_CTRL[2:0]: 000 ssa 001 vdda/4 010 vdda/2 011 vdda*3/4 100 vdda 101, 110, 111 unused
22	R/W	0	DETECT_EN: This bit controls the analog switch that connects a 50k resistor to the X+ signal. Setting this bit to 1 closes the analog switch
21	R/W	0	ADC_EN: Set this bit to 1 to enable the ADC
20-18	R/W	2	PANEL_DETECT_COUNT: Increasing this value increases the filtering on the panel detect signal using the timebase settings in bits [17:16] below.
17-16	R/W	0	PANEL_DETECT_FILTER_TB: 0 = count 1uS ticks, 1 = count 10uS ticks, 2 = count 100uS ticks. 3 = count 1mS ticks

Bit(s)	R/W	Default	Description
15-10	R/W	20	ADC_CLK_DIV: The ADC clock is derived by dividing the 27Mhz crystal by N+1. This value divides the 27Mhz clock to generate an ADC clock. A value of 20 for example divides the 27Mhz clock by 21 to generate an equivalent 1.28Mhz clock.
9-8	R/W	1	BLOCK_DLY_SEL: 0 = count 1uS ticks, 1 = count 10uS ticks, 2 = count 100uS ticks. 3 = count 1mS ticks
7-0	R/W	10	BLOCK_DLY: After all channels in the CHANNEL_LIST have been processed, the sampling engine will delay for an amount of time before re-processing the CHANNEL_LIST again. Combined with bits [9:8] above, this value is used to generate a delay between processing blocks of channels.

21.2.5. SAR_ADC_DELAY:INPUT / SAMPLING DELAY 0x21a4

As the CHANNEL_LIST is process, the input switches are set according to the requirements of the channel. After setting the switches there is a programmable delay before sampling begins. Additionally, each channel specifies the number of samples for that particular channel. The sampling rate is programmed below.

Bit(s)	R/W	Default	Description
15-10	R	0	unused
25-24	R/W	0	INPUT_DLY_SEL: 0 = 111nS ticks, 1 = count 1uS ticks, 2 = count 10uS ticks, 3 = count 100uS ticks
16-23	R/W	3	INPUT_DLY_CNY: For channels that acquire 2,4 or 8 samples, the delay between two samples is controlled by this count (N+1) combined with the delay selection in the two bits above.
15-10	R	0	unused
9-8	R/W	0	SAMPLE_DLY_SEL: 0 = count 1uS ticks, 1 = count 10uS ticks, 2 = count 100uS ticks. 3 = count 1mS ticks
7-0	R/W	9	SAMPLE_DLY_CNY: For channels that acquire 2,4 or 8 samples, the delay between two samples is controlled by this count (N+1) combined with the delay selection in the two bits above.

21.2.6. SAR_ADC_LAST_RD: Last Sample 0x21a5

For channel 0 and channel 1, (the special X/Y channels) the last sample pushed into the FIFO for each channel is saved in a register. This allows the software to see the last sample for channel 0 and channel 1 even when the FIFO overflows. For example, if we are sampling quickly and there is a gesture on the screen, we can use the contents of the FIFO to see the direction of the gesture and use the last sample values to see where the pen finally came to rest.

Bit(s)	R/W	Default	Description
31-24	R	0	unused
23-16	R	0	LAST_CHANNEL1
15-10	R	0	unused
9-0	R	0	LAST_CHANNEL0

21.2.7. SAR_ADC_FIFO_RD: Control Register #6 (FIFO RD) 0x21a6

Bit(s)	R/W	Default	Description
31-16	R	0	Unused
15	R	0	Unused
14-12	R	0	Channel ID. This value identifies the channel associated with the data in bits [9:0] below
11-10	R	0	Unused
9-0	R	0	Sample value: 9-bit raw or averaged ADC sample written to the FIFO.

21.2.8. SAR_ADC_AUX_SW:Channel 2~7 ADC MUX, Switch Controls 0x21a7

Channels 2 ~ 7 can program the ADC input mux to any selection between 0 and 7. This register allows the software to associate a mux selection with a particular channel. In addition to the ADC mux, there are a number of switches that can be set in any particular state. Channels 2 ~ 7 share a common switch setting. Channels 0 and 1 on the other hand have programmable switch settings (see other registers below).

