Datasheet

SiRFatlasV®
AT551 Datasheet

August 2010
Document Number: CS-130805-DS
Issue 6

INTRODUCTION

The SiRFatlasV® processor is the next generation GPS application processor for cost effective PND (portable navigation device). Using advanced low-power process technology, excellent hardware GPS baseband, rich connectivity and analog integration, SiRFatlasV delivers a high GPS performance, flexible system design, low power and low cost platform.

Table of Contents

List of Tables

List of Figures

CORE EXECUTION FEATURES

Computing Cores

- 500 MHz ARM11 with 16KB D-cache and I-cache, 16KB D-TCM and I-TCM
- 250 MHz Enhanced DSP for GPS
- Low power advanced 65nm process

Memory Subsystem

- 64-bit 250 MHz system bus with 16 DMA channels
- 1.8V 16-bit mDDR-333 support
- 1.8V 16-bit DDR2-400 support

Advanced Autonomous GPS

- 64 channels
- -161 dBm sensitivity
- Able to acquire and track Galileo signals on all channels

Display, Graphics, and Multimedia

- Capable of supporting up to 800x480 at 16 bits color
- Supports RGB565 or 16-bit CPU I/F TFT LCD panel
- Hardware VPP (Video Post Processor) for de-interlace, scalar, color space conversion
- Two hardware overlay layers

Peripherals and Interfaces

Power Integration

- Two switching DC/DC for core (1000mA) and DRAM (500mA)
- One high PSRR and low noise 300mA LDO for I/O and peripheral
- One high PSRR and low noise 150mA LDO for analog power
- One high PSRR and low noise 10mA LDO for PLL
- One high PSRR and low noise 100mA LDO for RF
- One high PSRR and low noise 10mA LDO for RTC
-

Audio Integration

- One mono differential audio output with 93db SNR and 80db THD (“A” weighted)
- One stereo audio output with 85db SNR and 70db THD (“A” weighted) connectivity

Connectivity

- One dedicated AC97/I^2S interface
- Two dedicated UART ports
- Two USP ports for PCM, DSP, I^2S, SPI, UART mode
- Two I^2C ports
- 12-bit ADC with 4-wire touch screen controller and 3 channel analog input, stream measurement mode for low cost audio input

NAND Flash Storage

- 8-bit NAND flash interface with 12/24bit configurable BCH HW ECC support
- Supports direct boot from SLC or MLC NAND
- Supports SLC 512B, 2KB, 4KB page size
- Supports MLC 2KB, 4KB, 8KB page size

SD/MMC/MMC+ Controller

- SDIO support for Wi-Fi, DVB-T/DVB-H/T--DMB/S-DMB
- Supports direct boot from SD/MMC+ Managed NAND
- Two 8-bit SD1.01/SD 2.1/MMC4.3 ports
- Two 4-bit SD1.01/SD2.1/MMC4.3 ports

USB Connectivity

- One USB 2.0 High Speed interfaces with on chip PHY
- Can be Host, Device or OTG
- Transfer up to 480Mbps

Packaging

- 10mm x 13mm 285 ball TFBGA with .65mm pitch

Temperature Range

- -20°C ~ +70°C extended commercial grade

BLOCK DIAGRAM

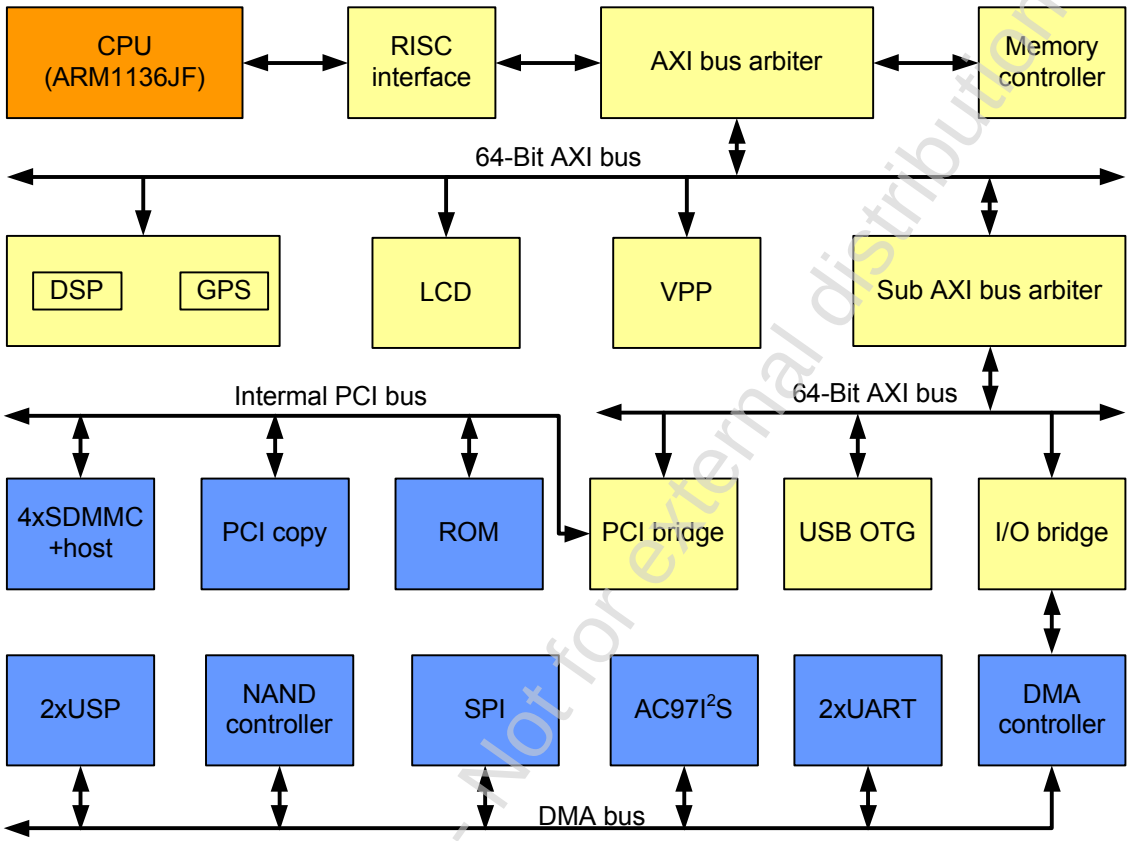


Figure 1: SiRFatlasV Block Diagram

RISC SUBSYSTEM

Overview

The RISC subsystem contains the ARM1136JF, 16KB I-Cache, 16KB D-Cache and RISC interface blocks as shown in Figure 2.

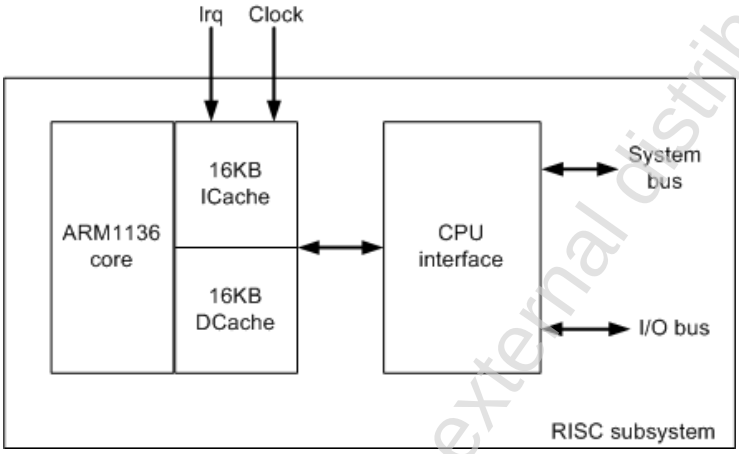


Figure 2: RISC Subsystem Block Diagram

ARM1136JF

The main CPU core of SiRFatlasV Applications Processor is ARM1136JF, which is based on the innovative ARM v6 architecture. It supports ARM Thumb® instruction sets, and features Jazelle® technology (which enables direct execution of Java bytecodes) SIMD DSP instructions on 16-bit or 8-bit data values in 32-bit registers.

The ARM1136JF-S processor core has the following features:

- 32-bit Integer unit
- Eight-stage pipeline
- Branch prediction with return stack
- Low-interrupt latency
- Instruction and Data Memory Management units (MMUs), manageable using micro TLB structures backed by a unified main TLB
- Instruction and Data L1 Caches, including a non-blocking Data cache with Hit-Under-Miss
- 64-bit interface to both L1 caches
- Write buffer (bypassable)
- Vector Floating Point coprocessor (VFP) for hardware acceleration of 3D graphics and other floating-point applications
- JTAG-based debug support

For detailed information on ARM1136JF, refer to ARM1136JF related documents available on the ARM web site http://www.arm.com.

RISC Interface

The RISC interface is an internal function block for converting the AHB Lite Bus protocol to SiRFatlasV system bus (C_AXI) and I/O bus.

Feature List

- Bus protocol convert between AHB bus and C-AXI bus
- Bus protocol convert between AHB bus and I/O bus
- Configurable I/O bus write width and read wait cycle
- Address mapping
- Illegal access protection
- Timeout
- Read pre-fetch

Functional Descriptions

Address Mapping

The ARM1136 has a 32-bit address bus, which will be translated by the RISC interface into either an access to data memory or memory-mapped registers. Bits <27:0> of the address are used as the physical address bus.

All address mapping registers must be inside the RISC interface. The RISC interface will be responsible for all address decoding before it issues the command cycles to the system. The boot ROM should have instructions about how to initialize these address-mapped registers. The programmer needs to provide the system with the initialization routine.

- ROM and compact flash

  The address mapping is defined starting from CPU reset vector (0x0000-0000). There is 512MB set aside for ROM. However, not all of this space can be used. For security reasons, this memory must be mirrored. Thus, the maximum size that can be set for the ROM cannot exceed 256MB. Also, because the mirrored memory is laid out onto two sequential 256MB segments, accessing 0x0000-0000 and 0x1000-0000 directly will yield the same result.

  NOTE – After boot-up, the address space from 0x0000-0000~0x0FFF-FFFF can be re-directed to the system memory (DDR2/MDDR) and user can only access the ROM/Flash through 0x1000-0000~0x1FFF-FFFF.

  The memory space from 0x2000-0000 to 0x47FF-FFFF is reserved.

- DSP shared memory

  The DSP shared memory takes 128MB address space (0x4800-0000~0x4FFF-FFFF).

- PCI address space

  The address between 0x5000-0000 and 0x5FFF-FFFF (256MB) is assigned to all the internal PCI devices. There is another 256MB space above it (0x6000-0000~0x6FFF-FFFF) assigned to the PCI ROM/Flash.

  NOTE – The ROM/Flash interface can be accessed as a PCI device too. In this case, user can access the ROM/Flash through PCI address space 0x6000-0000~0x6FFF-FFFF.

- Reserved space

  The address space between 0x7002-0000~0x7FFF-FFFF, is reserved. If the CPU reads from the reserved address space, a data abort operation will result. Writes to the reserved address space have no effects.

- TCM space
  - The address space between 0x7000-0000~0x7000-FFFF is assigned to ITCM of RISC.
  - The address space between 0x7001-0000~0x7001-FFFF is assigned to DTCM of RISC.

- Internal registers

  Every peripheral device occupies 64K-byte space starting from 2GB to 3GB (0x8000-0000~0xBFFF-FFFF).

- System memory

  The system memory is between 3GB and 4GB (0xC000-0000~0xFFFF-FFFF). But the maximum memory size is 512MB. The top 512MB (0xE000-0000~0xFFFF-FFFF) is Zero Bank (reading from this address range will return zero; writing to this address range has no effect.) Actual external memory size will depend on the memory volume connected to the SiRFatlasV's memory interface and users need to correctly set the configuration register in the memory controller before using the memory.

The following table shows the memory address mapping of SiRFatlasV.

Address Range	Usage	Resource Size
0xE000_0000~FFFF_FFFF	Zero Bank	512 MB
0xC000_0000~DFFF_FFFF	System Memory	512 MB
0x8000_0000~BFFF_FFFF	Internal Registers	1 GB
0x7000_0000~7000_FFFF	ITCM	64 KB
0x7001_0000~7001_FFFF	DTCM	64 KB
0x7002_0000~7FFF_FFFF	-	Reserved
0x6000_0000~6FFF_FFFF	PCI Flash/ROM	256 MB
0x5000_0000~5FFF_FFFF	Other PCI Devices	256 MB
0x4800_0000~4FFF_FFFF	DSP Shared Memory	128 MB
0x4000_0000~47FF_FFFF	-	128 MB
0x3000_0000~3FFF_FFFF	-	256 MB
0x2000_0000~2FFF_FFFF	-	256 MB
0x0000_0000~1FFF_FFFF	Flash/ROM	512 MB

Table 1: System Memory Mapping

Memory Bus Interface

The Memory Bus Interface is responsible for translating the ARM1136JF-S AHB bus protocol to the system bus protocol.

I/O Bus Interface

All the registers, interface to I/O devices, and FIFO storage in SiRFatlasV will be based on 32-bit addresses (except for some registers in the SDIO interface). Some of the registers are in System Clock Domain and others are in the I/O Clock Domain. For those registers in the I/O Clock Domain, RISC may insert wait states while accessing those devices. The users can change the number of wait states that can be programmed by setting the wait state register.

There are also some variable latency I/O devices, such as DSP IDMA port, etc. In this case, the I/O bus interface needs to be able to wait for a signal from the device to complete the access.

Internal Register Memory Mapping

Address Range	Device Mapped	Clock Domain
0x8000_0000~0x8000_FFFF	UART0	IOCLK
0x8001_0000~0x8001_FFFF	USP0	IOCLK
0x8002_0000~0x8002_FFFF	USP1	IOCLK
0x8003_0000~0x8003_FFFF	SPI	IOCLK
0x8004_0000~0x8004_FFFF	Reserved	-
0x8005_0000~0x8005_FFFF	Reserved	-
0x8006_0000~0x8006_FFFF	Codec Interface	IOCLK
0x8007_0000~0x8007_FFFF	NAND Flash Interface	IOCLK
0x8008_0000~0x8008_FFFF	Reserved	-
0x8009_0000~0x8009_FFFF	GPIO	IOCLK
0x800A_0000~0x800A_FFFF	GPS Interface	IOCLK
0x800B_0000~0x800B_FFFF	I^2C Interface	IOCLK
0x800C_0000~0x800C_FFFF	PCI Bridge	IOCLK
0x800D_0000~0x800D_FFFF	PCI_COPY	IOCLK
0x800E_0000~0x800E_FFFF	eFuse	IOCLK
0x800F_0000~0x800F_FFFF	PWM	IOCLK
0x8010_0000~0x8010_FFFF	UART1	IOCLK
0x8011_0000~0x8011_FFFF	Reserved	-
0x8012_0000~0x8012_FFFF	TSCIF	IOCLK
0x8013_0000~0x8013_FFFF	VPP	SYSCLK

Address Range	Device Mapped	Clock Domain
0x8014_0000~0x8014_FFFF	Reserved	-
0x8015_0000~0x8FFF_FFFF	Reserved	-
0x9000_0000~0x9000_FFFF	RISC Interface	SYSCLK
0x9001_0000~0x9001_FFFF	DSP Interface	SYSCLK
0x9002_0000~0x9002_FFFF	Interrupt Controller Interface	SYSCLK
0x9003_0000~0x9003_FFFF	Resource Sharing Controller	SYSCLK
0x9004_0000~0x9004_FFFF	CPUIO2RCT bridge	SYSCLK
0x9005_0000~0x9005_FFFF	OS Timer	SYSCLK
0x9006_0000~0x9006_FFFF	RTCIO domain	SYSCLK
0x9007_0000~0x9007_FFFF	Reset Controller	SYSCLK
0x9008_0000~0x9008_FFFF	CLK Controller	SYSCLK
0x9009_0000~0x9009_FFFF	Reserved	-
0x900A_0000~0x97FF_FFFF	Reserved	-
0x9800_0000~0x9800_FFFF	Reserved	-
0x9900_0000~0x9900_FFFF	USB	SYSCLK
0x9a00_0000~0x9FFC_FFFF	Reserved	-
0x9FFD_0000~0x9FFD_FFFF	DSP/GPS AXI arbiter	SYSCLK
0x9FFE_0000~0x9FFE_FFFF	Level2 AXI arbiter	SYSCLK
0x9FFF_0000~0x9FFF_FFFF	System Arbiter	SYSCLK
0xA000_0000~0xA000_FFFF	Memory Controller	SYSCLK
0xA001_0000~0xA001_FFFF	ROM Controller	IOCLK
0xA002_0000~0xA002_FFFF	Embedded ROM	IOCLK
0xA003_0000~0xAFFF_FFFF	Reserved	-
0xB000_0000~0xB000_FFFF	DMA Controller and I/O Bridge	IOCLK
0xB001_0000~0xB00F_FFFF	Reserved	-
0xB800_0000~0xB800_FFFF	LCD Controller	SYSCLK
0xB801_0000~0xB8FF_FFFF	Reserved	-
0xB900_0000~0xB9FF_FFFF	Reserved	-
0xBA00_0000~0xBA00_FFFF	Reserved	-
0xBA01_0000~0xBFFF_FFFF	Reserved	-

Table 2: Internal Memory Mapped Register Mapping

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O interface	0x90000000

Table 3: RISC Interface Base Address

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	RISCINT_FIFO_INVALID	FIFO invalid register
0x0004	Reserved	-
0x0008	Reserved	-
0x000C	RISCINT_BOOT_UP	Boot-up register
0x0010	RISCINT_WAIT1	Wait states register
0x0014	RISCINT_WAIT2	Wait states register
0x0018	RISCINT_WIDTH	Width control register
0x001C	RISCINT_TIMEOUT	Timeout register
0x0020	RISCINT_TIMEOUT_INT	Timeout interrupt register
0x0024	RISCINT_PREFETCH_EN	Pre-fetch enable register
0x0028	Reserved	-
0x002c	RISCINT_CPUSYNC_MODE	CPU synchronous mode
0x0030	Reserved	-
0x0034	RISCINT_TIMEOUT_ADDR	Timeout address
0x0038	RISCINT_1_4_ENABLE	1:4 mode enable register
Others	Reserved	-

Table 4: RISC Interface Register Mapping

Register Descriptions

- RISC Interface pre-fetch FIFO Invalid Register (RISCINT_FIFO_INVALID) – 0x0000

  The RISC interface has a read pre-fetch FIFO, set the pre-fetch FIFO invalid register to invalidate the data in the pre-fetch FIFO.

Bit	Name	Default	Description
0 (R/W)	RISCINT_FIFO_INVALID	1'b0	Pre-fetch FIFO Invalid bit. Set to 1 will invalidate the FIFO.

Table 5: RISC Interface Pre-fetch FIFO Invalid Register

- RISC Interface Boot-up Register (RISCINT_BOOT_UP) – 0x000C

  After boot-up, the boot program needs to do re-direct the ROM access to the shadowed memory space. Set COLD_BOOT to 1, RISC interface will re-direct ROM access in the address 0x0000_0000~0FFF_FFFF to shadowed memory 0xC0000~0xCFFFFFF, the access in address 0x1000_0000~1FFF_FFFF will be kept as ROM access.

  Due to the pipeline nature of the RISC core, after the boot-up register has been set, the user cannot access the ROM address space immediately. It needs to insert at least one NOP between them.

Bit	Name	Default	Description
0 (R/W)	COLD_BOOT	1'b0	Set to 1 when system is cold boot-up.
31:1	-	31'h0	Reserved.

Table 6: RISC Interface Boot-up Register

- RISC Interface Wait States Register (RISCINT_WAIT1) – 0x0010

  RISC I/O read cycle can be inserted with up to 16 wait states. There are 4 different wait values to be set at the same time. Each value is for some group of internal devices.

Bit	Name	Default	Description
7:0 (R/W)	WS0<7:0>	8'h0	-
15:8 (R/W)	WS1<7:0>	8'h1	Wait state number used by RISC Interface, system arbiter, MEMC, DSPIF, GPS, DSPGPS arbiter, LCD, and VPP.
23:16 (R/W)	WS2<7:0>	8'h2	Wait state number used by USB, interrupt controller, reset controller and clock controller.
31:24 (R/W)	WS3<7:0>	8'h3	Wait state number used by OS timer, RSC, CPUIO2RTC, and SUBAXI arbiter.

Table 7: RISC Interface Wait States Register

- RISC Interface Wait States Register (RISCINT_WAIT2) – 0x0014

  RISC I/O read cycle can be inserted with up to 16 wait states. There are four wait values to be set at the same time. Each value is for a particular group of internal devices.

Bit	Name	Default	Description
7:0 (R/W)	WS4<7:0>	8'h5	Wait state number used by NAND, CODEC, UART0, UART1, and SYS2PCI.
15:8 (R/W)	WS5<7:0>	8'h6	Wait state number used by USP0, USP1, DMA controller, and ROM controller.
23:16 (R/W)	WS6<7:0>	8'h7	Wait state number used by SPI, and TSC interface.
31:24 (R/W)	WS7<7:0>	8'h7	Wait state number used by other devices.

Table 8: RISC Interface Wait States Register

The actual wait cycle for WS4~WS7 is WSx+io_sys_ratio.

- sysclk: ioclk = 1:1 io_sys_ratio = 1
- sysclk: ioclk = 1:2 io_sys_ratio = 2
- sysclk: ioclk = 1:4 io_sys_ratio = 4

- RISC Interface Width Control Register (RISCINT_WIDTH) – 0x0018

  RISC I/O Bus can access two different clock domains, one is SYS_CLK and the other is IO_CLK. SYS_CLK can be 1, 2, or 4 times of IO_CLK. The write pulse width should be programmable.

Bit	Name	Default	Description
3:0 (R/W)	PW0<3:0>	4'h0	Write pulse width for System clock domain
7:4 (R/W)	PW1<3:0>	4'h1	Write pulse width for IO clock domain. When SYS_CLK:IO_CLK = 1, PW1 = 0 When SYS_CLK:IO_CLK = 2, PW1 = 1 When SYS_CLK:IO_CLK = 4, PW1 = 3
31:8	-	24'h0	Reserved

Table 9: RISC Interface Width Control Register

- RISC Interface Timeout Register (RISCINT_TIMEOUT) – 0x001C

  When RISC accesses I/O device and the devices provide no response within a certain period, the RISC interface will timeout and generate an interrupt.

Bit	Name	Default	Description
15:0 (R/W)	TO<15:0>	16'hFFFF	Timeout value in system clock.
30:16	-	15'h0	Reserved.
31 (R/W)	TO_EN	1'b0	Timeout enable. 1: Enables timeout check. 0: Disables timeout check (default).

Table 10: RISC Interface Timeout Register

- RISC Interface Timeout Interrupt Register (RISCINT_TIMEOUT_INT) – 0x0020

  When RISC accesses I/O devices and the devices provide no response within a certain period, the RISC interface will timeout and generate an interrupt.

Bit	Name	Default	Description
0 (R/W)	TO_INT	1'b0	Timeout interrupt status. 1: There is timeout interrupt. 0: No timeout interrupt. Write a 1 to this bit will clear the interrupt.
31:1	-	31'h0	Reserved.

Table 11: RISC Interface Timeout Interrupt Register

- RISC Interface Timeout Interrupt Register (RISCINT_TIMEOUT_ADDR) – 0x0034

  RISCINT_TIMEOUT_ADDR indicates the address when a timeout happens

Bit	Name	Default	Description
31:0	TO_ADDR<31:0>	31'h0	Indicates a timeout occurs when accessing an address

Table 12: RISC Interface Timeout Interrupt Register

- RISC Interface Pre-fetch Enable Register (RISCINT_PREFETCH_EN) – 0x0024

  There is a pre-fetch FIFO inside of the RISC interface. This FIFO can be enabled to speed up the CPU read.

Bit	Name	Default	Description
0 (R/W)	PREFETCH_EN	1'b1	Pre-fetch enable bit. 1: Enabled 0: Disabled
31:1	-	31'h0	Reserved

Table 13: RISC Interface Pre-fetch Enable Register

- RISC Interface CPU Synchronous Mode Control Register (RISCINT_CPUSYNC_MODE) – 0x002C

  ARM1136JF-S Provides two set control inputs that you can use to configure synchronous or asynchronous operation of each AHB clock domain (HCLKIRW and HCLKPD). Register RISCINT_CPUSYNC_MODE used to configure these two set inputs.

Bit	Name	Default	Description
0 (R/W)	SYNEN	1'b0	Core Clock synchronous enable 1: Enabled, SYNCENIRW=SYNCENPD=1 0: Disabled, SYNCENIRW=SYNCENPD=0
1 (R/W)	HSYNEN	1'b0	AHB Clock synchronous enable 1: Enabled, HSYNCENIRW=HSYNCENPD=1 0: Disabled, HSYNCENIRW=HSYNCENPD=0
31:2	-	30'h0	Reserved

Table 14: RISC Interface CPU Synchronous Mode Control Register

- RISC Interface Time Out Address Register (RISCINT_TIMEOUT_ADDR) – 0x0034

  When RISC interface timeout occurs, the current address will be latched to these registers.

Bit	Name	Default	Description
31:0 (R/W)	TIMEOUT_ADDR<31:0>	32'h0	Current address when timeout occurs

Table 15: RISC Interface Timeout Address Register

- RISC Interface 1:4 Mode Enable Register (RISCINT_1_4_ENABLE) – 0x0038

Bit	Name	Default	Description
0 (R/W)	1_4_ENABLE	1'b1	0: Disable SYSCLK:IOCLK 1:4 mode 1: Enable SYSCLK:IOCLK 1:4 mode
31:1	-	31'h0	reserved

Table 16: RISC Interface 1:4 Mode Enable Register

DSP SUBSYSTEM

DSP Core

Overview

The DSP subsystem in SiRFatlasV is designed for high-performance applications. It includes a 16-bit general purpose DSP core, a program and data memory, a system bus interface, a host interface, an FFT accelerator and an EPD (Energy Peak Detection).

Feature List

The DSP Subsystem in SiRFatlasV provides the following features:

- Faster speed (250MHz)
- Memory-driven DMA transfer
- 1x, 2x and 4x clock modes
- Controllable RTC, UART, USP, GPIO, Interrupt Controller, and all PCI devices
- Idle down mode
- High performance hardware FFT
- Hardware EPD
- Five interrupt sources

The flexible architecture and comprehensive instruction set allow the DSP to perform multiple operations simultaneously. In one processor cycle, the processor can:

- Generate the next program address
- Fetch the next instruction
- Perform one or two data moves
- Update one or two data address pointers
- Perform a computational operation

Functional Descriptions

For detailed information please contact a SiRF Application Engineer.

Register Definition

For detailed information please contact a SiRF Application Engineer.

GPS Baseband

Overview

GPS baseband in SiRFatlasV is a high performance coprocessor for GPS signal processing.

Feature List

- Basic features
  - Track mode
  - 58 correlators work in tracking mode
  - RF FIFO interface
  - Host interface
  - Supports up to 64 channels
  - Supports Galileo GPS
  - Match filter and the correlator can work together in acquisition mode
- DMA
  - Correlator DMA
  - Match filter DMA
- Match filter
  - Acquire mode
  - 2046 correlators

Functional Descriptions

For detailed information, contact a SiRF Application Engineer.

Register Definition

For detailed information, contact a SiRF Application Engineer.

Programming Guide

For detailed information, contact a SiRF Application Engineer.

MEMORY SUBSYSTEM

System Arbiter

Overview

The system bus in SiRFatlasV is a 64-bit high-performance, low-power bus. In the system arbiter architecture, there are eight bus masters: RISC, DSPGPS, VPP, LCD Controller, Sub-AXI system and three reserved bus masters. In addition to bus masters, the system arbiter architecture also includes a bus slave, the memory controller.

Bus Masters	Block Name
System Bus Master0	Reserved
System Bus Master1	RISC
System Bus Master2	DSPGPS
System Bus Master3	Reserved
System Bus Master4	Reserved
System Bus Master5	LCD
System Bus Master6	VPP
System Bus Master7	SUBAXI

Table 17: System Bus Master Allocation

When the bus master accesses the memory bus, it must own the bus first. To own the bus, the master must assert a request signal (M_WAVALID or M_RAVALID), then the arbiter can grant the bus to the master with a grant signal (M_WAREADY or M_RAREADY).

Feature List

- Arbitrates all system requests in the round-robin scheme
- Two priority groups for arbitration and the switch scheme between the two groups can be controlled
- Each master can be enabled or disabled
- Write protected when the write data operation is timed out
- Each master can be continuously responded by setting ACC numbers

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O Interface	0x9fff0000

Table 18: Sample Base Address

Register Descriptions

All registers in the System Arbiter are reserved by SiRF. For more information about these registers, please contact a SiRF Application Engineer. This document will not list register descriptions of the System Arbiter.

Memory Controller

Overview

The memory controller supports 16-bit DDR2/MDDR devices.

The memory controller is a complete embedded memory controller that interfaces directly to the input/output (I/O) drivers in the ASIC I/O pad ring that are connected directly to memory devices on custom integrated circuit boards. Communication to and from the memory devices is handled via the ASIC interface.

Feature List

- Supported Memory types:
  - 1.8V 200MHz DDR2
  - 1.8V 166MHz mobile DDR
- Total memory size is up to 128MB
- Support DDR2 with eight banks. In this configuration, memory BA2 should be connected with X_MA[13] on board.
- Fully pipelined command and read and write data interfaces to the controller
- Advanced bank look-ahead features for high memory throughput
- Interface to a standard AXI port
- A programmable register interface to control memory device parameters and protocols including auto pre-charge
- Full initialization of memory on controller reset
- Clock frequencies from 100MHz to 200MHz supported
- Delay Compensation Circuitry (DCC) for reliable data send and capture timing

Pin Descriptions

External Pin Descriptions

Pin Name	Pin Direction	Description
X_MD<15:0>	Bidirectional	Memory data
X_MA<13:0>	Output	Memory address X_MA<13> should be connected to BA<2> of external memory on board when working at eight bank mode.
X_M_BA<1:0>	Output	Bank signal
X_MCS_B	Output	Chip select signal
X_MDQM<1:0>	Output	Data mask signal
X_MDQS<1:0>	Bidirectional	Data strobe
X_MRAS_B	Output	Row address strobe signal
X_MCAS_B	Output	Column address strobe signal
X_MWE_B	Output	Write enable signal
X_MCLK_O	Output	Differential memory clock
X_MCLKB_O	Output	Differential memory clock
X_MCKE	Output	Memory clock enable

Table 19: External Pin Description

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O Interface	0xa0000000

Table 20: Sample Base Address

Register Mapping

There are 53 registers in the memory controller and 8 registers in the PHY wrap. All the registers should be configured correctly before setting the START bit in register MEM_CTL_06 and START bit in register MEMCIF_DLL_START. And after you set the two START bit, the memory controller will configure the memory device automatically. All masters should wait bit[4] (DRAM Initialization Complete) in MEM_CTL_47 to high before starting memory access.

The following table is address mapping of these registers:

RISC Address <11:0>	Register	Description
0x0000	MEM_CTL_00	AUTO_REFRESH_MODE AREFRESH AP ADDR_CMP_EN
0x0004	MEM_CTL_01	DLLLOCKREG CONCURRENTAP BANK_SPLIT_EN AXI0_FIFO_TYPE_REG
0x0008	MEM_CTL_02	FAST_WRITE DRIVE_DQ_DQS DLL_BYPASS_MODE
0x000C	MEM_CTL_03	NO_CMD_INIT INTRPTWRITEA INTRPTREADA INTRPTAPBURST
0x0010	MEM_CTL_04	PWRUP_SREFRESH_EXIT PRIORITY_EN POWER_DOWN PLACEMENT_EN
0x0014	MEM_CTL_05	RW_SAME_EN REG_DIMM_ENABLE RD2RD_TURN
0x0018	MEM_CTL_06	WRITEINTERP TRAS_LOCKOUT START SREFRESH
0x001C	MEM_CTL_07	CS_MAP AXI0_W_PRIORITY AXI0_R_PRIORITY WRITE_MODEREG
0x0020	MEM_CTL_08	ADDR_PINS MAX_CS_REG LOWPOWER_REFRESH_ENABLE

RISC Address <11:0>	Register	Description
0x0024	MEM_CTL_09	COLUMN_SIZE CASLAT AGE_COUNT
0x0028	MEM_CTL_10	TCKE Q_FULLNESS OUT_OF_RANGE_TYPE COMMAND_AGE_COUNT
0x002C	MEM_CTL_11	APREBIT TWTR TWR_INT TRRD
0x0030	MEM_CTL_12	INT_ACK INITAREF CASLAT_LIN_GATE CASLAT_LIN
0x0034	MEM_CTL_13	TRP TDAL MAX_COL_REG
0x0038	MEM_CTL_14	MAX_ROW_REG LOWPOWER_CONTROL LOWPOWER_AUTO_ENABLE
0x003C	MEM_CTL_15	TRC TMRD
0x0040	MEM_CTL_16	This register intentionally blank
0x0044	MEM_CTL_17	DLL_LOCK
0x0048	MEM_CTL_18	TRFC
0x004C	MEM_CTL_19	TRCD_INT TRAS_MIN
0x0050	MEM_CTL_20	TREF OUT_OF_RANGE_LENGTH
0x0054	MEM_CTL_21	AXI0_EN_SIZE_LT_WIDTH_INSTR
0x0058	MEM_CTL_22	This register intentionally blank
0x005C	MEM_CTL_23	LOWPOWER_EXTERNAL_CNT
0x0060	MEM_CTL_24	LOWPOWER_POWER_DOWN_CNT LOWPOWER_INTERNAL_CNT

RISC Address <11:0>	Register	Description
0x0064	MEM_CTL_25	LOWPOWER_SELF_REFRESH_CNT LOWPOWER_REFRESH_HOLD
0x0068	MEM_CTL_26	TDLL
0x006C	MEM_CTL_27	TRAS_MAX TPDEX
0x0070	MEM_CTL_28	TXSR TXSNR
0x0074	MEM_CTL_29	VERSION
0x0078	MEM_CTL_30	OUT_OF_RANGE_ADDR[31:0]
0x007C	MEM_CTL_31	SWAP_EN ENABLE_QUICK_SREFRESH ACTIVE_AGING OUT_OF_RANGE_ADDR[34:32]
0x0080	MEM_CTL_32	PORT_DATA_ERROR_TYPE CKE_DELAY TREF_ENABLE SWAP_PORT_RW_SAME_EN
0x0084	MEM_CTL_33	PORT_DATA_ERROR_ID PORT_CMD_ERROR_ID OUT_OF_RANGE_SOURCE_ID PORT_CMD_ERROR_TYPE
0x0088	MEM_CTL_34	EMRS2_DATA_0
0x008c	MEM_CTL_35	EMRS2_DATA_1
0x0090	MEM_CTL_36	TINIT
0x0094	MEM_CTL_37	PORT_CMD_ERROR_ADDR[31:0]
0x0098	MEM_CTL_38	ODT_ADD_TURN_CLK_EN EIGHT_BANK_MODE DQS_N_EN PORT_CMD_ERROR_ADDR[34:32]
0x009c	MEM_CTL_39	ODT_RD_MAP_CS0 DRAM_CLK_DISABLE REDUC ODT_ALT_EN
0x00a0	MEM_CTL_40	RTT_0 ODT_WR_MAP_CS1

RISC Address <11:0>	Register	Description
		ODT_WR_MAP_CS0 ODT_RD_MAP_CS1
0x00a4	MEM_CTL_41	W2R_SAMECS_DLY W2R_DIFFCS_DLY TRTP TDFI_DRAM_CLK_DISABLE
0x00a8	MEM_CTL_42	TDFI_CTRL_DELAY TDFI_CTRLUPD_MIN RDLAT_ADJ DRAM_CLASS
0x00ac	MEM_CTL_43	TDFI_PHY_WRLAT_BASE TDFI_PHY_WRLAT TDFI_PHY_RDLAT TDFI_DRAM_CLK_ENABLE
0x00b0	MEM_CTL_44	WRLAT_ADJ WRLAT TDFI_RDDATA_EN_BASE TDFI_RDDATA_EN
0x00b4	MEM_CTL_45	DLL_RST_ADJ_DLY TFAW OCD_ADJUST_PUP_CS_0 OCD_ADJUST_PDN_CS_0
0x00b8	MEM_CTL_46	INT_MASK TMOD
0x00bc	MEM_CTL_47	TDFI_CTRLUPD_MAX INT_STATUS
0x00c0	MEM_CTL_48	TDFI_PHYUPD_TYPE0 TDFI_PHYUPD_RESP
0x00c4	MEM_CTL_49	EMRS1_DATA_0 DLL_RST_RESP
0x00c8	MEM_CTL_50	EMRS3_DATA_0 EMRS1_DATA_1
0x00cc	MEM_CTL_51	MRS_DATA_0 EMRS3_DATA_1
0x00d0	MEM_CTL_52	TCPD MRS_DATA_1

RISC Address <11:0>	Register	Description
0x0200	MEMCIF_TIM_CONFIG	CMD_SEL RDEN_SEL GATE_SEL RESYNC_SEL BA2_SEL WDAT_SEL
0x0204	MEMCIF_DLL_START	START
0x0208	MEMCIF_DLL_STATUS	CLOCK FLOCK LOCK_VALUE
0x020c	MEMCIF_DLL_CONFIG1	DLL_ON FORCE INCREMENT START_POINT
0x0210	MEMCIF_DLL_CONFIG2	SHIFTC OFFSETC OFFSETD
0x0214	MEMCIF_DLL_CONFIG3	OFFSET3 OFFSET2 OFFSET1 OFFSET0
0x0218	MEMCIF_PHY_CONFIG	ADB_EN DRV_BUS FNC_FB HALF
0x021c	MEMCIF_ADB_CHECK	ADB_CHECK

Table 21: Address Mapping

Register Descriptions

- MEM_CTL_00 – OFFSET: 0x0

Bit	Name	Default	Description
24 (R/W)	AUTO_REFRESH_MODE	1'h0	Sets the mode for when the automatic refresh will occur. If auto_refresh_mode is set and a refresh is required to memory, the memory controller will delay this refresh until the end of the current transaction (if the transaction is fully contained inside a single page), or until the current

Bit	Name	Default	Description
			transaction hits the end of the current page. • 'b0 = Issue refresh on the next DRAM burst boundary, even if the current command is not complete. • 'b1 = Issue refresh on the next command boundary.
16 (W)	AREFRESH	1'h0	Initiates an automatic refresh to the DRAM devices based on the setting of the auto_refresh_mode parameter. If there are any open banks when this parameter is set, the Memory Controller will automatically close these banks before issuing the auto-refresh command. This parameter will always read back "0". • 'b0 = No action • 'b1 = Issue refresh to the DRAM devices
8 (R/W)	AP	1'h0	Enables auto pre-charge mode for DRAM devices. • 'b0 = Auto pre-charge mode disabled. Memory banks will stay open until another request requires this bank, the maximum open time (tras_max) has elapsed, or a refresh command closes all the banks. • 'b1 = Auto pre-charge mode enabled. All read and write transactions must be terminated by an auto pre-charge command. If a transaction consists of multiple read or write bursts, only the last command is issued with an auto pre-charge.
0 (R/W)	ADDR_CMP_EN	1'h0	Enables address collision/data coherency detection as a condition when using the placement logic to fill the command queue. • 'b0 = Disabled • 'b1 = Enabled

Table 22: MEM_CTL_00 – OFFSET: 0x0

- MEM_CTL_01 – OFFSET: 0x4

Bit	Name	Default	Description
24 (R)	DLLLOCKREG	1'h0	DLL lock/unlock. Read-only
16 (R/W)	CONCURRENTAP	1'h0	Enables concurrent auto pre-charge. Some DRAM devices do not allow one bank to be auto pre-charged while another bank is reading or writing. The JEDEC standard allows concurrent auto pre-charge. Set this parameter for the DRAM device being used. • 'b0 = Concurrent auto pre-charge disabled. • 'b1 = Concurrent auto pre-charge enabled.
8 (R/W)	BANK_SPLIT_EN	1'h0	Enables bank splitting as a condition when using the placement logic to fill the command queue.

Bit	Name	Default	Description
			• 'b0 = Disabled • 'b1 = Enabled
1:0 (R/W)	AXI0_FIFO_TYPE_REG	2'h0	Sets the relativity of the clock domains between the AXI port and the Memory Controller core clock. • 'b00 = Asynchronous • 'b01 = 2:1 Port:Core Pseudo-Synchronous • 'b10 = 1:2 Port:Core Pseudo-Synchronous • 'b11 = Synchronous

Table 23: MEM_CTL_01 – OFFSET: 0x4

- MEM_CTL_02 – OFFSET: 0x8

Bit	Name	Default	Description
24 (R/W)	FAST_WRITE	1'h0	Controls when the write commands are issued to the DRAM devices. • 'b0 = The memory controller will issue a write command to the DRAM devices when it has received enough data for one DRAM burst. In this mode, write data can be sent in any cycle relative to the write command. This mode also allows for multi-word write command data to arrive in non-sequential cycles. • 'b1 = The memory controller will issue a write command to the DRAM devices after the first word of the write data is received by the memory controller. The first word can be sent at any time relative to the write command. In this mode, multi-word write command data must be available to the memory controller in sequential cycles.
8 (R/W)	DRIVE_DQ_DQS	1'b0	Selects if the DQ output enables and DQS output enables will be driven active when the memory controller is in an idle state. • 'b0 = Leave the output enables de-asserted when idle. • 'b1 = Drive the output enables active when idle. If third party PHY is used, this bit can be ignored.
0 (R/W)	DLL_BYPASS_MODE	5'h10	Defines the behavior of the DLL bypass logic and establishes which set of delay parameters will be used. • 'b0 = Normal operational mode. • 'b1 = Bypass the DLL master delay line.

Bit	Name	Default	Description
			If third party PHY is used, this bit can be ignored.

Table 24: MEM_CTL_02 – OFFSET: 0x8

- MEM_CTL_03 – OFFSET: 0xC

Bit	Name	Default	Description
24 (R/W)	NO_CMD_INIT	1'h0	Disables DRAM commands until DLL initialization is complete and tdll has expired. • 'b0 = Issue only REF and PRE commands during DLL initialization of the DRAM devices. • 'b1 = Do not issue any type of command during DLL initialization of the DRAM devices.
16 (R/W)	INTRPTWRITEA	1'h0	Enables interrupting of a combined write with auto pre-charge command with another read or write command to the same bank before the first write command is completed. • 'b0 = Disable interrupting a combined write with auto pre-charge command with another read or write command to the same bank. • 'b1 = Enable interrupting a combined write with auto pre-charge command with another read or write command to the same bank.
8 (R/W)	INTRPTREADA	1'h0	Enables interrupting of a combined read with auto pre-charge command with another read command to the same bank before the first read command is completed. • 'b0 = Disable interrupting the combined read with auto pre-charge command with another read command to the same bank. • 'b1 = Enable interrupting the combined read with auto pre-charge command with another read command to the same bank.
0 (R/W)	INTRPTAPBURST	1'h0	Enables interrupting an auto pre-charge command with another command for a different bank. If enabled, the current operation will be interrupted. However, the bank will be pre-charged as if the current operation were allowed to continue. • 'b0 = Disable interrupting an auto pre-charge operation on a different bank. • 'b1 = Enable interrupting an auto pre-charge operation on a different bank.

Table 25: MEM_CTL_03 – OFFSET: 0xC

- MEM_CTL_04 – OFFSET: 0x10

Bit	Name	Default	Description
24 (R/W)	PWRUP_SREFRESH_EXIT	1'h0	Allows controller to exit power-down mode by executing a self-refresh exit instead of the full memory initialization. This parameter provides a means to skip full initialization when the DRAM devices are in a known self-refresh state. • 'b0 = Disabled • 'b1 = Enabled
16 (R/W)	PRIORITY_EN	1'h0	Enables priority as a condition when using the placement logic to fill the command queue. • 'b0 = Disabled • 'b1 = Enabled
8 (R/W)	POWER_DOWN	1'h0	When this parameter is written with a "1", the memory controller will complete processing of the current burst for the current transaction (if any), issue a pre-charge all command and then disable the clock enable signal to the DRAM devices. Any subsequent commands in the command queue will be suspended until this parameter is written with a "0". • 'b0 = Enable full power state. • 'b1 = Disable the clock enable and power down the memory controller.
0 (R/W)	PLACEMENT_EN	1'h0	Enables using the placement logic to fill the command queue. • 'b0 = Placement logic is disabled. The command queue is a straight FIFO. • 'b1 = Placement logic is enabled. The command queue will be filled according to the placement logic factors.

Table 26: MEM_CTL_04 – OFFSET: 0x10

- MEM_CTL_05 – OFFSET: 0x14

Bit	Name	Default	Description
16 (R/W)	RW_SAME_EN	1'h0	Enables read/write grouping as a condition when using the placement logic to fill the command queue. • 'b0 = Disabled • 'b1 = Enabled
8 (R/W)	REG_DIMM_ENABLE	1'h0	Enables registered DIMM operations to control the address and command pipeline of the memory controller. • 'b0 = Normal operation • 'b1 = Enable registered DIMM operation
0 (R/W)	RD2RD_TURN	1'h0	Adds an additional clock between back-to-back read operations to different chip selects. The extra clock is required for mobile DDR devices where tac_max > (period/2 + tac_min). Without this additional clock, the first read may drive DQS out at tac_max and the second read may drive DQS out at tac_min, resulting in a contention on the DQS line. • 'b0 = Disabled • 'b1 = Enabled

Table 27: MEM_CTL_05 – OFFSET: 0x14

- MEM_CTL_06– OFFSET: 0x18

Bit	Name	Default	Description
24 (R/W)	WRITEINTERP	1'h0	Defines whether the memory controller can interrupt a write burst with a read command. Some memory devices do not allow this functionality. • 'b0 = The device does not support read commands interrupting write commands. • 'b1 = The device does support read commands interrupting write commands.
16 (R/W)	TRAS_LOCKOUT	1'h0	Defines the tRAS lockout setting for the DRAM device. tRAS lockout allows the memory controller to execute auto pre-charge commands before the tras_min parameter has expired. • 'b0 = tRAS lockout not supported by memory device. • 'b1 = tRAS lockout supported by memory device.
8 (R/W)	START	1'h0	With this parameter set to 'b0, the memory controller will not issue any commands to the DRAM devices or respond to any signal activity except for reading and writing parameters. Once this parameter is set to 'b1, the memory controller will respond to inputs from the ASIC. When set,

Bit	Name	Default	Description
			the memory controller begins its initialization routine. • 'b0 = Controller is not in active mode. • 'b1 = Initiate active mode for the memory controller. Note: The user must wait for the initialization complete bit to be set in the int_status parameter and for the dfi_init_complete signal to be asserted from the PHY before submitting any transactions.
0 (R/W)	SREFRESH	1'h0	When this parameter is written with a 'b1, the DRAM device(s) will be placed in self-refresh mode. For this, the current burst for the current transaction (if any) will complete, all banks will be closed, the self-refresh command will be issued to the DRAM, and the clock enable signal will be de-asserted. The system will remain in self-refresh mode until this parameter is written with a 'b0. The DRAM devices will return to normal operating mode after the self-refresh exit time (txsr) of the device and any DLL initialization time for the DRAM is reached. The memory controller will resume processing of the commands from the interruption point. This parameter will be updated with an assertion of the srefresh_enter pin, regardless of the behavior on the register interface. To disable self-refresh again after a srefresh_enter pin assertion, the user will need to clear the parameter to 'b0. • 'b0 = Disable self-refresh mode. • 'b1 = Initiate self-refresh of the DRAM devices.

Table 28: MEM_CTL_06– OFFSET: 0x18

- MEM_CTL_07 – OFFSET: 0x1C

Bit	Name	Default	Description
25:24 (R/W)	CS_MAP	2'h0	Sets the mask that determines which chip select pins are active, with each bit representing a different chip select. The user address chip select field will be mapped into the active chip selects indicated by this parameter in ascending order from lowest to highest. This allows the memory controller to map the entire contiguous user address into any group of chip selects. Bit [0] of this parameter corresponds to chip select [0], bit [1] corresponds to chip select [1], etc. The number of chip selects, the number of bits set to 1 in this parameter, must be a power of 2 (20, 21, 22, etc.) Set this to 2'h1 and do not change it.
17:16 (R/W)	AXI0_W_PRIORITY	2'h0	Sets the priority of write commands from the AXI port. A value of 0 is the highest priority.
9:8 (R/W)	AXI0_R_PRIORITY	2'h0	Sets the priority of read commands from the AXI port. A value of 0 is the highest priority.
0 (R/W)	WRITE_MODEREG	1'h0	Supplies the EMRS data for each chip select to allow individual chips to set masked refreshing. When this parameter is written with a 'b1, the mode parameter(s) [EMRS register] within the DRAM devices will be written. Each subsequent write_modereg setting will write the EMRS register of the next chip select. This parameter will always read back as 'b0. The mode registers are automatically written at initialization of the memory controller. There is no need to initiate a mode register write after setting the start parameter in the memory controller unless some value in these registers needs to be changed after initialization. Note: This parameter may not be changed when the memory is in power-down mode (when the CKE input is de-asserted).

Table 29: MEM_CTL_07 – OFFSET: 0x1C

- MEM_CTL_08 – OFFSET: 0x20

Bit	Name	Default	Description
26:24 (R/W)	ADDR_PINS	3'h0	Defines the difference between the maximum number of address pins configured (16) and the actual number of pins being used. The user address is automatically shifted so that the user address space is mapped contiguously into the memory map based on the value of this parameter.
9:8 (R)	MAX_CS_REG	2'h0	Displays the maximum number of chip selects configured for this memory controller. This parameter is read-only.
1:0 (R/W)	LOWPOWER_REFRESH_ENABLE	2'h0	Sets whether refreshes will occur while the memory controller is in any of the low power modes. • 'b0 = Refreshes still occur • 'b1 = Refreshes do not occur

Table 30: MEM_CTL_08 – OFFSET: 0x20

- MEM_CTL_09 – OFFSET: 0x24

Bit	Name	Default	Description
26:24 (R/W)	COLUMN_SIZE	3'h0	Shows the difference between the maximum column width available (13) and the actual number of column pins being used. The user address is automatically shifted so that the user address space is mapped contiguously into the memory map based on the value of this parameter.
18:16 (R/W)	CASLAT	3'h0	Sets the CAS (Column Address Strobe) latency encoding that the memory uses. The binary value programmed into this parameter is dependent on the memory device, since the same caslat value may have different meanings to different memories. This will be programmed into the DRAM devices at initialization. The CAS encoding will be specified in the DRAM spec sheet, and should correspond to the caslat_lin parameter.
2:0 (R/W)	AGE_COUNT	3'h0	Holds the initial value of the master aging-rate counter. When using the placement logic to fill the command queue, the command aging counters will be decremented one each time the master aging-rate counter counts down age_count cycles.

Table 31: MEM_CTL_09 – OFFSET: 0x24

- MEM_CTL_10 – OFFSET: 0x28

Bit	Name	Default	Description
26:24 (R/W)	TCKE	3'h0	Defines the minimum CKE pulse width, in cycles.
18:16 (R/W)	Q_FULLNESS	3'h0	Defines quantity of data that will be considered full for the command queue.
10:8 (R)	OUT_OF_RANGE_TYPE	3'h0	Holds the type of command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.
2:0 (R/W)	COMMAND_AGE_COUNT	3'h0	Holds the initial value of the command aging counters associated with each command in the command queue. When using the placement logic to fill the command queue, the command aging counters decrease 1 each time the master aging-rate counter counts down age_count cycles.

Table 32: MEM_CTL_10 – OFFSET: 0x28

- MEM_CTL_11 – OFFSET: 0x2C

Bit	Name	Default	Description
27:24 (R/W)	APREBIT	4'h0	Define the location of the auto pre-charge bit in the DRAM address in decimal encoding
18:16 (R/W)	TWTR	3'h0	Define DRAM TWTR parameter in cycles
10:8 (R/W)	TWR_INT	3'h0	Define DRAM TWR parameter in cycles
2:0 (R/W)	TRRD	3'h0	Define DRAM TRRD parameter in cycles

Table 33: MEM_CTL_11 – OFFSET: 0x2C

- MEM_CTL_12 – OFFSET: 0x30

Bit	Name	Default	Description
27:24 (W)	INT_ACK	4'h0	Controls the clearing of the int_status parameter. If any of the int_ack bits are set to 1, the corresponding bit in the int_status parameter will be set to 0. Any int_ack bits written with a 0 will not alter the corresponding bit in the int_status parameter. This parameter will always read back as 0.
19:16	INITAREF	4'h0	Defines the number of auto-refresh commands needed by the DRAM devices to satisfy the initialization sequence.

Bit	Name	Default	Description
(R/W)			
11:8 (R/W)	CASLAT_LIN_GATE	4'h0	Adjusts the data capture gate open time by 1/2 cycle increments. This parameter is programmed differently than caslat_lin when there are fixed offsets in the flight path between the memories and the memory controller for clock gating. When caslat_lin_gate is a larger value than caslat_lin, the data capture window will become shorter. A caslat_lin_gate value smaller than caslat_lin may have no effect on the data capture window, depending on the fixed offsets in the ASIC and the board. • 'b0000 - 'b0001 = Reserved • 'b0010 = 1 cycle • 'b0011 = 1.5 cycles • 'b0100 = 2 cycles • 'b0101 = 2.5 cycles • 'b0110 = 3 cycles • 'b0111 = 3.5 cycles • 'b1000 = 4 cycles • 'b1001 = 4.5 cycles • 'b1010 = 5 cycles • 'b1011 = 5.5 cycles • 'b1100 = 6 cycles • 'b1101 = 6.5 cycles • 'b1110 = 7 cycles • 'b1111 = 7.5 cycles If third party PHY is used, ignore this.
3:0 (R/W)	CASLAT_LIN	4'h0	Sets the CAS latency linear value in 1/2 cycle increments. This sets an internal adjustment for the delay from when the read command is sent from the memory controller to when data will be received back. The window of time in which the data is captured is a fixed length. The caslat_lin parameter adjusts the start of this data capture window. Not all linear values will be supported for the memory devices being used. Refer to the specification for the memory devices being used. • 'b0000 - 'b0001 = Reserved • 'b0010 = 1 cycle • 'b0011 = 1.5 cycles • 'b0100 = 2 cycles • 'b0101 = 2.5 cycles • 'b0110 = 3 cycles • 'b0111 = 3.5 cycles • 'b1000 = 4 cycles • 'b1001 = 4.5 cycles • 'b1010 = 5 cycles

Bit	Name	Default	Description
			• 'b1011 = 5.5 cycles • 'b1100 = 6 cycles • 'b1101 = 6.5 cycles • 'b1110 = 7 cycles • 'b1111 = 7.5 cycles

Table 34: MEM_CTL_12 – OFFSET: 0x30

- MEM_CTL_13 – OFFSET: 0x34

Bit	Name	Default	Description
19:16 (R/W)	TRP	4'h0	Define DRAM TRP parameter in cycles
11:8 (R/W)	TDAL	4'h0	Define DRAM TDAL parameter in cycles
3:0 (R)	MAX_COL_REG	4'h0	Defines the maximum width of column address in the DRAM devices. This value can be used to set the column_size parameter. This parameter is read-only. column_size = max_col_reg - <number of column bits in memory device>.
3:0 (R)	Reserved	4'h0	

Table 35: MEM_CTL_13 – OFFSET: 0x34

- MEM_CTL_14 – OFFSET: 0x38

Bit	Name	Default	Description
28:24 (R)	MAX_ROW_REG	5'h0	Defines the maximum width of the memory address bus (number of row bits) for the memory controller. This value can be used to set the addr_pins parameter. This parameter is read-only. addr_pins = max_row_reg - <number of row bits in memory device>.
20:16 (R/W)	LOWPOWER_CONTROL	5'h0	Enables the individual low power modes of the device. • Bit [4] = Controls memory power-down mode (Mode 1). • Bit [3] = Controls memory power-down with memory clock gating mode (Mode 2). • Bit [2] = Controls memory self-refresh mode (Mode 3). • Bit [1] = Controls memory self-refresh with memory clock gating mode (Mode 4).

Bit	Name	Default	Description
			• Bit [0] = Controls memory self-refresh with memory and controller clock gating mode (Mode 5). For all bits: • 'b0 = Disabled. • 'b1 = Enabled.
12:8 (R/W)	LOWPOWER_AUTO_ ENABLE	5'h0	Enables automatic entry into the low power modes of the memory controller. • Bit [4] = Controls memory power-down mode (Mode 1). • Bit [3] = Controls memory power-down with memory clock gating mode (Mode 2). • Bit [2] = Controls memory self-refresh mode (Mode 3). • Bit [1] = Controls memory self-refresh with memory clock gating mode (Mode 4). • Bit [0] = Controls memory self-refresh with memory and controller clock gating mode (Mode 5). For all bits: • 'b0 = Automatic entry into this mode is disabled. The user may enter this mode manually by setting the associated lowpower_control bit. • 'b1 = Automatic entry into this mode is enabled. The mode will be entered automatically when the proper counters expire, and only if the associated lowpower_control bit is set.

Table 36: MEM_CTL_14 – OFFSET: 0x38

- MEM_CTL_15 – OFFSET: 0x3C

Bit	Name	Default	Description
12:8 (R/W)	TRC	5'h0	Define DRAM TRC parameter in cycles
4:0 (R/W)	TMRD	5'h0	Define DRAM TMRD parameter in cycles

Table 37: MEM_CTL_15 – OFFSET: 0x3C

- MEM_CTL_16 – OFFSET: 0x40

Bit	Name	Default	Description
31:0	-	-	Obsolete

Table 38: MEM_CTL_16 – OFFSET: 0x40

- MEM_CTL_17 – OFFSET: 0x44

Bit	Name	Default	Description
29:24 (R)	DLL_LOCK	6'h0	Defines the actual number of delay elements used to capture one full clock cycle. This parameter is automatically updated every time a refresh operation is performed. This parameter is read-only. If third party PHY is used, this parameter can be ignored.

Table 39: MEM_CTL_17 – OFFSET: 0x44

- MEM_CTL_18 – OFFSET: 0x48

Bit	Name	Default	Description
30:24 (R/W)	TRFC	7'h0	Define DRAM TRFC parameter in cycles

Table 40: MEM_CTL_18 – OFFSET: 0x48

- MEM_CTL_19 – OFFSET: 0x4C

Bit	Name	Default	Description
31:24 (R/W)	TRCD_INT	8'h0	Define DRAM TRCD parameter in cycles
23:16 (R/W)	TRAS_MIN	8'h0	Define DRAM TRAS_MIN parameter in cycles

Table 41: MEM_CTL_19 – OFFSET: 0x4C

- MEM_CTL_20 – OFFSET: 0x50

Bit	Name	Default	Description
29:16 (R/W)	TREF	14'h0	Define DRAM TREF parameter in cycles
6:0 (R)	OUT_OF_RANGE_LENGTH	7'h0	Holds the length of the command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.

Table 42: MEM_CTL_20 – OFFSET: 0x50

- MEM_CTL_21– OFFSET: 0x54

Bit	Name	Default	Description
31:16 (R/W)	AXI0_EN_SIZE_LT_WIDTH_INSTR	16'h0	Allows the port to accept size less than width transactions on the AXI port from requestor Z. Each bit Z corresponds to requestor Z, meaning that if bit [0] is set, then requestor 0 for the AXI port will be able to send size less than width transactions. • 'b0 = Requestor Z may only issue size equal to width instructions. • 'b1 = Requestor Z can issue size equal to width or size less than width instructions.

Table 43: MEM_CTL_21– OFFSET: 0x54

- MEM_CTL_22 – OFFSET: 0x58

Bit	Name	Default	Description
31:0	-	-	Obsolete

Table 44: MEM_CTL_22 – OFFSET: 0x58

- MEM_CTL_23 – OFFSET: 0x5C

Bit	Name	Default	Description
31:16 (R/W)	LOWPOWER_EXTERNAL_CNT	16'h0	Counts the number of idle cycles before memory self-refresh with memory clock gating low power mode.

Table 45: MEM_CTL_23 – OFFSET: 0x5C

- MEM_CTL_24 – OFFSET: 0x60

Bit	Name	Default	Description
31:16 (R/W)	LOWPOWER_POWER_DOWN_CNT	16'h0	Counts the number of idle cycles before memory power-down or power-down with memory clock gating low power mode.
15:0 (R/W)	LOWPOWER_INTERNAL_CNT	16'h0	Counts the number of idle cycles before memory self-refresh with memory and controller clock gating low power mode.

Table 46: MEM_CTL_24 – OFFSET: 0x60

- MEM_CTL_25 – OFFSET: 0x64

Bit	Name	Default	Description
31:16 (R/W)	LOWPOWER_SELF_REFRESH_CNT	16'h0	Counts the number of cycles to the next memory self-refresh low power mode.
15:0 (R/W)	LOWPOWER_REFRESH_HOLD	16'h0	Sets the number of cycles that the Memory Controller will wait before attempting to re-lock the DLL when using the controller clock gating mode low power mode. **NOTE –** This counter is used only in the deepest low power mode.

Table 47: MEM_CTL_25 – OFFSET: 0x64

- MEM_CTL_26 – OFFSET: 0x68

Bit	Name	Default	Description
15:0 (R/W)	TDLL	16'h0	Defines the DRAM DLL lock time in cycles.

Table 48: MEM_CTL_26 – OFFSET: 0x68

- MEM_CTL_27 – OFFSET: 0x6C

Bit	Name	Default	Description
31:16 (R/W)	TRAS_MAX	16'h0	Defines the DRAM TRAS_MAX parameter in cycles.
15:0 (R/W)	TPDEX	16'h0	Defines the DRAM TPDEX parameter in cycles.

Table 49: MEM_CTL_27 – OFFSET: 0x6C

- MEM_CTL_28 – OFFSET: 0x70

Bit	Name	Default	Description
31:16 (R/W)	TXSR	16'h0	Defines the DRAM TXSR parameter in cycles.
15:0 (R/W)	TXSNR	16'h0	Defines the DRAM TXSNR parameter in cycles.

Table 50: MEM_CTL_28 – OFFSET: 0x70

- MEM_CTL_29 – OFFSET: 0x74

Bit	Name	Default	Description
31:0 (R)	VERSION	32'h2040	Controller version number. Read-Only.

Table 51: MEM_CTL_29 – OFFSET: 0x74

- MEM_CTL_30 – OFFSET: 0x78

Bit	Name	Default	Description
31:0 (R)	OUT_OF_RANGE_ADDR[31:0]	32'h0	Holds the address of the command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.

Table 52: MEM_CTL_30 – OFFSET: 0x78

- MEM_CTL_31 – OFFSET: 0x7C

Bit	Name	Default	Description
24 (R/W)	SWAP_EN	1'b0	Enables swapping of the active command for a new higher-priority command when using the placement logic. • 'b0 = Disabled • 'b1 = Enabled
16 (R/W)	ENABLE_QUICK_SREFRESH	1'b0	When this bit is set, the memory initialization sequence will be interrupted and self-refresh mode will be entered. This is used to place the memory devices into self-refresh mode when a power loss is detected during the initialization process. • 'b0 = Continue memory initialization. • 'b1 = Interrupt memory initialization and enter self-refresh mode.
8 (R/W)	ACTIVE_AGING	1'b0	Enables aging of commands in the command queue when using the placement logic to fill the command queue. The total number of cycles required to decrement the priority value on a command by one is the product of the values in the age_count and command_age_count parameters. • 'b0 = Disabled • 'b1 = Enabled
2:0 (R)	OUT_OF_RANGE_ADDR[34:32]	2'h0	Holds the address of the command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.

Table 53: MEM_CTL_31 – OFFSET: 0x7C

- MEM_CTL_32 – OFFSET: 0x80

Bit	Name	Default	Description
26:24 (R)	PORT_DATA_ERROR_TYPE	3'h0	Defines the type of error and the access type that caused the port data error condition. If multiple bits are asserted to 'b1, then multiple errors were found. This parameter is read-only. Bit [2] = Reserved. • Bit [1] = Write data interleaved beyond supported interleaving depth. • Bit [0] = Data Overflow. The write data quantity exceeded the Maximum_Byte_Request configured option.
18:16 (R/W)	CKE_DELAY	3'h0	Sets the number of additional cycles of delay to include in the CKE signal cke_status for status reporting. The default delay is 0 cycles.
8	TREF_ENABLE	1'b0	Enables refresh commands. If command refresh mode is configured, then refresh commands will be automatically issued based on the internal tref counter and any refresh commands sent through the command interface or the register interface. • 'b0 = Refresh commands disabled. • 'b1 = Refresh commands enabled.
0	SWAP_PORT_RW_SAME_EN	1'b0	When swapping in enabled (swap_en is set to 'b1), this parameter enables swapping of similar command types (read with a read, write with a write) between commands on the same port. This effectively enables interleaving. • 'b0 = Disabled • 'b1 = Enabled

Table 54: MEM_CTL_32 – OFFSET: 0x80

- MEM_CTL_33 – OFFSET: 0x84

Bit	Name	Default	Description
28:24 (R)	PORT_DATA_ERROR_ID	5'h0	Holds the source ID of the command that caused a port data error condition. For AXI ports, the source ID is comprised of the Port ID and the Requestor ID, where the Requestor ID is the axi0_BID for write response errors, the axi0_RID for read data errors, or the axi0_WID for write data errors. This parameter is read-only.
20:16 (R)	PORT_CMD_ERROR_ID	5'h0	Holds the source ID of the command that caused a port command error condition. For AXI ports, the source ID is comprised of the Port ID and the Requestor ID, where the Requestor ID is the

Bit	Name	Default	Description
			axi0_AWID for write commands or the axi0_ARID for read commands. This parameter is read-only.
12:8 (R)	OUT_OF_RANGE_SOURCE_ID	5'h0	Holds the Source ID of the command that caused an out-of-range interrupt request to the memory devices. This parameter is read-only.
3:0 (R)	PORT_CMD_ERROR_TYPE	4'h0	Defines the type of error and the access type that caused the port command error condition. If multiple bits are asserted to 'b1, then multiple errors were found. This parameter is read-only. • Bit [3] = Narrow transfer requested for a requestor Y whose axiY_en_size_lt_width_instr parameter is clear. • Bit [2] = The byte count of the wrap command is not a log-2 value. • Bit [1] = Either the starting or ending address of the wrap command is not aligned to a beat size. • Bit [0] = The size of the command (axi0_AWSIZE/ axi0_ARSIZE) is greater than port data width.

Table 55: MEM_CTL_33 – OFFSET: 0x84

- MEM_CTL_34 – OFFSET: 0x88

Bit	Name	Default	Description
31:16 (R/W)	EMRS2_DATA_0	16'h0	Holds the EMRS2 data written during memory initialization for chip select 0. The contents of this parameter will be programmed into the DRAM at initialization or when the write_modereg parameter is written with a "1". Consult the DRAM specification for the correct settings for this parameter.

Table 56: MEM_CTL_34 – OFFSET: 0x88

- MEM_CTL_35 – OFFSET: 0x8c

Bit	Name	Default	Description
15:0 (R/W)	EMRS2_DATA_1	16'h0	Holds the EMRS2 data written during memory initialization for chip select 1. The contents of this parameter will be programmed into the DRAM at initialization or when the write_modereg parameter is written with a "1". Consult the DRAM specification for the correct settings for this parameter.

Table 57: MEM_CTL_35 – OFFSET: 0x8c

- MEM_CTL_36 – OFFSET: 0x90

Bit	Name	Default	Description
23:0 (R/W)	TINIT	24'h0	Defines the DRAM initialization time, in cycles.

Table 58: MEM_CTL_36 – OFFSET: 0x90

- MEM_CTL_37 – OFFSET: 0x94

Bit	Name	Default	Description
31:0 (R)	PORT_CMD_ERROR_ADDR[31:0]	32'h0	Holds the address of the command that caused a port command error condition. This parameter is read-only.

Table 59: MEM_CTL_37 – OFFSET: 0x94

- MEM_CTL_38 – OFFSET: 0x98

Bit	Name	Default	Description
24 (R/W)	ODT_ADD_TURN_CLK_EN	1'h0	Adds a turn-around clock between back-to-back reads or back-to-back writes to different chip selects. The additional clock may be needed at higher clock frequencies. The "turn off" and "turn on" time of termination resistors are not scalable. At higher clock frequencies, it is possible that these times may overlap, resulting in two active resistors while the DQS line is still active. This could compromise the signal integrity of the DQS signal. The additional clock prevents this overlap. • 'b0 = No additional clocking required. • 'b1 = Additional clock added for back-to-back reads or back-to-back writes that occur to different banks.
16 (R/W)	EIGHT_BANK_MODE	1'b0	Indicates that the memory devices have eight banks. • 'b0 = Memory devices have 4 banks. • 'b1 = Memory devices have 8 banks.
8 (R/W)	DQS_N_EN	1'b0	Enables differential data strobe signals from the DRAM. • 'b0 = Single-ended DQS signal from the DRAM. • 'b1 = Differential DQS signal from the DRAM.
2:0 (R)	PORT_CMD_ERROR_ADDR[34:32]	3'h0	Holds the address of the command that caused a port command error condition. This parameter is read-only.

Table 60: MEM_CTL_38 – OFFSET: 0x98

- MEM_CTL_39 – OFFSET: 0x9c

Bit	Name	Default	Description
25:24 (R/W)	ODT_RD_MAP_CS0	2'h0	Sets up which (if any) chip(s) will have their ODT termination active while a read occurs on chip select 0. • Bit [1] = CS1 will have active ODT termination when chip select 0 is performing a read. • Bit [0] = CS0 will have active ODT termination when chip select 0 is performing a read.
17:16 (R/W)	DRAM_CLK_DISABLE	2'h0	Sets value for the DFI output signal dfi_dram_clk_disable. Bit [0] controls CS0, Bit [1] controls CS1. For each bit: • 'b0 = Memory clock/s should be active. • 'b1 = Memory clock/s should be disabled.
8 (R/W)	REDUC	1'b0	Controls the width of the memory datapath. When enabled, the upper half of the memory buses (DQ, DQS and DM) are unused and relevant data only exists in the lower half of the buses. This parameter expands the Memory Controller for use with memory devices of the configured width or half of the configured width. • 'b0 = Standard operation using full memory bus. • 'b1 = Memory datapath width is half of the maximum size.
0 (R/W)	ODT_ALT_EN	1'b0	Enables the use of the non-DFI compliant alternative ODT internal signal odt_alt, which is externally viewed as the signal reserved0. This signal is only required if the user intends to use a CAS latency of 3 with ODT support. • 'b0 = ODT support with CAS latency 3 is not supported. • 'b1 = ODT support with CAS latency 3 is supported but is not DFI compliant. This disables the interrupt bit for ODT-with-CAS3 and disables the OVL error.

Table 61: MEM_CTL_39 – OFFSET: 0x9c

- MEM_CTL_40 – OFFSET: 0xa0

Bit	Name	Default	Description
25:24 (R/W)	RTT_0	2'h0	Defines the On-Die termination resistance for all DRAM devices. The Memory Controller can not be set for different termination values for each chip select. • 'b00 = Termination disabled • 'b01 = 75 Ohm • 'b10 = 150 Ohm • 'b11 = Reserved
17:16 (R/W)	ODT_WR_MAP_CS1	2'h0	Sets up which (if any) chip(s) will have their ODT termination active while a write occurs on chip select 1. • Bit [1] = CS1 will have active ODT termination when chip select 1 is performing a write. • Bit [0] = CS0 will have active ODT termination when chip select 1 is performing a write.
9:8 (R/W)	ODT_WR_MAP_CS0	2'h0	Sets up which (if any) chip(s) will have their ODT termination active while a write occurs on chip select 0. • Bit [1] = CS1 will have active ODT termination when chip select 0 is performing a write. • Bit [0] = CS0 will have active ODT termination when chip select 0 is performing a write.
1:0 (R/W)	ODT_RD_MAP_CS1	2'h0	Sets up which (if any) chip(s) will have their ODT termination active while a read occurs on chip select 1. • Bit [1] = CS1 will have active ODT termination when chip select 1 is performing a read. • Bit [0] = CS0 will have active ODT termination when chip select 1 is performing a read.

Table 62: MEM_CTL_40 – OFFSET: 0xa0

- MEM_CTL_41 – OFFSET: 0xa4

Bit	Name	Default	Description
26:24 (R/W)	W2R_SAMECS_DLY	3'h0	Defines the number of additional clocks of delay to insert between a write and a read to the different chip selects.
18:16 (R/W)	W2R_DIFFCS_DLY	3'h0	Defines the number of additional clocks of delay to insert between a write and a read to the same chip select.
10:8 (R/W)	TRTP	3'h0	Defines the DRAM tRTP (read to pre-charge time) parameter, in cycles.

Bit	Name	Default	Description
2:0 (R/W)	TDFI_DRAM_CLK_DISABLE	3'h0	Holds the DFI tdram_clk_disable timing parameter. This parameter should be programmed with the number of cycles that the PHY requires to disable the clock after the dfi_dram_clk_disable signal is asserted.

Table 63: MEM_CTL_41 – OFFSET: 0xa4

- MEM_CTL_42 – OFFSET: 0xa8

Bit	Name	Default	Description
27:24 (R/W)	TDFI_CTRL_DELAY	4'h0	Holds the DFI tctrl_delay timing parameter. This parameter should be programmed with the number of cycles that the PHY requires to send a power-down or self-refresh command to the DRAM devices.
19:16 (R)	TDFI_CTRLUPD_MIN	4'h0	Holds the DFI tctrlupd_min timing parameter. This parameter is read-only.
11:8 (R/W)	RDLAT_ADJ	4'h0	Adjusts the relative timing between DFI read commands and the dfi_rddata_en signal to conform to PHY timing requirements. When this parameter is programmed to 0x0, dfi_rddata_en will assert one cycle after the dfi_address.
3:0 (R/W)	DRAM_CLASS	4'h0	Selects the mode of operation. • 'b0000 = Reserved • 'b0001 = Mobile DDR • 'b0100 = DDR2 • All other settings reserved.

Table 64: MEM_CTL_42 – OFFSET: 0xa8

- MEM_CTL_43 – OFFSET: 0xac

Bit	Name	Default	Description
27:24 (R/W)	TDFI_PHY_WRLAT_BASE	4'h0	Used to adjust the tdfi_phy_wrlat parameter if the PHY requires greater delay from write command to write data.
19:16 (R)	TDFI_PHY_WRLAT	4'h0	Holds the calculated value of the tphy_wrlat timing parameter. tdfi_phy_wrlat = tdfi_phy_wrlat_base + (wrlat - 1) + reg_dimm_enable. This parameter is read-only.
11:8 (R/W)	TDFI_PHY_RDLAT	4'h0	Holds the tphy_rdlat timing parameter.

Bit	Name	Default	Description
3:0 (R/W)	TDFI_DRAM_CLK_ENABLE	4'h0	Holds the DFI tdram_clk_enable timing parameter. This parameter is currently unused.

Table 65: MEM_CTL_43 – OFFSET: 0xac

- MEM_CTL_44 – OFFSET: 0xb0

Bit	Name	Default	Description
27:24 (R/W)	WRLAT_ADJ	4'h0	Adjusts the relative timing between DFI write commands and the dfi_wrdata_en signal to conform to PHY timing requirements. When this parameter is programmed to 0x0, dfi_wrdata_en will assert on the same cycle as the dfi_address.
19:16 (R/W)	WRLAT	4'h0	Defines the write latency from when the write command is issued to the time the write data is presented to the DRAM devices, in cycles. **NOTE –** This parameter must be set to 0x1 when used in mobile DDR mode.
11:8 (R/W)	TDFI_RDDATA_EN_BASE	4'h0	Used to adjust the tdfi_rddata_en parameter if the PHY requires greater delay from read command to read data enable.
3:0 (R)	TDFI_RDDATA_EN	4'h0	Holds the calculated value of the trddata_en timing parameter. tdfi_rddata_en = tdfi_rddata_en_base + (caslat_lin [3:1] - 1) + reg_dimm_enable + caslat_lin [0]. This parameter is read-only.

Table 66: MEM_CTL_44 – OFFSET: 0xb0

- MEM_CTL_45 – OFFSET: 0xb4

Bit	Name	Default	Description
31:24 (R/W)	DLL_RST_ADJ_DLY	8'h0	Specifies the minimum number of cycles after the master delay value is programmed before the DLL reset may be asserted.
20:16 (R/W)	TFAW	5'h0	Defines the DRAM tFAW parameter, in cycles.
12:8 (R/W)	OCD_ADJUST_PUP_CS_0	5'h0	Sets the off-chip driver (OCD) pull-up adjustment settings for the DRAM devices. The memory controller will issue OCD adjust commands to the DRAM devices during power up. • Bit [4] = Increment (1) or decrement (0) OCD settings.

Bit	Name	Default	Description
			• Bits [3:0] = Number of OCD adjust commands to issue.
4:0 (R/W)	OCD_ADJUST_PDN_CS_0	5'h0	Sets the off-chip driver (OCD) pull-down adjustment settings for the DRAM devices. The memory controller will issue OCD adjust commands to the DRAM devices during power up. • Bit [4] = Increment (1) or decrement (0) OCD settings. • Bits [3:0] = Number of OCD adjust commands to issue.

Table 67: MEM_CTL_45 – OFFSET: 0xb4

- MEM_CTL_46 – OFFSET: 0xb8

Bit	Name	Default	Description
16:8 (R/W)	INT_MASK	9'h0	Active-high mask bits that control the value of the memory controller_int signal. This mask is inverted and then logically AND'ed with the outputs of the int_status parameter.
7:0 (R/W)	TMOD	8'h0	Defines the number of cycles of wait time between mode commands. For write leveling, this is defined as the number of cycles of wait time after a MRS command to the ODT enable.

Table 68: MEM_CTL_46 – OFFSET: 0xb8

- MEM_CTL_47 – OFFSET: 0xbc

Bit	Name	Default	Description
29:16 (R)	TDFI_CTRLUPD_MAX	14'h0	Holds the DFI tctrlupd_max timing parameter. This parameter is read-only.
8:0 (R)	INT_STATUS	9'h0	Shows the status of all possible interrupts generated by the memory controller. The MSB is the result of a logical OR of all the lower bits. This parameter is read-only. The int_status bits correspond to these interrupts: • Bit [8] = Logical OR of all lower bits. • Bit [7] = DLL unlock condition detected. • Bit [6] = ODT enabled and CAS Latency 3 programmed error detected. This is an unsupported programming option. • Bit [5] = Both DDR2 and Mobile modes have been enabled.

Bit	Name	Default	Description
			• Bit [4] = DRAM initialization complete. • Bit [3] = Error was found with command data channel in a port. • Bit [2] = Error was found with command channel in a port. • Bit [1] = Multiple accesses outside the defined PHYSICAL memory space detected. • Bit [0] = A single access outside the defined PHYSICAL memory space detected.

Table 69: MEM_CTL_47 – OFFSET: 0xbc

- MEM_CTL_48 – OFFSET: 0xc0

Bit	Name	Default	Description
29:16 (R)	TDFI_PHYUPD_TYPE0	14'h0	Holds the DFI tphyupd_type0 timing parameter. This parameter is read-only.
13:0 (R)	TDFI_PHYUPD_RESP	14'h0	Holds the DFI tphyupd_resp timing parameter. This parameter is read-only.

Table 70: MEM_CTL_48 – OFFSET: 0xc0

- MEM_CTL_49 – OFFSET: 0xc4

Bit	Name	Default	Description
31:16 (R/W)	EMRS1_DATA_0	16'h0	Holds the EMRS1 data written during memory initialization for chip select 0. The contents of this parameter will be programmed into the DRAM at initialization or when the write_modereg parameter is written with a 1. Consult the DRAM specification for the correct settings for this parameter.
15:0 (R/W)	DLL_RST_DELAY	16'h0	Sets the number of cycles that the reset must be held asserted for the DLL.

Table 71: MEM_CTL_49 – OFFSET: 0xc4

- MEM_CTL_50 – OFFSET: 0xc8

Bit	Name	Default	Description
31:16 (R/W)	EMRS3_DATA_0	16'h0	Holds the EMRS3 data written during memory initialization for chip select 0. The contents of this parameter will be programmed into the DRAM at

Bit	Name	Default	Description
			initialization or when the write_modereg parameter is written with a 1. Consult the DRAM specification for the correct settings for this parameter.
15:0 (R/W)	EMRS1_DATA_1	16'h0	Holds the EMRS1 data written during memory initialization for chip select 1. The contents of this parameter will be programmed into the DRAM at initialization or when the write_modereg parameter is written with a 1. Consult the DRAM specification for the correct settings for this parameter.

Table 72: MEM_CTL_50 – OFFSET: 0xc8

- MEM_CTL_51 – OFFSET: 0xcc

Bit	Name	Default	Description
31:16 (R/W)	MRS_DATA_0	16'h0	Holds the MRS data written during memory initialization for chip select 0. This mode parameter should contain the necessary MRS fields in the correct bit locations. The Memory Controller ignores the programmed value of the DLL Reset bit in this parameter. An internal state machine controls this bit and only sets the DLL Reset bit during initialization. The contents of this parameter, except the DLL Reset bit, will be programmed into the DRAM at initialization or when the write_modereg parameter is written with a 1. Consult the DRAM specification for the correct settings for this parameter.
15:0 (R/W)	EMRS3_DATA_1	16'h0	Holds the EMRS3 data written during memory initialization for chip select 1. The contents of this parameter will be programmed into the DRAM at initialization or when the write_modereg parameter is written with a 1. Consult the DRAM specification for the correct settings for this parameter.

Table 73: MEM_CTL_51 – OFFSET: 0xcc

- MEM_CTL_52 – OFFSET: 0xd0

Bit	Name	Default	Description
31:16 (R/W)	TCPD	16'h0	Defines the clock enable to pre-charge delay time for the DRAM devices, in cycles.
15:0 (R/W)	MRS_DATA_1	16'h0	Holds the MRS data written during memory initialization for chip select 1. This mode parameter should contain the necessary MRS fields in the correct bit locations. The Memory Controller ignores

Bit	Name	Default	Description
			the programmed value of the DLL Reset bit in this parameter. An internal state machine controls this bit and only sets the DLL Reset bit during initialization. The contents of this parameter, except the DLL Reset bit, will be programmed into the DRAM at initialization or when the write_modereg parameter is written with a 1. Consult the DRAM specification for the correct settings for this parameter.

Table 74: MEM_CTL_52 – OFFSET: 0xd0

- MEMCIF_TIM_CONFIG – OFFSET: 0x200

Bit	Name	Default	Description
28 (R/W)	CMD_SEL	1'b0	DFI control and address signal from controller to PHY will be delayed for CMD_SEL cycles. If CMD_SEL is 0, those signals will be directly transmit to PHY
26:24 (R/W)	RDEN_SEL	3'h0	The dfi_rddata_en signal will be delayed for RDEN_SEL cycles to generate ctrl_rden signal for PHY. If RDEN_SEL is 0, ctrl_rden is the same timing as dfi_rddata_en • 000 : same as ctrl_rden • 001 : 1 cycle delay • 010 : 2 cycles delay • 011 : 3 cycles delay • 100 : 4 cycles delay • 101 : 5 cycles delay • 110 : 6 cycles delay • 111 : 7 cycles delay
17:16 (R/W)	GATE_SEL	2'h0	The dfi_rddata_en signal will be delayed for GATE_SEL cycles to generate ctrl_gate signal for PHY. If GATE_SEL is 0, ctrl_gate is the same timing as dfi_rddata_en • 00 : same as dfi_rddata_en • 01 : 1 cycle delay • 10 : 2 cycles delay • 11 : 3 cycles delay
11:8 (R/W)	RESYNC_SEL	4'h0	The ctrl_resync signal will be asserted for one cycle when dfi_ctrlupd_req remains active for RESYNc_SEL cycles.

Bit	Name	Default	Description
4 (R/W)	BA2_SEL	1'h0	To support eight bank mode, memory address A13 will act as BA2 if this bit is 1.
1:0 (R/W)	WDAT_SEL	2'h0	The dfi_wrdata will be delayed for WDAT_SEL to generate ctrl_data for PHY.

Table 75: MEMCIF_TIM_CONFIG – OFFSET: 0x200

- MEMCIF_DLL_START – OFFSET: 0x204

Bit	Name	Default	Description
0 (R/W)	START	1'b0	This bit controls PHY interface signal ctrl_start When setting this bit to 1, ctrl_start signal is high to make the DLL run and lock. When setting this bit to 0, DLL stops running.