Bit(s)	R/W	Default	Description
31-26	R	0	unused
25-23	R/W	7	Channel 7 ADC_MUX setting when channel 7 is being measured.
22-20	R/W	7	Channel 6 ADC_MUX setting when channel 6 is being measured.
19-17	R/W	7	Channel 5 ADC_MUX setting when channel 5 is being measured.
16-14	R/W	6	Channel 4 ADC_MUX setting when channel 4 is being measured.
13-11	R/W	0	Channel 3 ADC_MUX setting when channel 3 is being measured.
10-8	R/W	1	Channel 2 ADC_MUX setting when channel 2 is being measured.

Bit(s)	R/W	Default	Description
7	R	0	unused
6	R/W	0	VREF_P_MUX setting when channel 2,3..7 is being measured
5	R/W	0	VREF_N_MUX setting when channel 2,3..7 is being measured
4	R/W	0	MODE_SEL setting when channel 2,3..7 is being measured
3	R/W	1	YP_DRIVE_SW setting when channel 2,3..7 is being measured
2	R/W	1	XP_DRIVE_SW setting when channel 2,3..7 is being measured
1	R/W	0	YM_DRIVE_SW setting when channel 2,3..7 is being measured
0	R/W	0	YM_DRIVE_SW setting when channel 2,3..7 is being measured

21.2.9. SAR_ADC_CHAN_10_SW:Channel 0, 1 ADC MUX, Switch Controls 0x21a8

Channels 0 and 1 have independent programmable switch settings when either/both of these channels are being measured.

Bit(s)	R/W	Default	Description
31-26	R	0	unused
25-23	R/W	2	Channel 1 ADC MUX setting
22	R/W	0	Channel 1 VREF_P_MUX
21	R/W	0	Channel 1 VREF_N_MUX
20	R/W	0	Channel 1 MODE_SEL
19	R/W	1	Channel 1 YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
18	R/W	1	Channel 1 XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
17	R/W	0	Channel 1 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
16	R/W	0	Channel 1 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
15-10	R		unused
9-7	R/W	3	Channel 0 ADC MUX setting
6	R/W	0	Channel 0 VREF_P_MUX
5	R/W	0	Channel 0 VREF_N_MUX
4	R/W	0	Channel 0 MODE_SEL
3	R/W	1	Channel 0 YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
2	R/W	1	Channel 0 XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
1	R/W	0	Channel 0 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
0	R/W	0	Channel 0 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND

21.2.10. SAR_ADC_DETECT_IDLE_SW:DETECT / IDLE Mode switches 0x21a9

IDLE MODE:

When nothing is being measured, the switches should be put into a safe state. This safe state is accomplished using bits [9:0] below.

DETECT MODE:

When bit [26] is set, the input muxes / switches are configured according to the bits below. Typically the software configures the switches below to correspond to the detect touch mode. That is, Y- internal MOSFET is closed so that the Y plane of the touch screen is connected to Ground. Additionally, the DETECT_EN bit (different register) set to 1 so that the 50k resistor to VDD is connected to X+. In this configuration, the detect comparator connected to the 50k resistor will be weakly pulled up to

VDD through the 50k resistor. If the user touches the screen, the X and Y planes of the touch screen will contact causing the X+ signal to be pulled to ground.

Bit(s)	R/W	Default	Description
31-27	R	0	unused
26	R/W	0	DETECT_SW_EN: If this bit is set, then bits [25:16] below are applied to the analog muxes/switches of the touch panel controller.
25-23	R/W	5	DETECT MODE ADC MUX setting
22	R/W	0	DETECT MODE VREF_P_MUX setting
21	R/W	0	DETECT MODE VREF_N_MUX setting
20	R/W	0	DETECT MODE MODE_SEL setting
19	R/W	1	DETECT MODE YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
18	R/W	1	DETECT MODE XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
17	R/W	0	DETECT MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
16	R/W	0	DETECT MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
15-10	R		Unused
9-7	R/W	5	IDLE MODE ADC MUX setting
6	R/W	0	IDLE MODE VREF_P_MUX setting
5	R/W	0	IDLE MODE VREF_N_MUX setting
4	R/W	0	IDLE MODE MODE_SEL setting
3	R/W	1	IDLE MODE YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
2	R/W	1	IDLE MODE XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
1	R/W	0	IDLE MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
0	R/W	0	IDLE MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND

21.2.11. SAR_ADC_DELTA_10:Delta Mode Deltas 0x21aa

Bit(s)	R/W	Default	Description
31-28	R	0	unused
27	R/W		TEMP_SEL
26	R/W		TS_REVE[1]
25-16	R/W	0	Channel 1 delta value when delta processing for channel 1 is enabled.
15	R/W		TS_REVE[0]
14-11	R/W		TS_C[3:0]
10	R/W	0	TS_VBG_EN
9-0	R/W	0	Channel 0 delta value when delta processing for channel 0 is enabled.

22. ETHERNET MAC

22.1. Overview

The Ethernet MAC controller provides a complete Ethernet interface from the chip to a Reduced Gigabit Media Independent Interface (RGMII) or Reduced Media Independent Interface (RMII) compliant Ethernet PHY.

22.2. Features

- 10/100/1000 MAC 3.70a
- RGMII/RMII
- AHB 32 bits internal bus
- RX FIFO 4KB, TX FIFO 2KB
- 2 MAC addresses
- EEE
- Power Management

22.3. Timing Specification

22.3.1. Management Data Timing:

Figure 19. Management Data Timing Diagram

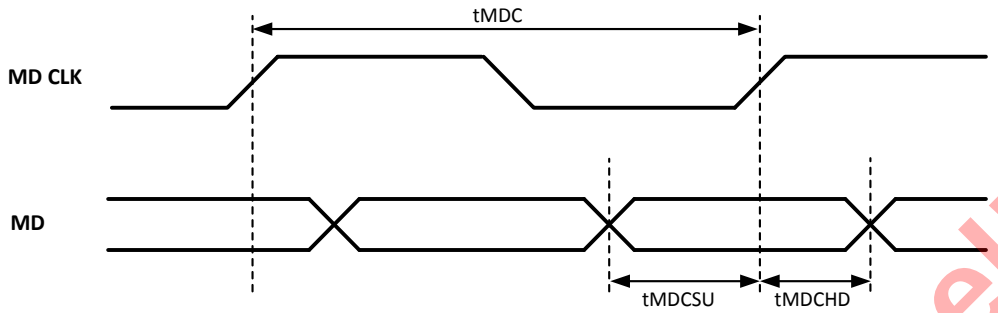


Table 21. Management Data Timing Specification

Symbol	Description	Min.	Typ.	Max.	Unit	Notes
tMDC	MDC clock Period	400	500		ns	From MAC
tMDCSU	Setup time to rising edge of MDC	10			ns	
tMDCHD	Hold time to rising edge of MDC	10			ns	