Table 76: MEMCIF_DLL_START – OFFSET: 0x204

- MEMCIF_DLL_STATUS – OFFSET: 0x208

Bit	Name	Default	Description
24 (R)	CLOCK	1'b0	Status of PHY interface signal ctrl_clock. It means DLL coarse lock {CLOCK, FLOCK} • 'b00 = DLL is not locked. • 'b01 = Impossible value. • 'b10 = DLL is locked and phase error is less than 160ps • 'b11 = DLL is locked and phase error is less than 80ps.
16 (R)	FLOCK	1'b0	Status of PHY interface signal ctrl_flock. It means DLL fine lock
9:0 (R)	LOCK_VALUE	10'h0	Status of PHY interface signal ctrl_lock_value. It means locked delay line encoding value • Bit[9:2] : number of delay cells for coarse lock • Bit[1:0] : control value for fine lock

Table 77: MEMCIF_DLL_STATUS – OFFSET: 0x208

- MEMCIF_DLL_CONFIG1 – OFFSET: 0x20c

Bit	Name	Default	Description
24 (R/W)	DLL_ON	1'b0	This bit controls PHY interface signal ctrl_dll_on. This bit should be kept high for normal operation. When set this bit to 0, DLL is turned off and ctrl_clock and ctrl_flock become high. This bit should be kept set before seting START in MEMCIF_DLL_START register.
23:16 (R/W)	FORCE	8'h0	When DLL_ON is low, this field is used for DLL to generate phase shifted clock instead of ctrl_lock_value[9:2]
15:8 (R/W)	INCREMENT	8'h0	Increase amount of start point for DLL to trace lock. Recommended values: • 100MHz: [8'h10, 8'h4f] • 133MHz: [8'h10, 8'h3b] • 166MHz: [8'h10, 8'h2f] • 200MHz: [8'h10, 8'h28]
7:0 (R/W)	START_POINT	8'h0	Initial DLL start point for DLL to trace lock. This is number of delay cells. Recommended values: • 100MHz: [8'h10, 8'h4f] • 133MHz: [8'h10, 8'h3b] • 166MHz: [8'h10, 8'h2f] • 200MHz: [8'h10, 8'h28]

Table 78: MEMCIF_DLL_CONFIG1 – OFFSET: 0x20c

- MEMCIF_DLL_CONFIG2 – OFFSET: 0x210

Bit	Name	Default	Description
18:16 (R/W)	SHIFTC	3'h0	This field controls GATEout signal delay amount. GATEout signal is used to generate clean read dqs signal. • 'b000: T/128 • 'b001: T/64 • 'b010: T/32 • 'b011: T/16 • 'b100: T/8 • 'b101: T/4 • 'b110: T/2 • 'b111: T

Bit	Name	Default	Description
14:8 (R/W)	OFFSETC	7'h0	GATEout signal offset amount. OFFSETC[6] = 1: GATEout delay amount – OFFSETC[5:0]*tFS OFFSETC[6] = 0: GATEout delay amount + OFFSETC[5:0]*tFS if HALF = 0 tFS = tCK/LOCK_VALUE[9:0] if HALF = 1 tFS = tCK*0.5/LOCK_VALUE[9:0]
6:0 (R/W)	OFFSETD	7'h0	Offset amount for 270 degree clock generation OFFSETD[6] = 1: 270 degree clock delay amount – FFSETD[5:0]*tFS OFFSETD[6] = 0: 270 degree clock delay amount + OFFSETD[5:0]*tFS

Table 79: MEMCIF_DLL_CONFIG2 – OFFSET: 0x210

- MEMCIF_DLL_CONFIG3 – OFFSET: 0x214

Bit	Name	Default	Description
30:24 (R/W)	OFFSET3	7'h0	90 degree clock in rd_slice_3 offset amount. OFFSET3[6] = 1: 90 degree delay amount – OFFSET3[5:0]*tFS OFFSET3[6] = 0: 90 degree delay amount + OFFSET3[5:0]*tFS
22:16 (R/W)	OFFSET2	7'h0	90 degree clock in rd_slice_2 offset amount. OFFSET3[6] = 1: 90 degree delay amount – OFFSET3[5:0]*tFS OFFSET3[6] = 0: 90 degree delay amount + OFFSET3[5:0]*tFS
14:8 (R/W)	OFFSET1	7'h0	90 degree clock in rd_slice_1 offset amount. OFFSET3[6] = 1: 90 degree delay amount – OFFSET3[5:0]*tFS OFFSET3[6] = 0: 90 degree delay amount + OFFSET3[5:0]*tFS
6:0 (R/W)	OFFSET0	7'h0	90 degree clock in rd_slice_0 offset amount. OFFSET3[6] = 1:

Bit	Name	Default	Description
			90 degree delay amount – OFFSET3[5:0]*tFS OFFSET3[6] = 0: 90 degree delay amount + OFFSET3[5:0]*tFS

Table 80: MEMCIF_DLL_CONFIG3 – OFFSET: 0x214

- MEMCIF_PHY_CONFIG – OFFSET: 0x218

Bit	Name	Default	Description
24 (R/W)	ADB_EN	1'h0	If this bit is set, PHY stats 32-bit CRC generation according to control and address signals(ctrl_adct[18:0], ctrl_ras, ctrl_cas, ctrl_we and ctrl_cke). If this bit is cleared, it stops CRC generation.
19:16 (R/W)	DRV_BUS	4'h0	Set high to drive DQ/DM/DQS signals. For normal operation this field should be 0.
9:8 (R/W)	FNC_FB	2'h0	This field controls FNC Feedback test mode when PHY interface signals {mode_phy, mode_nand, mode_scan, mode_bypass, mode_mux} is 5'b00000 2'b0x: Normal operation mode. 2'b10: External FNC Feedback test mode. 2'b11: Internal FNC Feedback test mode.
0 (R/W)	HALF	1'h0	If this bit is set, low-speed mode for DLL will turn on, DLL can run at low speed (80MHz ~ 100MHz).

Table 81: MEMCIF_PHY_CONFIG – OFFSET: 0x218

- MEMCIF_ADB_CHECK – OFFSET: 0x21c

Bit	Name	Default	Description
31:0 (R)	ADB_CHECK	32'h0	This field is output of 32-bit CRC for ctrl_adct[18:0], ctrl_ras,ctrl_cas, ctrl_we and ctrl_cke (CRC is generated while ctrl_adb_en is high). This field is used for the chip test purpose.

Table 82: MEMCIF_ADB_CHECK – OFFSET: 0x21c

SYSTEM CONTROL MODULES

Mode Configuration Pins

The mode configuration pins (X_TEST_MODE[5:0]) control the mode of SiRFatlasV, as listed in the table below.

When SiRFatlasV is set to Normal, ARM JTAG or Function ATE mode, X_TEST_MODE [3:0] will control the boot configuration.

X_TEST_MODE[5:0] pins has no internal pull-up/down resistor, so there is no default value when the system is powered on. It is recommended to use 0~10Kohm resistor to pull-up/down these pins.

Working Mode	Sub Mode	Feature 1	Feature 0
X_TEST_MODE[5:4]	X_TEST_MODE[3:2]	X_TEST_MODE[1]	X_TEST_MODE[0]
2'B00: Normal 2'B01: Normal With ARM JTAG 2'B10: Function ATE	2'b00: Embedded ROM NAND boot (SLC)	1'b0: 8-bit	1'b0: 2KB page
			1'b1: 4KB page
		1'b1: LBA-NAND	1'bx: 2KB page
	2'b01: NAND boot (SLC)	1'bx: Don't Care	1'b0: 512B page
			1'b1: 2KB page
	2'b10: Embedded ROM NAND boot (MLC)	1'b0: ECC 12-bit per 1KB	1'b0: 2KB page
			1'b1: 4KB page
		1'b1: ECC 24-bit per 1KB	1'b0: 4KB page
			1'b1: 8KB page
	2'b11: Embedded ROM SD/MMC boot	-	-
2'B11: TEST MODE	2'b00: Scan mode	2'b00: INTEST_AC_DC	
		2'b01: EXTEST_AC_DC	
		2'b10: Reserved	
		2'b11: Reserved	
	2'b01: Reserved	-	-
	2'b10: Chip test 1	1'b0: Boundary Scan	1'b0: NandTree
			1'b1: BSD
		1'b1: Macro Test0	1'b0: BIST
			1'b1: eFuse (x_reset_b = 1'b0)
	2'b11: Chip test 2	1'b0: Macro Test1	1'b0: USB PHY stand alone test

Working Mode	Sub Mode	Feature 1	Feature 0
			1'b1: TSC/PLL test
		1'b1: Macro Test2	1'b0: MEM PHY test
			1'b1: Reserved

Table 83: Mode Configure Table

Clock Controller

Overview

The clock controller module is used to control PLL and generate block clocks, which consists of four sources, XINW (32KHz), XIN (24MHz), PLL1 clock and PLL2 clock. There are four reference clock groups (ck_cpu_rf, ck_sys_rf, ck_mem_rf, and ck_sd_rf) which can select its own clock source, six clock groups (CPU, SYS, DSP, I/O, MEM, SD) which can set up its clock divider. Every block clock is gated through the clock controller module.

Feature List

- Glitch-free source clock switch
- Glitch-free clock divide
- Glitch-free clock gating
- RISC I/O interface
- PLL control

Pin Description

External Pin Descriptions

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_XIN	I	-	-	-	External input clock
X_XINW	I	-	-	-	External 32KHz input clock
X_RESET_B	I	-	-	-	Power-on reset input

Table 84: Clock Controller External Pin Descriptions

Functional Descriptions

Block Diagram

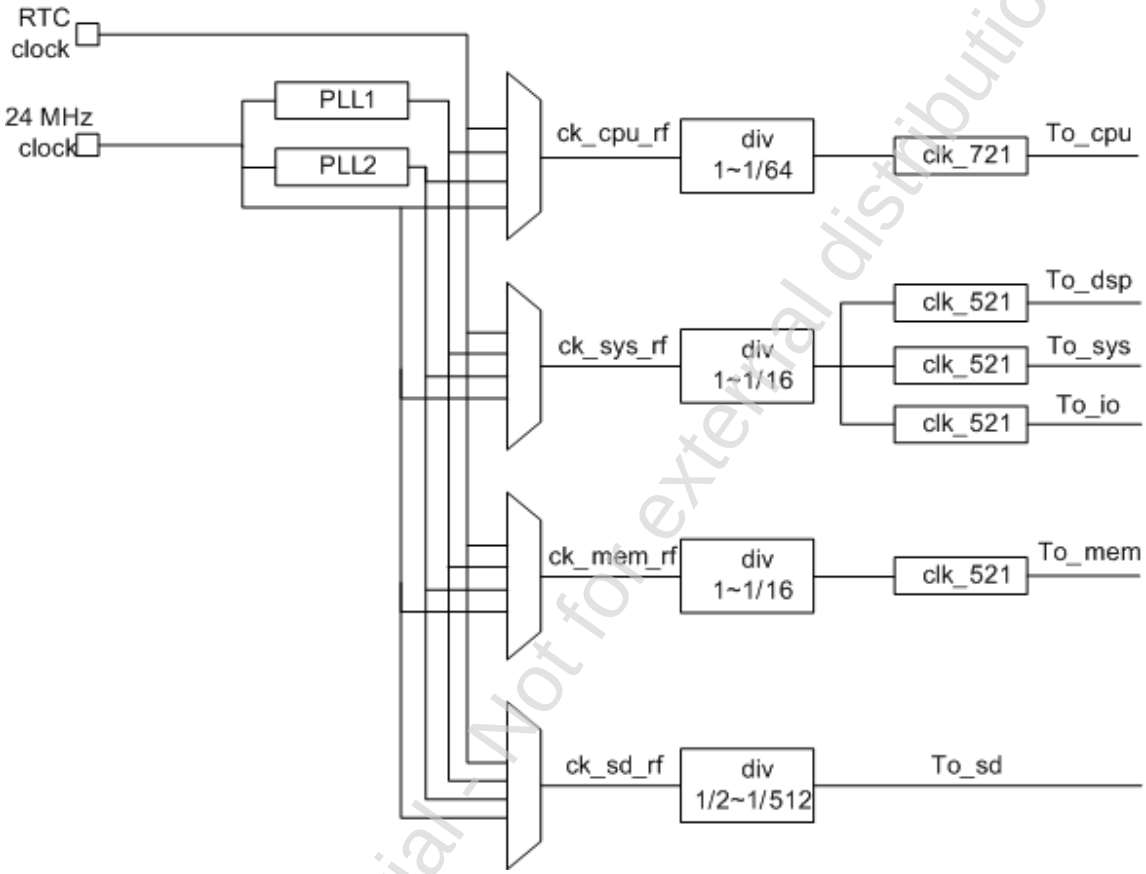


Figure 3: CLKC Block Diagram

CLKC

CLK_MUX_421

There are six 4 to 1 MUXs, one for each of the following clock domains: CPU clock, system clock, memory clock, SDC clock, CKO_0, and CKO_1.

Each MUX contains four input clock sources:

- High-speed oscillator (X_XIN)
- Real-time oscillator (X_XINW)
- PLL #1 (PLL1)
- PLL #2 (PLL2)

There is a 2-bit select signal for each MUX to choose one of the four input sources. These signals are controlled by the CLKC_CLK_SWITCH register. The most important feature is that the switch should be set glitch-free.

PLLTOP

This block is the PLL (Phase Locked Loop) analog part where there are two similar PLL analogs in the module.

The PLL is a PLL frequency synthesizer. The PLL provides frequency multiplication capabilities. Its output clock frequency FOUT is related to the input clock frequency FIN by the following equation:

FOUT = (m × FIN) / (p × 2^s)

Where FOUT is the output clock frequency and FIN is the input clock frequency. M, P and S are the decimal values for programmable dividers. PLL consists of a PFD (Phase Frequency Detector), a Charge Pump, a VCO (Voltage Controlled Oscillator), a 6-bit Pre-divider, a 10-bit Main-divider and a 3-bit Post-scalar

- Feature
  - 1.2V single power supply
  - Output frequency range: 40MHz ~ 1000MHz
  - Duty ratio: 40% to 60% (all tuned range)
  - Power down mode
  - Bypass mode (FOUT=FIN)
  - Programmable dividers

- Electrical characteristics

Characteristics	Symbol	Min	Typ	Max	Unit
Input frequency	FIN	4	-	300	MHz
Output frequency	FOUT	40	-	1000	MHz
VCO output frequency	FVCO	800	-	1600	MHz
Output clock duty ratio	TOD	40	-	60	%
Locking time	TLT	-	-	300	us
PLL period jitter	TJP	-	-	60ps@1G	psp-p

Table 85: PLL Electrical Characteristics

- PLL programming guide
  - Even after PLL power supply becomes stable at t1. RESETB must keep the logical low for enough time (Δt >1us)
  - Whenever any register value is updated, PLL needs a period called “Locking Time” to generate a target frequency. Actually PLL generates an unknown frequency during the “Locking Time”. However, PLL behavioral model is described to generate the low state during

this period to prevent a simulation problem. It is recommended to disable PLL clock path during “Locking Time”.

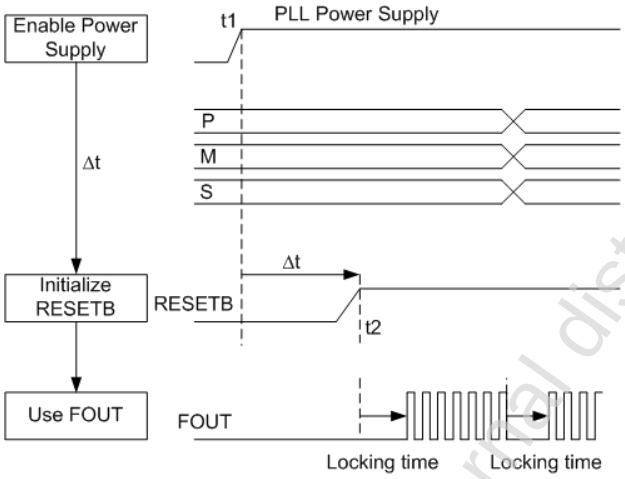


Figure 4: PLL Program Timing

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
Clock controller register base address through RISC I/O	0x90080000

Table 86: CLKC Interface Address Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	CLKC_CLK_EN	Clock Controller Clock Enable
0x0084	-	Reserved
0x0008	CLKC_PLL1_CONFIG	Clock Controller PLL1 Configuration Register
0x000C	CLKC_PLL2_CONFIG	Clock Controller PLL2 Configuration Register
0x0010	CLKC_SYS_RATIO	Clock Controller System Clock Ratio Register
0x0014	CLKC_PRF_RATIO	Clock Controller Peripheral Clock Ratio Register
0x0018	CLKC_CLK_SWITCH	Clock Controller Clock Switch Register

Table 87: Register Mapping

Register Descriptions

- Clock Controller Clocks Enable Register (CLKC _CLK_EN) – 0x000

  Users can enable/disable the clock of majority logic in SiRFatlasV, except the RISC core, memory controller, and some other system control modules.

Bit	Name	Default	Description
0 (R/W)	DSP_EN	1'b0	DSP Core clock enable: 1: Enable 0: Disable
1 (R/W)	ROMEFUSE_EN	1'b1	ROM Controller and EFUSE clock enable
2 (R/W)	DMA_EN	1'b0	DMA Controller clock enable
3 (R/W)	LCD_EN	1'b0	LCD Controller clock enable
4 (R/W)	GPS_EN	1'b0	GPS clock enable
5	-	-	Reserved
6 (R/W)	PCICOPY_EN	1'b0	PCI_COPY clock enable
7	-	-	Reserved
8 (R/W)	CODEC_EN	1'b0	CODEC clock enable
9 (R/W)	UART0_EN	1'b0	UART0 clock enable
10 (R/W)	USP0_EN	1'b0	Serial Port 0 clock enable
11 (R/W)	USP1_EN	1'b0	Serial Port 1 clock enable
12 (R/W)	SPI_EN	1'b0	SPI Port clock enable
13	-	-	Reserved
14 (R/W)	UART1_EN	1'b0	UART1 clock enable
15	-	-	Reserved
16 (R/W)	VPP_EN	1'b0	Video Preprocessor clock enable
17 (R/W)	SDIO01_EN	1'b0	SDIO01 interface clock enable
18 (R/W)	NAND_EN	1'b1	NAND Flash interface clock enable
19 (R/W)	I2C_EN	1'b0	I^2C0 and I^2C1 interface clock enable
20	-	-	Reserved
21 (R/W)	GPIO_EN	1'b0	GPIO clock enable
22 (R/W)	PCI_EN	1'b0	PCI clock enable
23	-	-	Reserved
24	-	-	Reserved
25	-	-	Reserved

Bit	Name	Default	Description
26 (R/W)	USB_EN	1'b0	USB clock enable
27	-	1'b1	Reserved
28	-	-	Reserved
29	-	-	Reserved
30 (R/W)	SDIO23	1'b0	SDIO23 interface clock enable
31 (R/W)	MF_EN	1'b0	Match filter clock enable

Table 88: Clock Controller Clocks Enable Register (CLKC_CLK_EN)

- Clock Controller PLL1 Configuration Register (CLKC_PLL1_CONFIG) – 0x008

  Both PLL1 and PLL2 can be programmed to different frequencies. Users need to make sure that the clock source has been switched to another PLL when programming a PLL. Switch back the clock source only after the PLL is stable.

  Clock frequency is determined by the following equation:

  NOTE – FOUT = (m × FIN) / (p × 2^s)

  For example, if FIN = 12MHz, and m = 250, p = 3, s = 1, then the PLL output will be 500MHz.

  PLL Output frequency range: 40MHz - 1.0GHz

  Reset the PLL before using it, for example:

```
rCLKC_PLL1_CONFIG |= 0x2000000; // clear reset
for(io_i=1;io_i<=PLL_STABLE_TIME;io_i++); // Wait 300us
rCLKC_PLL1_CONFIG = PLL_500M;
rCLKC_PLL2_CONFIG = PLL2_332M;
for(io_i=1;io_i<=PLL_STABLE_TIME;io_i++); // Wait 300us lock time
```

Bit	Name	Default	Description
9:0 (R/W)	m<9:0>	10'h190	Value of 10-bit programmable main-divider.
12:10	-	3'h0	Reserved.
18:13 (R/W)	p<5:0>	6'h3	Value of 6-bit programmable pre-divider.
21:19 (R/W)	s<2:0>	3'h3	Value of 3-bit programmable post-scale.
22 (R/W)	BP	1'b1	1: Bypass the PLL. 0: Not bypass the PLL.

Bit	Name	Default	Description
23 (R/W)	-	1'b0	Reserved.
24 (R/W)	-	1'b0	Reserved.
25 (R/W)	RESETB	1'b0	Reset control: 1: No reset. 0: Reset.
31:26	-	-	Reserved.

Table 89: Clock Controller PLL1 Configuration Register (CLKC_PLL1_CONFIG)

- Clock Controller PLL2 Configuration Register (CLKC_PLL2_CONFIG) – 0x00c
  It is totally same as PLL1 configuration.

Bit	Name	Default	Description
9:0 (R/W)	m<9:0>	10'h190	Value of 10-bit programmable main-divider.
12:10	-	3'h0	Reserved.
18:13 (R/W)	p<5:0>	6'h3	Value of 6-bit programmable pre-divider.
21:19 (R/W)	s<2:0>	3'h3	Value of 3-bit programmable post-scale.
22 (R/W)	BP	1'b1	1: Bypass the PLL. 0: Not bypass the PLL.
23 (R/W)	-	1'b0	Reserved.
24 (R/W)	-	1'b0	Reserved.
25 (R/W)	RESETB	1'b0	Reset control: 1: No reset. 0: Reset.

Table 90: Clock Controller PLL2 Configuration Register (CLKC_PLL2_CONFIG)

- Clock Controller System CLK Ratio Register (CLKC_SYS_RATIO) – 0x010
  This register is used to configure the clock frequency by programming the divide parameter from the clock source.

  The time between pre-write and next-write, the registers should at least 4 times the lowest clock frequency cycle.

For example: Assume that the CPU_CLK 0 division is 500 MHz, 2-ns period, and you are writing the register to switch it into 3 divisions. After writing CPU_RATIO, you should wait at least 512 ns (2 ns X 64 X 4) before you can write to the register again.

Bit	Name	Default	Description
3:0 (R/W)	CPU_RATIO<3:0>	4'h0	The clock ratio PLL_OUT[1]: 0: 1 1: 1/2 2: 1/4 3: 1/8 4: 1/16 5: 1/32 6: 1/64
7:4 (R/W)	DSP_RATIO<3:0>	4'h0	PLL_OUT:DSPCLK ratio 0: 1 1: 1/2 2: 1/4 3: 1/8 4: 1/16
11:8 (R/W)	SYS_RATIO<3:0>	4'h0	PLL_OUT:SYSCLK ratio (same as DSP)
15:12 (R/W)	IO_RATIO<3:0>	4'h0	PLL_OUT:IOCLK ratio (same as DSP)
19:16 (R/W)	MEM_RATIO<3:0>	4'h0	PLL_OUT:MEMCLK ratio (same as DSP)
31:20	-	-	Reserved

Table 91: Clock Controller System CLK Ratio Register (CLKC_SYS_RATIO)

The constraints below must be followed:

- The frequency of DSPCLK can only be 1X 2X or 4X of SYSCLK.
- The frequency of IOCLK can only be 1X, 1/2X or 1/4X of SYSCLK.

NOTE – Any configurations that violate the above constraints will create unexpected results.

[1] PLL_OUT is determined by CLKC_CLK_SWITCH register: it can be 24MHz crystal, 32KHz crystal, PLL1, or PLL2.

- Clock Controller Peripheral CLK Ratio Register (CLKC_PRF_RATIO) – 0x014

  This register is used to configure the clock frequency by programming the divide parameter from the clock source.

Bit	Name	Default	Description
7:0 (R/W)	SD_RATIO<7:0>	8'h0	PLL_OUT: SD CLK ratio (range from 1/2X to 1/512X) Fout = Fin / ((SD_RATIO + 1) x 2)
31:8	-	-	Reserved

Table 92: Clock Controller Peripheral CLK Ratio Register (CLKC_PRF_RATIO)

- Clock Controller Clocks Switch Register (CLKC_CLK_SWITCH) – 0x018

  The time between pre-write and next-write the registers should be at least four times of the 32KHZ clock cycle intervals.

Bit	Name	Default	Description
1:0 (R/W)	CPU_CS	2'b00	CPU clock source select. 00: 24MHz crystal. 01: Select PLL1 as the system clock source. 10: Select PLL2 as the system clock source. 11: 32KHz.
3:2 (R/W)	SYS_CS	2'b00	System Clock Source Select. Definition is the same as the above.
5:4 (R/W)	MEM_CS	2'b00	Memory Clock Source Select. Definition is the same as the above.
7:6 (R/W)	SD_CS	2'b00	SD Clock Source Select. Definition is the same as the above.
31:8	-	24'h0	Reserved.

Table 93: Clock Controller Clocks Switch Register (CLKC_CLK_SWITCH)

Power Controller

Overview

Power controller (PWRC) is used to control the deep sleep/hibernation and wake-up process and memory pad retention function. SiRFatlasV can be woken up by key and RTC. The glitch removing logic is also involved. PWRC can control the power according to sleep/wakeup mode through power enable and dram enable pins. For sleep, SiRFatlasV support deep sleep mode and hibernation mode. In deep sleep mode, the memory will be in self-refresh mode; while in hibernation mode, the memory will be power down. PWRC still support low battery control function.

Feature List

- Deep-sleep, hibernation, and wake-up control
- RISC I/O interface

- Power control

Pin Description

External Pin Descriptions

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_SYSTEM_EN	O	-	-	Output low	All chip powers except VDD_RTC/VDDIO_RTC/VDDIO_MEM control signal
X_DRAM_EN	O	-	-	Output low	Control memory chip and memory I/O power
X_ON_KEY_B	I	-	-	Input	Main sleep/wakeup key
X_EXT_ON	I	-	-	Input	Extension sleep/wakeup control
X_LOW_BATT_B	I	-	-	Input	Battery low power pin
X_RTC_RST_B	I	-	-	Input	RTC domain reset signal

Table 94: PWRC External Pin Description

Functional Descriptions

Function Modes

SiRFatlasV comes with multiple power status/modes as described below.

- RTC cold boot
  Chip power on at first time. In the mode, only VDD_RTC (including its I/O) has power which includes SYSRTC and PWRC function.

- Normal mode
  Chip goes into normal usage. In the mode, all powers are on.

- Deep-sleep mode
  In this mode, VDD_RTC (including its I/O) has power, and memory pads (VDDIO_MEM) also has power, which works in retention mode to keep a fix status. Other powers are off.

- Hibernation mode
  Same as deep-sleep mode except memory pads also power off.

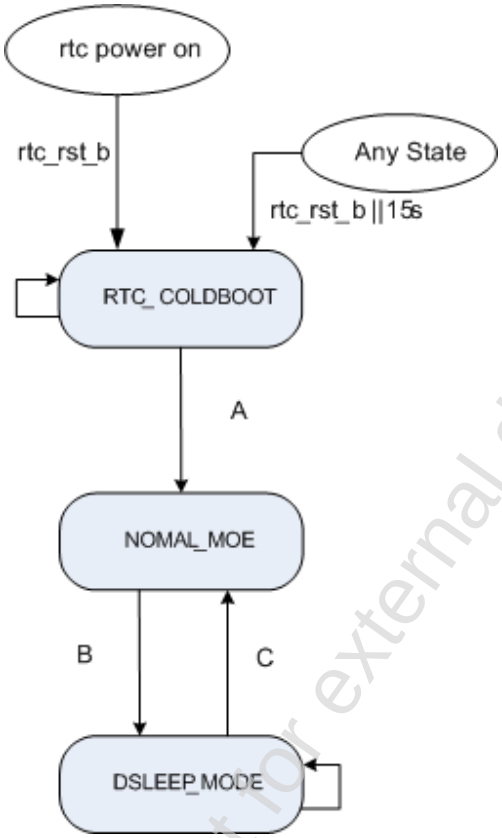


Figure 5: SiRFatlasV Function Mode Switch Diagram

Mode State Description

RTC_COLDBOOT: When x_rtc_rst_b is valid or x_on_key_b 15s event happen, the chip will back to the state. In the state, x_system_en and x_dram_en will be invalid.

NORMAL_MODE: All power on and normal run state. In the state, x_system_en and x_dram_en will be valid.

DSLEEP_MODE: Keep deep-sleep or hibernation mode. In the state, x_system_en will be invalid, and x_dram_en is valid or not according to deep sleep mode or hibernation mode.

Mode Switch Description

1. After the VDDRTC first power on, when x_rtc_rst_b is valid, the chip will go into RTC_COLDBOOT state.
2. In RTC_COLDBOOT, if PWRC detect that x_on_key_b is low level over 320ms or x_ext_on is high level over 320ms, and there is no low battery event happen, then the state will switch to NORMAL_MODE. For 15S reset event, it first needs invalid x_on_key_b to end the 15S event, then re-check 320ms low level. (Point A in the above figure.)
3. In NORMAL_MODE, if PWRC detects that x_on_key_b has a negative edge and keeps for 320ms after de-bounce or x_ext_on has a positive/negative edge and keeps for 320ms after de-bounce or low battery event, the PWRC can generate interrupts to notify software, then software will configure PWRC into deep-sleep or hibernation mode. (Point B in the above figure.)

4. In DSLEEP_MODE, if PWRC detect x_on_key_b has a negative edge and keeps for 320ms after de-bounce or x_ext_on has positive edge and keeps for 320ms after de-bounce or RTC alarm, and there is no low battery event, then the PWRC will switch to NORMAL_MODE state. (Point C in Figure 5.)

In any state, if x_on_key_b detects a 15 second low level, the state will switch to RTC_COLDBOOT.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
PWRC internal register base address through RISC I/O	0x9006c000

Table 95: PWRC Interface Address Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	PWRC_PDN_CTRL	PWRC Power Down Control
0x0004	PWRC_PON_STATUS	PWRC Power On Status
0x0008	PWRC_TRIGGER_EN	PWRC Trigger Enable
0x000C	PWRC_INT_STATUS	PWRC Interrupt Status
0x0010	PWRC_INT_MASK	PWRC Interrupt Mask
0x0014	PWRC_PIN_STATUS	PWRC Pins Status
0x0018	PWRC_SCRATCH_PAD	PWRC Software Use

Table 96: Register Mapping

Register Descriptions

- PWRC Power Down Control Register (PWRC_PDN_CTRL) – 0x0000

Bit	Name	Default	Description
0 (R/W)	START_SLEEP	1'b0	Power down control 1: Begin to go into deep sleep or hibernation mode 0: Reserved Note: When software writes 1, the PWRC will begin deep sleep or hibernation and the bit will be cleared by hardware.
1 (R/W)	PDN_MODE	1'b0	Power-down mode 1: Hibernation mode 0: Deep-sleep mode
2 (R/W)	DRAM_HOLD	1'b0	Memory pad retention control When power mode is deep sleep mode, the hardware will automatically set this bit and let memory pads keep a program status in power down mode. After wakeup, software should clear it before using memory. After clearing it, software should wait for at least 10us to ensure the memory pads stable. 1: Do not set to 1, and let the hardware set it. 0: Clear it
31:3 (R)	-	29'h0	Reserved

Table 97: PWRC Power down Control Register

- PWRC Power On Status Register (PWRC_PON_STATUS) – 0x0004

Bit	Name	Default	Description
0 (R/W)	WARM_BOOT	1'b0	1: Wakeup from deep sleep or hibernation 0: Cold boot Writing 0 will clear it.
1 (R/W)	ON_KEY	1'b0	Trigger to power on by ON_KEY event 1: Happen 0: Not happen Writing 0 will clear it.
2 (R/W)	EXT_ON	1'b0	Trigger to power on by EXT_ON event 1: Happen

Bit	Name	Default	Description
			0: Not happen Writing 0 will clear it.
3 (R/W)	RTC_ALARM0	1'b0	Trigger wake-up by the RTC alarm0 event. 1: Happen 0: Not happen Writing 0 will clear it.
4 (R/W)	RTC_ALARM1	1'b0	Trigger wake-up by the RTC alarm1 event. 1: Happen 0: Not happen Writing 0 will clear it.
31:5	-	27'h0	Reserved

Table 98: PWRC Power on Status Register

- PWRC Trigger Enable Register (PWRC_TRIGGER_EN) – 0x0008

Bit	Name	Default	Description
0 (R)	-	1'b0	Reserved
1 (R/W)	ON_KEY_EN	1'b1	Enable X_ON_KEY_B to trigger to wakeup in deep sleep/hibernation mode or generate interrupt in normal mode 1: Enable 0: Disable
2 (R/W)	EXT_ON_EN	1'b1	Enable X_EXT_ON to trigger to wakeup in deep sleep/ hibernation or generate interrupt in normal mode 1: Enable 0: Disable
3 (R/W)	RTC_ALARM0_EN	1'b0	Enable RTC alarm0 to trigger to wakeup 1: Enable 0: Disable
4 (R/W)	RTC_ALARM1_EN	1'b0	Enable RTC alarm1 to trigger to wakeup 1: Enable 0: Disable
5 (R/W)	LOW_BATT_EN	1'b1	Enable X_LOW_BATT_B to trigger to generate interrupt in normal mode 1: Enable 0: Disable

Bit	Name	Default	Description
31:6	-	26'b0	Reserved

Table 99: PWRC Trigger Enable Register

- PWRC Interrupt Status Register (PWRC_INT_STATUS) – 0x000C

Bit	Name	Default	Description
0 (R)	-	1'b0	Reserved
1 (R/W)	ON_KEY	1'b0	1: Interrupt happen 0: Clear
2 (R/W)	EXT_ON	1'b0	1: Interrupt happen 0: Clear
3 (R)	-	1'b0	Reserved
4 (R)	-	1'b0	Reserved
5 (R/W)	LOW_BATT	1'b0	1: Interrupt happen 0: Clear
31:6	-	26'b0	Reserved

Table 100: PWRC Interrupt Status Register

- PWRC Interrupt Mask Register (PWRC_INT_MASK) – 0x0010

Bit	Name	Default	Description
0 (R)	-	1'b0	Reserved
1 (R/W)	ON_KEY	1'b0	0: Mask the interrupt 1: Not mask
2 (R/W)	EXT_ON	1'b0	0: Mask the interrupt 1: Not mask
3 (R)	-	1'b0	Reserved
4 (R)	-	1'b0	Reserved
5 (R/W)	LOW_BATT	1'b0	0: Mask the interrupt 1: Not mask
31:19	-	-	Reserved

Table 101: PWRC Interrupt Mask Register

- PWRC Pin Status Register (PWRC_PIN_STATUS) – 0x0014

Bit	Name	Default	Description
0 (R)	-	1'b0	Reserved
1 (R)	ON_KEY	-	Directly read X_ON_KEY_B pin status
2 (R)	EXT_ON	-	Directly read X_EXT_ON pin status
3 (R)	-	1'b0	Reserved
4 (R)	-	1'b0	Reserved
5 (R)	LOW_BATT	-	Directly read X_LOW_BATT_B pin status
31:11	-	21'b0	Reserved

Table 102: PWRC Pin Status Register

- PWRC Scratch Pad Register (PWRC_SCRATCH_PAD) – 0x0018

Bit	Name	Default	Description
31:0 (R/W)	SCRATCH_PAD	32'b0	For software to save data.

Table 103: PWRC Scratch Pad Register

PWR (Power Management Module)

Overview

The PWR (Power Management Module) is used to control the TSC analog domain and the OSC/RF/CKO_0/1 pad.

Feature List

- RISC I/O interface

Functional Descriptions

The Module is only used to control the touch screen analog domain and 24MHz OSC pad, RF clock pad enable or disable, and CKO_0/1 pad source select.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
PWR internal register base address through RISC I/O	0x90060000

Table 104: PWR Interface Address Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	PWR CONTROL	Power management control register
0x0004	PWR_OSC_CONFIG	Power management oscillator configuration register

Table 105: PWR Register Mapping

Register Descriptions

- Power Management Control Register (PWR_CTRL) – 0x0000

Bit	Name	Default	Description
0 (W)	SLEEP	1'b0	Should be 0.
1 (R/W)	RF_PWR_OFF_EN	1'b0	Should be 0.
2 (R/W)	CPH_EN	1'b0	Should be 0.
3 (R/W)	PEN_DETECT_EN	1'b0	0: Disables the touch screen pen detect function. 1: Enables the touch screen pen detect function.
4 (R/W)	TSC_ISO_EN	1'b1	0: Touch screen power isolation disabled. 1: Touch screen power isolation enabled. **NOTE**: Before disabling the touch screen power isolation, the power of touch screen should be power on.
5 (R/W)	PAD_RF_CLK_EN	1'b1	0: Pad is for other use. 1: Pad is for RF_CLK input. For more detail, please check pin share document.

Bit	Name	Default	Description
6 (R/W)	MEMC_CLKOE_CTRLEN	1'b0	Reserved
7 (R/W)	CKO_32K_EN	1'b0	0: Pad X_CKO_0 does not select 32KHz (XINW) clock output. 1: Pad X_CKO_0 selects 32KHz (XINW) clock output.
8 (R/W)	CKO_24M_EN	1'b0	0: Pad X_CKO_0 does not select 24MHz (XIN) clock output. 1: Pad X_CKO_0 selects 24MHz (XIN) clock output. **NOTE**: If CKO_32K_EN is valid, no matter CKO_24M_EN is valid or not, the CKO outputs 32KHz clock.
15:9 (R)	-	7'b0	Reserved
23:16 (R/W)	GPIO_WAKUP_MODE	8'h0	Reserved
31:24 (R/W)	GPIO_WAKEUP_LEVEL	8'h0	Reserved

Table 106: Power Management Control Register (PWR_CTRL)

- Power Management Oscillator Configuration Register (PWR_OSC_CONFIG) – 0x0004

Bit	Name	Default	Description
0 (R)	OSC_OK	1'b0	Reserved
1 (R)	OSC_PD	1'b0	Reserved
2 (R/W)	OSC_FO	1'b0	Reserved
3 (R/W)	OSC_FD	1'b0	Force 24MHz oscillator power down. Used to force a 24MHz oscillator power down during normal operating mode. Before enabling this bit, switch the clock source to the 32KHz oscillator.
4 (R/W)	OPD_EN	1'b0	Reserved
31:5	-	27'h0	Reserved

Table 107: Power Management Oscillator Configuration Register (PWR_OSC_CONFIG)

RSC (Resource Sharing Control Module)

Overview

The RSC (Resource Sharing Module) is used to control PAD-MUX and PAD driving strength. The PAD-MUX is for the chip's pad share logic. The pad function is determined by the control register (configured by software).

Refer to the PAD MUX section for more details.

- Pad MUX Program
  Software will determine which device uses the pad. Program the static pad mux as below:
  - Write RSC_PIN_MUX to switch the pad MUX to the right device controller (Codec's pad MUX control signal exists in the codec controller, this is because AC97 and I^2S are both in the same module)
  - Check if the GPIO pad enable needs to be configured.
  - Start related device controllers such as SPI, PWM, and so on.

Feature List

- Static selection for PAD share

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RSC internal register base address through RISC I/O	0x90030000
PADCTRL internal register base address through RISC I/O	0x90034000

Table 108: RSC Interface Address Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	-	Reserved
0x0004	RSC_PIN_MUX	RSC Pin Multiplex Register
0x0008	-	Reserved
0x000C	-	Reserved
0x4000	PADCTRL_PAD_OUT0	PAD Output Group 0 Control
0x4004	PADCTRL_PAD_OUT1	PAD Output Group 1 Control
0x4008	PADCTRL_PAD_OUT2	PAD Output Group 2 Control

RISC Address <11:0>	Register	Description
0x400c	PADCTRL_PAD_OUT3	PAD Output Group 3 Control
0x4010	PADCTRL_PAD_OUT4	PAD Output Group 4 Control
0x4014	PADCTRL_PAD_OUT5	PAD Output Group 5 Control
0x4018	PADCTRL_PAD_OUT6	PAD Output Group 6 Control
0x401c	PADCTRL_PAD_OUT7	PAD Output Group 7 Control
0x4020	PADCTRL_PAD_OE0	PAD Output Enable Group 0 Control
0x4024	PADCTRL_PAD_OE1	PAD Output Enable Group 1 Control
0x4028	PADCTRL_PAD_OE2	PAD Output Enable Group 2 Control
0x402c	PADCTRL_PAD_OE3	PAD Output Enable Group 3 Control
0x4030	PADCTRL_PAD_OE4	PAD Output Enable Group 4 Control
0x4034	PADCTRL_PAD_OE5	PAD Output Enable Group 5 Control
0x4038	PADCTRL_PAD_OE6	PAD Output Enable Group 6 Control
0x403C	PADCTRL_PAD_OE7	PAD Output Enable Group 7 Control
0x4040	PADCTRL_PAD_PS0	PAD Input Pull Select Group 0 Control
0x4044	PADCTRL_PAD_PS1	PAD Input Pull Select Group 1 Control
0x4048	PADCTRL_PAD_PS2	PAD Input Pull Select Group 2 Control
0x404C	PADCTRL_PAD_PS3	PAD Input Pull Select Group 3 Control
0x4050	PADCTRL_PAD_PS4	PAD Input Pull Select Group 4 Control
0x4054	PADCTRL_PAD_PS5	PAD Input Pull Select Group 5 Control
0x4058	PADCTRL_PAD_PE0	PAD Input Pull Enable Group 0 Control
0x405C	PADCTRL_PAD_PE1	PAD Input Pull Enable Group 1 Control
0x4060	PADCTRL_PAD_PE2	PAD Input Pull Enable Group 2 Control
0x4064	PADCTRL_PAD_PE3	PAD Input Pull Enable Group 3 Control
0x4068	PADCTRL_PAD_PE4	PAD Input Pull Enable Group 4 Control
0x406C	PADCTRL_PAD_PE5	PAD Input Pull Enable Group 5 Control
0x4070	PADCTRL_PAD_CD0	PAD Driving Bit 0 Control
0x4074	PADCTRL_PAD_CD1	PAD Driving Bit 1 Control
0x4078	PADCTRL_PAD_MEMPAD	Memory Pad Driving Control

Table 109: RSC Register Mapping

Register Descriptions

- Pin Multiplex Register (RSC_PIN_MUX) – 0x4

Bit	Name	Default	Description
0	-	-	Reserved
1 (R/W)	PAD_I2S_EXTCLK_EN	1'b0	I^2S external clock enable: 0: x_usclk_0 1: i2s_extclk
2 (R/W)	PAD_I2S_6CHN_EN	1'b0	I^2S 6channel enable: 0: USP interface 1: I^2S data[2:1] interface
3 (R/W)	PAD_UTMI_DRVVBUS_EN	1'b0	USB VBUS control pin enable: 1'b0: x_utfs_0 1'b1: USB utmi drvvbus
4	-	-	Reserved
5 (R/W)	PAD_I2S_MCLK_EN	1'b0	I^2S MCLK pin enable. 0: CKO_1 uses the pin 1: MCLK uses the pin
20:6	-	15'h0	Reserved
21 (R/W)	PAD_ROM_EN	1'b0	ROM interface enable. 1'b0: LCD interface. 1'b1: ROM interface.
22 (R/W)	PAD_SPI_0_EN	1'b0	SPI enable. 1'b0: SD 3 interface. 1'b1: SPI interface.
23 (R/W)	PAD_SD2_EN	1'b0	SD2 interface enable. 1'b0: SD0 or NAND interface decided by boot mode. 1'b1: SD2 interface.
24 (R/W)	PAD_UART0_STREAM_EN	1'b0	UART0 stream control enable 1'b0: USP or I^2S 6 channel interface decided by PAD_I2S_6CHN_EN. 1'b1: UART0 steam control interface.
27:25	-	-	Reserved
28 (R/W)	PAD_SPK_EN	1'b0	X_URXD_1, X_UTFS_1 output source control: 1'b0 used as other function 1'b1 used as speaker

Bit	Name	Default	Description
			NOTE: If GPIO0_PAD_EN[26:25] == 2'b11, the pads are used as GPIO even if PAD_SPK_EN is 1'b1.
29 (R/W)	PAD_HEADPHONE_EN	1'b0	X_USCLK_1, X_UTXD_1 output source control: • 1'b0 is used as other function • 1'b1 is used as headphone **NOTE**: If GPIO0_PAD_EN[24:23] == 2'b11, the pads are used as GPIO even if PAD_HEADPHONE_EN is 1'b1.
31:30	-	-	Reserved

Table 110: Pin Multiplex Register (RSC_PIN_MUX)

- PAD Output Group 0 Register (PADCTRL_PAD_OUT0) – 0x4000

Bit	Name	Default	Description
[31:0] (R/W)	PAD_OUT[31:0]	32'h0	They are used to control pad output value before entering deep-sleep mode. Configure them before writing deep-sleep commands. • 1'b0: If PADCTRL_PAD_OE0 related bit is 1'b1, the pad will output 0 • 1'b1: If PADCTRL_PAD_OE0 related bit is 1'b1, the pad will output 1 For the bit relationship between PADCTRL with pin name, see the Pin Share Table.

Table 111: PAD Output 0 Register (PADCTRL_PAD_OUT0)

- PAD Output Group 1~7 Register (PADCTRL_PAD_OUT1~7) – 0x4004~0x401C

Bit	Name	Default	Description
[31:0] (R/W)	PAD_OUT[63:32]	32'h0	They are used to control pad output value before entering deep-sleep mode. Configure them before writing deep-sleep commands. • 1'b0: If PADCTRL_PAD_OE1~7 related bit is 1'b1, the pad outputs 0. • 1'b1: If PADCTRL_PAD_OE1~7 related bit is 1'b1, the pad outputs 1. For the bit relationship between PADCTRL with pin name, see the Pin Share Table.
	PAD_OUT[95:64]		
	PAD_OUT[127:96]		
	PAD_OUT[159:128]		
	PAD_OUT[191:160]		
	PAD_OUT[223:192]		
	PAD_OUT[255:224]		

Table 112: PAD Output 1~7 Register (PADCTRL_PAD_OUT1~7)

- PAD Output Enable Group 0 Register (PADCTRL_PAD_OE0) – 0x4020

Bit	Name	Default	Description
[31:0] (R/W)	PAD_OE[31:0]	32'hffffffff	They are used to control pad output enable before entering deep-sleep mode. Configure them before writing deep-sleep commands. • 1'b0: The pad is input. • 1'b1: The pad is output. For the bit relationship between PADCTRL with pin name, see the Pin Share Table.

Table 113: PAD Output Enable 0 Register (PADCTRL_PAD_OE0)

- PAD Output Enable Group 1~7 Register (PADCTRL_PAD_OE1~7) – 0x4024~0x403C

Bit	Name	Default	Description
[31:0] (R/W)	PAD_ OE[63:32]	32'hffffffff	They are used to control pad output enable before entering deep-sleep mode. Configure them before writing deep-sleep commands. • 1'b0: The pad will be as input. • 1'b1: The pad will be as output. For the bit relationship between PADCTRL with pin name, see the Pin Share Table.
	PAD_ OE[95:64]		
	PAD_ OE[127:96]		
	PAD_ OE[223:192]		
	PAD_ OE[255:224]		
	PAD_ OE[159:128]	32'h7fffffff	
	PAD_ OE[191:160]	32'hfffffffa	

Table 114: PAD Output Enable 1~7 Register (PADCTRL_PAD_OE1~7)

- PAD Pull Select Group 0 Register (PADCTRL_PAD_PS0) – 0x4040

Bit	Name	Default	Description
[31:0] (R/W)	PAD_PS[31:0]	32'h0	They are used to control pad input pull type when pull enabled before entering deep-sleep mode. Configure them before writing deep-sleep commands. • 1'b0: The pad is pull-low. • 1'b1: The pad is pull-high. For the bit relationship between PADCTRL with pin name, see the Pin Share Table.

Table 115: PAD Pull Select Group 0 Register (PADCTRL_PAD_PS0)

- PAD Pull Select Group 1~5 Register (PADCTRL_PAD_PS1~5) – 0x4044~0x4054

Bit	Name	Default	Description
[31:0] (R/W)	PAD_ PS[63:32]	32'h0	They are used to control pad input pull type when pull enabled before entering deep-sleep mode. Configure them before writing deep-sleep commands. • 1'b0: The pad is pull-low. • 1'b1: The pad is pull-high. For the bit relationship between PADCTRL with pin name, see the Pin Share Table.
	PAD_ PS[95:64]		
	PAD_ PS[127:96]		
	PAD_ PS[159:128]		
	PAD_ PS[191:160]	32'h4	

Table 116: PAD Pull Select Group 1~5 Register (PADCTRL_PAD_PS1~5)

- PAD Pull Enable Group 0 Register (PADCTRL_PAD_PE0) – 0x4058

Bit	Name	Default	Description
[31:0] (R/W)	PAD_PE[31:0]	32'h0	They are used to control pad input pull enabled before entering deep-sleep mode. Configure them before writing deep-sleep commands. • 1'b0: The pad pull is disabled. • 1'b1: The pad pull is enabled. For the bit relationship between PADCTRL with pin name, see the Pin Share Table.

Table 117: PAD Pull Enable Group 0 Register (PADCTRL_PAD_PE0)

- PAD Pull Enable Group 1~5 Register (PADCTRL_PAD_PE1~5) – 0x405c~0x406c

Bit	Name	Default	Description
[31:0] (R/W)	PAD_ PE[63:32]	32'h0	They are used to control pad input pull enabled before entering into deep sleep mode. You need configure them before write deep-sleep command. • 1'b0: The pad pull is disabled. • 1'b1: The pad pull is enabled. For the bit relationship between PADCTRL with pin name, please see Pin Share Table for detail.
	PAD_ PE[95:64]		
	PAD_ PE[127:96]		
	PAD_ PE[159:128]		
	PAD_ PE[191:160]	32'h4	

Table 118: PAD Pull Enable Group 1~5 Register (PADCTRL_PAD_PE1~5)

- PAD Driving Bit 0 Register (PADCTRL_PAD_CD0) – 0x4070

Bit	Name	Default	Description
14:0 (R/W)	PAD_CD0	15'hbfcb	They are used to control pad driving bit0. The driving value depends on pad cell type.
31:15	-	-	Reserved.

Table 119: PAD Driving Bit 0 Register (PADCTRL_PAD_CD0)

- PAD Driving Bit 1 Register (PADCTRL_PAD_CD1) – 0x4074

Bit	Name	Default	Description
14:0 (R/W)	PAD_CD1	15'h0000	They are used to control pad driving bit1. The driving value depends on pad cell type.
31:15	-	-	Reserved.

Table 120: PAD Driving Bit 1 Register (PADCTRL_PAD_CD1)

- PAD Memory Driving Register (PADCTRL_PAD_MEMPAD) – 0x4078

Bit	Name	Default	Description
1:0 (R/W)	PAD_MEMPAD_DATA	2'b01	They are used to control memory pad (x_md, x_mdqs, x_mdqm) driving. The driving value depends on the pad cell type and voltage. When VDD=1.8v (typical): CD1=0,CD0=0 Isink 4.9mA CD1=0,CD0=0 Isource -3.5mA CD1=0,CD0=1 Isink 9.8mA CD1=0,CD0=1 Isource -7.1mA CD1=1,CD0=0 Isink 14.8mA CD1=1,CD0=0 Isource -10.6mA CD1=1,CD0=1 Isink 19.7mA CD1=1,CD0=1 Isource -14.2mA
3:2 (R/W)	PAD_MEMPAD_CLK	2'b01	They are used to control memory pad (x_mclk_o, x_mclkb_o) driving. The driving value depends on the pad cell type and voltage. When VDD=1.8v (typical): CD1=0,CD0=0 Isink 4.9mA

<table>
<tr><th>Bit</th><th>Name</th><th>Default</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td>Isource</td><td>-3.5mA</td></tr>
<tr><td rowspan="2">CD1=0,CD0=1</td><td>Isink</td><td>9.8mA</td></tr>
<tr><td>Isource</td><td>-7.1mA</td></tr>
<tr><td rowspan="2">CD1=1,CD0=0</td><td>Isink</td><td>14.8mA</td></tr>
<tr><td>Isource</td><td>-10.6mA</td></tr>
<tr><td rowspan="2">CD1=1,CD0=1</td><td>Isink</td><td>19.7mA</td></tr>
<tr><td>Isource</td><td>-14.2mA</td></tr>
</table>
</td></tr>
<tr><td>5:4 (R/W)</td><td>PAD_MEMPAD_CTRL</td><td>2’b01</td><td>
They are used to control memory pad (x_ma, x_mcke, x_mwe_b, x_mcas_b, x_mras_b, x_m_ba) driving. The driving value depends on the pad cell type and voltage.
When VDD=1.8v (typical):
<table>
<tr><td rowspan="2">CD1=0,CD0=0</td><td>Isink</td><td>4.9mA</td></tr>
<tr><td>Isource</td><td>-3.5mA</td></tr>
<tr><td rowspan="2">CD1=0,CD0=1</td><td>Isink</td><td>9.8mA</td></tr>
<tr><td>Isource</td><td>-7.1mA</td></tr>
<tr><td rowspan="2">CD1=1,CD0=0</td><td>Isink</td><td>14.8mA</td></tr>
<tr><td>Isource</td><td>-10.6mA</td></tr>
<tr><td rowspan="2">CD1=1,CD0=1</td><td>Isink</td><td>19.7mA</td></tr>
<tr><td>Isource</td><td>-14.2mA</td></tr>
</table>
</td></tr>
<tr><td>31:6</td><td>-</td><td>-</td><td>Reserved</td></tr>
</table>

Table 121: PAD Memory Driving Register (PADCTRL_PAD_MEMPAD)

Programming Guide

Flowcharts for PADMUX Configuration

The following flowcharts illustrate how the pad is configured. Every chart will have a ready state named Normal, which indicates the SoC chip is working in a normal state.

- Programming steps for general function pins are:

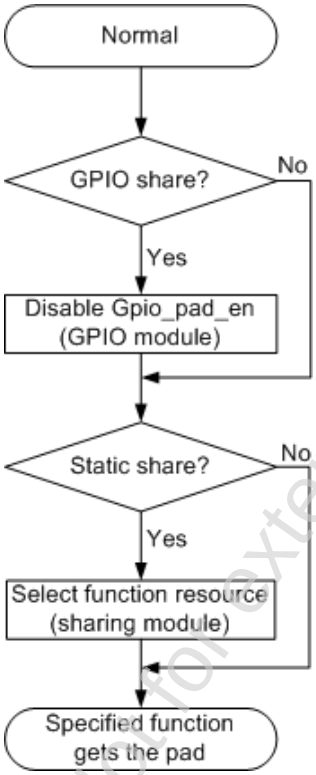


Figure 6: Pad MUX Function Switch Flowchart

In Figure 6, “GPIO share” indicates that the pads must be either MUXed as GPIO or other function pads before the register to disable its GPIO function is set. “Static share” means the pad function can be freely configured.

- Programming steps for AC97/I^2S:

  Although AC97 and I^2S share the same PAD logic, however neither of which shares the same PAD logic with GPIOs. It is therefore recommended to configure the register to either AC97 or I^2S.

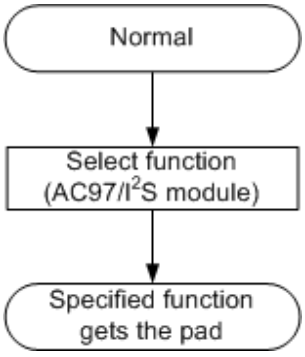


Figure 7: Pad MUX AC97/I^2S Switch Flowchart

- Programming steps for GPIO:

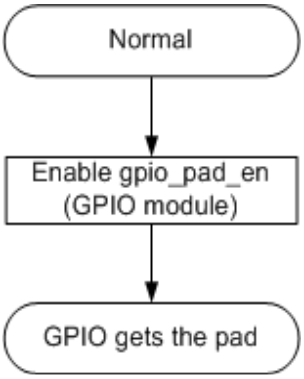


Figure 8: Pad MUX GPIO Switch Flowchart

- Programming steps for PAD Controlling registers:

  The PAD control registers (PADCTRL_PAD_OUTn, PADCTRL_PAD_OEn, PADCTRL_PAD_PEn, and PADCTRL_PAD_PSn) are used to control all PAD's status before system entering deep-sleep mode. The following diagram shows the PAD status changes and configures before the system entering into Deep Sleep Mode.

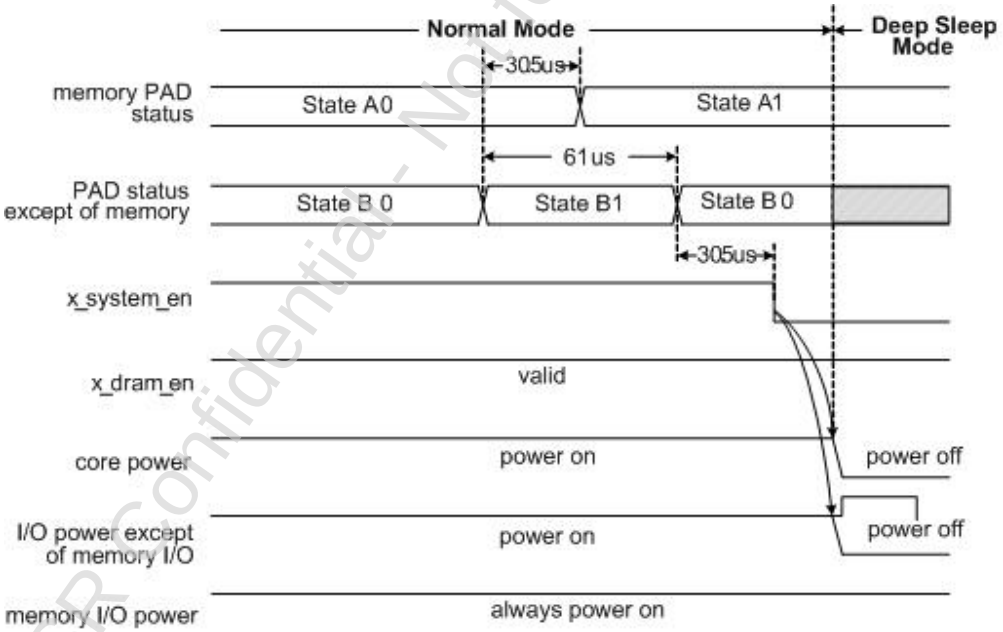


Figure 9: Pads Status before Entering Deep Sleep Mode

The PADs status in above figure:

- State A0

  Memory PADs status after the last memory operation before entering Deep-Sleep mode. This status is controlled by the memory controller.

- State B0

  PADs status except memory after the last PADs related function block operation before entering the deep-sleep mode. This status is controlled by the PADs function block.

- State A1

  Memory PADs status corresponding to the configuration of the PAD control registers.

- State B1

  PADs status except memory corresponding to the configuration of the PAD control registers. It is recommended to configure the State B1 same with the State B0 for each PAD.

For the PADs status before entering into Hibernation Mode, it's not affect by the PAD control registers, as shown in the following diagram.

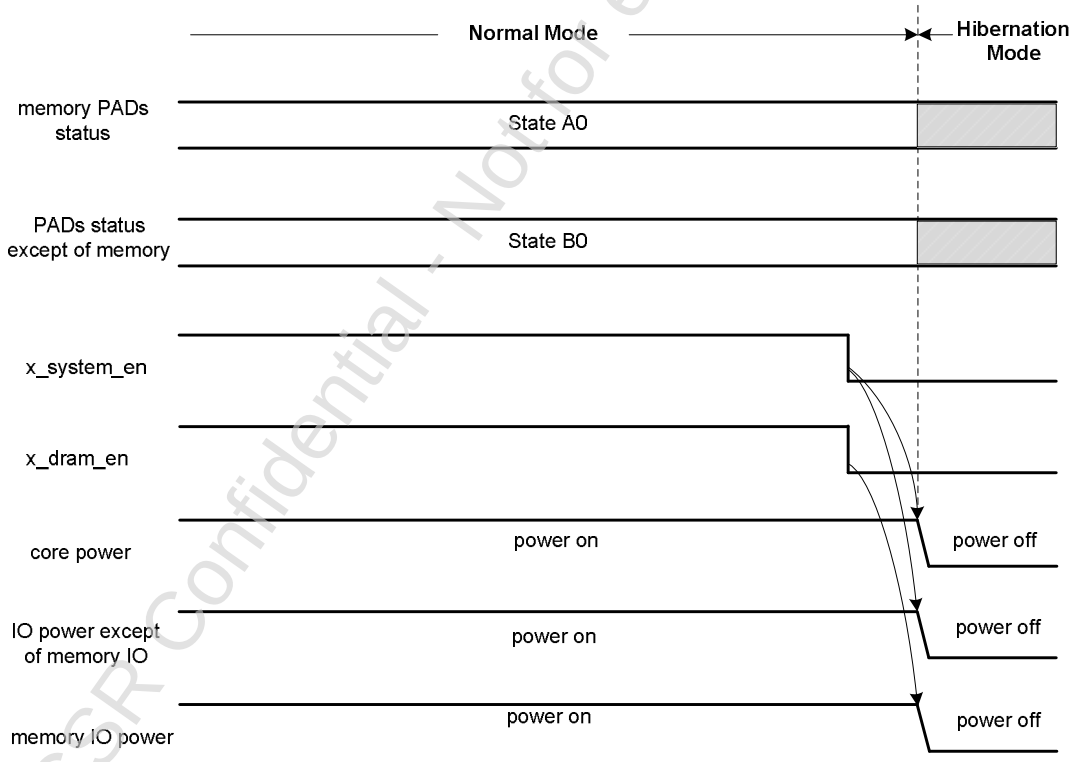


Figure 10: Pads Status before Entering Hibernation Mode

Reset Controller

Overview

The reset controller manages various reset sources in SiRFatlasV. The reset controller block controls all resets in SiRFatlasV except for the touch screen PHY and RTC reset. RISC can access it through RISC I/O.

Pin Descriptions

External Pin Descriptions

The following table shows the pins used in the reset controller and their functions. The reset controller pins are multiplexed with other devices. Because the default function of the pads is GPIO, you need to disable the corresponding GPIOx_PAD_EN if these pads are going to work in the reset controller function mode.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_RESET_B	I	X_RESET_B MBR(bist_reset)	X_RESET_B	Input without pull	Power on reset
X_GPIO_12	I	GPIO NSRST	GPIO	Input pull-up	JTAG nsrst
X_GPIO_14	I	GPIO NTRST	GPIO	Input pull-up	JTAG ntrst
X_SCAN_EN	I	X_SCAN_EN	X_SCAN_EN	Input pull-down	Scan mode enable
X_GPIO_8	I	GPIO Warm_rst	GPIO	Input pull-up	Chip warm reset.

Table 122: Reset Controller External Pin Descriptions

Functional Descriptions

Reset Type

There are 7 reset types:

- Power-on reset

  Power-on reset is asserted through the X_RESET_B pin which is intended to be used for power-on reset. It is a full-chip reset so that every register on the chip (except for the system RTC and touch screen PHY) will be reset to its default value and the content in the memory will be lost.

- nSRST reset

  nSRST reset is asserted through the nSRST pin and is intended to be used for ICE reset the entire chip. It's a full-chip reset so that every register on the chip (except for the cp14 registers in

arm1136 and RTC, power controller and touch screen PHY) will be reset to its default value and the content in the memory will be lost.

- nTRST reset

  nTRST reset is asserted through the nTRST pin and is intended to be used for ICE reset the TAP controller.

- Bist reset

  Bist-reset is asserted through the bistresetn pin that is intended to be used for cpu bist reset only. The output pin is rst_b[35].

- Software reset

  There is a RESET_SWR register which can be programmed by the user to reset most of the blocks on-chip. Users can choose to apply reset to each block separately. The assertion of system-reset bit (bit31: SYS_RST) will reset the chip. Software reset will not cause memory data loss.

- Watchdog reset

  Watchdog reset is generated when TIMER_WATCHDOG_EN is set and TIMER_MATCH_5 matches the OS timer counter. The result of a Watchdog reset is the same as the System software reset.

- Warm reset

  Warm reset is assert through the warm_rst_b pin, the function is as blow

  i. warm_rst_b goes low to tell RISC start warm reset process.
  ii. warm_rst_b signal:
    a. forces the memory controller to control the memory enter self-refresh mode.
    b. triggers a hardware counter to delay some time to ensure memory has enter the self-refresh mode (hardware logic will also check cke signal to make sure it has enter the self-refresh mode).
  iii. After make sure memory has enter self-refresh. Generate an internal reset signal with the same action as watch_dog reset.

Reset Scheme

Each block in SiRFatlasV has its own reset control signal. As described above, the reset has 7 sources: Power-on-reset, nSRST-reset, nTRST-reset, BIST-reset, Software-reset, Watchdog-reset, and Warm reset. The following table shows that how each module is controlled by these reset sources.

Module	Power-on reset	Warm-reset	Bist reset	nTRST	nSRST	Software reset		Watchdog reset
						System software reset	Block software reset	
RISC Core	Yes	Yes	Yes	No	Yes	Yes	No	Yes
RISC Interface	Yes	Yes	No	No	Yes	Yes	No	Yes
Interrupt	Yes	Yes	No	No	Yes	Yes	No	Yes

Module	Power-on reset	Warm-reset	Bist reset	nTRST	nSRST	Software reset		Watchdog reset
Controller								
CLKC	Yes	Yes	No	No	Yes	Yes	No	Yes
GPSRTC	Yes	No	No	No	No	No	No	No
SYSRTC	No	No	No	No	No	No	No	No
Timer	Yes	Yes	No	No	Yes	Yes	No	Yes
Reset controller	Yes	No	No	No	No	No	No	No
GPIO	Yes	Yes	No	No	Yes	Yes	No	Yes
System Arbiter	Yes	Yes	No	No	Yes	Yes	No	Yes
RISC CP14 Registers	Yes	No	No	No	No	No	No	No
Bist Logic	Yes	Yes	Yes	No	Yes	Yes	No	Yes
RISC TAP controller	Yes	No	No	Yes	No	No	No	No
Others	Yes	Yes	No	No	Yes	Yes	Yes	Yes

Table 123: Reset Table

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O Interface	0x90070000

Table 124: Reset Controller Base Address

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	RESET_SR	Reset controller software reset register
0x0004	RESET_STATUS	Reset controller status register
0x0008	RESET_SR1	Reset controller software reset register1
0x000c	WARM_TIME_OUT	Warm reset timeout counter

Table 125: Reset Controller Register Mapping

Register Descriptions

- Reset Controller Software Reset Register (RESET_SWR) – 0x0000

  Writing a 1 to each bit from bit30 to bit0 will cause the corresponding block to be reset. Writing a 1 to the most-significant bit (SYS_RST) will cause all on-chip resources to reset except for the Power Controller, Reset controller and RTC. Writing a 0 to the software-reset bit will clear the corresponding reset. Special attention should be paid to restrict access to this register by programming MMU permissions. For reserved bits, writes produce no effects.

  The software reset program sequence is as follows:

  - For ASYNC reset block:
    a. Disable the current block.
    b. Stop the clock of current block.
    c. Assert the reset of the current block.
    d. Release the reset of the current block.
    e. Enable the clock of the current block.
    f. Enable the current block.
  - For synchronizing the reset block:
    a. Disable the current block.
    b. Stop the clock of the current block.
    c. Assert the reset of the current block.
    d. Enable the clock of the current block.
    e. Release the reset of the current block.
    f. Enable the current block.

  This register will be reset to the default value by power-on reset only.

Bit	Name	Default	Description
0 (R/W)	DSP_RST	1'b0	DSP Core software reset: 1: Reset
1 (R/W)	DIFACE_RST	1'b0	DSP Interface software reset
2 (R/W)	ROM_RST	1'b0	ROM Controller software reset
3 (R/W)	PCI_RST	1'b0	SYS2PCI and PCI arbiter software reset
4 (R/W)	DMA_RST	1'b0	DMA Controller software reset
5 (R/W)	LCD_RST	1'b0	LCD Controller software reset
6 (R/W)	Reserved	1'b0	Reserved
7 (R/W)	GPS_RST	1'b0	GPS software reset
8 (R/W)	Reserved	1'b0	Reserved
9 (R/W)	NAND_RST	1'b0	NAND flash software reset
10 (R/W)	COPY_RST	1'b0	PCI COPY software reset
11 (R/W)	I2C0_RST	1'b0	I^2C0 software reset

Bit	Name	Default	Description
12 (R/W)	Reserved	1'b0	Reserved
13 (R/W)	CODEC_RST	1'b0	Audio Codec software reset
14 (R/W)	UART0_RST	1'b0	UART 0 software reset
15 (R/W)	USP0_RST	1'b0	USP 0 software reset
16 (R/W)	USP1_RST	1'b0	USP 1 software reset
17 (R/W)	SPI_RST	1'b0	SPI software reset
18 (R/W)	Reserved	1'b0	Reserved
19 (R/W)	I2C1_RST	1'b0	I^2C1 software reset
20 (R/W)	VPP_RST	1'b0	VPP software reset
21 (R/W)	SD0_RST	1'b0	SD controller 0 software reset
22 (R/W)	PWM_RST	1'b0	PWM software reset
23 (R/W)	Reserved	1'b0	Reserved
24 (R/W)	UART1_RST	1'b0	UART 1 software reset
25 (R/W)	Reserved	1'b0	Reserved
26 (R/W)	SD1_RST	1'b0	SD controller 1 software reset
27 (R/W)	USBOTG_RST	1'b0	USBOTG software reset
28 (R/W)	Reserved	1'b0	Reserved
29 (R/W)	Reserved	1'b0	Reserved
30 (R/W)	Reserved	1'b0	Reserved
31 (R/W)	SYS_RST	1'b0	System software reset (including all blocks except the power controller, resource sharing controller, real time clock, and GPIO).

Table 126: Reset Controller Software Reset Register

- Reset Controller Status Register (RESET_STATUS) – 0x0004

  Users can use the reset controller to reset status register (RESET_STATUS) and to determine the last cause or causes of the reset.

  Each RESET_STATUS status bit is set by a different reset sources and can be cleared by writing a 0 back to that bit. After Power-on reset, all status bits will be zero except for the HWR bit. For reserved bits, writes will be ignored and reads will return zeroes.

Bit	Name	Default	Description
0 (R/W)	HWR	1'b1	Hardware reset: 0: Hardware reset has not occurred since the last time CPU cleared this

Bit	Name	Default	Description
			bit. 1: Hardware reset has occurred since the last time CPU cleared this bit.
1 (R/W)	SWR	1'b0	Software reset: 0: Software reset has not occurred since the last time CPU cleared this bit. 1: Software reset has occurred since the last time CPU cleared this bit.
2 (R/W)	WDR	1'b0	Watchdog reset: 0: Watchdog reset has not occurred since the last time CPU cleared this bit. 1: Watchdog reset has occurred since the last time CPU cleared this bit.
3 (R/W)	WMR	1'b0	Warm reset: 0: Warm reset has not occurred since the last time CPU cleared this bit. 1: Warm reset has occurred since the last time CPU cleared this bit.
4 (R)	-	-	Reserved
5 (R)	WARM Reset Timeout	1'b0	Warm reset timeout: 0: The last is warm reset and memory is in selfrefresh mode now. 1: The last is not a successful warm reset and the memory is not in selfrefresh mode now.
31:6	-	26'h0	Reserved

Table 127: Reset Controller Status Register

- Reset Controller Software Reset Register 1 (RESET_SWR1) – 0x0008
  This register will be reset to default value by power-on reset only.

Bit	Name	Default	Description
0 (R/W)	MEM_RST	1'b0	Memory controller software reset
1	-	-	Reserved
2 (R/W)	TSCIF_RST	1'b0	TSC interface software reset
3.(R/W)	DAC_RST	1'b0	DAC interface software reset.
31:4	-	-	Reserved

Table 128: Reset Controller Software Reset Register 1

- Warm Reset time out Counter (WARM_TIME_OUT) – 0x000C

Bit	Name	Default	Description
31:0 (R/W)	WARM_TIME_OUT_CNT	32'h50	Warm reset time out counter: This register will decrease from the default value or the set value (the value can be set by RISC). When the counter is decreased to 0, a warm reset will occur even if the memory did not enter self refresh mode. This register will be set to the default value after a warm reset.

Table 129: Warm Reset Timeout Counter Register

Interrupt Controller

Overview

In the SiRFatlasV processor, both the RISC and the DSP core can accept external interrupts. In addition, RISC and DSP can interrupt each other. The RISC core contains two external interrupt inputs, FIQ and IRQ. FIQ allows for fast interrupt processing by providing five additional dedicated general-purpose registers. The DSP core on the other hand, has up to six external interrupt pins, IRQ2n, IRQ1n, IRQ0n, IRQL1n, IRQL0n, and IRQEn. The interrupt controller will collect external interrupts and MUX them to DSP and RISC. Then both DSP and RISC can access the interrupts through DSP I/O or RISC I/O access.

Feature List

- Supports RISC and DSP interrupt
- Supports RISC FIQ and IRQ Level select
- Supports interrupt priority with interrupt ID in RISC interrupt

Functional Descriptions

Block Diagram

The interrupt hierarchy of SiRFatlasV is a two-level structure. The first level is responsible for the masking/unmasking of all enabled interrupts and sends the interrupt to the processor; the second level is implemented in the source device (the device generates the first level interrupt bit). The second-level interrupt status register gives additional information about the interrupt and is used inside the service routine. The enabling of interrupts is performed inside the source device.

For RISC, the first level of structure is represented by the interrupt controller IRQ pending register (INT_IRQ_PENDING) and the interrupt controller FIQ pending register (INT_FIQ_PENDING). The INT_IRQ_PENDING register contains the interrupts programmed to generate an IRQ interrupt, whereas the INT_FIQ_PENDING register contains all valid interrupts programmed to generate an FIQ interrupt. The routing is programmed via the RISC interrupt controller level register (INT_RISC_LEVEL).

The following diagram illustrates the RISC part of the interrupt controller block. For the DSP part, it is similar but not included in the diagram below:

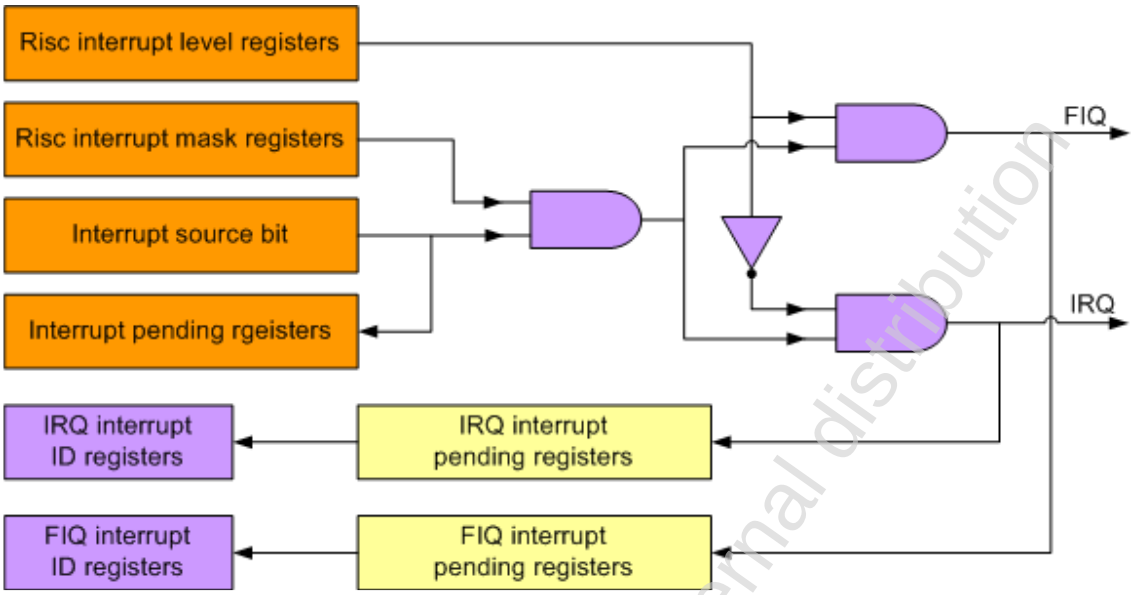


Figure 11: Block Diagram of the Interrupt Controller (RISC)

For DSP interrupts, the RISC→DSP interrupt is connected to IRQ1n and the FFT interrupt is connected to IRQL0n. The GPS interrupt is connected to IRQ2n and IRQ0n

For other interrupt sources, IRQL1n is used for low-level sensitive interrupts of the DSP core. This is a simpler process compared to RISC, because all qualified interrupt bits will be stored in the interrupt controller pending registers (INT_PENDING0 and INT_PENDING1). DSP can read the corresponding interrupt pending registers followed by the status register in the device. It uses a different interrupt mask register from RISC so that DSP can be interrupted independently by different sources with RISC.

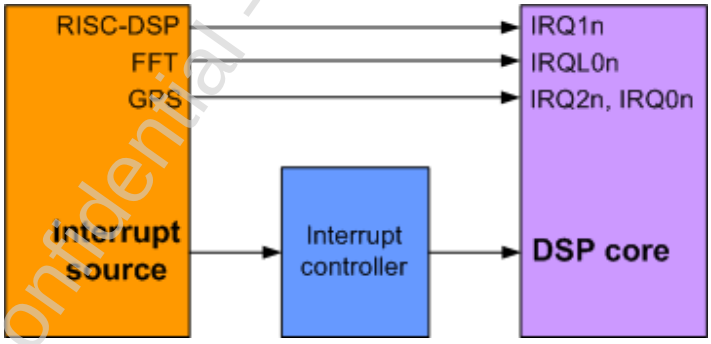


Figure 12: Block Diagram of the Interrupt Controller (DSP)

Interrupt mechanism

The interrupt source should be synchronous with two level flip-flop (clock is system clock), because the interrupt source maybe be generated from a different clock domain. Therefore there will be a three-clock (system clock) latency between interrupt source and the interrupt signal that connect to CPU (IRQ or FIQ).

The interrupt signal that is connected to CPU will be cleared after three clock cycle (system clock), when the CPU clears the interrupt source by writing to the interrupt status registers.

Therefore, after the CPU clears the interrupt source, it must wait at least three system clock cycles before it exits the ISR or the CPU will reenter the ISR with the same interrupt.

The DSP cannot check the GPIO interrupt pending through INT_PENDINGx register. Therefore if the DSP accepts the interrupt, check the interrupt status registers in GPIO block instead.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
DSP I/O interface	0x000
RISC I/O Interface	0x90020000

Table 130: Interrupt Controller Base Address

Register Mapping

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x0000	0x00~01	INT_PENDING0	Read only. RISC Interrupt Pending Register 0
0x0004	0x02~03	INT_PENDING1	Read only. RISC Interrupt Pending Register 1
0x0008	-	INT_IRQ_PENDING0	Read only. IRQ Pending Register 0 Default: 0x00000000
0x000C	-	INT_IRQ_PENDING1	Read only. IRQ Pending Register 1 Default: 0x00000000
0x0010	-	INT_FIQ_PENDING0	Read only. FIQ Pending Register 0 Default: 0x00000000
0x0014	-	INT_FIQ_PENDING1	Read only. FIQ Pending Register 1 Default: 0x00000000
0x0018	-	INT_RISC_MASK0	RISC Mask Register 0 Default: 0x00000000
0x001C	-	INT_RISC_MASK1	RISC Mask Register 1 Default: 0x00000000
0x0020	-	INT_RISC_LEVEL0	RISC Level Register 0

RISC Address <11:0>	DSP Address <6:0>	Register	Description
			Default: 0x00000000
0x0024	-	INT_RISC_LEVEL1	RISC Level Register 1 Default: 0x00000000
0x0028	0x14~15	INT_DSP_MASK0	DSP Mask Register 0 Default: 0x00000000
0x002C	0x16~17	INT_DSP_MASK1	DSP Mask Register 1 Default: 0x00000000
0x0030	0x18	INT_DSP_ACCEN	DSP access enable. This register is a read-only register for DSP. Only RISC can read or write to it. Default: 0x00000000
0x0034	0x20	CHIP_ID	This is a read-only register, it contains the CHIP ID. Default: Depending on chip version
0x0038	-	INT_ID_IRQ	Interrupt ID of IRQ
0x003c	-	INT_ID_FIQ	Interrupt ID of FIQ
Others	Others	-	Reserved

Table 131: Register Mapping

Register Descriptions

- Interrupt Controller Pending Register 0 (INT_PENDING) – RISC: 0x0000 DSP: 0x00~0x01

  The INT_PENDING0 is a 32-bit read-only register that shows all active interrupts in the system. These bits are not affected by the state of the mask register. The following table shows the pending interrupt source assigned to each bit position in the INT_PENDING0. For more details on the second-level interrupts, please read the relevant section.

  Bits within INT_PENDING0 are read-only, and represent the logical OR of status bits for a given interrupt within the source unit. Once an interrupt has been serviced, the handler will clear the pending interrupt at the source by writing a 1 to the necessary status bit. Clearing the interrupt status bit at the source will automatically clear the corresponding INT_IRQ_PENDING and INT_FIQ_PENDING flag unless there are no other interrupt status bits set within the source unit. All interrupt source status bits are cleared by writing a 1 to them. Writing a 0 to an interrupt status bit produces no effects.

  The default value is unknown upon power on.

  NOTE – Writing 1 to this register bit will generate an interrupt (only for testing purpose).

Bit	Source Module	Description	Sensitive Type
0 (R)	Timer	OS Timer 0 interrupt request	Level
1 (R)		OS Timer 1 interrupt request	Level
2 (R)		OS Timer 2 interrupt request	Level
3 (R)	CPU	TCM interrupt request	Level
4 (R)	Reserved	Reserved	Level
5 (R)	Reserved	Reserved	Level
6 (R)	GPSDSP	GPS service request	Level
7 (R)		DSP to RISC interrupt request	Level
8 (R)		FFT interrupt request	Level
9 (R)	Reserved	Reserved	Level
10 (R)	USBOTG	USBOTG interrupt request	Level
11 (R)	DMA Controller	DMA Controller interrupt request	Level
12 (R)	Reserved	Reserved	Level
13 (R)	SPI	Serial Port interrupt request	Level
14 (R)	Reserved	Reserved	Level
15 (R)	UART0	UART0 interrupt request	Level
16 (R)	UART1	UART1 interrupt request	Level
17 (R)	Reserved	Reserved	Level
18 (R)	USP0	USP0 interrupt request	Level
19 (R)	USP1	USP1 interrupt request	Level
20 (R)	SD1	SD controller 1 interrupt request	Level
21 (R)	I^2C	I^2C interrupt request	Level
22 (R)	System	CPU interface interrupt request	Level
23 (R)		SYSARB interrupt request	Level
24 (R)		Memory controller interrupt request	Level
25 (R)		SYS2PCI interrupt request	Level
26 (R)		IOBG interrupt request	Level
27 (R)		PCIARB interrupt request	Level
28 (R)	PMUIRQ	ARM PMU interrupt request	Level
29 (R)	LCD Controller	LCD Controller interrupt request	Level
30 (R)	VPP	VPP interrupt request	Level

Bit	Source Module	Description	Sensitive Type
31 (R)	PWRC	Power controller interrupt request.	Level

Table 132: Interrupt Controller Pending Register 0

- Interrupt Controller Pending Register 1 (INT_PENDING1) – RISC: 0x0004 DSP: 0x02~0x03

Bit	Source Module	Description	Sensitive Type
0 (R)	TSCIF	TSC interface interrupt request	Level
1 (R)	CODEC	CODEC interrupt request	Level
2 (R)	ROMIF	ROMIF interrupt request	Level
3 (R)	Reserved	Reserved	Level
4 (R)	SD0	SD controller 0 interrupt request	Level
5 (R)	PCI_COPY	PCI COPY interrupt request	Level
6 (R)	NAND Flash	NAND flash interrupt request	Level
7 (R)	GPIO	GPIO0 interrupt request	Configurable
8 (R)		GPIO1 interrupt request	Configurable
9 (R)		GPIO2 interrupt request	Configurable
10 (R)		GPIO3 interrupt request	Configurable
11 (R)		GPIO4 interrupt request	Configurable
12 (R)	Timer	OS timer 3 interrupt request	Level
13 (R)		OS timer 4 interrupt request	Level
14 (R)		OS timer 5 interrupt request	Level
15 (R)	RTC	RTC-ALARM 0 interrupt request	Level
16 (R)		RTC-TIC interrupt request	Level
17 (R)		RTC-ALARM 1 interrupt request	Level
18 (R)		GPSRTC-ALARM 0 interrupt request	Level
19 (R)		GPSRTC-ALARM 1 interrupt request	Level
20 (R)		GPSRTC-TIC interrupt request	Level
21 (R)	Reserved	Reserved	Level
31:22	-	Reserved	

Table 133: Interrupt Controller Pending Register 1

- Interrupt Controller IRQ/FIQ Pending Register (INT_IRQ_PENDING0~1/INT_FIQ_PENDING0~1) – RISC: 0x0008~0x000C/0x0010~0x0014

  INT_IRQ_PENDING and INT_FIQ_PENDING both contain one flag per interrupt that indicates an interrupt request has been made by a unit. Inside the interrupt service routine, the INT_IRQ_PENDING and INT_FIQ_PENDING registers are read by RISC to determine the interrupt source. In general, software will then read the status registers within the interrupted device to determine how the interrupt should be serviced.

  The following table lists the bit locations corresponding to the interrupt pending status flags in INT_IRQ_PENDING. Then the following table will show the bit locations corresponding to the interrupt pending status flags in the INT_FIQ_PENDING register. These are read-only registers.

Bit	Name	Default	Description
31:0 (R)	IRQ0<31:0>	32'h0	IRQ pending bits: 0: No interrupt pending 1: Interrupt pending

Table 134: INT_IRQ_PENDING0 – RISC 0x0008

Bit	Name	Default	Description
20:0 (R)	IRQ1<20:0>	21'h0	IRQ pending bits: 0: No interrupt pending 1: Interrupt pending
31:21	-	11'h0	Reserved

Table 135: INT_IRQ_PENDING1 – RISC 0x000C

Bit	Name	Default	Description
31:0 (R)	FIQ0<31:0>	32'h0	FIQ pending bits: 0: No interrupt pending 1: Interrupt pending

Table 136: INT_FIQ_PENDING0 – RISC 0x0010

Bit	Name	Default	Description
20:0 (R)	FIQ1<20:0>	21'h0	FIQ pending bits: 0: No interrupt pending 1: Interrupt pending
31:21	-	11'h0	Reserved

Table 137: INT_FIQ_PENDING1 – RISC 0x0014

- Interrupt Controller RISC Mask Register (INT_RISC_MASK) – RISC: 0x0018~0x001C

  Mask bits serve two purposes. First, they allow periodic software polling of interruptible sources while preventing them from actually causing an interrupt. Second, they allow the interrupt handler routine to prevent interrupts of lower priority from occurring while still maintaining a list of pending interrupts that may have occurred previously or during the servicing of another interrupt.

  The INT_RISC_MASK is not initialized at reset. A question mark indicates that the values are unknown at reset. The following table shows the bit locations corresponding to interrupt mask bits.

Bit	Name	Default	Description
31:0 (W/R)	RM0<31:0>	32'h0	RISC interrupts mask bits: 0: Pending interrupt is masked from becoming active. 1: Pending interrupt is allowed to become active.

Table 138: INT_RISC_MASK0 – RISC 0x0018

Bit	Name	Default	Description
20:0 (W/R)	RM1<20:0>	21'h0	RISC interrupts mask bits: 0: Pending interrupt is masked from becoming active. 1: Pending interrupt is allowed to become active.
31:21		11'h0	Reserved.

Table 139: INT_RISC_MASK1 – RISC: 0x001C

- Interrupt RISC LEVEL Register (INT_RISC_LEVEL0/INT_RISC_LEVEL1>) – RISC: 0x0020/0x0024

Bit	Name	Default	Description
31:0 (W/R)	RM0<31:0>	32'h0	RISC interrupts level bits: 0: The interrupt level is IRQ. 1: The interrupt level is FIQ.

Table 140: INT_RISC_LEVEL0 – RISC: 0x0020

Bit	Name	Default	Description
20:0 (W/R)	RM1<20:0>	21'h0	RISC interrupts level bits: 0: The interrupt level is IRQ. 1: The interrupt level is FIQ.
31:21		11'h0	Reserved.

Table 141: INT_RISC_LEVEL1 – RISC: 0x0024

- Interrupt Controller DSP Mask Register 0(INT_DSP_MASK0) – RISC: 0x002C, DSP: 0x14~0x15

Bit	Name	Default	Description
31:0 (R)	DM0<31:0>	32'h0	DSP interrupts mask bits: 0: Pending interrupt is masked from becoming active. 1: Pending interrupt is allowed to become active.

Table 142: Interrupt Controller DSP Mask Register 0

- Interrupt Controller DSP Mask Register 1(INT_DSP_MASK1) – RISC: 0x002C, DSP: 0x16~0x17
  The definition of each bit is the same as the above.

Bit	Name	Default	Description
20:0 (W/R)	DM1<20:0>	21'h0	DSP interrupts mask bits: 0: Pending interrupt is masked from becoming active. 1: Pending interrupt is allowed to become active.
31:21		11'h0	Reserved.