22.3.2. RMII Timing:

Figure 20. RMII Timing Diagram

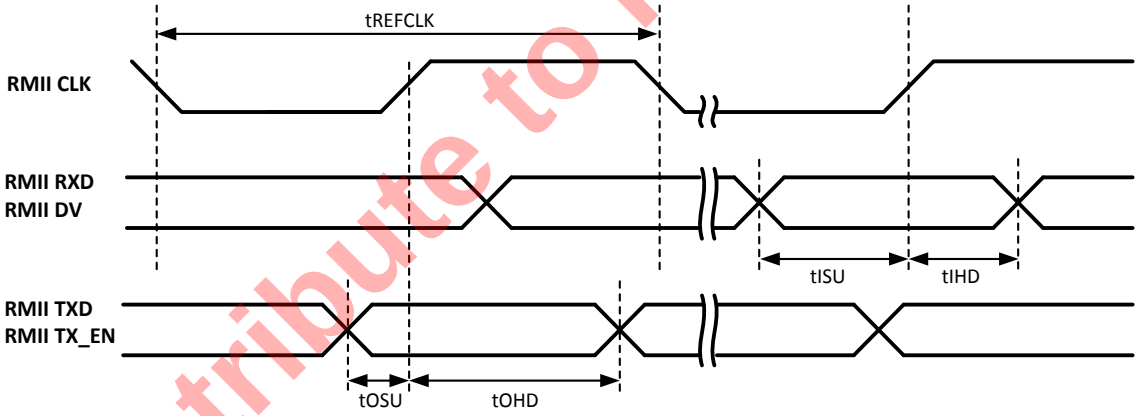


Table 22. RMII Timing Specification

Symbol	Description	Min.	Typ.	Max	Unit	Notes
tREFCLK	RMII clock period		20		ns	50MHz from PHY
tOSU	TXD & TX_EN setup time to rising edge of RMII clock	1.8	10		ns	To PHY
tOHD	TXD & TX_EN hold time to rising edge of RMII clock	1.4	10		ns	To PHY
tISU	RXD & DV setup time to rising edge of RMII clock	1.0	10		ns	From PHY
tIHD	RXD & DV hold time to rising edge of RMII clock	1.0	10		ns	From PHY

22.4. Register Description

22.4.1. PRG_ETHERNET_ADDR0 0x2050

Bit(s)	R/W	Default	Description
31	R/W	0	Set AHB to DDR interface as urgent.
30	R/W	0	RGMII mode Use RX_CLK as TX_CLK.
29-27	R/W	0	RMII & RGMII mode Select one signal from {RXDV, RXD[3:0]} to calibrate.
26	R/W	0	RMII & RGMII mode 0: test falling edge 1: test rising edge
25	R/W	0	RMII & RGMII mode Start calibration logic
24-20	R/W	0	RMII & RGMII mode 5 bits correspondent to {RXDV, RXD[3:0]}, set to 1 will delay the data capture by 1 cycle.
19-15	R/W	0	Set bit14 to 0. RMII & RGMII mode Capture input data at clock index equal to adj_delay.
14	R/W	0	Set RXDV and RXD setup time, data is aligned with index 0. When set to 1, auto delay and skew
13	R/W	0	RMII & RGMII mode Enable data delay adjustment and calibration logic.
12	R/W	0	RMII & RGMII mode Enable TX_CLK and PHY_REF_CLK generator.
11	R/W	0	RMII mode Use inverted internal clk_rmii_i to generate 25/2.5 tx_rx_clk.
10	R/W	0	Generate 25MHz clock for PHY
9-7	R/W	0	RMII & RGMII mode, M8Baby internal clock source is mp2_clk_out only. 000: invalid value. 001: mp2_clk_out is 250MHz. 010: mp2_clk_out is 500MHz. … Mp2_clk_out is “ratio” *250MHz.
6-5	R/W	0	RGMII mode, TX_CLK related to TXD 00: clock delay 0 cycle. 01: clock delay ¼ cycle. 10: clock delay ½ cycle. 11: clock delay ¾ cycle.
4	R	0	Unused
3	R/W	0	RMII mode CLK_RMII RGMII mode RX_CLK Use inverted signal when set to 1.
2	R/W	0	Sideband Descriptor Endianness Control Function: When set high, this signal configures the DMA to transfer descriptors in reverse endianness of the data format. When low (by default), the descriptors are transferred in the same endian format as the data. This signal is sampled during active reset (including soft-reset) only and ignored after reset is de-asserted.
1	R/W	0	Sideband Data Endianness Control Function: When set high, this signal configures the DMA to transfer data in big-endian format. When low (by default), the data is transferred in little-endian format. This signal is sampled during active reset (including soft-reset) only and ignored after reset is de-asserted.
0	R/W	0	PHY Interface Select Function: These pins select one of the multiple PHY interfaces of MAC. This is sampled only during reset assertion and ignored after that. 1: internal value 001: RGMII 0: internal value 100: RMII

22.4.2. PRG_ETHERNET_ADDR1 0x2051

Bit(s)	R/W	Default	Description
31-16	R	0	Unused
15	R/W	0	The result is valid
14	R/W	0	The results is rising edge test or falling edge test.
13-11	R/W	0	The signal under test.

Bit(s)	R/W	Default	Description
10	R/W	0	The Calibration logic is waiting for event.
9-5	R/W	0	The RX_CLK length in 1ns.
4-0	R/W	0	Signal switch position in 1ns.