Table 143: Interrupt Controller DSP Mask Register 1

- Interrupt DSP Access Enable Register (INT_DSP_ACCEN) – RISC: 0x0030, DSP: 0x18
  By default, all the register can only be written to RISC only. In order for DSP to write the INT_DSP_MASK register, INT_DSP_ACCEN register should be set by RISC.

Bit	Name	Default	Description
0 (W/R)	DSP_EN	1'b0	DSP Access enable
31:1	-	31'h0	Reserved

Table 144: Interrupt DSP Access Enable Register

- Chip ID Register (CHIP_ID) – RISC: 0x0034, DSP: 0x020
  This register is read-only, it contain the CHIP ID for different chip versions.

Bit	Name	Default	Description
15:0 (R)	CHIP_ID<15:0>	16'h1230	CHIP ID
31:16	-	-	Reserved

Table 145: Chip ID Register

- Interrupt Controller ID Register (INT_ID_IRQ) – RISC: 0x0038

  The Interrupt Controller IRQ ID register returns the IRQ interrupt ID from the pending interrupts. Each time RISC reads this register, it will return the highest priority enabled pending interrupt.

Bit	Name	Default	Description
7:0	INT_ID_IRQ<7:0>	8'hff	Interrupt ID of IRQ
31:8	-	-	Reserved

Table 146: Interrupt Controller ID Register

- Interrupt Controller ID Register (INT_ID_FIQ) – RISC: 0x003C

  The Interrupt Controller FIQ ID register returns the FIQ interrupt ID from the pending interrupts. Each time RISC reads this register, it will return the highest priority enabled pending interrupt.

Bit	Name	Default	Description
7:0	INT_ID_FIQ<7:0>	8'hff	Interrupt ID of FIQ
31:16	-	-	Reserved

Table 147: Interrupt Controller ID Register

The following table provides detailed information about the priority and descriptions for each ID (the smaller the ID value, the higher the interrupt priority).

ID Value	Description
0	Interrupt from TIMER0
1	Interrupt from TIMER1
2	Interrupt from TIMER2
3	Interrupt from TCM
4	Reserved
5	Reserved
6	Interrupt from GPS
7	Interrupt from DSP to RISC
8	Interrupt from FFT
9	Reserved
10	Interrupt from USB
11	Interrupt from DMA
12	Reserved

ID Value	Description
13	Interrupt from SPI
14	Reserved
15	Interrupt from UART0
16	Interrupt from UART1
17	Reserved
18	Interrupt from USP0
19	Interrupt from USP1
20	Interrupt from SD controller 1
21	Interrupt from I^2C
22	Interrupt from RISC_I/F
23	Interrupt from SYSARB
24	Interrupt from MEM_CTRL
25	Interrupt from SYS2PCI
26	Interrupt from IOBG
27	Interrupt from PCIARB
28	Interrupt from CPU_PMU
29	Interrupt from LCD
30	Interrupt from VPP
31	Interrupt from PWRC
32	Interrupt from TSCIF
33	Interrupt from Codec
34	Interrupt from ROMIF
35	Reserved
36	Interrupt from SD controller 0
37	Interrupt from PCI_COPY
38	Interrupt from NAND
39	Interrupt from GPIO0
40	Interrupt from GPIO1
41	Interrupt from GPIO2
42	Interrupt from GPIO3
43	Interrupt from GPIO4

ID Value	Description
44	Interrupt from Timer 3
45	Interrupt from Timer 4
46	Interrupt from Timer 5
47	Interrupt from RTC-ALARM 0
48	Interrupt from RTC-TIC
49	Interrupt from RTC-ALARM 1
50	Interrupt from GPSRTC- ALARM 0
51	Interrupt from GPSRTC-ALARM 1
52	Interrupt from GPSRTC-TIC
53	Reserved

Table 148: Interrupt Priority and ID Table

INT_PENDING, INT_IRQ_PENDING, INT_FIQ_PENDING, INT_DSP_PENDING are all read-only registers, they only perform combinational logic of interrupt inputs from each internal block. To clear the interrupt, the interrupt in the block needs to be cleared, which will also generate an interrupt. These registers will change as soon as the interrupt signal from the internal block is changed.

The INT_FIQ_PENDING register is created by INT_PENDING, INT_RISC_MASK, and INT_RISC_LEVEL.

If the value in other registers is changed, then INT_ FIQ _PENDING will also be changed. This requires special attention from the user. While in an interrupt routine a user writes the INT_RISC_MASK register to mask the interrupt from certain blocks, INT_FIQ _PENDING will also be changed to 0. As a result, users may not be able to read this register to determine interrupt status. Instead users should read the INT_PENDING register.

The INT_IRQ_PENDING register is created by INT_PENDING, INT_RISC_MASK, and ~INT_RISC_LEVEL.

If the value in other registers is changed, then INT_IRQ_PENDING will also be changed. This requires special attention from the user in that while in an interrupt routine a user writes the INT_RISC_MASK register to mask the interrupt from certain blocks, INT_IRQ_PENDING will also be changed to 0. As a result, users may not be able to read this register to determine the interrupt status. Instead users should read the INT_PENDING register.

OS Timer

Overview

SiRFatlasV provides six OS timers which can be programmed independently. These OS timers are generated by a 64-bit counter (TIMER_COUNTER), which is clocked by a non-system clock. The frequency of this clock can be set by configuring the TIMER_DIV register. If the value of TIMER_COUNTER matches any one of the 6 timers, then an interrupt may be generated to the RISC if the corresponding enable bit is set.

The 6th timer can also act as a Watchdog timer when the Watchdog mode is enabled. In this mode, when this timer matches TIMER_COUNTER value (lower 32 bits), a watchdog event will be generated to reset most of the on-chip modules.

Feature List

- Six programmable OS timers
- 64-bit timer counter
- Watchdog timer

Register Definitions

Register Address Mapping

Access Type	Address Mapping
RISC I/O Interface	0x90050000

Table 149: OS Timer Base Address

RISC Address <11:0>	Register	Description
0x0000	TIMER_COUNTER_LO	OS Timer Counter Low Register
0x0004	TIMER_COUNTER_HI	OS Timer Counter High Register
0x0008	TIMER_MATCH_0	OS Timer Match Register 0
0x000C	TIMER_MATCH_1	OS Timer Match Register 1
0x0010	TIMER_MATCH_2	OS Timer Match Register 2
0x0014	TIMER_MATCH_3	OS Timer Match Register 3
0x0018	TIMER_MATCH_4	OS Timer Match Register 4
0x001C	TIMER_MATCH_5	OS Timer Match Register 5
0x0020	TIMER_STATUS	OS Timer Status Register
0x0024	TIMER_INT_EN	OS Timer Interrupt Enable Register
0x0028	TIMER_WATCHDOG_EN	OS Timer Interrupt Enable Register
0x002C	TIMER_DIV	OS Timer Division Register
0x0030	TIMER_LATCH	OS Timer Latch Register
0x0034	TIMER_LATCHED_LO	OS Timer Latched Low Register
0x0038	TIMER_LATCHED_HI	OS Timer Latched High Register

Table 150: OS Timer Register Mapping

Register Descriptions

- OS Timer Counter Low Register (TIMER_COUNTER_LO) – 0x0000

  The low register of the OS timer counter is the low 32-bit of the 64-bit-counter that increments on rising edges of the timer clock. This counter can be read or written at any time. It is recommended that this register be set to write-protected through the MMU protection mechanisms.

Bit	Name	Default	Description
31:0 (R/W)	CN<31:0>	32'h0	OS Timer counter low 32-bit value

Table 151: OS Timer Counter Low Register

- OS Timer Counter High Register (TIMER_COUNTER_HI) – 0x0004

  The OS timer count high register is the high 32-bit of the 64-bit-counter that increments on rising edges of the timer clock. This counter can be read or written at any time. It is recommended that this register be set to write-protected through the MMU protection mechanisms.

Bit	Name	Default	Description
31:0 (R/W)	CN<63:32>	32'h0	OS Timer counter high 32-bit value

Table 152: OS Timer Counter High Register

- OS Timer Match Register 0-5 (TIMER_MATCH_x) – 0x0008~0x001C

  These registers are 32 bits wide and are accessible by the RISC. They are compared with the lower 32-bit of TIMER_COUNTER following every timer clock cycle. If any of these registers matches the counter, then the corresponding status bit in the TIMER_STATUS will be set.

Bit	Name	Default	Description
31:0 (R/W)	MA<31:0>	32'h0	OS Timer match value

Table 153: OS Timer Match Register 0-5

- OS Timer Status Register (TIMER_STATUS) – 0x0020

  These bits are set when the Timer match event occurs and cleared by writing a one to it. Writing zeros to this register produces no effects. All reserved bits are read as zeros and are unaffected by the write operation.

Bit	Name	Default	Description
0 (R/W)	M0	1'b0	Match status channel 0. 0: The OS timer match register<0> has not matched the OS timer counter since it was last cleared. 1: The OS timer match register<0> has matched the OS timer counter.
1 (R/W)	M1	1'b0	Match status channel 1. 0: The OS timer match register<1> has not matched the OS timer counter since it was last cleared. 1: The OS timer match register<1> has matched the OS timer counter.
2 (R/W)	M2	1'b0	Match status channel 2. 0: The OS timer match register<2> has not matched the OS timer counter since it was last cleared. 1: The OS timer match register<2> has matched the OS timer counter.
3 (R/W)	M3	1'b0	Match status channel 3. 0: The OS timer match register<3> has not matched the OS timer counter since it was last cleared. 1: The OS timer match register<3> has matched the OS timer counter.
4 (R/W)	M4	1'b0	Match status channel 4. 0: The OS timer match register<4> has not matched the OS timer counter since it was last cleared. 1: The OS timer match register<4> has matched the OS timer counter.
5 (R/W)	M5	1'b0	Match status channel 5. 0: The OS timer match register<5> has not matched the OS timer counter since it was last cleared. 1: The OS timer match register<5> has matched the OS timer counter.
31:6	-	26'h0	Reserved.

Table 154: OS Timer Status Register

- OS Timer Interrupt Enable Register (TIMER_INT_EN) – 0x0024

  Each match register has a corresponding enable bit. Clearing an enable bit will prevent the corresponding interrupt status bit being generated in TIMER_STATUS register.

Bit	Name	Default	Description
0 (R/W)	E0	1'b0	Interrupt enable channel 0. This bit is set by the software and allows a match between match register 0 and the OS timer to assert interrupt bit M0 in TIMER_STATUS.
1 (R/W)	E1	1'b0	Interrupt enable channel 1
2 (R/W)	E2	1'b0	Interrupt enable channel 2
3 (R/W)	E3	1'b0	Interrupt enable channel 3
4 (R/W)	E4	1'b0	Interrupt enable channel 4
5 (R/W)	E5	1'b0	Interrupt enable channel 5
31:6	-	26'h0	Reserved

Table 155: OS Timer Interrupt Enable Register

- OS Timer Watchdog Enable Register (TIMER_WATCHDOG_EN) – 0x0028

Bit	Name	Default	Description
0 (R/W)	WME1	1'b0	Watchdog match enable. 0: The OS timer match register<5> does not cause a watchdog reset. 1: The OS timer match register<5> causes a reset.
31:1	-	31'h0	Reserved

Table 156: OS Timer Watchdog Enable Register

- OS Timer Division Register (TIMER_DIV) – 0x002C

  To generate the timer clock, the clock input needs to be divided from the system clock domain.

  TIMER_DIV = (SYS_CLK/TIMER_CLK)/2 –1

Bit	Name	Default	Description
15:0 (R/W)	DIV<15:0>	16'h0	OS timer clock division
31:16	-	16'h0	Reserved

Table 157: Timer Division Register

- OS Timer Latch Register (TIMER_LATCH) – 0x0030

  When user writes 1'b1 into this register bit, the value of the current OS timer counter will be latched into TIMER_LATCHED_LO and TIMER_LATCHED_HI.

Bit	Name	Default	Description
0 (W)	LATCH	1'b0	OS Timer Counter latch
31:1 (R/W)	-	31'h0	Reserved

Table 158: OS Timer Latch Register

- OS Timer Latched Low Register (TIMER_LATCHED_LO) – 0x0034

  The OS timer latched low register is the low 32-bit value of the 64-bit counter that latched when users write a 1'b1 into the TIMER_LATCH register.

Bit	Name	Default	Description
31:0 (R/W)	LA<31:0>	32'h0	The low 32-bit latched value

Table 159: OS Timer Latched Low Register

- OS Timer Latched High Register (TIMER_LATCHED_HI) – 0x0038

  The OS timer latched high register is the high 32-bit value of the 64-bit counter that latched when users write a 1'b1 into the TIMER_LATCH register.

Bit	Name	Default	Description
31:0 (R/W)	LA<63:32>	32'h0	The high 32-bit latched value

Table 160: OS Timer Latched High Register

RTC

Overview

System RTC and GPS RTC are provided in SiRFatlasV, but the two timers are for different purposes and have different precision requirements.

Feature List

- RISC/DSP I/O interface
- Different power modes supported: normal, deep sleep, hibernation
- RTC: SYS_RTC, GPS_RTC

Pin Description

External Pin Descriptions

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
XINW	Input	-	-	-	RTC clock input
XOUTW	Output	-	-	-	RTC clock output
X_RTC_RST_B	Input	-	-	-	System RTC will be reseted by this signal
X_RESET_B	Input	-	-	-	GPS RTC will be reseted by this signal

Table 161: External Pins of RTC

Functional Descriptions

Module Descriptions

System RTC

There is a programmable 16-bit divider (RTC_DIV) that divides the input 32.768KHz clock to the frequency that users need (E.g. 1 Hz). The divided real-time clock will be used to drive a 32-bit counter (RTC_COUNTER) that provides users with the actual time.

In each cycle of the divided real-time clock, there is a Hertz interrupt generated to the RISC. Users can also configure an alarm (RTC_ALARM). When RTC_COUNTER matches the alarm, there will be an alarm interrupt generated to the RISC.

The system RTC can generate an alarm wake-up signal to notify the power controller to wake up from deep-sleep/hibernation mode.

GPS RTC

There are 3 stages for GPS RTC block:

- Clock source select
- Clock divider
- Counting

There is a 2-to-1 MUX for selecting different clock sources:

- 32.768KHz input
- TCXO input from GPS RF

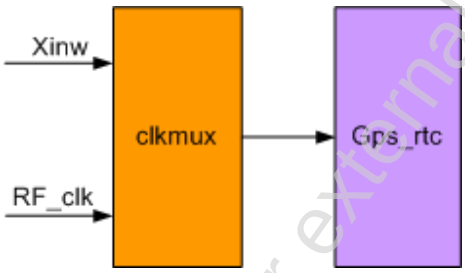


Figure 13: Clock Source MUX

There are two register bits for MUX to choose a clock source and the register bits are configured by software. Configuration will vary according to the scenario adopted in the system.

A clock divider follows the 2-to-1 MUX. The purpose of the clock divider is to divide the clock output of the 2-to-1 MUX to a lower frequency clock so that it can be used by the internal logic of GPS RTC. The divider is also a 16-bit programmable register. Different sources have different dividers, and different duty cycles for GPS may also have different divider configurations.

The GPS RTC function is described as follows:

- General requirements:

  The divided clock is used to drive a 32-bit counter that provides the GPS sub-system with an accurate time.

  GPS RTC may also be used as system RTC, it needs to implement all function required by the system RTC. In each cycle of the divided real-time clock, there is a Hertz interrupt generated to the RISC. For configuring an alarm, use AGPS_RTC_ALARM. When AGPS_RTC_COUNTER matches the alarm, there will be an alarm interrupt generated to the RISC. The interrupt to RISC can also be masked off.

- Special requirements:

  GPS RTC register can be accessed both by RISC and DSP.

  The GPS baseband is generating interrupts at every 1ms while the GPS is running. There is a 32-bit register in the GPS RTC block which can be used to latch the GPS RTC timer counter when

the baseband interrupt is generated. The 1ms interrupt may exist for several cycles. The latched data should be from the 1st cycle in which the interrupt is asserted.

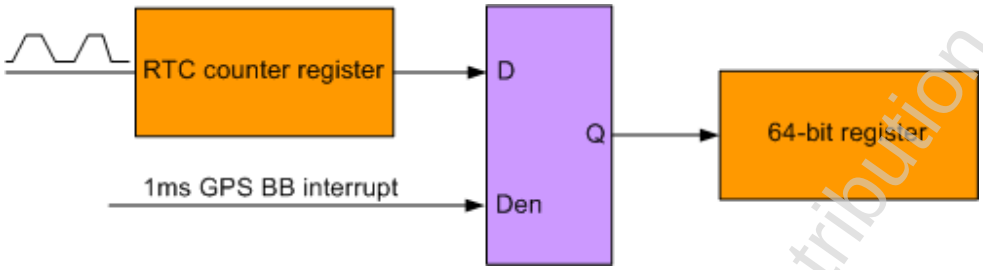


Figure 14: GPS Diagram for Counter Saving

Function Modes

- Normal mode:
  Both the SYS_RTC and the GPS_RTC module work in power-on state.
- Deep-sleep and hibernation mode:
  SYS_RTC module is powered on.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
SYS_RTC register base address through RISC I/O	0x90068000
GPS_RTC register base address through RISC I/O	0x90064000
SYS_RTC register base address through DSP I/O	0xa0
GPS_RTC register base address through DSP I/O	0x80

Table 162: RTC Interface Address Mapping

Register Mapping

RISC Address <11:0>	DSP Address <11:0>	Register	Description
SYS_RTC			
0x0000	0x01~0x00	RTC_COUNTER	RTC counter register
0x0004	0x03~0x02	RTC_ALARM0	RTC alarm0 register
0x0008	0x05~0x04	RTC_STATUS	RTC status register
0x000C	0x07~0x06	RTC_DIV	RTC division register
0x0010	-	-	Reserved
0x0014	-	-	Reserved
0x0018	0x0d~0x0c	RTC_ALARM1	RTC alarm1 register
0x001c	0x0f~0x0e	RTC_CLOCK_SWITCH	RTC clock switch
GPS_RTC			
0x0000	0x01~0x00	RTC_GPS_COUNTER	RTC counter register for GPS
0x0004	0x03~0x02	RTC_GPS_ALARM	RTC alarm register for GPS
0x0008	0x05~0x04	RTC_GPS_STATUS	RTC status register for GPS
0x000C	0x07~0x06	RTC_GPS_DIV	RTC division register for GPS
0x0010	0x09~0x08	RTC_GPS_LATCH	RTC GPS latch register for GPS
0x0014	-	-	Reserved
0x0018	0x0d~0x0c	RTC_GPS_ALARM1	RTC alarm1 register
0x001c	0x0f~0x0e	RTC_GPS_CLOCK_SWITCH	RTC clock switch
CPU_IO_BRIDGE			
0x0000	-	CPUIOBG_BE_SYNCING	RISC I/O Bridge syncing register
DSP IO BRIDGE			
-	0x00	DSPIOBG_BE_SYNCING	DSP I/O Bridge syncing register

Table 163: RTC Register Mapping

The register definition for RTC of GPS is the same as the system RTC from RTC_GPS_COUNTER to RTC_GPS_DIV.

All the SYSRTC register has no default value since SYSRTC has no any reset source, and for GPSRTC the registers only can be reset by X_RESET_B.

Register Descriptions

- RTC Counter Register (RTC_COUNTER) – 0x0000

  The RTC counter register (RTC_COUNTER) is a read/write register. The counter may be written by the RISC at any time.

NOTE – Because of ASYNC, the counter register read operation has some limitations. The SYS_CLK frequency should be eight times the counter toggle rate. For example: the GPSRTC uses RF_CLK (16.368MHZ) as clock source, and the divide register is set to 0, which means the lowest frequency of SYS_CLK is (16.368/2)*8 = 65.472 MHZ.

Bit	Name	Default	Description
31:0 (R/W)	CN<31:0>	32'h0	Real-time Counter value

Table 164: RTC Counter Register

- RTC Alarm0 Register (RTC_ALARM0) – 0x0004

  The real-time clock alarm register is a 32-bit register accessible by RISC.

  In each cycle of the divided real-time clock, this register is compared to RTC_COUNTER. If they are matched and the enable bit (ALE) is set, then the alarm bit in the RTC status register will be set.

Bit	Name	Default	Description
31:0 (R/W)	AL<31:0>	32'h0	Alarm0 value

Table 165: RTC Alarm0 Register

- RTC Status Register (RTC_STATUS) – 0x0008

  Writing 1's to AL and HZ will clear the bits.

Bit	Name	Default	Description
0 (R/W)	AL0	1'b0	RTC alarm0 detected: 0: No alarm has been detected. 1: An alarm has been detected (RTC Counter matches RTC_ALARM). The interrupt is also a wake-up source.
1 (R/W)	HZ	1'b0	1 Hz rising-edge detected: 0: No rising edges have been detected. 1: A rising edge has been detected.
2 (R/W)	AL0E	1'b0	RTC alarm0 interrupt enable: 0: RTC alarm interrupt is not enabled. 1: RTC alarm interrupt is enabled.
3 (R/W)	HZE	1'b0	1 Hz interrupt enable:

Bit	Name	Default	Description
			0: The 1-Hz interrupt is not enabled. 1: The 1-Hz interrupt is enabled.
4 (R/W)	AL1	1'b0	RTC alarm1 detected: 0: No alarm has been detected. 1: An alarm has been detected (RTC Counter matches RTC_ALARM). The interrupt is also a wake-up source.
5	-	-	Reserved
6 (R/W)	AL1E	1'b0	RTC alarm1 interrupt enable: 0: RTC alarm interrupt is not enabled. 1: RTC alarm interrupt is enabled.
31:7	-	-	Reserved.

Table 166: RTC Status Register

- RTC Division Register (RTC_DIV) – 0x000C

  To generate a 1-Hz divided real-time clock, the clock input needs to be divided from the external real-time crystal (RTC_CLK=32.768 KHz):

  RTC_DIV = RTC_CLK/2 – 1

  To generate a real-time clock in frequency other than 1Hz, for example, X-Hz, the register needs to be set up as the following:

  RTC_DIV = (RTC_CLK/X)/2 – 1

NOTE – The divide register will not be set to smaller than 16 when using RF_CLK as GPSRTC clock source. Because when the division is too small, the total counter time becomes very short.

Bit	Name	Default	Description
15:0 (R/W)	DIV<15:0>	16'h0	Division value
31:16	-	16'h0	Reserved

Table 167: RTC Division Register

- RTC Latch Register (RTC_GPS_LATCH) – 0x0010

  RTC_GPS_COUNTER will be latched once it detects a GPS baseband 1ms interrupt. This register is only valid for GPS RTC.

Bit	Name	Default	Description
31:0 (R)	LATCH<31:0>	32'h0	Latch value

Table 168: RTC Latch Register

- RTC Alarm1 Register (RTC_ALARM) – 0x00018

  The real-time clock alarm register is a 32-bit register accessible by RISC.

  In each cycle of the divided real-time clock, this register is compared to RTC_COUNTER. If they are matched and the enable bit (ALE) is set, then the alarm bit in the RTC status register will be set.

Bit	Name	Default	Description
31:0 (R/W)	AL<31:0>	32'h0	Alarm1 value

Table 169: RTC Alarm1 Register

- RTC Clock Switch Control Register (RTC_CLK_CTRL) – 0x001c

  The register is only for GPS RTC counter clock control.

Bit	Name	Default	Description
1:0 (R/W)	CLK_SWITCH	2'h3	GPSRTC: 0: Reserved 1: RF_CLK 2: Reserved 3: RTC_CLK (32khz)
2 (R/W)	CLK_EN	1'h1	GPSRTC: 0: Clock disable 1: Clock enable
31:3	-	-	Reserved

Table 170: RTC Clock Switch Control Register

RTC I/O Bridge

Overview

The module is a bridge between the RTC clock domain and the system/DSP clock domain.

Feature List

- I/O interface
- Modules can be divided according to clock domain
- Arbitrary clock frequency

Functional Descriptions

Block Diagram

The RTC I/O bridge module is a bridge between the RTC clock domain and the system/DSP clock domain. It is used to synchronize the I/O bus signal, and is combined by FASTEND and SLOWEND 2 modules to separate two different clock domains.

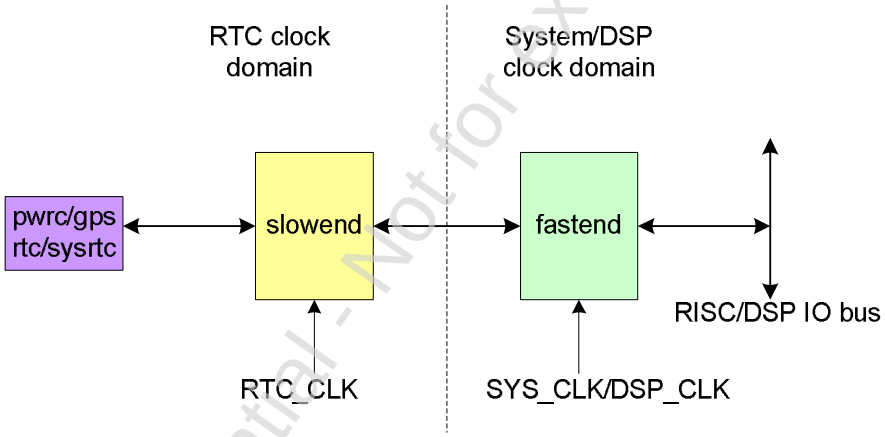


Figure 15: Block Diagram of the RTC I/O Bridge

Module Descriptions

FASTEND

The module is an interface with system/DSP clock domain I/O bus interface, which has two primary functions, one is for generating handshake signals, and the other is for latching the value to the fast clock domain in order to ensure that the slow clock domain can sample them.

SLOWEND

The module is an interface with the I/O bus interface of the RTC clock domain. It provides two primary functions, one for responding to handshake signals, the other one for interfacing with the I/O interface of the RTC clock domain.

Flowchart for Accessing RTC Clock Domain

For SYSRTC, GPSRTC, and PWRC are in the RTC clock domain, they have a different access method from other modules.

1. Check RTC I/O Bridge is idle or not
2. If the bridge is idle, software can read/write RTC domain register
3. Check if the I/O bridge is idle or not. If idle, the operation is done; if not, the operation is processing.

NOTE – The 3 steps should be contained in one critical section.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O Base Address	0x90040000
DSP I/O Base Address	0xe0

Table 171: Sample Base Address

Register Mapping

Address <15:0>	Register	Description
0x0000	BE_SYNCING	I/O Bridge busy flag

Table 172: Address Mapping

Register Descriptions

- RTC I/O Bridge SYNC Flag (BE_SYNCING) – Offset: 0x0

Bit	Name	Default	Description
0 (R)	BE_SYNCING	1'b0	The value is bridge busy flag: 1: Busy 0: Idle
31:1	-	31'b0	Reserved

Table 173: RTC IOBG BE_SYNCING REG

GPIO

Overview

The GPIO is a programmable, general purpose I/O controller which is used to implement functions not implemented with dedicated controllers and only requiring simple input or output signals.

In SiRFatlasV, there is a total of 55 GPIOs; each GPIO pin can be configured as input or output independently. When the GPIO is configured as input, it can also be enabled as an interrupt source (either edge or level triggered). For the GPIO pin pull function, all the GPIO can be configured with or without the pull independently.

In SiRFatlasV, five GPIO groups are reserved. In each GPIO group, there is one control register for each GPIO, one pad enable register for each GPIO group, one interrupt status register for each GPIO group and one DSP enable control register for GIOP group 0. All GPIO pins are allocated into different groups.

NOTE – Only group0 can be accessed by the DSP interface.

Feature List

- GPIO programmed as input can be the interrupt source for CPU or DSP (group 0).
- Rise edge, fall edge, as well as low-level and high level interrupt are all supported.

Pin Description

Pin MUX of GPIO

The following table shows the pin MUX of each GPIO pin:

Group #	GPIO #	Pin Name
0	15~0	X_GPIO<15:0>
	16	RESERVED
	17	RESERVED
	18	X_USCLK_0
	19	X_UTXD_0
	20	X_URXD_0
	21	X_UTFS_0
	22	X_URFS_0
	23	X_USCLK_1
	24	X_UTXD_1
	25	X_URXD_1
	26	X_UTFS_1

Group #	GPIO #	Pin Name
	27	X_URFS_1
	28	RESERVED
	29	RESERVED
	30	RESERVED
	31	RESERVED
1	0	X_SD_DAT_3[2]
	1	X_SD_DAT_3[3]
	2	X_SD_CLK_3
	3	X_SD_CMD_3
	4	RESERVED
	5	RESERVED
	6	RESERVED
	7	X_SD_CD_B_1
	8	X_SD_VCC_ON_1
	9	X_SD_WP_B_1
	10	X_SD_CLK_1
	11	X_SD_CMD_1
	12	X_SD_DAT_1[0]
	13	X_SD_DAT_1[1]
	14	X_SD_DAT_1[2]
	15	X_SD_DAT_1[3]
	16	RESERVED
	17	RESERVED
	18	RESERVED
	19	RESERVED
	20	X_CKO_1
	21	RESERVED
	22	RESERVED
	23	RESERVED
	24	RESERVED

Group #	GPIO #	Pin Name
	25	RESERVED
	26	RESERVED
	27	RESERVED
	28	X_SD_DAT_3[0]
	29	X_SD_DAT_3[1]
	30	RESERVED
	31	RESERVED
2	0	RESERVED
	1	RESERVED
	2	RESERVED
	3	RESERVED
	4	RESERVED
	5	RESERVED
	6	RESERVED
	7	RESERVED
	8	RESERVED
	9	RESERVED
	10	RESERVED
	11	RESERVED
	12	RESERVED
	13	RESERVED
	14	RESERVED
	15	RESERVED
	16	RESERVED
	17	RESERVED
	18	RESERVED
	19	RESERVED
	20	RESERVED
	21	RESERVED
	22	RESERVED

Group #	GPIO #	Pin Name
	23	RESERVED
	24	RESERVED
	25	RESERVED
	26	RESERVED
	27	RESERVED
	28	RESERVED
	29	RESERVED
	30	RESERVED
	31	RESERVED
3	0	RESERVED
	1	RESERVED
	2	RESERVED
	3	RESERVED
	4	RESERVED
	5	RESERVED
	6	RESERVED
	7	RESERVED
	8	RESERVED
	9	RESERVED
	10	RESERVED
	11	RESERVED
	12	RESERVED
	13	RESERVED
	14	RESERVED
	15	X_RXD_0
	16	X_RXD_1
	17	RESERVED
	18	X_SDA_0
	19	X_SCL_0
	20	X_GPS_SGN

Group #	GPIO #	Pin Name
	21	X_GPS_MAG
	22	X_GPS_SAMPLE_CLK
	23	X_L_DE
	24	RESERVED
	25	RESERVED
	26	RESERVED
	27	RESERVED
	28	RESERVED
	29	RESERVED
	30	RESERVED
	31	RESERVED
4	0	RESERVED
	1	RESERVED
	2	RESERVED
	3	RESERVED
	4	RESERVED
	5	RESERVED
	6	RESERVED
	7	RESERVED
	8	RESERVED
	9	RESERVED
	10	RESERVED
	11	RESERVED
	12	RESERVED
	13	RESERVED
	14	RESERVED
	15	RESERVED
	16	RESERVED
	17	RESERVED
	18	RESERVED

Group #	GPIO #	Pin Name
	19	RESERVED
	20	RESERVED
	21	RESERVED
	22	RESERVED
	23	RESERVED
	24	X_AC97_BIT_CLK
	25	X_AC97_DOUT
	26	X_AC97_DIN
	27	X_AC97_SYNC
	28	X_GPS_CLK
	29	RESERVED
	30	RESERVED
	31	RESERVED

Table 174: GPIO Pin MUX

Functional Descriptions

Block Diagram

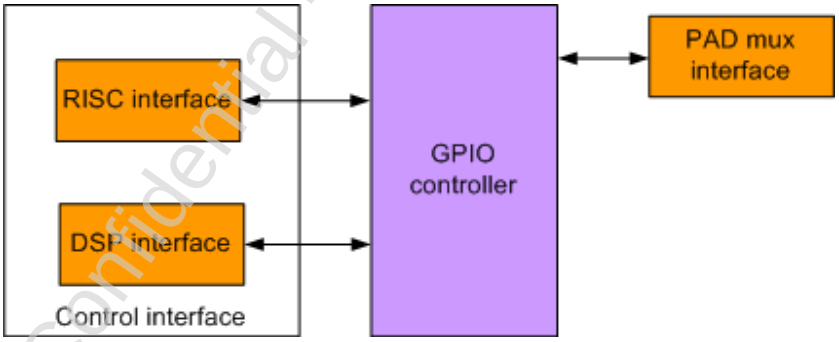


Figure 16: GPIO Block Diagram

Register Definitions

The GPIO Pin Output Enable bit (OUT_EN) is used to set the GPIO pin to output or input. When programmed as an output, the pin can be set high by writing 1 to the corresponding bit in the GPIO Pin Data output bit (OUT) and set low by writing 0 to the same register. The GPIO Pin Data output register can be written regardless of its I/O configuration.

If a pin is configured as an input, the pin becomes output when it is reconfigured as output. The GPIO pin state can be validated by reading the GPIO Pin data input bit (DATA_IN). This register can be read at any time even it is in output state.

The GPIO pull control register (PULL_EN, PULL_TYPE) is used to configure the GPIO input pin to determine its status. Please refer to the register description.

GPIO supports two types of interrupts: edge and level triggered. You can set the GPIO interrupt type bit (INT_TYPE) to choose an interrupt type. To enable the interrupt, use the GPIO interrupt enable bit (RISC_INT_EN / DSP_INT_EN). To set the interrupt type defined in GPIO_INT_TYPE(high-level/rising-edge), use the GPIO high-level trigger bit (INT_HT). To set low-level or falling edge interrupt type defined in GPIO_INT_TYPE, use the GPIO low-level trigger bit (INT_LT). The GPIO interrupt status bit (INT_STATUS) is used to read interrupt status from the GPIO.

The GPIO DSP access enable register (GPIO_DSP_EN) is used to define whether a GPIO is accessed by DSP or by RISC. Only GPIO group 0 can be accessed by DSP.

Each GPIO pin has a control register, a GPIO PAD enable register that controls PAD MUX, and an interrupt status register.

Register Address Mapping

Base Address

Access Type	Address Mapping
GPIO register base address through RISC I/O	0x80090000
GPIO register base address through DSP I/O (group 0 only)	0x200

Table 175: GPIO Controller Base Address Mapping

Register Mapping

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x000	0x00	GPIO0_CTRL0	GPIO control for GPIO0 bit0
0x004	0x01	GPIO0_CTRL1	GPIO control for GPIO0 bit1
0x008	0x02	GPIO0_CTRL2	GPIO control for GPIO0 bit2
0x00C	0x03	GPIO0_CTRL3	GPIO control for GPIO0 bit3
0x010	0x04	GPIO0_CTRL4	GPIO control for GPIO0 bit4
0x014	0x05	GPIO0_CTRL5	GPIO control for GPIO0 bit5
0x018	0x06	GPIO0_CTRL6	GPIO control for GPIO0 bit6
0x01C	0x07	GPIO0_CTRL7	GPIO control for GPIO0 bit7
0x020	0x08	GPIO0_CTRL8	GPIO control for GPIO0 bit8

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x024	0x09	GPIO0_CTRL9	GPIO control for GPIO0 bit9
0x028	0x0A	GPIO0_CTRL10	GPIO control for GPIO0 bit10
0x02C	0x0B	GPIO0_CTRL11	GPIO control for GPIO0 bit11
0x030	0x0C	GPIO0_CTRL12	GPIO control for GPIO0 bit12
0x034	0x0D	GPIO0_CTRL13	GPIO control for GPIO0 bit13
0x038	0x0E	GPIO0_CTRL14	GPIO control for GPIO0 bit14
0x03C	0x0F	GPIO0_CTRL15	GPIO control for GPIO0 bit15
0x040	0x10	GPIO0_CTRL16	Reserved
0x044	0x11	GPIO0_CTRL17	Reserved
0x048	0x12	GPIO0_CTRL18	GPIO control for GPIO0 bit18
0x04C	0x13	GPIO0_CTRL19	GPIO control for GPIO0 bit19
0x050	0x14	GPIO0_CTRL20	GPIO control for GPIO0 bit20
0x054	0x15	GPIO0_CTRL21	GPIO control for GPIO0 bit21
0x058	0x16	GPIO0_CTRL22	GPIO control for GPIO0 bit22
0x05C	0x17	GPIO0_CTRL23	GPIO control for GPIO0 bit23
0x060	0x18	GPIO0_CTRL24	GPIO control for GPIO0 bit24
0x064	0x19	GPIO0_CTRL25	GPIO control for GPIO0 bit25
0x068	0x1A	GPIO0_CTRL26	GPIO control for GPIO0 bit26
0x06C	0x1B	GPIO0_CTRL27	GPIO control for GPIO0 bit27
0x070	0x1C	GPIO0_CTRL28	Reserved
0x074	0x1D	GPIO0_CTRL29	Reserved
0x078	0x1E	GPIO0_CTRL30	Reserved
0x07C	0x1F	GPIO0_CTRL31	Reserved
0x080	-	GPIO0_DSP_EN	GPIO group0 DSP control enable
0x084	-	GPIO0_PAD_EN	GPIO group0 PAD enable
0x088	-	-	Reserved
0x08C	-	GPIO0_INT_STATUS	GPIO group0 interrupt status for RISC
0x90~0xFC	-	-	Reserved
0x100	-	GPIO1_CTRL0	GPIO control for GPIO1 bit0

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x104	-	GPIO1_CTRL1	GPIO control for GPIO1 bit1
0x108	-	GPIO1_CTRL2	GPIO control for GPIO1 bit2
0x10C	-	GPIO1_CTRL3	GPIO control for GPIO1 bit3
0x110	-	GPIO1_CTRL4	Reserved
0x114	-	GPIO1_CTRL5	Reserved
0x118	-	GPIO1_CTRL6	Reserved
0x11C	-	GPIO1_CTRL7	GPIO control for GPIO1 bit7
0x120	-	GPIO1_CTRL8	GPIO control for GPIO1 bit8
0x124	-	GPIO1_CTRL9	GPIO control for GPIO1 bit9
0x128	-	GPIO1_CTRL10	GPIO control for GPIO1 bit10
0x12C	-	GPIO1_CTRL11	GPIO control for GPIO1 bit11
0x130	-	GPIO1_CTRL12	GPIO control for GPIO1 bit12
0x134	-	GPIO1_CTRL13	GPIO control for GPIO1 bit13
0x138	-	GPIO1_CTRL14	GPIO control for GPIO1 bit14
0x13C	-	GPIO1_CTRL15	GPIO control for GPIO1 bit15
0x140	-	GPIO1_CTRL16	Reserved
0x144	-	GPIO1_CTRL17	Reserved
0x148	-	GPIO1_CTRL18	Reserved
0x14C	-	GPIO1_CTRL19	Reserved
0x150	-	GPIO1_CTRL20	GPIO control for GPIO1 bit 20
0x154	-	GPIO1_CTRL21	Reserved
0x158	-	GPIO1_CTRL22	Reserved
0x15C	-	GPIO1_CTRL23	Reserved
0x160	-	GPIO1_CTRL24	Reserved
0x164	-	GPIO1_CTRL25	Reserved
0x168	-	GPIO1_CTRL26	Reserved
0x16C	-	GPIO1_CTRL27	Reserved
0x170	-	GPIO1_CTRL28	GPIO control for GPIO1 bit 28
0x174	-	GPIO1_CTRL29	GPIO control for GPIO1 bit 29

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x178	-	GPIO1_CTRL30	Reserved
0x17C	-	GPIO1_CTRL31	Reserved
0x180	-	-	Reserved
0x184	-	GPIO1_PAD_EN	GPIO group1 pad enable
0x188	-	-	Reserved
0x18C	-	GPIO1_INT_STATUS	GPIO group1 interrupt status for RISC
0x190~0x1FC	-	-	Reserved
0x200	-	GPIO2_CTRL0	Reserved
0x204	-	GPIO2_CTRL1	Reserved
0x208	-	GPIO2_CTRL2	Reserved
0x20C	-	GPIO2_CTRL3	Reserved
0x210	-	GPIO2_CTRL4	Reserved
0x214	-	GPIO2_CTRL5	Reserved
0x218	-	GPIO2_CTRL6	Reserved
0x21C	-	GPIO2_CTRL7	Reserved
0x220	-	GPIO2_CTRL8	Reserved
0x224	-	GPIO2_CTRL9	Reserved
0x228	-	GPIO2_CTRL10	Reserved
0x22C	-	GPIO2_CTRL11	Reserved
0x230	-	GPIO2_CTRL12	Reserved
0x234	-	GPIO2_CTRL13	Reserved
0x238	-	GPIO2_CTRL14	Reserved
0x23C	-	GPIO2_CTRL15	Reserved
0x240	-	GPIO2_CTRL16	Reserved
0x244	-	GPIO2_CTRL17	Reserved
0x248	-	GPIO2_CTRL18	Reserved
0x24C	-	GPIO2_CTRL19	Reserved
0x250	-	GPIO2_CTRL20	Reserved
0x254	-	GPIO2_CTRL21	Reserved

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x258	-	GPIO2_CTRL22	Reserved
0x25C	-	GPIO2_CTRL23	Reserved
0x260	-	GPIO2_CTRL24	Reserved
0x264	-	GPIO2_CTRL25	Reserved
0x268	-	GPIO2_CTRL26	Reserved
0x26C	-	GPIO2_CTRL27	Reserved
0x270	-	GPIO2_CTRL28	Reserved
0x274	-	GPIO2_CTRL29	Reserved
0x278	-	GPIO2_CTRL30	Reserved
0x27C	-	GPIO2_CTRL31	Reserved
0x280	-	-	Reserved
0x284	-	GPIO2_PAD_EN	GPIO group2 pad enable
0x288	-	-	Reserved
0x28C	-	GPIO2_INT_STATUS	GPIO group2interrupt status for RISC
0x290~0x2FC	-	-	Reserved
0x300	-	GPIO3_CTRL0	Reserved
0x304	-	GPIO3_CTRL1	Reserved
0x308	-	GPIO3_CTRL2	Reserved
0x30C	-	GPIO3_CTRL3	Reserved
0x310	-	GPIO3_CTRL4	Reserved
0x314	-	GPIO3_CTRL5	Reserved
0x318	-	GPIO3_CTRL6	Reserved
0x31C	-	GPIO3_CTRL7	Reserved
0x320	-	GPIO3_CTRL8	Reserved
0x324	-	GPIO3_CTRL9	Reserved
0x328	-	GPIO3_CTRL10	Reserved
0x32C	-	GPIO3_CTRL11	Reserved
0x330	-	GPIO3_CTRL12	Reserved
0x334	-	GPIO3_CTRL13	Reserved

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x338	-	GPIO3_CTRL14	Reserved
0x33C	-	GPIO3_CTRL15	GPIO control for GPIO3 bit15
0x340	-	GPIO3_CTRL16	GPIO control for GPIO3 bit16
0x344	-	GPIO3_CTRL17	Reserved
0x348	-	GPIO3_CTRL18	GPIO control for GPIO3 bit18
0x34C	-	GPIO3_CTRL19	GPIO control for GPIO3 bit19
0x350	-	GPIO3_CTRL20	GPIO control for GPIO3 bit 20
0x354	-	GPIO3_CTRL21	GPIO control for GPIO3 bit 21
0x358	-	GPIO3_CTRL22	GPIO control for GPIO3 bit 22
0x35C	-	GPIO3_CTRL23	GPIO control for GPIO3 bit 23
0x360	-	GPIO3_CTRL24	Reserved
0x364	-	GPIO3_CTRL25	Reserved
0x368	-	GPIO3_CTRL26	Reserved
0x36C	-	GPIO3_CTRL27	Reserved
0x370	-	GPIO3_CTRL28	Reserved
0x374	-	GPIO3_CTRL29	Reserved
0x378	-	GPIO3_CTRL30	Reserved
0x37C	-	GPIO3_CTRL31	Reserved
0x380	-	-	Reserved
0x384	-	GPIO3_PAD_EN	GPIO group3 pad enable
0x388	-	-	Reserved
0x38C	-	GPIO3_INT_STATUS	GPIO group3 Interrupt status for RISC
0x390~0x3FC	-	-	Reserved
0x400	-	GPIO4_CTRL0	Reserved
0x404	-	GPIO4_CTRL1	Reserved
0x408	-	GPIO4_CTRL2	Reserved
0x40C	-	GPIO4_CTRL3	Reserved
0x410	-	GPIO4_CTRL4	Reserved
0x414	-	GPIO4_CTRL5	Reserved

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x418	-	GPIO4_CTRL6	Reserved
0x41C	-	GPIO4_CTRL7	Reserved
0x420	-	GPIO4_CTRL8	Reserved
0x424	-	GPIO4_CTRL9	Reserved
0x428	-	GPIO4_CTRL10	Reserved
0x42C	-	GPIO4_CTRL11	Reserved
0x430	-	GPIO4_CTRL12	Reserved
0x434	-	GPIO4_CTRL13	Reserved
0x438	-	GPIO4_CTRL14	Reserved
0x43C	-	GPIO4_CTRL15	Reserved
0x440	-	GPIO4_CTRL16	Reserved
0x444	-	GPIO4_CTRL17	Reserved
0x448	-	GPIO4_CTRL18	Reserved
0x44C	-	GPIO4_CTRL19	Reserved
0x450	-	GPIO4_CTRL20	Reserved
0x454	-	GPIO4_CTRL21	Reserved
0x458	-	GPIO4_CTRL22	Reserved
0x45C	-	GPIO4_CTRL23	Reserved
0x460	-	GPIO4_CTRL24	GPIO control for GPIO4 bit 24
0x464	-	GPIO4_CTRL25	GPIO control for GPIO4 bit 25
0x468	-	GPIO4_CTRL26	GPIO control for GPIO4 bit 26
0x46C	-	GPIO4_CTRL27	GPIO control for GPIO4 bit 27
0x470	-	GPIO4_CTRL28	GPIO control for GPIO4 bit 28
0x474	-	GPIO4_CTRL29	Reserved
0x478	-	GPIO4_CTRL30	Reserved
0x47C	-	GPIO4_CTRL31	Reserved
0x480	-	-	Reserved
0x484	-	GPIO4_PAD_EN	GPIO group4 pad enable
0x488	-	-	Reserved

RISC Address <11:0>	DSP Address <6:0>	Register	Description
0x48C	-	GPIO4_INT_STATUS	GPIO group4 interrupt status for RISC
0x490~0x4FC	-	-	Reserved

Table 176: GPIO Register Mapping

Register Descriptions

- GPIO control register (GPIO_CTRLx)
  Each GPIO_CTRLx register controls one GPIO pin.

Bit	Name	Default	Description
0 (R/W)	INT_LT	1'b0	GPIO Interrupt low/falling edge trigger enable: 0: Either the falling-edge interrupt or the low-level interrupt is disabled. 1: If GPIO_INT_TYPE is set, then the falling-edge interrupt is enabled; if GPIO_INT_TYPE is de-asserted, then the low-level interrupt is enabled.
1 (R/W)	INT_HT	1'b0	GPIO Interrupt high/rising edge trigger enable: 0: Either the rising-edge interrupt or the high-level interrupt is disabled. 1: If GPIO_INT_TYPE is set, then the rising-edge interrupt is enabled; if GPIO_INT_TYPE is de-asserted, then the high-level interrupt is enabled.
2 (R/W)	INT_TYPE	1'b0	GPIO Interrupt Type Select bits: 0: Interrupt is level-triggered. 1: Interrupt is edge-triggered.
3 (R/W)	RISC_INT_EN	1'b0	Enables the GPIO to generate interrupts to RISC when the interrupt status is set: 0: Interrupt is disabled. 1: Interrupt is enabled.
4 (R/W)	INT_STATUS	1'b0	GPIO Interrupt Status: 0: No interrupts. 1: There is an interrupt request. Write one to the bit will clear the interrupt status bit.
5 (R/W)	OUT_EN	Refer to Table: Pin Status	GPIO Data output enable: 0: The GPIO is input pin. 1: The GPIO is output pin.
6 (R/W)	OUT	1'b0	GPIO data output.
7 (R)	DATA_IN	1'bx	GPIO data input.

Bit	Name	Default	Description
8 (R/W)	PULL_EN	Refer to Table : Pin Status	Enables the GPIO Pull function: 1: Enable 0: Disable
9 (R/W)	PULL_TYPE	Refer to Table: Pin Status	GPIO pull type selection when the pull function is enabled: 1: Pull High 0: Pull low
10 (R/W)	DSP_INT_EN	1'b0	Exists only for GPIO group0; GPIO will generate different interrupts to DSP and RISC. There are different interrupt enable registers for DSP and RISC. Enables the GPIO to generate interrupts to DSP when the interrupt status is set: 0: Interrupt is disabled. 1: Interrupt is enabled.
31:11	-	-	Reserved

Table 177: GPIO Control Register Definition

NOTE – When GPIO is set as input (OUT_EN=1'b0), it can be configured as pull function according to the setting of PULL_EN and PULL_TYPE. However, when the pull function is enabled and before switching into another full function, it must disable PULL_EN first. For example, if the current GPIO status is pull-high, before switching to pull-low, it must set PULL_EN=1'b0, then setting PULL_TYPE=1'b0 and enable the pull function (PULL_EN=1'b1).

- GPIO DSP Control Enable Register (GPIO0_DSP_EN)
  This control register only exist for GPIO group0.

Bit	Name	Default	Description
31:0 (R/W)	DSP_EN<31:0>	32'h0	GPIO DSP control enable: 1: The GPIO pin is controlled by DSP. 0: The GPIO pin is controlled by RISC.

Table 178: DSP Control Enable Register Definition

- GPIO PAD Enable Register (GPIOx_PAD_EN)
  Each bit in GPIOx_PAD_EN controls the pad MUX of corresponding GPIO pin in group x, for example:
  - GPIO0_PAD_EN<0> controls the pad MUX of GPIO pin 0 in group 0
  - GPIO2_PAD_EN<21> controls the pad MUX of GPIO pin 85 (32*2+21) in group 2

All the GPIO is enabled at the default status except of GPIO3[16:15], and GPIO4[28:24].

So the default value of GPIO pad enable in AtlasV is:

GPIO0_PAD_EN: 32'bxxxx_1111_1111_11xx_1111_1111_1111_1111

GPIO1_PAD_EN: 32'bxx11_xxxx_xxx1_xxxx_1111_1111_1xxx_1111

GPIO2_PAD_EN: 32'bxxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx

GPIO3_PAD_EN: 32'bxxxx_xxxx_1111_11x0_0xxxx_xxxx_xxxx_xxxx

GPIO4_PAD_EN: 32'bxxx0_0000_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx

NOTE – 1'bx indicates the value of this bit is reserved.

Refer to the section of Pin Sequence and Ball Assignment for which pins to enable as GPIO by default for different modes.

Bit	Name	Default	Description
31:0 (R/W)	PAD_EN<31:0>	Refer to the section of Pin Sequence and Ball Assignment	GPIO PAD MUX control: 1: The pin is used by GPIO. 0: The pin is used by other blocks.

Table 179: GPIO PAD Enable Register Definition

- GPIO INTERRUPT STATUS Register (GPIOx_ INT_STATUS)
  This register contains the interrupt status for all GPIO pins in group x.

  This register is set as read-only. Writing to this register will produce no effects. To clear the interrupt, write the INT_STATUS bit in GPIO control register.

Bit	Name	Default	Description
31:0 (R)	INT_STATUS<31:0>	32'h0	GPIO interrupt status: 0: No interrupts. 1: There is an interrupt request.

Table 180: GPIO Interrupt Status Register Definition

eFuse

The module is the interface of the programmable electrical fuse macro, whose primary function is to carry out program and read operations. For detailed information please contact a SiRF Application Engineer.

Touch Screen Controller

Overview

The Touch Screen (TS) Controller is used for driving the touch screen to perform coordinates and pressure measurements. It supports a 4-wire resistive touch screen interface which can be used for PDAs, Cell phones, Smart handheld devices, MP3 players and so on. There are three auxiliary inputs which are supported by the Touch Screen Controller for the other measurements, such as temperature, battery and so on. Also stream mode is supported through one of auxiliary inputs. The stream mode is used for converting analog input at a configured 8K/16K/24K/32K/40K/48K SPS (Samples per Second) conversion rate, which can be used for recording stream from microphone.

Feature List

- 12-bit resolution
- 600kSPS maximum conversion rate
- 3.3 V (Typical) analog supply, 1.2 V (Typical) digital supply
- X direction measurement
- Y direction measurement
- Auto-sequential measurement
- Touch pressure measurement
- Supports 4-wire touch screen
- Three auxiliary inputs for general purpose measurements (e.g. temperature, battery, etc.)
- Stream mode with configured sampling rate

Pin Description

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_XP	Analog	-	Touch Screen	-	Touch-Screen X+ terminal
X_XN	Analog	-	Touch Screen	-	Touch-Screen X- terminal
X_YP	Analog	-	Touch Screen	-	Touch-Screen Y+ terminal
X_YN	Analog	-	Touch Screen	-	Touch-Screen Y- terminal
X_REF_ADC	Analog	-	Touch Screen	-	Reference voltage for single-ended topology
X_AUX0	Analog	-	Auxiliary Input	-	Auxiliary input
X_AUX1	Analog	-	Auxiliary input	-	Auxiliary input
X_AUX2	Analog	-	Auxiliary input	-	Auxiliary input

Table 181: TSC External Pin Descriptions

Functional Descriptions

Block Diagram

The TSC is a mix-signal design. It has an Analog Module which is built around a 12-bit ADC.

The block diagram of the TSC is shown in the Figure below.

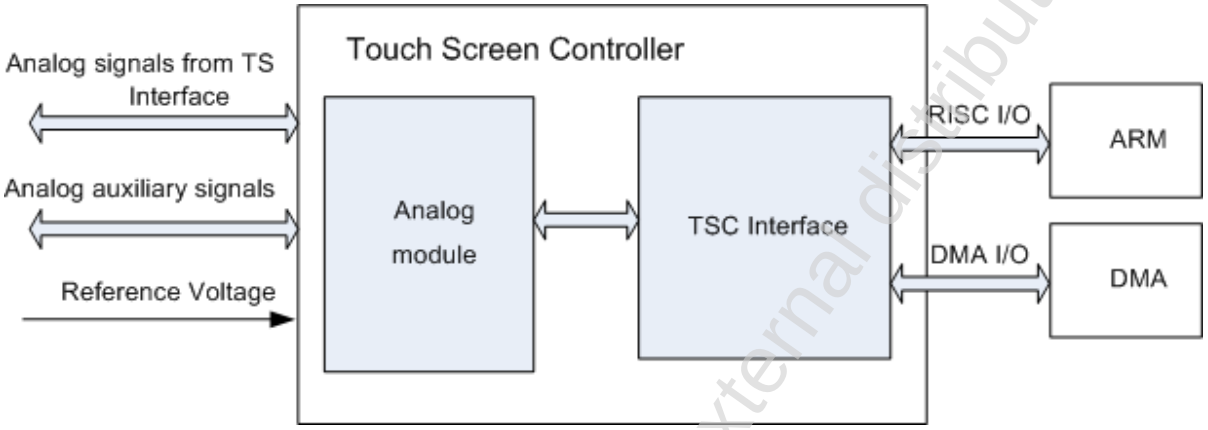


Figure 17: TSC Block Diagram

Module Descriptions

Top Level of TSC

The Touch Screen Controller supports a 4-wire resistive TS interface; it also supports three auxiliary inputs for general purpose measurements.
While connected to the 4-wire touch screen, the controller can work in either polling mode or continuous mode, which can be set by ARM through the IO interface. The interrupt signal (intr) is used to notify the ARM that either a pen-down is detected or data is available for measurement. When the screen is touched, the pen-down interrupt register (PEN_INTR) will be set by the controller. The controller will then begin measurement according to the corresponding register set by ARM. The TS Controller will notify ARM by setting the data interrupt register (DATA_INTR) when it finishes measurement, then ARM can read the measured data from the registers.
For the auxiliary input measurement, the first step is to set the touch screen power management register (PRP) to the corresponding mode, and the ADCSEL register will be set to select a corresponding auxiliary measurement. The controller will signal ARM through the DATA_INTR register as the data is available on the auxiliary data registers. Following figure illustrates the top-level of TS Controller which consists of two main modules, Analog module and TSC interface. The analog module has its own power domain.

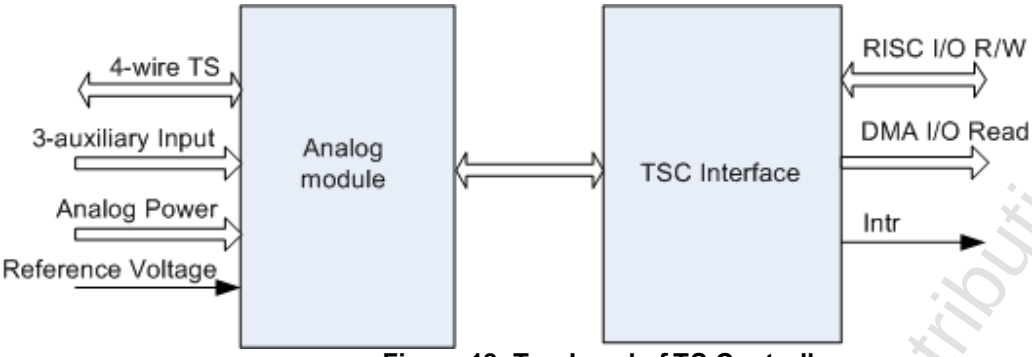


Figure 18: Top Level of TS Controller

Analog Module

The analog module is built around a 12-bit ADC for TS applications.

In Analog module, the driver switches which consist of several nMOS and pMOS are used for controlling the analog input terminal of 4-wire touch screen. The input multiplexer is used for selecting the corresponding analog input for ADC sampling.

The pen-down detector function is implemented through a Pull-up resistance and a pMOS gate; detailed implementation of this function will be described later.

The current consumption of Analog module in normal operation is 14.85mW and 66uW in power-down mode.

Principles of Operation

Pen-Down Detection

The pen-down detection is implemented separately from the resistance and conductance of the touch screen plate; it is made with a simple analog circuit - a pMOS gate and a resistance. To implement pen-down detection, simply set the puon pin to logic high. Before executing pen detection, the first step is pre-charge. At the pre-charge step, the xppsw driver should be opened and the XP node should be changed.

After pre-charge the second step is discharge. The touch screen interface should close the xppsw driver and the discharge will begin through opening the ynnsw driver after pre-charge. The purpose of discharge is to discharge the charge on decoupling capacitors in order to sample pen-down interrupts. When the touch screen is untouched, the internal XP is pulled High and there will be no current flows. The penirq output will be at the logic state High.

When the screen is touched, the internal XP will therefore be pulled down and penirq will output logic Low to indicate that the screen is being touched.

The next picture illustrates a simplified schematic of pen-down detection on a 4-wire touch screen.

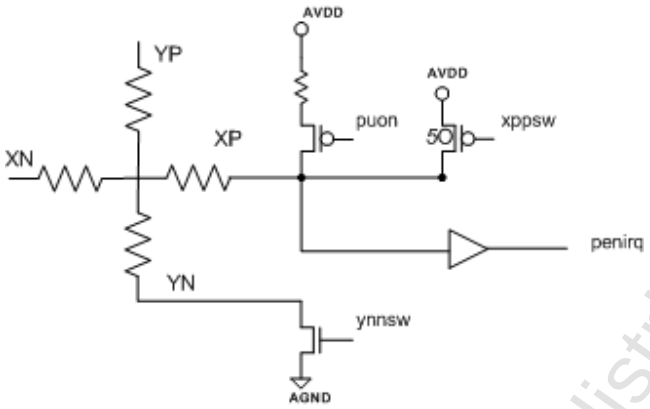


Figure 19: Pen-Down Detection on 4-Wire Touch Screen

Y/X Coordinate Measurement

The principle of operation is illustrated in the following figure.

For the Y coordinate measurement, the YP pin is internally driven to AVDD and YN to AGND. So the Y coordinate is a potential divder and the voltage at the touched point is proportional to its Y coordinate, this voltage can be measured on the XP terminal because no more current is flowing through it.

X coordiante measurement is similar to that of the Y coordinate.

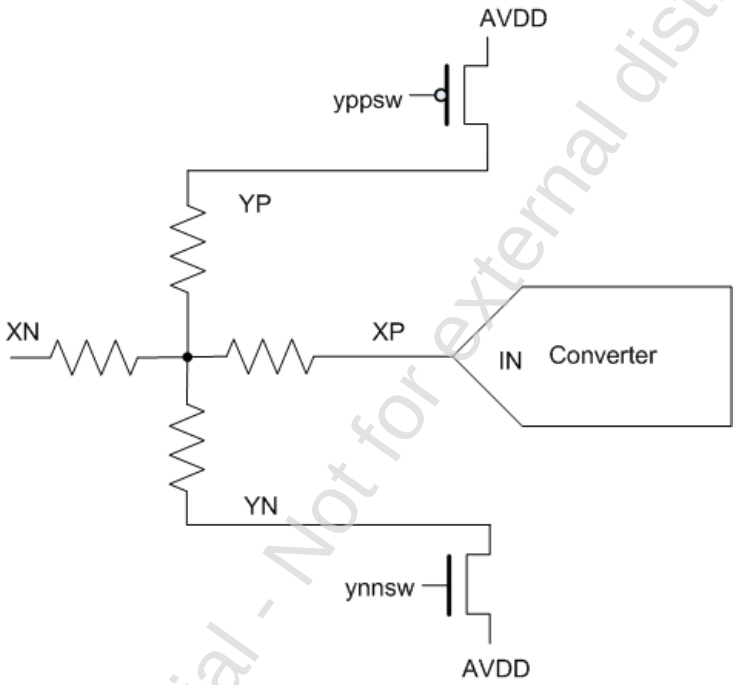


Figure 20: Y Coordinate Measurement on 4-Wire Touch Screen

Pressure Measurement

Pressure Measurement is implemented by calculating contact resistance between the top and bottom plates of the touch screen. This calculation requires three auxiliary measurements, measurement of the X position and two additional cross panel measurements of the touch screen. All these measurements are performed using the switch drivers. You also need to know the value of the X plate resistance.

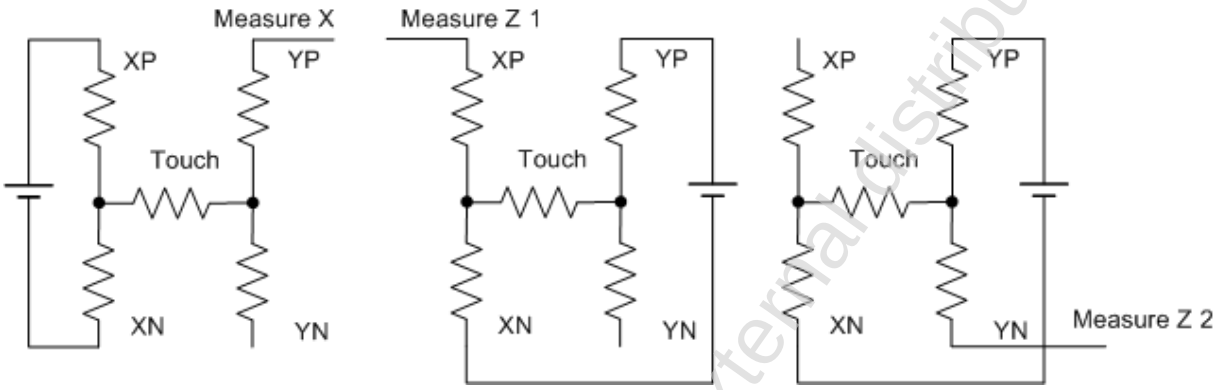


Figure 21: Pressure Measurement Block Diagram

Using the following equation to calculate the touch resistance:

$$R_{TOUCH} = R_{X-Plate} * \frac{X}{1024}\left(\frac{Z2}{Z1} - 1\right)$$

The second method requires the knowledge of both the X-Plate and Y-Plate resistance, as well as measurement of the X-Position and Y-Position, and Z1. The following equation is used to calculate touch resistance:

$$R_{TOUCH} = \frac{R_{X-Plate} * X}{1024}\left(\frac{1024}{Z1} - 1\right) - R_{Y-Plate} * \left(1 - \frac{Y}{1024}\right)$$

Auxiliary Measurement

Auxiliary inputs are used for the measurement of temperature, battery and so on. There are three pins for the auxiliary measurement. When using auxiliary measurement, the reference voltage of ADC topology will be the voltage connected on the ADC's reference PAD.

Stream Mode of TSC

The stream mode is an alone mode compared with coordinate measurement and auxiliary measurement. Although only one ADC is used for the stream mode, coordinate and auxiliary measurement, time-division multiplexing technology is used to ensure that steam mode can work together with these two measurement at the same time period.

In stream mode, ADC converts analog input at certain conversion rate which can be configured by software. Any one of auxliary inputs can be configured as analog input in stream mode. The converted data by ADC is stored into FIFO first, and then transferred into memory through DMA. The figure below shows the data stream path in stream mode.

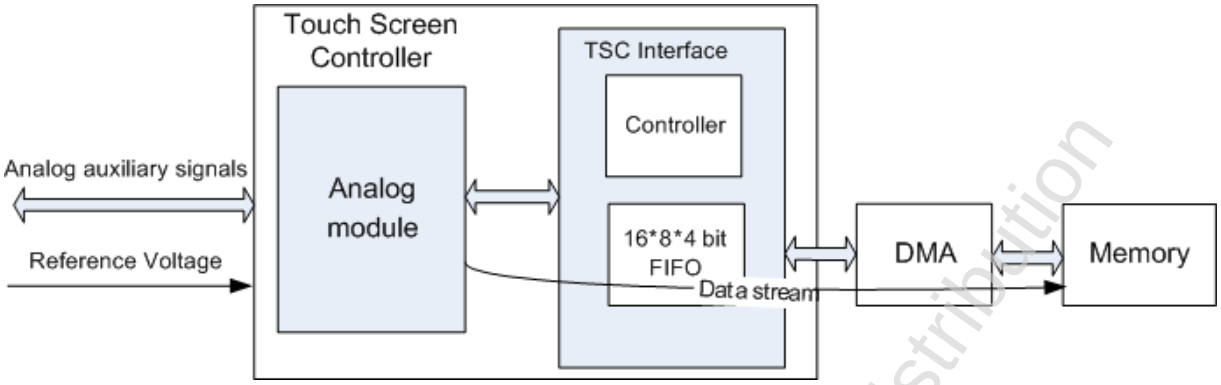


Figure 22: Data Flow in Stream Mode

The FIFO's width is 16-bit and ADC's data out is 12-bit. So each time, after ADC finish one data convertion, this data is wrapped into a 16-bit and then filled into FIFO. Register WRAP_CTRL is used to control the wrap format.
When setting WRAP_CTRL to different value, the 16-bit data format in FIFO data shows in the blow table.

WRAP_CTRL=3'b000	**Bit**	**15:4**		**3:0**
	Data	ADC Conversion Data[11:0]		4'b0
WRAP_CTRL=3'b001	Bit	15	14:3	2:0
	Data	1'b0	ADC Conversion Data[11:0]	3'b0
WRAP_CTRL=3'b010	Bit	15:14	13:2	1:0
	Data	2'b0	ADC Conversion Data[11:0]	2'b0
WRAP_CTRL=3'b011	Bit	15:13	12:1	0
	Data	3'b0	ADC Conversion Data[11:0]	1'b0
WRAP_CTRL=3'b100	Bit	15:12		11:0
	Data	4'b0		ADC Conversion Data[11:0]

Table 182: 16-Bit Data Format in the FIFO

Operation of the TS Controller

The TS Controller includes two parts of circuit:

- Pen detector: detects whether the screen is touched.
- Digitizer: is used for measurement.

A TS Controller operation is defined as the process of controlling these two parts of circuit.

Analog Module Disable Mode

This mode is the default status of the TS controller or when register **PRP** is set to 2'b00. Both the pen-detector and digitizer are disabled in this mode. In addition, ADC will be powered down and all driver switches will be opened.

Pen Detector Enabled while Digitizer Is Disabled (Wake-Up ADC)

This mode is used for the coordinate measurement (X/Y/Z). However it does not support auxiliary measurement.

Writing PRP to 2'b01 will automatically enter this mode. As the digitizer is disabled, the ADC will be powered off and all driver switches will be turned off except for the one that drives ynnsw (drives the yn to the agnd). The puon pin will be set to logic low to monitor whether or not the screen is touched.

When the screen is touched, the PENIRQ pin is pulled to logic high to alert the TSC interface. Upon receiving the penirq signal from Analog Module, the TSC Interface will generate a pen-down interrupt (pen_intr) to ARM. Then ARM will set the coordinate condition and initiate measurement. This shows that the TSC interface enables the digitizer through the control signals to the Analog Module.

TSC has two measurement modes: polling and continuous. For the polling measurement mode, the measurement may occur only when the ARM chip writes the register Poll once. In continuous mode, the measurement will be repeated when meeting measurement conditions (such as pen-down).

- Polling Measurement of the X Coordinate

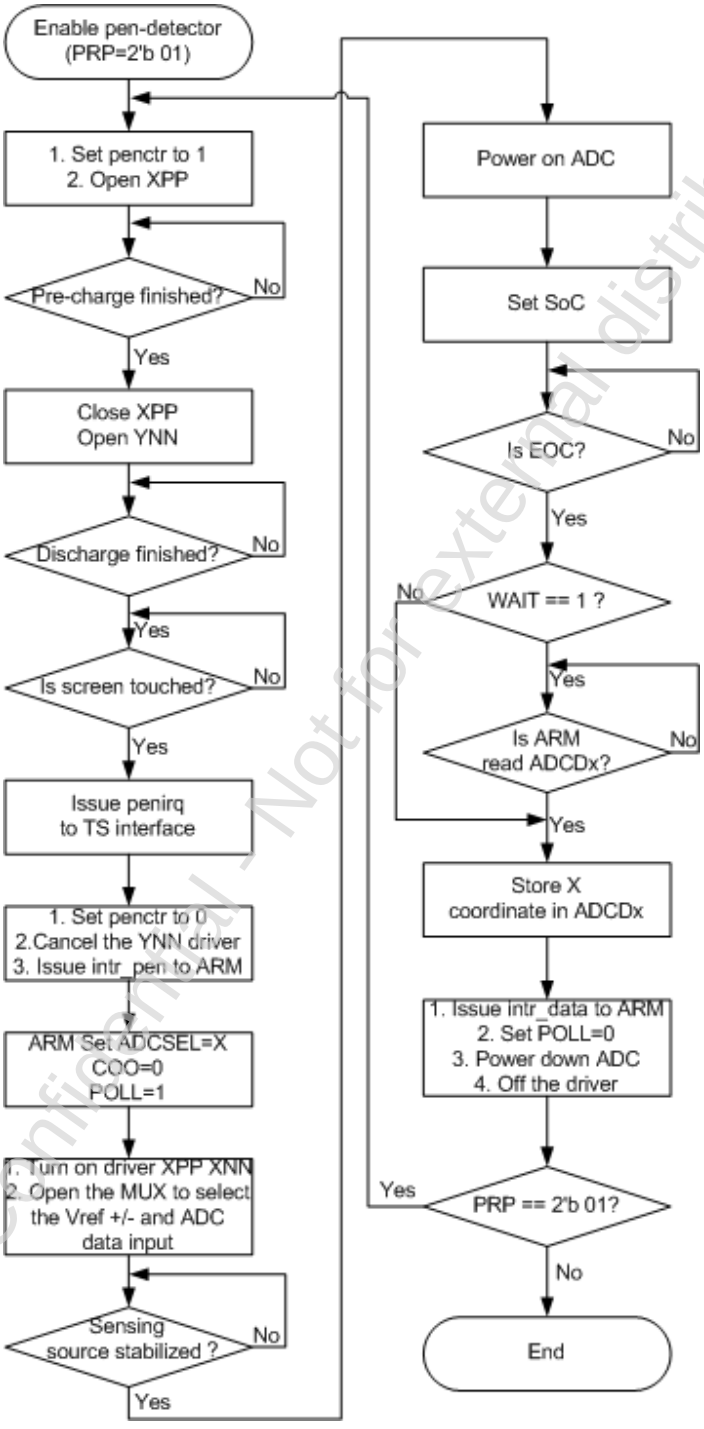


Figure 23: X Coordinate Polling Measurement Flowchart

- Polling Measurement of the Y Coordinate

  The polling measurement of the Y coordinate is similar to that of the X coordinate.

- Polling Measurement of the Z Coordinate

  Touch pressure can be determined indirectly by measuring the contact resistance on the Z coordinate. Therefore pressure measurement requires making three measurements as this process:

  - Measurement of the X position
  - Two additional cross-panel measurements of the TS

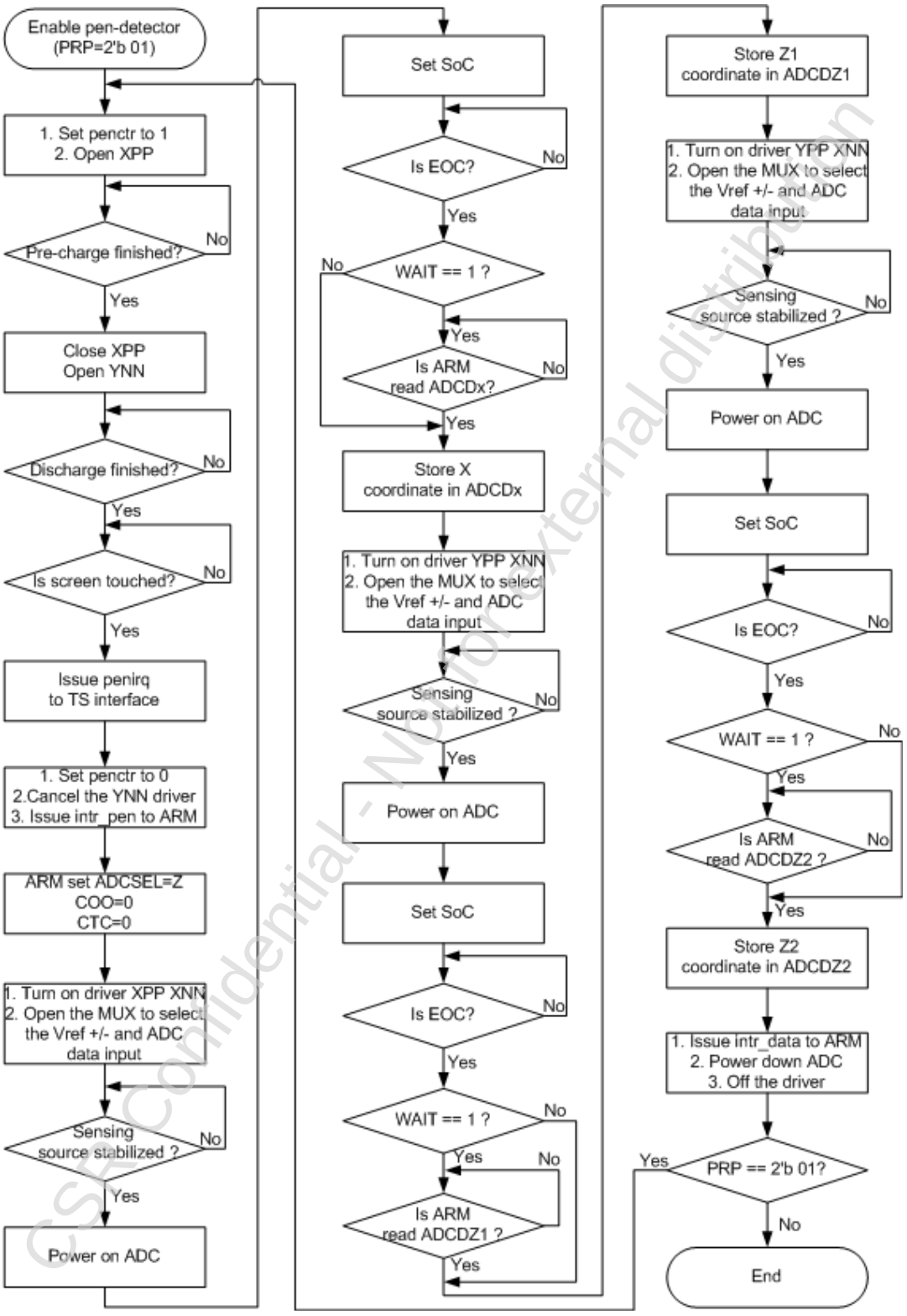


Figure 24: Z Coordinate Polling Measurement Flowchart

- Polling Measurement of XY Coordinates

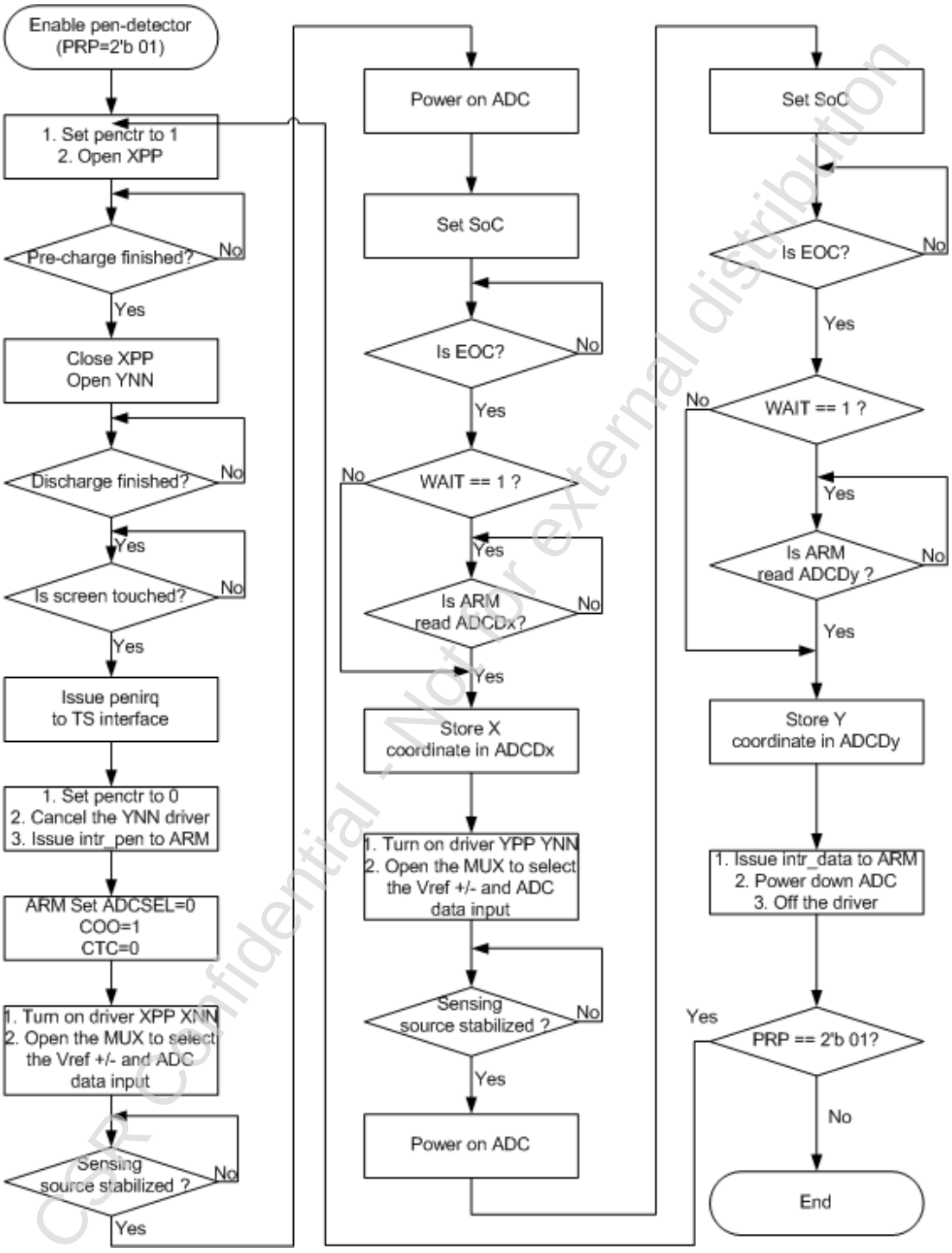


Figure 25: XY Coordinate Polling Measurement Flowchart

- Polling Measurement of XYZ Coordinates
  The polling measurement of XYZ is similar to that of the XY coordinate.
- Continuous Measurement of the X Coordinate

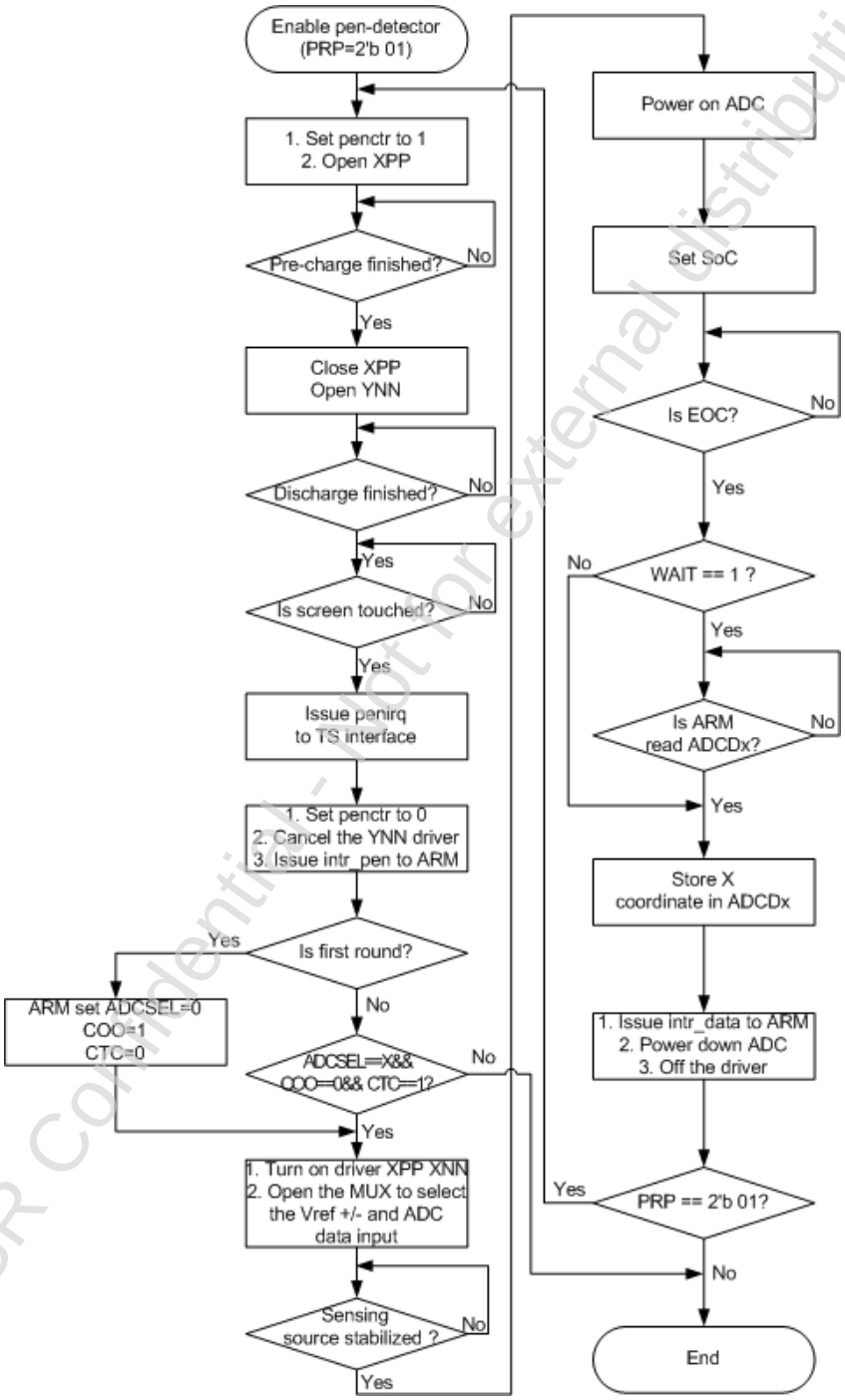


Figure 26: X Coordinate Continuous Measurement Flowchart

- Continuous Measurement of the Y and Z Coordinates
  The continuous measurement of Y and Z is similar to that of the X coordinate.

- Continuous Measurement of XY Coordinate

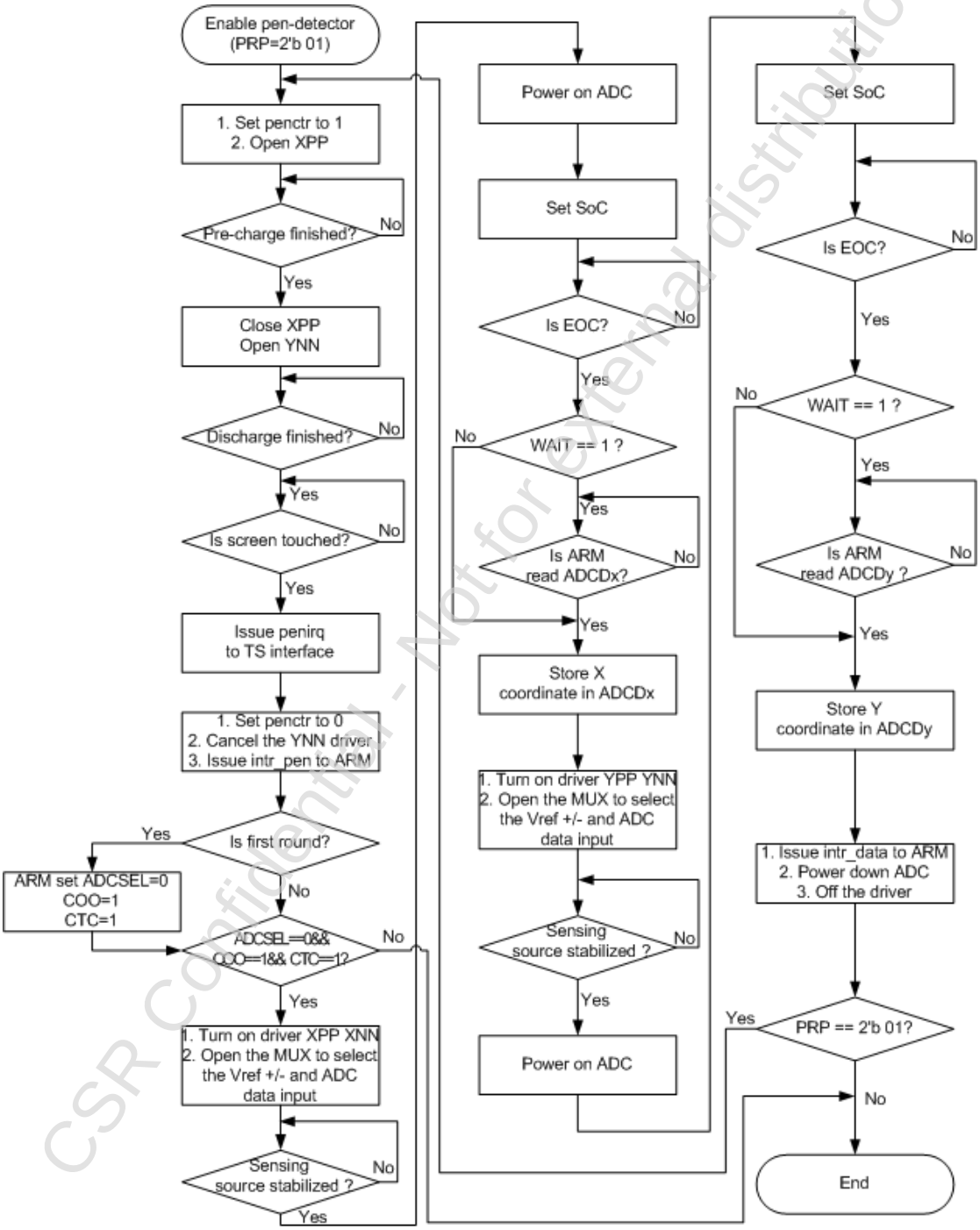


Figure 27: XY Coordinate Continuous Measurement Flowchart

- Continuous Measurement of XYZ Coordinate
  The continuous measurement of XYZ is simular to that of the XY coordinate.

Pen-Detector Enable while Digitizer Disable (No Wake-Up ADC)

This mode is only for the pen-detector.

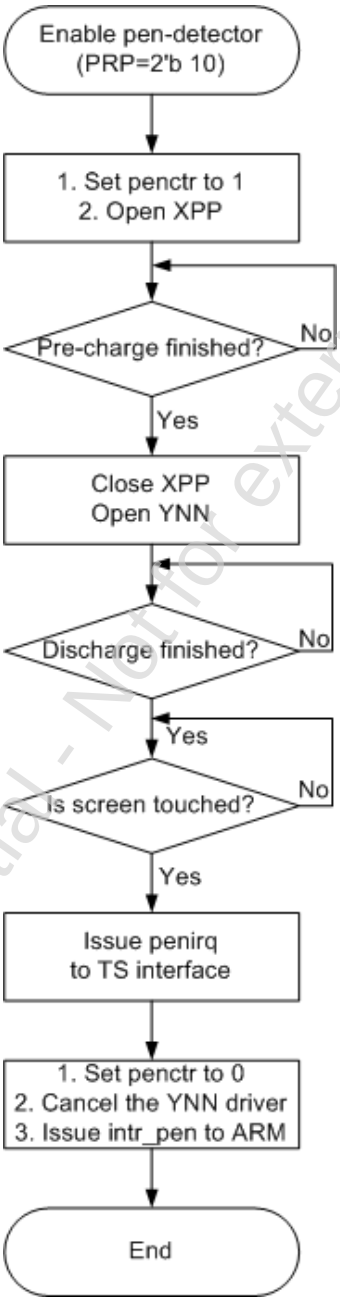


Figure 28: Pen-Detector Enable while Digitizer Disable (No Wake-Up on Pen-Down)

Both Pen-Detector and Digitizer Enable

This mode can perform either coordinate or auxiliary measurement according to the value of the ADCSEL0 register. However it is not recommended to use this mode for coordinate measurements due to power consumption issues.

- Polling Measurement of the X Coordinate

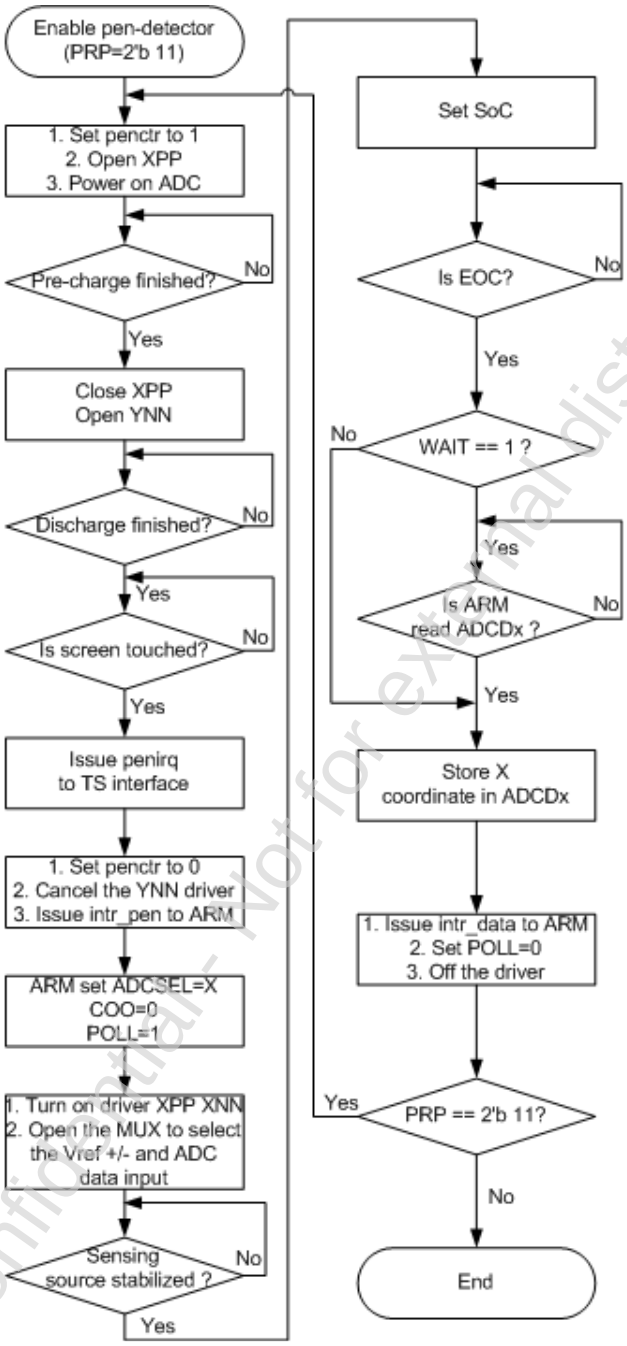


Figure 29: X Coordinate Polling Measurement Flowchart (PRP=2'b11)

- Polling Measurement of Auxiliary Input

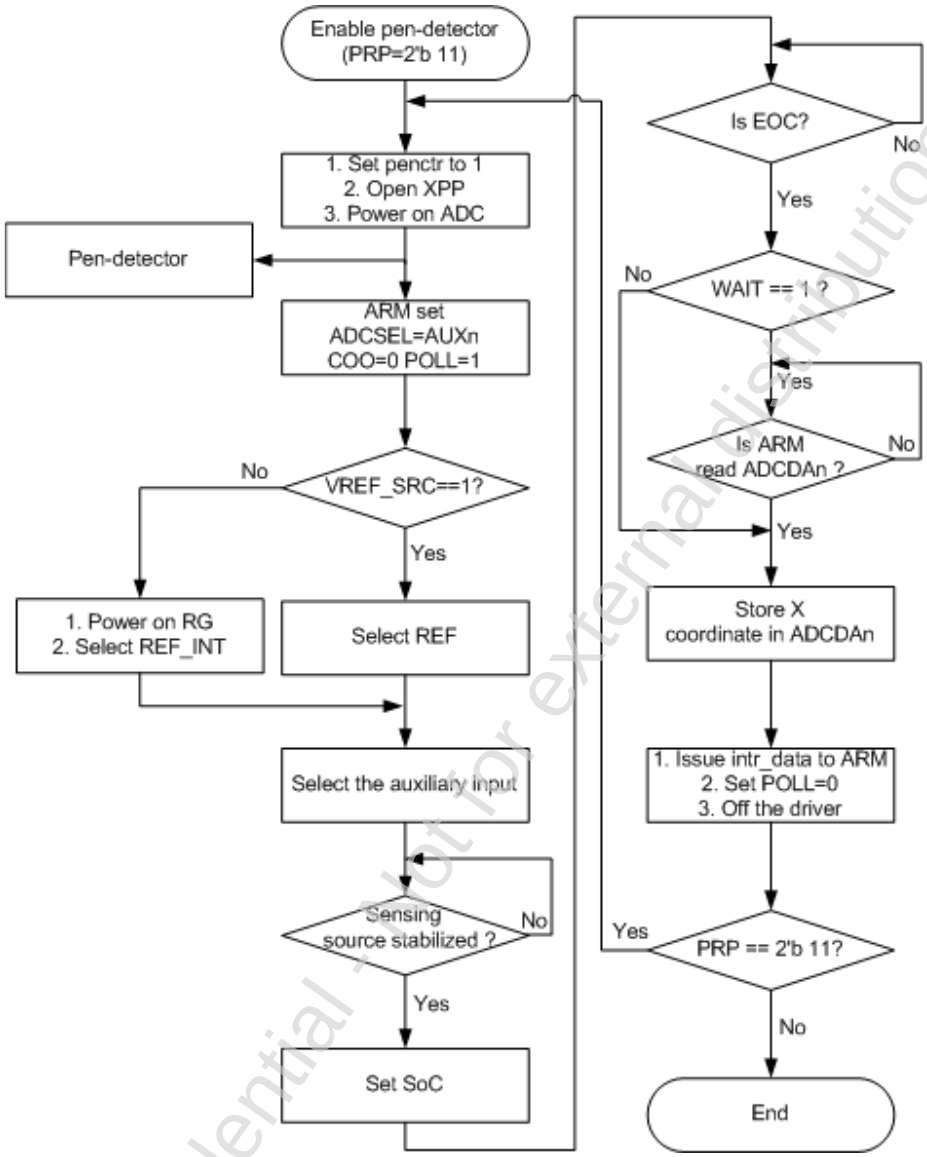


Figure 30: Auxiliary Input Polling Measurement Flowchart (PRP=2'b11)

- Continuous Measurement of the X Coordinate

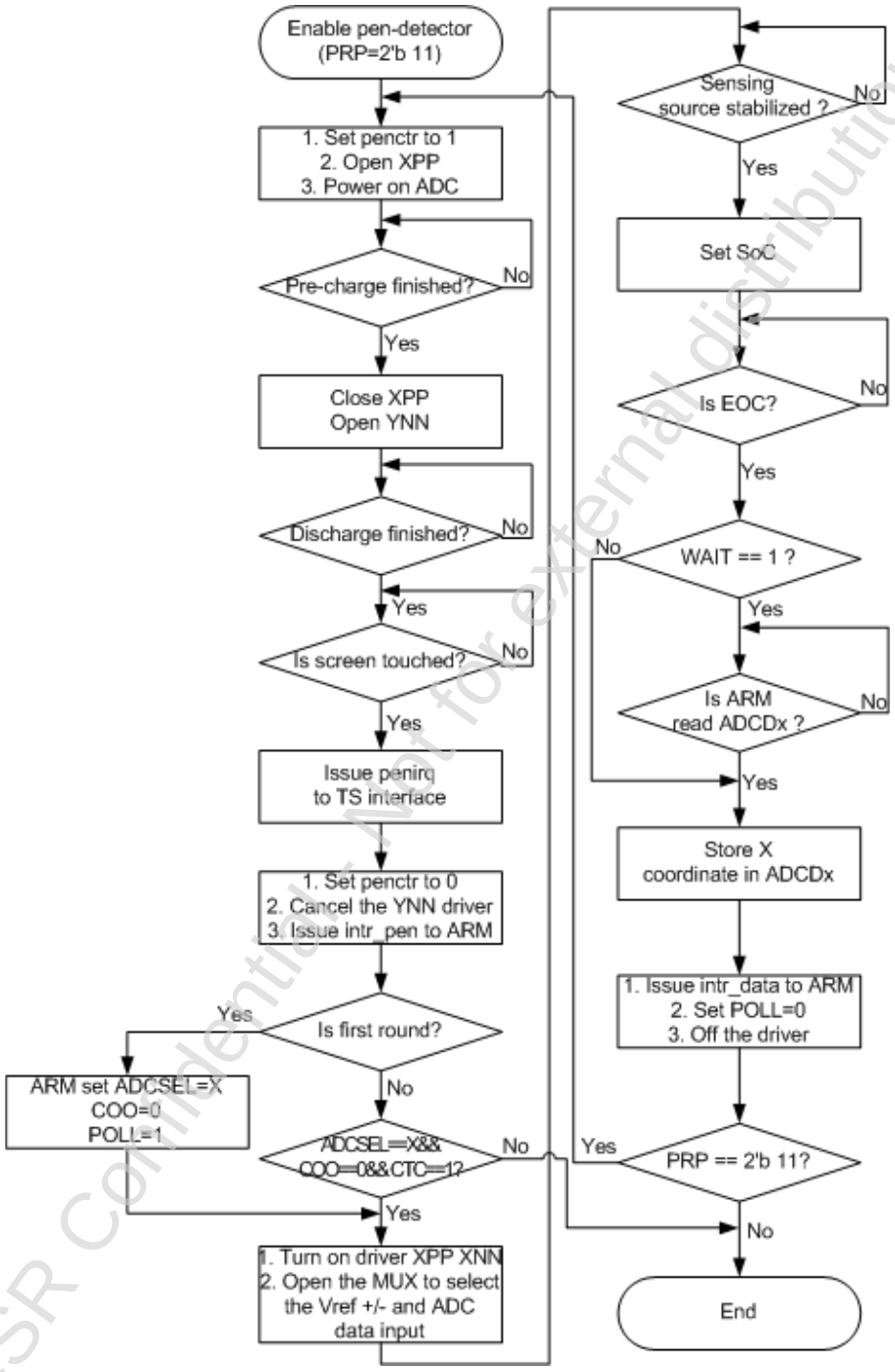


Figure 31: X Coordinate Continuous Measurement Flowchart (PRP=2'b11)

- Stream Mode Measurement
  The blow figure shows the flowchart of the stream mode measurement.

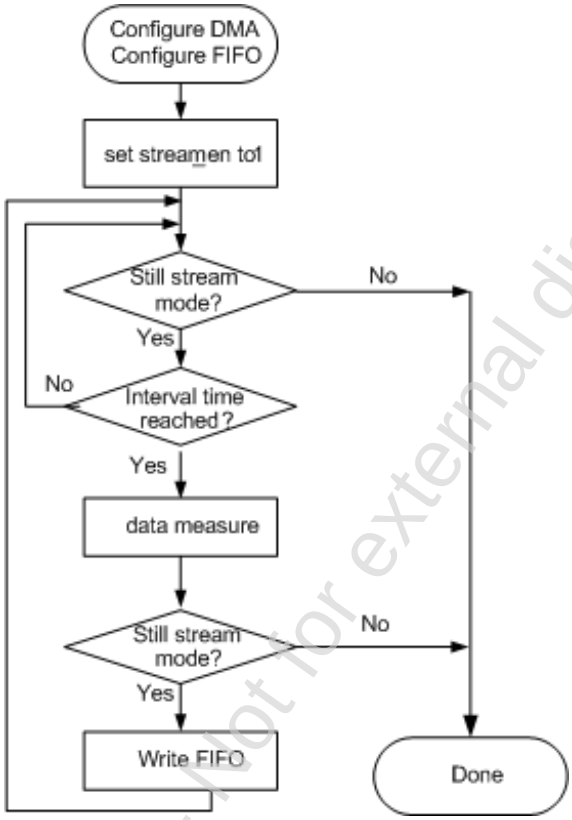


Figure 32: Stream Mode Measurement Flowchart

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
TSC-based address through RISC I/O	0x80120000

Table 183: TSC Address Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	TSC_REG1	TSC control register 1
0x0004	TSC_REG2	TSC control register 2
0x0008	TSC_INTR	TSC interrupt register
0x000C	TSC _ BACK_C	TSC Read Back coordinate register
0x0010	TSC _ BACK_Z	TSC Read Back pressure register
0x0014	TSC _ BACK_A0	TSC Read Back auxiliary0 register
0x0018	TSC _ BACK_A1	TSC Read Back auxiliary1 register
0x001C	TSC _ BACK_A2	TSC Read Back auxiliary2 register
0x0020~0040	-	Reserved
0x0044	TSC_STREAM_CTRL	Stream Mode control
0x0048	TSC_FIFO_LEVEL_CHK	FIFO level check
0x004C	TSC_FIFO_OP	FIFO operation
0x0050	TSC_FIFO_STS	FIFO status register
0x0054	TSC_FIFO_INT_EN	FIFO interrupt enable
Others	-	Reserved

Table 184: TSC Register Mapping

Detailed register mapping of the TS controller (empty cells indicate reserved registers):

A[12:0]	31:16	15	14	13	12	11	10	9	8	7:4	3:0	DEFAULT
0x0000	-	POLL	ADCSEL0			COO	CTC	-		DEL_SET	-	32'h0
0x0004	-	PRP		-	-	-	-	WAIT_READ		DEL_PRE	DEL_DIS	32'h0

Table 185: Detailed TSC Register Mapping (1)

A[12:0]	31:9	8	7:6	5	4	3:2	1	0	DEFAULT
0x0008	-	--	-	PEN_INTR_EN	DATA_INTR_EN		PEN_INTR	DATA_INTR	32'h0

Table 186: Detailed TSC Register Mapping (2)

A[12:0]	31	30	29	28:20	19:10	9:0	DEFAULT
0x000C	PNDN	VALID_Y	VALID_X	-	ADCDY	ADCDX	32'h0
0x0010	-	VALID_Z2	VALID_Z1	-	ADCDZ2	ADCDZ1	32'h0
0x0014	-	VALID_A0	-	-	-	ADCDA0	32'h0
0x0018	-	VALID_A1	-	-	-	ADCDA1	32'h0
0x001C	-	VALID_A2	-	-	-	ADCDA2	32'h0

Table 187: Detailed TSC Register Mapping (3)

Register Description

TSC Power Management

To save power, both the digitizer and the pen-down detector of Analog module can be disabled when they are not being used.

When the digitizer is disabled, ADC will be powered off and all driver switches will be turned off except for the one that drives the YNN (drives the YN to the AGND). When the pen-down detector is disabled, the YNN driver switch will be turned off in order to disable the pen-down detection function.

The status of the digitizer and the pen-down detector is controlled by the following bits:

TSC Power Management

Bit	Name	Default	Description
15:14 (R/W) 0x0004	PRP	2'b00	Analog Module power management. 00: Pen digitizer off, pen detector off (default). 01: Pen digitizer off, pen detector enable, pen digitizer wakes up when pen-down is detected. 10: Pen digitizer off, pen detector enable, no wake-up pen digitizer when pen down is detected. 11: Both the pen digitizer and the detector are enabled.

Table 188: TSC Power Management

TS Initiation of Measurement

The TSC interface supports both polling and continuous mode for measurement. In a polling mode, each measurement is initiated individually by writing logic H to the POLL bit. This bit will automatically reset itself to logic L when the measurement process is completed.

- TS Initiation of Measurement

Bit	Name	Default	Description
10 (R/W) 0x0000	CTC	1'b0	0 : Polling mode 1 : Continuous mode
15 (R/W) 0x0000	POLL	1'b0	Writing 1 initiates measurement in polling mode

Table 189: TS Initiation of Measurement

TS Measurement Types

The ADCSEL0 register determines which type of measurement is performed.

When COO is set to 1'b0, there will only be one type of measurement both in polling or continuous mode. When COO is set to 1'b1, it is easy to obtain coordinate pairs rather than single coordinate measurement.

In polling coordinate mode (COO=1'b1, CTC=1'b0), the TS controller performs the X measurement, followed by a Y measurement and an additional measurement determined by ADCSEL0 before stopping. In continuous coordinate mode (COO=1'b1, CTC=1'b1), the TS controller continuously repeats a sequence consisting of a X coordinate measurement, followed by a Y coordinate measurement and an additional measurement determined by ADCSEL0.

- TS Measurement Types Control Registers

Bit	Name	Default	Description
14:12 (R/W) 0x0000	ADCSEL0	3'b0	Measurement type (ADC input selector) 000: No measurement 001: X coordinate measurement 010: Y coordinate measurement 011: Z coordinate measurement 100: AUX0 measurement 101: AUX1 measurement 110: AUX2 measurement 111: Reserved
11 (R/W) 0x0000	COO	1'h0	Enable coordinate mode 0: Single measurement according to ADCSEL0. 1: X, then Y, then additional measurement indicated by ADCSEL0.

Table 190: TS Measurement Types Control Register

TS Read-Back Register

There are three types of read back data from the TS controller.

- Pen status, which indicates whether the pen is up or down
- Output data from ADC
- Data valid

  When the ADC data is output to the data registers, the corresponding valid registers will be set to 1'b1, then the data valid registers will reset themselves (set to 1'b0) after the corresponding data registers are read by ARM.

The WAIT_READ register indicates whether or not to overwrite the new data to the unread data register.

Bit	Name	Default	Description
31 (R) 0x000C	PNDN	1'b0	Pen status 0 : Pen Up 1 : Pen Down
11:0 (R) 0x000C	ADCDX	12'b0	Touch Screen X coordinate measurement data ADCDX[11] = MSB ADCDX[0] = LSB
23:12 (R) 0x000C	ADCDY	12'b0	Touch Screen Y coordinate measurement data ADCDY[23] = MSB ADCDY[12] = LSB

Bit	Name	Default	Description
11:0 (R) 0x0010	ADCDZ1	12'b0	Touch Screen pressure measurement data ADCDZ1[11] = MSB ADCDZ1[0] = LSB
23:12 (R) 0x0010	ADCDZ2	12'b0	Touch Screen pressure measurement data ADCDZ2[23] = MSB ADCDZ2[12] = LSB
11:0 (R) 0x0014	ADCDA0	12'b0	Touch Screen auxiliary 0 measurement data ADCDA0[11] = MSB ADCDA0[0] = LSB
11:0 (R) 0x0018	ADCDA1	12'b0	Touch Screen auxiliary 1 measurement data ADCDA1[11] = MSB ADCDA1[0] = LSB
11:0 (R) 0x001C	ADCDA2	12'b0	Touch Screen auxiliary 2 measurement data ADCDA2[11] = MSB ADCDA2[0] = LSB
29 (R) 0x000C	VALID_X	1'h0	X measurement data valid 0: Invalid 1: Coordinate data is valid
30 (R) 0x000C	VALID_Y	1'h0	Y measurement data valid 0: invalid 1: coordinate data is valid
29 (R) 0x0010	VALID_Z1	1'h0	Z1 measurement data valid 0: Invalid 1: Coordinate data is valid
30 (R) 0x0010	VALID_Z2	1'h0	Z2 measurement data valid 0: Invalid 1: Coordinate data is valid
30 (R) 0x0014	VALID_A0	1'h0	Auxiliary 0 measurement data valid 0: Invalid 1: Auxiliary 0 data is valid
30 (R) 0x0018	VALID_A1	1'h0	Auxiliary 1 measurement data valid 0: Invalid 1: Auxiliary 1 data is valid
30 (R) 0x001C	VALID_A2	1'h0	Auxiliary 2 measurement data valid 0: Invalid 1: Auxiliary 2 data is valid

Bit	Name	Default	Description
9 (R/W) 0x0004	WAIT_READ	1'h0	0: No effects (new ADC data overwrites unread data) 1: New data is being held back, and measurement is delayed until data registers are read

Table 191: TS Read-Back Register

TS Interrupt Status Registers

When the screen is touched, the pen interrupt will be initiated (as long as it is enabled). However, the INT_PEN register can still be set to logic H regardless of whether or not the pen interrupt is enabled. After the measurement data is stored into the corresponding data register according to its measurement type, the data ready interrupt will be initiated if it is enabled. The INT_DATA register can still be set to logic H regardless of the enable/disable state of the data interrupt.

- Touch Screen Interrupt Status Register

Bit	Name	Default	Description
0 (R/W) 0x0008	INT_DATA	1'b0	Data ready interrupt status: 0: No interrupt. 1: Data ready interrupt exists. Writing 1 to this bit will clear the interrupt and writing 0 produces no effects to the interrupt status.
1 (R/W) 0x0008	INT_PEN	1'b0	Pen-down interrupt status: 0: No interrupt. 1: Pen-down interrupt exists. Writing 1 to this bit will clear the interrupt and writing 0 produces no effects to the interrupt status.
4 (R/W) 0x0008	INT_DATA_EN	1'b0	Data ready interrupt enable: 0: Data ready interrupt is disabled. 1: Data ready interrupt is enabled.
5 (R/W) 0x0008	INT_PEN_EN	1'b0	Pen-down interrupt enable: 0: Pen-down interrupt is disabled. 1: Pen-down interrupt is enabled.

Table 192: Touch Screen Interrupt Status Register

Time Control Registers

In order to conduct accurate measurements, the setting time and pre-charge time of the touch screen must be set. The pre-charge time indicates the required time between closing the switch to pre-charge the pen-detector and opening this switch. While the setting time indicates the required time between closing the switch to applying a voltage across the touch screen and sampling the input signals by ADC.

- Time Control Register

Bit	Name	Default	Description
7:4 (R/W) 0x0000	DEL_SET	4'b0	The touch screen setting time before ADC begins to sample input signals: 0000 : No delay 0001 : 8 us 0010 : 16 us 0011 : 32 us 0100 : 48 us 0101 : 64 us 0110 : 96 us 0111 : 128 us 1000 : 192 us 1001 : 256 us 1010 : 384 us 1011 : 512 us 1100 : 768 us 1101 : 1 ms 1110 : 1.28 ms 1111 : 1.5 ms
3:0 (R/W) 0x0004	DEL_DIS	4'b0	The discharge time before the pen detector is initiated: 0000 : No delay 0001 : 8 us 0010 : 16 us 0011 : 32 us 0100 : 48 us 0101 : 64 us 0110 : 96 us 0111 : 128 us 1000 : 192 us 1001 : 256 us 1010 : 384 us 1011 : 512 us 1100 : 768 us 1101 : 1 ms 1110 : 1.28 ms 1111 : 1.5 ms

Bit	Name	Default	Description
7:4 (R/W) 0x0004	DEL_PRE	4'b0	The pre-charge time before the pen detector is initiated: 0000 : No delay 0001 : 8 us 0010 : 16 us 0011 : 24 us 0100 : 32 us 0101 : 48 us 0110 : 64 us 0111 : 96 us 1000 : 128 us 1001 : 160 us 1010 : 192 us 1011 : 256 us 1100 : 320 us 1101 : 384 us 1110 : 448 us 1111 : 500 us

Table 193: Time Control Register

Stream Mode Registers

- TSC Stream Mode Control Register (TSC_STREAM_CTRL) -0x0044

Bit	Name	Default	Description
0 (R/W)	STREAM_EN	1'b0	Stream mode enable. 0: Stream mode is disabled. 1: Stream mode is enabled.
3:1 (R/W)	RATE_CTRL	3'b0	Sampling rate of stream control. 000: 8K SPS 001: 16K SPS 010: 24K SPS 011: 32K SPS 100: 40K SPS 101: 48K SPS Other: 8K SPS
6:4 (R/W)	CHN_SEL	3'b0	Stream channel selection. 000: AUX0 as stream mode channel. 001: AUX1 as stream mode channel.

Bit	Name	Default	Description
			010: AUX2 as stream mode channel. Other: AUX0 as stream mode channel.
9:7 (R/W)	WRAP_CTRL	3'b0	This register controls how to wrap 16-bit FIFO with 12-bit ADC output data in stream mode. 000: Low 4-bit of FIFO data is filled with zero. 001: Low 3-bit and High 1-bit of FIFO data is filled with zero. 010: Low 2-bit and High 2-bit of FIFO data is filled with zero. 011: Low 1-bit and High 3-bit of FIFO data is filled with zero. 100: High 4-bit of FIFO data is filled with zero. Other: Low 4-bit of FIFO data is filled with zero.
31:10 (R)	-	22'b0	Reserved

Table 194: TSC Stream Mode Control Register

- TSC FIFO level check Register (TSC_FIFO_LEVEL_CHK) -0x0048

Bit	Name	Default	Description
3:0 (R/W)	FIFO_SC	4'h4	Stop check in Double word.
9:4 (R)	-	6'h0	Reserved.
13:10 (R/W)	FIFO_LC	4'h8	Low check in Double word.
19:14 (R)	-	6'h0	Reserved.
23:20 (R/W)	FIFO_HC	4'hC	High check in Double word.
31:24 (R)	-	8'h0	Reserved.

Table 195: TSC FIFO level check Register

- TSC FIFO Operation Register (TSC_FIFO_OP) -0x004C

Bit	Name	Default	Description
0 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is set.
1 (R/W)	FIFO_RESET	1'b0	Set to 1 to stop the FIFO and reset the FIFO internal status including its relevant interrupt status. Set to 0 in normal operation.
31:2 (R)	-	30'h0	Reserved.

Table 196: TSC FIFO Operation Register

- TSC FIFO Status Register (TSC_FIFO_STS) -0x0050

Bit	Name	Default	Description
0 (R)	FIFO_FULL	1'b0	FIFO full status: 1: FIFO is in full state 0: FIFO is not in full state. It indicates the current FIFO full status, once the FIFO status is changed, the status bit will be cleared automatically.
1 (R)	FIFO_EMPTY	1'b1	FIFO empty status: 1: FIFO is in empty state 0: FIFO is not in empty state It indicates the current FIFO empty status, once the FIFO status is changed, the status bit will be cleared automatically.
2 (R/W)	FIFO_OFLOW	1'b0	FIFO overflow status: 1: FIFO overflow takes place. 0: FIFO is not overflow. It indicates the current FIFO over flow, once the FIFO status is changed, the status bit will be cleared automatically. Also user can write 1'b1 to clear the register bit after the FIFO overflow takes place.
3 (R/W)	FIFO_UFLOW	1'b0	FIFO underflow status: 1: FIFO underflow takes place. 0: FIFO is not underflow. It indicates the current FIFO under flow, once the FIFO status is changed, the status bit will be cleared automatically. Also user can write 1'b1 to clear the register bit after the FIFO overflow takes place.
9:4 (R)	FIFO_LEVEL	6'h0	The byte count of the valid data in the FIFO varies from 0 to 63 bytes. In case FIFO is full, the value of this register will be set to 0, thus users must concatenate the FIFO_FULL bit with this value to determine the actual data count in it.
31:10 (R)	-	22'h0	Reserved.

Table 197: TSC FIFO Status Register

- TSC FIFO Interrupt Register (TSC_FIFO_INTR) -0x0054

Bit	Name	Default	Description
0 (R/W)	INT_FIFO_FULL_EN	1'b0	1: FIFO full interrupt is enabled. 0: FIFO full interrupt is disabled.
1 (R/W)	INT_FIFO_EMPTY_EN	1'b0	1: FIFO empty interrupt is enabled. 0: FIFO empty interrupt is disabled.
2 (R/W)	INT_FIFO_OFLOW_EN	1'b0	1: FIFO overflow interrupt is enabled. 0: FIFO overflow interrupt is disabled.
3 (R/W)	INT_FIFO_UFLOW_EN	1'b0	1: FIFO underflow interrupt is enabled. 0: FIFO underflow interrupt is disabled.
4 (R/W)	INT_FIFO_OFLOW	1'b0	FIFO overflow interrupt: 1: FIFO overflow takes place. 0: FIFO is not overflow. User can write 1'b1 to clear the register bit after the FIFO overflow takes place. After the overflow, the hardware can not clear this bit even the FIFO is out of overflow.
5 (R/W)	INT_FIFO_UFLOW	1'b0	FIFO underflow interrupt: 1: FIFO underflow takes place. 0: FIFO is not underflow. Users need to write 1'b1 to clear the register bit after the FIFO underflow takes place. After the underflow, the hardware can not clear this bit even the FIFO is out of underflow.
31:6 (R)	-	26'h0	Reserved.

Table 198: TSC FIFO Interrupt Enable Register

GRAPHICS AND MULTIMEDIA SUBSYSTEM

LCD Controller

Overview

The LCD controller is used for displaying images to an external display panel and can be used on GPS Navigation/Telematics systems, as well as PDAs and smart phones.

Feature List

- Active color mode, supported source pixel data up to 16777216 colors (24 bits), and 16 bits output data bus
- Resolution up to 1024x1024 pixels is supported, but is limited by memory bus bandwidth
- Two layers (typically one for primary surface, the other for overlay surface)
- One hardware cursor layer (2bits/pixel, 64X64 or 32X32 pixels each)
- On-chip RGB to YUV hardware conversion and interface to the TV encoder (8-bit 4:2:2 format or 16-bit 4:2:2 format)
- Programmable frame and line clock polarity, pulse width and starting location; programmable polarity for output enable and toggle frequency of the L_DE output pin
- Programmable pixel clock frequency and polarity
- Master and slave modes are supported
- Depending on the type of panel used, the LCD controller can be programmed to use 8- and 16-pixel data output pins

Pin Description

External Pin Descriptions

The following table shows the pins used in the LCD controller and their functions. The LCD controller pins are multiplexed with other devices.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_LDD<15:0>	O	ROM	X_LDD<15:0> : LCD	output low	LCD Data Bus: Provides 8- and 16-bit data at a time to the LCD panel. Either the bottom eight pins (LDD<7:0>) or the bottom 16 pins (LDD<15:0>) will be used. For active displays, all used data pins represent a single pixel.
X_L_PCLK	I/O	ROM	LCD	Input pull down	LCD Shift / Pixel Clock (programmable polarity) for screen: Used by the LCD panel to latch the pixel data and control signals.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_L_LCK	I/O	ROM	LCD	Input pull down	LCD Line Clock (programmable polarity) for screen: Used by the LCD display to signal the beginning of a line. Horizontal sync.
X_L_FCK	I/O	ROM	LCD	Input pull down	LCD Frame Clock (programmable polarity) for screen: Used by the LCD display to signal the start of a new frame. Vertical sync.
X_L_DE	O	GPIO/ROM	GPIO	Output low	AC Bias or Data Enable for screen: Data Enable (DE).

Table 199: LCD External Pin Descriptions

Functional Descriptions

Block Diagram

Data flow:

There are two layers within the LCD controller; these layers request data from the external frame buffer. LCD_DMA_arbiter arbitrates those requests and grants a layer, then sends the request to the system arbiter. The system arbiter then grants that request and provides data from the frame buffer. Data from the cursor layer is provided by the SRAM configured by CPU. These two layers and the cursor layer will then perform alpha blending, which will result in RGB to YUV conversion. Finally the results will be sent to the LCD panel or a TV encoder.

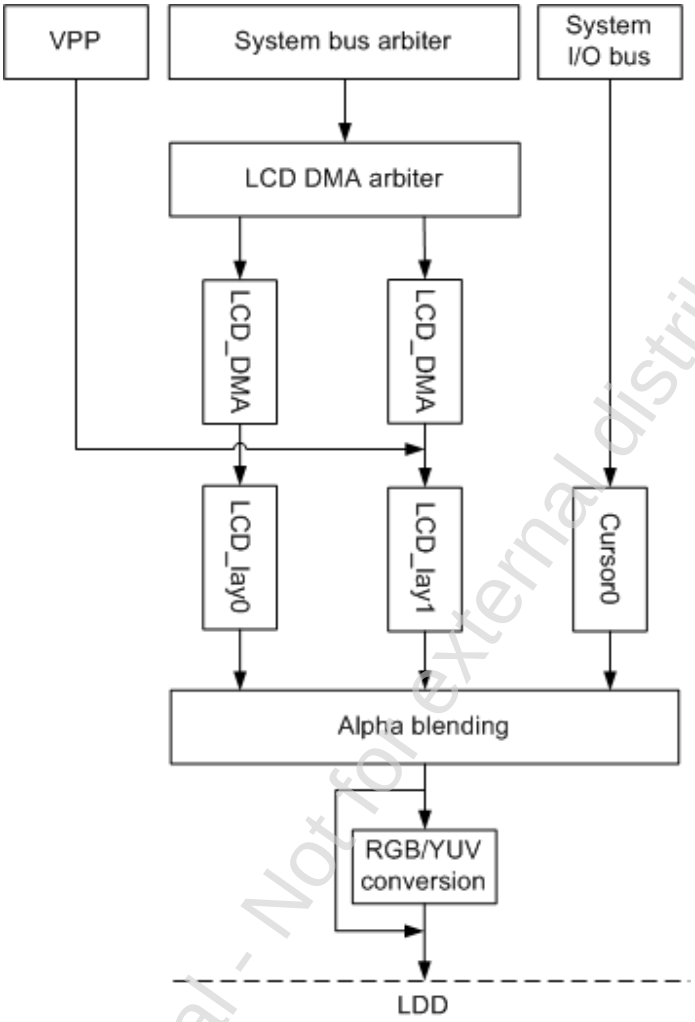


Figure 33: LCD Controller Block Diagram

Top Module

There are three interfaces in the LCD block:

- The layer interface, which processes the pixel data from the frame buffer, the two layers are independent.
- The cursor interface, which processes the cursor data from the internal SRAM.
- The screen interface, which processes timing control with an external panel and outputs the required pixel data.

Layer Interface

Layer Display

There are total of three layers which can be displayed on the screen. The following figure is a display example.

If different layers are overlapped, then alpha blending should be applied. However at some points when no layers are overlapped, alpha blending is still applicable with pixel value 0 (of those layers that contain no pixels).

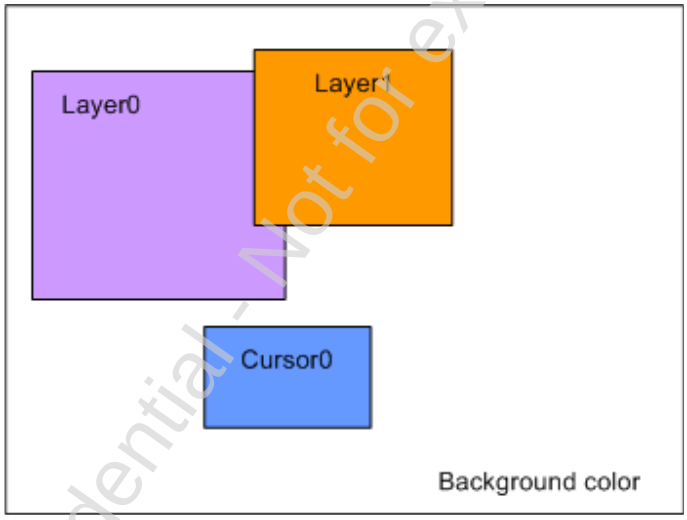


Figure 34: Display Example

Different layer can be configured to different pixel data format. For example, layer 0 pixel format is 16-bit RGB565 data, layer1 pixel format can be 24-bit RGB888 data.

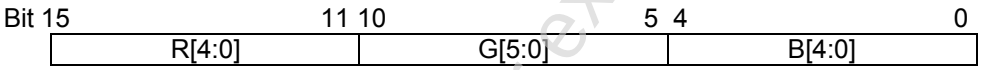
Layer Data Format from Frame Buffer

The LCD controller supports several formats that contain raw RGB data. Each layer can be programmed independently to support one pixel format.

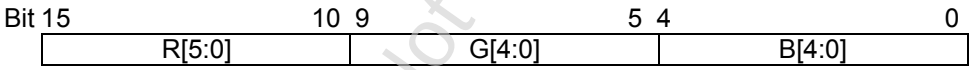
Memory Depth	RGB Format	Description
16-bit	RGB565 (5-bit R, 6-bit G and 5-bit B) RGB655 (6-bit R, 5-bit G and 5-bit B)	A: Alpha value T: T-bit R: Red G: Green B: Blue
18-bit	RGB666 (6-bit R, 6-bit G and 6-bit B)	
32-bit	RGB888 (8-bit Reserved, 8-bit R, 8-bit G and 8-bit B) TRGB888 (1-bit T, 8-bit R, 8-bit G and 8-bit B) ARGB888 (8-bit A, 8-bit R, 8-bit G and 8-bit B)	

Table 200: Layer Pixel Data Format

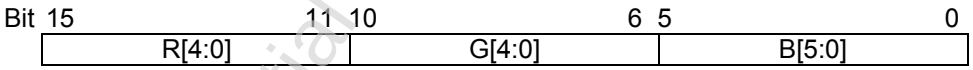
- Data format for 16-bit RGB565:



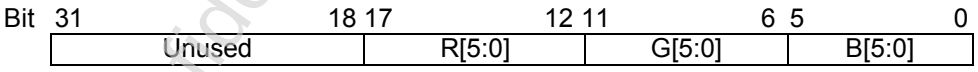
- Data format for 16-bit RGB655:



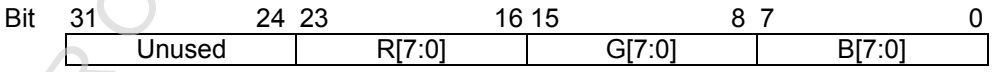
- Data format for 16-bit RGB556:



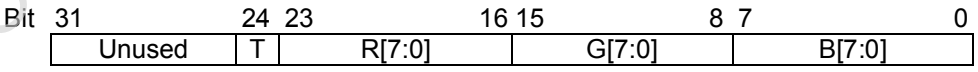
- Data format for 18-bit RGB:



- Data format for 32-bit RGB:



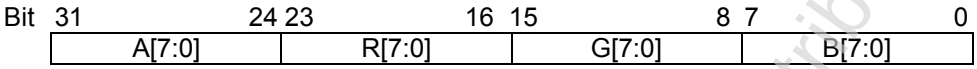
- Data format for 32-bit TRGB888:



T-bit:

- 1: The alpha value for this pixel is 0x00;
- 0: The alpha value of this layer is being used.

- Data format for 32-bit ARGB8888:



Encoded pixel data of the frame buffer is stored in the off-chip memory (usually DRAM), and is transferred to the LCD controller's FIFO. User can select how the LCD views the ordering of the frame buffer pixel by programming big/little-endian selection bits.

There are two levels of endian selection for a dword data from the memory. The following figures are examples for these two levels in different bit/pixel formats:

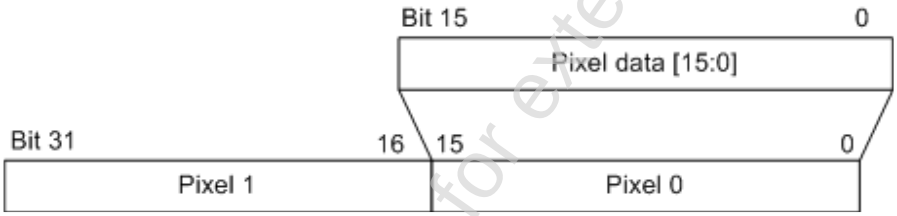


Figure 35: 16-Bit/Pixel Data in a Dword (Little-Endian)

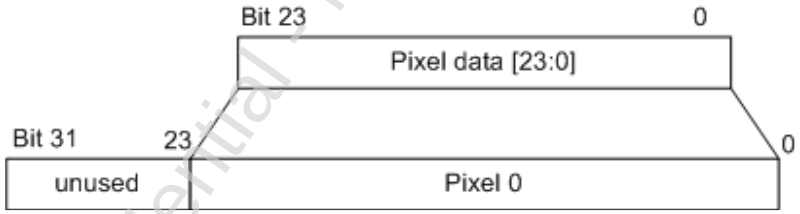


Figure 36: 24-Bit/Pixel Data in a Dword (Little-Endian)

The required bandwidth of each layer is depends on the pixel data format. For example, when one layer is configured to 320x240 resolution with RGB565 pixel data format, the required bandwidth should be 320x240x2, each RGB565 pixel data occupies 2 bytes at the memory side.

Color Key

Layer0 to layer1 support the color key function, two registers (CKEYB and CKEYS) are used to specify a color key range. If a color of the overlay layer is between CKEYB and CKEYS, then the color will be transparent and the pixel from the lower layer will be shown instead.

When the pixel data format of layer is configured to TRGB mode, the color will be transparent and the pixel from the lower layer will be shown instead if the bit24 of pixel data is 1.

Alpha Blending and Keying Principle

When a front layer and a background layer are being mixed, the mixer output will contain partial background and partial foreground.

Alpha blending gives the flowing result:

Mixer_out = alpha*FG + (1 –alpha)*BG

Where BG stands for background and FG stands for foreground.

In the formula, alpha is a variable between 0 and 1. The alpha value is decided by the LCD_Lx_ALPHA of the foregourd or the alpha from pixel when the pixel data format of foreground is configured to ARGB mode.

If the alpha variable is 1, mixer_out equals to FG (which means only the current layer will be displayed, the same applies to alpha variable 0). On the other hand, if the alpha variable is 0, mixer_out equals to BG.

If the alpha variable is less than 1 and greater than 0, then the mixer output will contain both BG and FG.

When the pixel data format of layer is configured to ARGB mode, the alpha value specified by LCD_Lx_ALPHA is invalid and the hardware will get the alpha value from each pixel data.

Therefore the output data should be:

Mixer_out = alpha_from_pixel x FG + (1 - alpha_from_pixel) x BG

Alpha Blending Implementation

By default, layer0 is the bottom layer, and layer1 is on top of layer0. The cursor layer is the top layer. Users can select and configure one of the three layers to be the top layer. For example, if layer1 is configured to be the top layer, then the layers will be shown in the following sequence:

Layer0 is the bottom layer, then cursor, on top of the cursor layer is layer1.

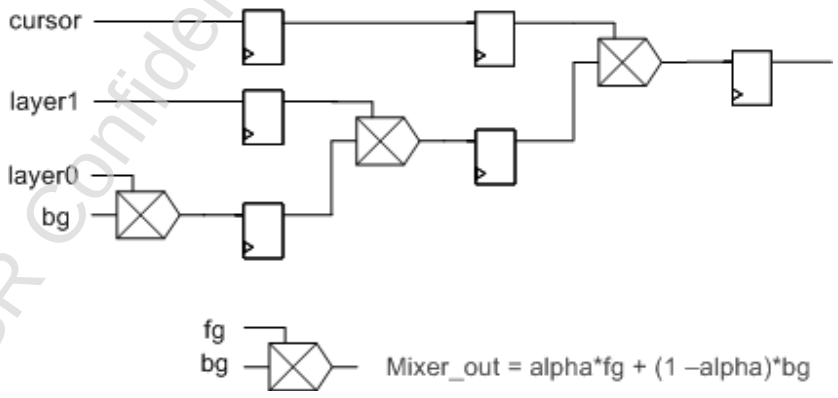


Figure 37: Default Alpha Blending Implementation

The default alpha blending implementation is shown in the figure above. Alpha values can be different values for different layers. Alpha blending operations are pipelined.

The following figure illustrates a typical implementation of alpha blending. In this example, layer1 is configured as the top layer.

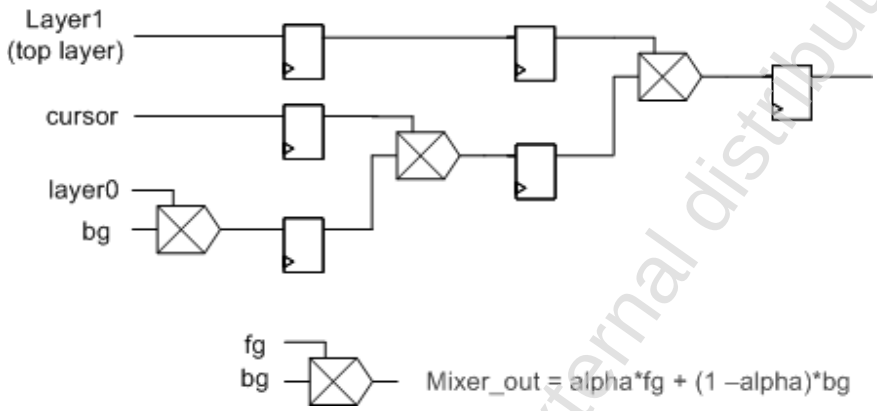


Figure 38: Alpha Blending Implementation Example (Top Layer = Layer 1)

DMA Operation

The LCD controller accesses the frame buffer data through its two 2-dimensional DMA channels (layer0-layer1) with each corresponding to one layer. The X size determines how many consecutive DMA bursts of data are in a line, whereas the Y size determines how many lines are required for the current display. The SKIP value determines the number of bytes to skip at the end of a line in order to get to the beginning of the next line, and the base address specifies the starting address of DMA. The start address for each line must be a QWORD alignment address.

A control bit allows each DMA to operate in continuous mode. This means that the DMA will repeat itself automatically (with the exact same settings) after it is completed for continuous display. This saves the software from always having to generate DMA requests. Clearing this bit can stop the continuous DMA.

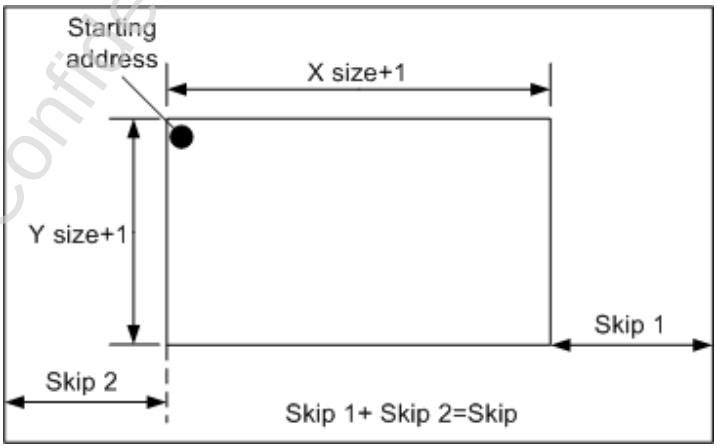


Figure 39: LCD Controller DMA Memory 2-D Layout

In case the QWORD width of the output image is not a multiple of the DMA burst length, some dummy spaces will then need to be reserved at each line in order for the frame buffer QWORD width to be a multiple of the DMA burst length. The following figure illustrates this example:

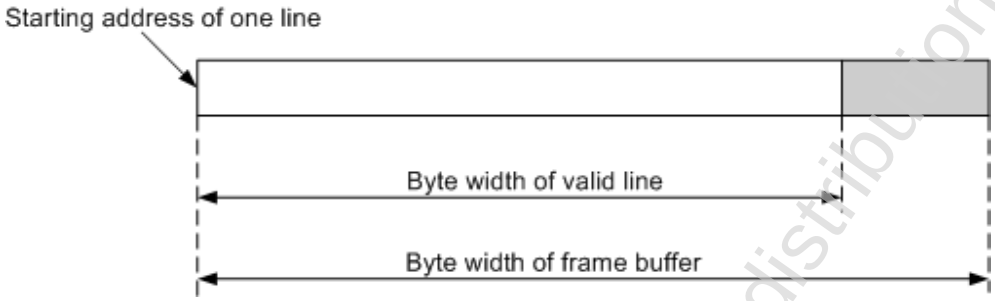


Figure 40: LCD Controller Line Frame Buffer Layout

A DMA chain is also supported; each layer has an extra memory address register. When the DMA chain and continuous the DMA mode are enabled, DMA will first be processed from address0 and then address1 after the last DMA is completely processed (with same X and Y configurations of last DMA), then address0 will be followed. This function can be used for both TV output mode and ping-pong frame buffer mode.

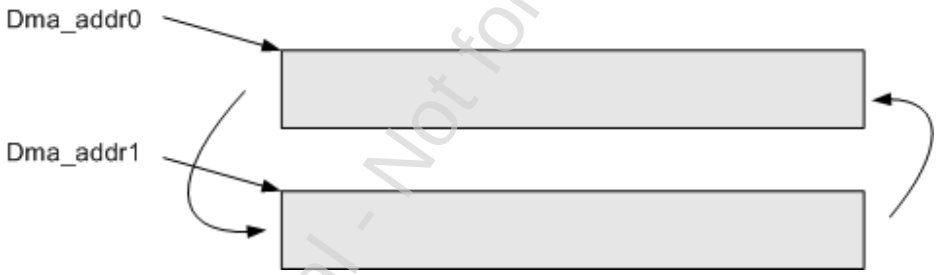


Figure 41: LCD Controller DMA Chain

Each Layer (layer0-1) has a FIFO with 256*64 bits.

DMA burst length can be configured to the following sizes: 4 QWord, 8 QWord, 16 QWord.

Data from VPP

Layer1 input data can comes directly from the VPP module. If the DEST bit of the VIDEO_PRE_CTRL register is set, then VPP will output data to layer1 of the LCD controller directly and DMA of layer1 will be stopped. Layer0 will keep fetching data from the frame buffer.

Hardware Cursors Interface

The LCD controller provides a hardware cursor that can be disabled or configured to the following modes:

- 32x32x2bpp 2-color and transparency mode
- 64x64x2bpp 2-color and transparency mode
- 32x32x2bpp 4-color mode
- 64x64x2bpp 4-color mode
- 32x32x2bpp 3-color and transparency mode
- 64x64x2bpp 3-color and transparency mode

The cursor data is stored in its internal 256x32 bit SRAM. Users can enable, disable and configure the cursor by programming the Cursor Control register.

32x32x2bpp and 64x64x2bpp 2-color and transparency modes , these two modes follow the Microsoft® Windows cursor data plane structure. Each pixel has 2 bits, which represent four colors:

- Two colors are used for drawing a cursor.
- The third color is used for transparency. This allows the main display image behind the cursor to show through.
- The fourth color is for inverted transparency. This allows the main display image behind the cursor to show through, but with inverted color value.

Bits/Pixel	Color Displayed at Corresponding Pixel Position
00	Cursor color 0
01	Cursor color 1
10	Transparent. The pixel of the main display image behind the cursor is shown through.
11	Transparent, but inverted. The pixel of the main display image behind the cursor is shown through with inverted color.

Table 201: 2-Color and Transparency Mode Data Format

Bits/Pixel	Color Displayed at Corresponding Pixel Position
00	Cursor color 0
01	Cursor color 1
10	Cursor color 2
11	Cursor color 3

Table 202: 4-Color Mode Data Format

Bits/Pixel	Color Displayed at Corresponding Pixel Position
00	Cursor color 0
01	Cursor color 1
10	Cursor color 2
11	Transparent

Table 203: 3-Color and Transparency Mode Data Format

The following figure shows one of the formats in which the pixel data is stored in the internal SRAM:

Bit	31 30	29 28	27 26	7 6	5 4	3 2	1 0
Addr0	**Pixel 15**	**Pixel 14**	**Pixel 13**	**-**	**Pixel 3**	**Pixel 2**	**Pixel 1**	**Pixel 0**
Addr4	Pixel 31	Pixel 30	Pixel 29	-	Pixel 19	Pixel18	Pixel 17	Pixel 16

Screen Interface

RGB to YUV Conversion

This function mainly is targeted for video encoders. There is a set of registers which contain coefficients to convert RGB outputs to YUV. The conversion is a 3x3 matrix multiplication shown as follows:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} C11 & C12 & C13 \\ C21 & C22 & C23 \\ C31 & C32 & C33 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

NOTE – This multiplication is a signed multiplication.

The basic function unit: Out=Cx1*R+Cx2*G+Cx3*B (x = 1, 2, 3). Three such computation units are used to get Y, U and V. Currently the booth-4 coding multiplier and CSA adders are used to implement such function unit. This function can be bypassed if necessary.

Data Format Output to LCD Panels

The following table describes the LCD pin allocation for various types of external LCD panels. It is independent with the pixel format of each layer. Whatever the LCD panel type is, the pixel format of layer can be programmed to required type according to the system requirement.

Data Type	Pins	Pixel Format
8-bit	X_LDD<7:0>	RGB□X_LCD<7:0> YCrYCb □X_LDD<7:0>
16-bit	X_LDD<15:0>	YcrCb□X_LDD<15:0> RGB565 R<7:3>→X_LDD<15:11>

Data Type	Pins	Pixel Format
		G<7:2>→X_LDD<10:5> B<7:3>→X_LDD<4:0>

Table 204: Data Pins Allocation for Different Output Pannels

	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	…
Row 0	R00	G00	B00	R01	G01	B01	…
Row 1	R10	G10	B10	R11	G11	B11	…
Row 2	R20	G20	B20	R21	G21	B21	…
Row 3	R30	R31	B31	R31	G31	B31	…

Table 205: Pixel Output Order for 8-Bit RGB Mode

	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5
Row 0	Y00	Cr00	Y01	Cb01	Y02	Cr02
Row 1	Y10	Cr10	Y11	Cb11	Y12	Cr12
Row 2	Y20	Cr20	Y21	Cb21	Y22	Cr22
Row 3	Y30	Cr30	Y31	Cb31	Y32	Cr32

Table 206: Pixel Output Order for 8-Bit YCrCb Mode

	Column 0	Column 1	Column 2
Row 0	Y00 Cr00	Y01 Cb01	Y02 Cr02
Row 1	Y10 Cr10	Y11 Cb11	Y12 Cr12
Row 2	Y20 Cr20	Y21 Cb21	Y22 Cr22
Row 3	Y30 Cr30	Y31 Cb31	Y32 Cr32

Table 207: Pixel Output Order for 16-Bit YCrCb Mode

	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5
Row 0	P00	P01	P02	P03	P04	P05
Row 1	P10	P11	P12	P13	P14	P15
Row 2	P20	P21	P22	P23	P24	P25
Row 3	P30	P31	P32	P33	P34	P35

Table 208: Pixel Output Order for 16-Bit RGB Mode

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
LCD controller base address through RISC I/O	0xb8000000

Table 209: LCD Controller Address Mapping

Register Mapping

RISC Address <12:0>	Register	Description
0x0000	LCD_S0_HSYNC_PERIOD	Period for Master Mode Horizontal Sync
0x0004	LCD_S0_HSYNC_WIDTH	Width for Master Mode Horizontal Sync
0x0008	LCD_S0_VSYNC_PERIOD	Period for Master Mode Vertical Sync
0x000C	LCD_S0_VSYNC_WIDTH	Width for Master Mode Vertical Sync
0x0010	LCD_S0_ACT_HSTART	Horizontal start position for active screen
0x0014	LCD_S0_ACT_VSTART	Vertical start position for active screen
0x0018	LCD_S0_ACT_HEND	Horizontal end position for active screen
0x001C	LCD_S0_ACT_VEND	Vertical end position for active screen
0x0020	LCD_S0_OSC_RATIO	OSC_PIXCLK divider ratio register
0x0024	LCD_S0_TIM_CTRL	Timing control register
0x0028	LCD_S0_TIM_STATUS	Timing control status register
0x002C	LCD_S0_HCOUNT	Horizontal counter value
0x0030	LCD_S0_VCOUNT	Vertical counter value
0x0034	LCD_S0_BLANK	Blanking register

RISC Address <12:0>	Register	Description
0x0038	LCD_S0_BACK_COLOR	Background color register
0x003C	LCD_S0_DISP_MODE	Display mode and format register
0x0040	LCD_S0_LAYER_SEL	Select layer display at screen0
0x0044	LCD_S0_RGB_SEQ	Sequence for RGB outputs
0x0048	LCD_S0_RGB_YUV_COEF1	RGB to YUV Coefficient 1
0x004C	LCD_S0_RGB_YUV_COEF2	RGB to YUV Coefficient 2
0x0050	LCD_S0_RGB_YUV_COEF3	RGB to YUV Coefficient 3
0x0054	LCD_S0_YUV_CTRL	RGB to YUV conversion control register
0x0058	LCD_S0_TV_FIELD	Screen control for TV mode
0x005C	LCD_S0_INT_LINE	Interrupt line number
0x0060	LCD_S0_LAYER_STATUS	Layer enable status
0x00F0	LCD_DMA_STATUS	DMA status.
0x00F4	LCD_INT_MASK	Interrupt enable.
0x00F8	LCD_INT_CTRL	Interrupt status.
0x00FC	LCD_SCR_CTRL	Enable the screen.
0x0100	LCD_L0_CTRL	Control register for L0
0x0104	LCD_L0_HSTART	Horizontal start position for L0
0x0108	LCD_L0_VSTART	Vertical start position for L0
0x010C	LCD_L0_HEND	Horizontal end position for L0
0x0110	LCD_L0_VEND	Vertical end position for L0
0x0114	LCD_L0_BASE0	Screen memory base register0
0x0118	LCD_L0_BASE1	Screen memory base register1
0x011C	LCD_L0_XSIZE	X Size for screen DMA
0x0120	LCD_L0_YSIZE	Y Size for screen DMA
0x0124	LCD_L0_SKIP	Skip value for screen DMA
0x0128	LCD_L0_DMA_CTRL	Control register of the DMA
0x012C	LCD_L0_ALPHA	L0 Alpha blending control register
0x0130	LCD_L0_CKEYB	L0 color key big RGB value
0x0134	LCD_L0_CKEYS	L0 color key small RGB value
0x0138	LCD_L0_FIFO_CHK	L0 FIFO control register
0x013C	LCD_L0_FIFO_STATUS	Status register of FIFO

RISC Address <12:0>	Register	Description
0x0200	LCD_L1_CTRL	Control register for L1
0x0204	LCD_L1_HSTART	Horizontal start position for L1
0x0208	LCD_L1_VSTART	Vertical start position for L1
0x020C	LCD_L1_HEND	Horizontal end position for L1
0x0210	LCD_L1_VEND	Vertical end position for L1
0x0214	LCD_L1_BASE0	Screen Memory Base Register0
0x0218	LCD_L1_BASE1	Screen Memory Base Register1
0x021C	LCD_L1_XSIZE	X Size for screen DMA
0x0220	LCD_L1_YSIZE	Y Size for screen DMA
0x0224	LCD_L1_SKIP	Skip value for screen DMA
0x0228	LCD_L1_DMA_CTRL	Control register of the DMA
0x022C	LCD_L1_ALPHA	L1 Alpha blending control register
0x0230	LCD_L1_CKEYB	L1 color key big RGB value
0x0234	LCD_L1_CKEYS	L1 color key small RGB value
0x0238	LCD_L1_FIFO_CHK	L1 FIFO control register
0x023C	LCD_L1_FIFO_STATUS	Status register of FIFO
0x1000	LCD_CUR0_CTRL	Control register for CUR0
0x1004	LCD_CUR0_HSTART	Horizontal start position for CUR0
0x1008	LCD_CUR0_VSTART	Vertical start position for CUR0
0x100C	LCD_CUR0_HEND	Horizontal end position for CUR0
0x1010	LCD_CUR0_VEND	Vertical end position for CUR0
0x1014	LCD_CUR0_COLOR0	Color 0
0x1018	LCD_CUR0_COLOR1	Color 1
0x101C	LCD_CUR0_COLOR2	Color 2
0x1020	LCD_CUR0_COLOR3	Color 3
0x1024	LCD_CUR0_ALPHA	CUR0 alpha blending control register
0x1028	LCD_CUR0_FIFO_RDPTR	Read address of SRAM
0x1400~0x17FC	LCD_CUR0_FIFODATA	Read/write screen FIFO

Table 210: LCD Controller Register Definition

Register Description

Sync Signal Generation Registers

In the slave mode, the horizontal and vertical sync signals are output by the display. However in the master mode, the LCD controller will generate sync signals based on parameters such as period and width of each sync pulse.

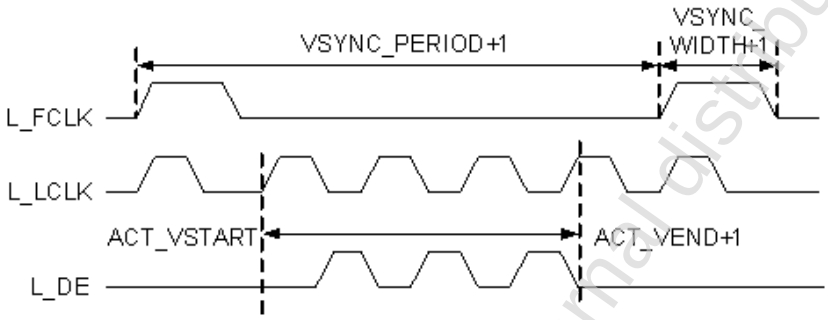


Figure 42: VSYNC Output

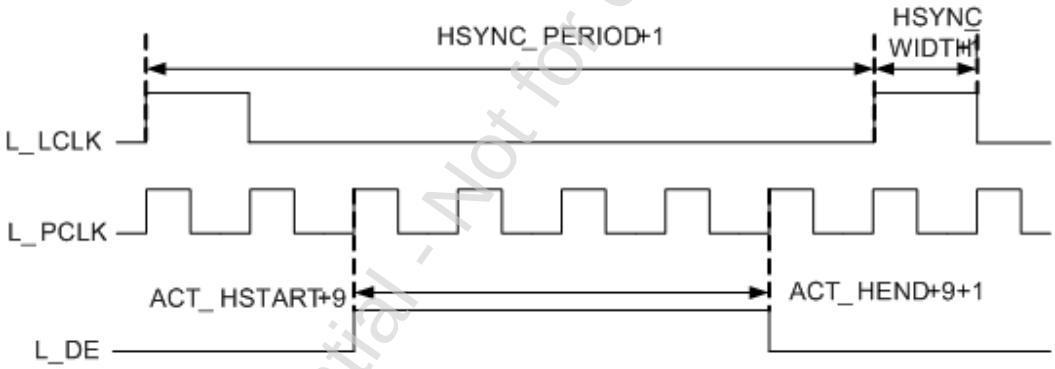


Figure 43: HSYNC Output

- Period for Master Mode Horizontal Sync (LCD_S0_HSYNC_PERIOD) – 0x0000

Bit	Name	Default	Description
10:0 (R/W)	HSYNC_PERIOD	11'h0	In the master mode where the LCD controller generates sync signals, this value will define the period of the horizontal sync pulse in numbers of pixels. If this register is set to 60, then the sync period is 61 pixel clocks. In slave mode, this register produces no effects.
31:11	-	21'h0	Reserved.

Table 211: Period for Master Mode Horizontal Sync (LCD_S0_HSYNC_PERIOD)

- Width for Master Mode Horizontal Sync (LCD_S0_HSYNC_WIDTH) – 0x0004

Bit	Name	Default	Description
10:0 (R/W)	HSYNC_WIDTH	11'h0	For master mode, where LCD controller generates the sync signals, this value will define the width (minus 1) of the horizontal sync pulse in terms of number of pixels. If this register is set to 8, then the width of the sync pulse is 9 pixel clocks. In slave mode, this register produces no effects.
31:11	-	21'h0	Reserved.

Table 212: Width for Master Mode Horizontal Sync (LCD_S0_HSYNC_WIDTH)

- Period for Master Mode Vertical Sync (LCD_S0_VSYNC_PERIOD) – 0x0008

Bit	Name	Default	Description
10:0 (R/W)	VSYNC_PERIOD	11'h0	In the master mode where the LCD controller generates the sync signals, this value will define the period of the vertical sync pulse in terms of number of lines. In slave mode, this register produces no effects. The actual vertical line number is VSYNC_PERIOD+1.
31:11	-	21'h0	Reserved.

Table 213: Period for Master Mode Vertical Sync (LCD_S0_VSYNC_PERIOD)

- Width for Master Mode Vertical Sync (LCD_S0_VSYNC_WIDTH) – 0x000C

Bit	Name	Default	Description
10:0 (R/W)	VSYNC_WIDTH	11'h0	In the master mode where the LCD controller generates the sync signals, this value will define the width of the horizontal sync pulse, in either number of lines or number of pixels (on bit 11 above). In the slave mode, this register produces no effects. The actual width is VSYNC_WIDTH+1 (lines or pixels)
11 (R/W)	WIDTH_UNIT	1'b0	1: Vertical sync pulse width defined below is in number of lines. 0: Vertical sync pulse width defined below is in number of pixels. Usually, this bit is set to 1.
31:12	-	20'h0	Reserved

Table 214: Width for Master Mode Vertical Sync (LCD_S0_VSYNC_WIDTH)

The following conditions must be fulfilled:

- HSYNC_PERIOD > HSYNC_END
- VSYNC_PERIOD > VSYNC_END
- HSYNC_PERIOD > MAX (L0_HEND~L3_HEND, CUR0_HEND)
- VSYNC_PERIOD > MAX (L0_VEND~L3_VEND, CUR0_VEND)

Active Region Definition Registers

The active region of the LCD display is specified by a rectangle defined by two coordinates, which are the horizontal with vertical start position, and the horizontal with vertical end position, as shown in the diagram below. The LCD display contains internal counters, which count pixels by pixel clock and lines by horizontal sync. The pixel counts are cleared by horizontal sync and the line counts are cleared by vertical sync.

The following conditions must be satisfied for displaying an M (pixels) x N (lines) layer output in 18-bit RGB mode:

- M = LCD_Ln_HEND–LCD_Ln_HSTART+1
- N = LCD_Ln_VEND–LCD_Ln_VSTART+1

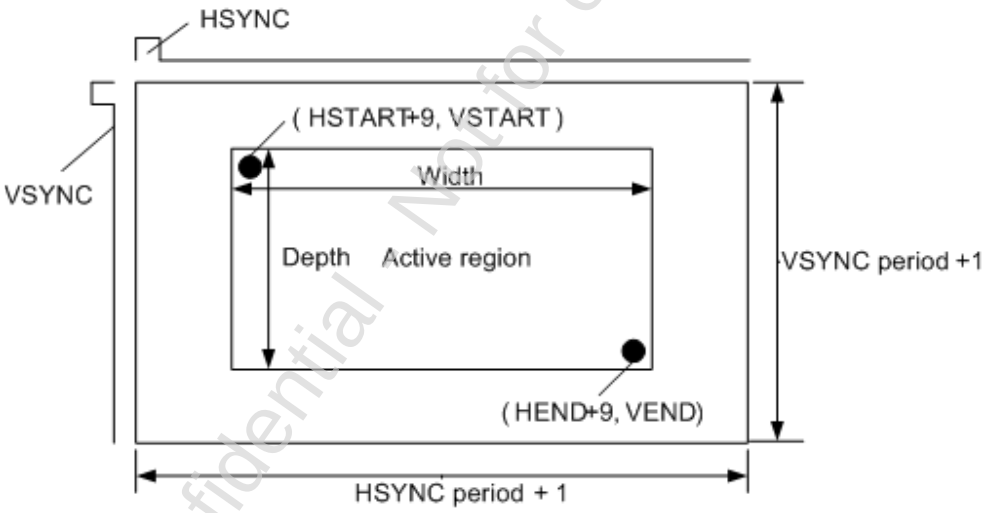


Figure 44: Screen Active Region

As show in the figure above, only pixels inside the active window will be visible, pixels outside the active window will be clipped.

- Horizontal Start Position for Active Screen (LCD_S0_ACT_HSTART) – 0x0010

Bit	Name	Default	Description
10:0 (R/W)	ACT_HSTART	11'h0	Horizontal start position (in pixel number) This value, along with the horizontal end position and vertical start and end positions, defines the rectangular region of the active window. The X_L_DE will be valid after (ACT_HSTART+9) pixel clocks when the X_L_LCK valid.
31:11	-	21'h0	Reserved

Table 215: Horizontal Start Position for Active Screen (LCD_S0_ACT_HSTART)

- Vertical Start Position for Active Screen (LCD_S0_ACT_VSTART) – 0x0014

Bit	Name	Default	Description
10:0 (R/W)	ACT_VSTART	11'h0	Vertical start position (in line number)
31:11	-	21'h0	Reserved

Table 216: Vertical Start Position for Active Screen (LCD_S0_ACT_VSTART)

- Horizontal End Position for Active Screen (LCD_S0_ACT_HEND) – 0x0018

Bit	Name	Default	Description
10:0 (R/W)	ACT_HEND	11'h0	In pixel clock numbers In 8-bit RGB mode: ACT_HSTART + (WIDTH – 1)*3 In 8-bit YUV mode: ACT_HSTART + (WIDTH – 1)*2 Others: ACT_HSTART + WIDTH – 1
31:11	-	21'h0	Reserved

Table 217: Horizontal End Position for Active Screen (LCD_S0_ACT_HEND)

- Vertical End Position for Active Screen (LCD_S0_ACT_VEND) – 0x001C

Bit	Name	Default	Description
10:0 (R/W)	ACT_VEND	11'h0	In line numbers. ACT_VSTART + DEPTH - 1
31:11	-	21'h0	Reserved

Table 218: Vertical End Position for Active Screen (LCD_S0_ACT_VEND)

Screen Control Registers

The following registers specify modes and parameters for the display. For example:

- The timing control registers determine whether the pixel clocks and the sync signals are driven by the LCD controller or by the display, and whether the signals are active high or active low.
- The display mode registers set the display data and the output format.
- The RGBSEQ register determines the pixel sequence for those displays that require one color at a time.
- The BLANK registers define the output value for the blank region.

Also, to accommodate certain displays, the pixel clock can be masked by setting the mask pixel clock bits (bit8 of LCD_S0_TIM_CTRL). The HSYNC signal can also be masked by setting bit 9 of LCD_S0_TIM_CTRL.

- OSC_PIXCLK Divider Ratio Register (LCD_S0_OSC_RATIO) – 0x0020

Bit	Name	Default	Description
9:0 (R/W)	DIV_RATIO	10'h0	The pixel clock is divided from the system clock when Prism-lite drives the pixclk: Fpixclk = Fsys_clk / (DIV_RATIO+1) The minimum DIV_RATIO value is 1.
11:10	-	2'b00	Reserved
12 (R/W)	HALF_DUTY	1'b0	1: Generates pixclk of 50% duty cycle when the divider ratio (DIV_RATIO<9:0>) is even. 0: Generates pixclk is not 50% duty cycle clock when the divider ratio (DIV_RATIO<9:0>) is even.
15:13		3'b000	Reserved
16 (R/W)	PCLK_CTRL	1'b0	When setting this bit, the output pixel clock can be latched when one of the layer FIFO lengths is 0 0: Pixel clock work normal. 1: Pixel clock will be latched when FIFO is about to be underflowed.
31:16	-	15'h0	Reserved

Table 219: OSC_PIXCLK Divider Ratio Register (LCD_S0_OSC_RATIO)

- Timing Control Register (LCD_S0_TIM_CTRL) – 0x0024

Bit	Name	Default	Description
0	-	1'b0	Reserved
1 (R/W)	PCLK_IO	1'b0	Pixel clock master mode 1: LCD controller drives the pixel clock. 0: The display drives the pixel clock.
2 (R/W)	PCLK_POLAR	1'b0	Invert pixel clock. In master mode, this register inverts the internal pixel clock before output. In slave mode, it inverts the input pixel clock before it is used by the internal logic. 1: Inverts pixel clock (i.e. pixel clock ½ phase off). 0: Do not invert pixel clock.
3 (R/W)	PCLK_EDGE	1'b0	Determines whether the pixel output changes on the rising or falling edge of the internal pixel clock. 1: Pixel changes on the rising edge of the clock. 0: Pixel changes on the falling edge of the clock.
4 (R/W)	HSYNC_IO	1'b0	Horizontal sync signal master mode: 1: LCD controller drives the horizontal sync. 0: The display drives horizontal sync.
5 (R/W)	HSYNC_POLAR	1'b0	This register inverts the horizontal sync signal. In master mode, it inverts the horizontal sync signal before it is output. In slave mode, it inverts the horizontal sync signal before it is used by the internal logic. 1: Horizontal sync signal is active low. 0: Horizontal sync signal is active high.
6 (R/W)	VSYNC_IO	1'b0	Vertical sync signal master mode: 1: LCD controller drives the vertical sync. 0: The display drives the vertical sync.
7 (R/W)	VSYNC_POLAR	1'b0	This register inverts the vertical sync signal. In master mode, it inverts the vertical sync signal before it is output. In slave mode, it inverts the vertical sync signal before it is used by the internal logic. 1: Vertical sync signal is active low. 0: Vertical sync signal is active high.
8 (R/W)	PCLK_MASK	1'b0	Test Purpose register, software writes this with 0 Remark: Mask pixel clock control1: 1: Pixel clock is masked to 0 whenever the pixel output data is invalid. • This signal only masks the pixel clock to the external pin. The internal pixel clock is not masked, so the logic operation will remain the same.

Bit	Name	Default	Description
			• Masking pixclk is useful for certain display modes, such as STN.
9 (R/W)	HSYNC_MASK	1'b0	Test Purpose register, software writes this with 0 Remark: Mask HSYNC control: 1: Disables the HSYNC during vertical blank time. 0: Enables the HSYNC during vertical blank time.
12:10 (R/W)	SYNC_DLY	3'h0	Delays Hsync, Vsync and bias output for the number of system clocks. (ranging from 1 to 7) It only delays signals in one pixel clock to satisfy setup or hold time requirements.
31:13	-	-	Reserved.

Table 220: Timing Control Register (LCD_S0_TIM_CTRL)

Suggested settings:

- 0x52, //master mode, sends data at falling edge
- 0xfe, //master mode, invert all, sends data at rising edge
- 0x08, //slave all, sends data at rising edge

- Timing Control Status Register (LCD_S0_TIM_STATUS) – 0x0028
  This register is for reference and testing purposes only, software shall disregard this register.

Bit	Name	Default	Description
1:0 (R)	RGB_SEQ_STA	2'b00	The current value of the RGB select index for choosing which color to output based on the RGB sequence (for displays which require outputs of a single color at a time).
2 (R)	VSYNC_STA	1'b0	The current value of the internal vertical sync signal.
3 (R)	HSYNC_STA	1'b0	The current value of the internal horizontal sync signal.
4 (R)	PCLK_STA	1'b0	The current value of internal pixel clock.
31:5	-	-	Reserved.

Table 221: Timing Control Status Register (LCD_S0_TIM_STATUS)

- Horizontal Counter Value (LCD_S0_HCOUNT) – 0x002C
  This register is only for reference and testing purposes.

Bit	Name	Default	Description
10:0 (R)	HCOUNT	11'h0	The current value of the internal horizontal counter.
31:11	-	21'h0	Reserved.

Table 222: Horizontal Counter Value (LCD_S0_HCOUNT)

- Vertical Counter Value (LCD_S0_VCOUNT) – 0x0030
  This register is only for reference and testing purposes.

Bit	Name	Default	Description
10:0 (R)	VCOUNT	11'h0	The current value of the internal vertical counter.
31:11	-	21'h0	Reserved.

Table 223: Vertical Counter Value (LCD_S0_VCOUNT)

- Blanking Register (LCD_S0_BLANK) – 0x0034

Bit	Name	Default	Description
23:0 (R/W)	Blank Value	24'h0	Pixel value to be used for the inactive region: Bits 23:16 – R value Bits 15:8 – G value Bits 7:0 – B value
24 (R/W)	Blank Valid	1'b0	Use blank value 1: Uses the blank value defined below when not in active region. 0: Displays the last pixel value when not in active region.
31:25	-	7'h0	Reserved.

Table 224: Blanking Register (LCD_S0_BLANK)

- Background Color Register (LCD_S0_BACK_COLOR) – 0x0038

Bit	Name	Default	Description
23:0 (R/W)	Background Value	24'h0	The pixel value to be used for the background color: Bits 23:16 – R value Bits 15:8 – G value Bits 7:0 – B value
31:24	-	8'h0	Reserved

Table 225: Background Color Register (LCD_S0_BACK_COLOR)

- Display Mode and Format Register (LCD_S0_DISP_MODE) – 0x003C

Bit	Name	Default	Description
0 (R/W)	FRAME_VALID	1'b0	When this bit is set, the frame will be considered as valid and a valid counter will be generated. However when this bit is cleared, the frame will then be considered as invalid and the counters will not change, this will result in no output data. Please note that when the bit value changes in the middle of a frame, the change will not take effect until the beginning of the next frame. Normally this bit should be set to true.
3:1 (R/W)	OUT_FORMAT	3'b011	Output format: 000: 8-bit rgbrgb 001: 8-bit yuv422 010: 16-bit yuv422 011: 16-bit rgb565
6:4 (R/W)	TOP_LAYER	3'h6	This register determines which layer will be the top layer: 000: Layer0 001: Layer1 110: Cursor0
31:7	-	-	Reserved

Table 226: Display Mode and Format Register (LCD_S0_DISP_MODE)

- Layer Select Register (LCD_S0_LAYER_SEL) – 0x0040

Bit	Name	Default	Description
7:0 (R/W)	LAYER_SEL	8'h00	Bit0: Layer0 is enabled. Bit1: Layer1 is enabled. Bit6: Cursor0 is enabled. If the corresponding bit is set, its corresponding layer will be enabled. If one of the 5 layers needs to be reconfigured, first disable this bit, then wait until the corresponding bit of LAYER_STATUS changes to 0. Only after the respective layer is enabled, will the LCD controller read data from the memory and send it to the LCD panel.
31:8	-	21'h0	Reserved.

Table 227: Layer Select Register (LCD_S0_LAYER_SEL)

- Sequence for RGB Output (LCD_S0_RGB_SEQ) – 0x0044

Bit	Name	Default	Description
5:0 (R/W)	EVEN_RGBSEQ	6'h0	These bits represent the RGB output sequence for even number lines. The meaning is the same as that of ODD_RGBSEQ.
11:6 (R/W)	ODD_RGBSEQ	6'h0	For displays that require only one color per pixel, these bits will represent the RGB output sequence for odd number lines. Each two bits represent a color: 00: Red 01: Green 10: Blue For example, 000110 means the sequence is R-G-B-R-G-B, and 100100 represents B-G-R-B-G-R, etc For displays that require all three color values per pixel, this register will produce no effects.
31:12	-	20'h0	Reserved.

Table 228: Sequence for RGB Output (LCD_S0_RGB_SEQ)

YUV Output Registers

These registers support YUV output instead of RGB, and are mainly targeted for video encoders. The first set of registers contains the coefficients to use for converting RGB outputs to YUV. The conversion is a 3x3 matrix multiplication process as shown below:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} C11 & C12 & C13 \\ C21 & C22 & C23 \\ C31 & C32 & C33 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Please note that since this is a signed multiplication, the coefficients must be 8-bit signed values. In addition, these RGB values may need to be turned into signed values by inverting the MSB of the each component. Also, since YUV outputs are initially signed values, it may be necessary to invert the MSB again to output unsigned values. These inversion options and other control signals for the YUV output are included in the YUV_CTRL register.

The format for the coefficients is as follows:

- Bit 7: the sign bit
- Bits 6-0: the fraction bits

- RGB to YUV Coefficient 1 (LCD_S0_RGB_YUV_COEF1) – 0x0048

Bit	Name	Default	Description
23:16 (R/W)	COEF1	8'h0	Coefficient 11 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
15:8 (R/W)	COEF2	8'h0	Coefficient 12 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
7:0 (R/W)	COEF3	8'h0	Coefficient 13 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
31:24	-	8'h0	Reserved.

Table 229: RGB to YUV Coefficient 1 (LCD_S0_RGB_YUV_COEF1)

The suggested coefficients are 0x264B0E which translates into the following formula:

Y = 0.296875 * R + 0.5859375*G + 0.109375*B

- RGB to YUV Coefficient 2 (LCD_S0_RGB_YUV_COEF2) – 0x004C

Bit	Name	Default	Description
23:16 (R/W)	COEF1	8'h0	Coefficient 21 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
15:8 (R/W)	COEF2	8'h0	Coefficient 22 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.

Bit	Name	Default	Description
7:0 (R/W)	COEF3	8'h0	Coefficient 23 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
31:24	-	8'h0	Reserved.

Table 230: RGB to YUV Coefficient 2 (LCD_S0_RGB_YUV_COEF2)

The suggested coefficients are 0xEBD640 as shown by the following formula:

U = (-0.1640625)*R + (-0.328125)*G + 0.5*B

- RGB to YUV Coefficient 3 (LCD_S0_RGB_YUV_COEF3) – 0x0050

Bit	Name	Default	Description
23:16 (R/W)	COEF1	8'h0	Coefficient 31 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
15:8 (R/W)	COEF2	8'h0	Coefficient 32 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
7:0 (R/W)	COEF3	8'h0	Coefficient 33 for the RGB to YUV conversion matrix. This value should be an 8-bit signed value.
31:24	-	8'h0	Reserved.

Table 231: RGB to YUV Coefficient 3 (LCD_S0_RGB_YUV_COEF3)

The suggested coefficients are 0x40CBF6 shown in the following formula:

V = 0.5*R + (-0.4140625)*G + (-0.078125)*B

- RGB to YUV Conversion Control Register (LCD_S0_YUV_CTRL) – 0x0054

Bit	Name	Default	Description
2:0 (R/W)	YUV_U_CONV	3'b000	This register inverts the MSB of the YUV components to convert signed results to unsigned outputs: Bit 2: Invert Y Bit 1: Invert U Bit 0: Invert V ITU601 accepts positive YUV, and Y is always positive while UV is signed after conversion, this means both U and V need inversion. This bit will usually be set as 3'b011.
5:3 (R/W)	RGB_S_CONV	3'b000	This register inverts the MSB of the RGB components to convert signed to unsigned when the input RGB is signed: Bit 5: invert R

Bit	Name	Default	Description
			Bit 4: Invert G Bit 3: Invert B **NOTE**: The rgb2yuv conversion accepts unsigned RGB and signed coefficients only, so this bit remains as 3'b000.
7:6 (R/W)	YUV_SEQ	1'b0	For 8-bit 4:2:2 output: 00: Send data as YUYV sequence. 01: Send data as YVYU sequence. 10: Send data as UYVY sequence. 11: Send data as VYUY sequence.
8 (R/W)	RGB_YUV	1'b0	1: Converts RGB to YUV. 0: Bypasses RGB to YUV conversion.
9 (R/W)	Even_UV	1'b0	1: Keeps the even pixel UV (U1V1 mode). 0: Keeps the even pixel U and odd pixel V (U1V2 mode).
11:10	-	-	Reserved.
12 (R)	EVENFIELD	1'b0	This bit is only used when in TV mode: 1: The current field is the even field of a frame. 0: The current field is the odd field of a frame.
31:13	-	21'h0	Reserved.

Table 232: RGB to YUV Conversion Control Register (LCD_S0_YUV_CTRL)

- Screen Control for External TV Encoder Register (LCD_S0_TV_FIELD) – 0x0058

Bit	Name	Default	Description
10:0 (R/W)	TV_HSTART	11'h0	TV_HSTART: The start pixel number for the active TV display
11	-	1'b0	Reserved.
22:12 (R/W)	TV_VSTART	11'h0	TV_VSTART: Vertical Start Position (in line number) for the active TV display.
23	-	1'b0	Reserved.
24 (R/W)	TV_F_VALID	1'b0	1: Work in TV mode. When in master mode, TV_HASTART and TV_VSTART will both work. 0: Work in normal mode This is used when using TV 2 fields mode – defining the position of EVEN fields, refer to Figure 10 for details.
31:25	-	-	Reserved.

Table 233: Screen Control for External TV Encoder Register (LCD_S0_TV_FIELD)

- Interrupt Line Number (LCD_S0_INT_LINE) – 0x005C

Bit	Name	Default	Description
10:0 (R/W)	LINE_NUM	11'h0	This value represents the line number in which a screen interrupt is generated. This allows the user to specify a line during which an interrupt will be generated for each frame. The interrupt is generated at the beginning of that particular line.
11	-	1'b0	Reserved.
12 (R/W)	INT_LINE_VALID	1'b0	This bit determines whether the line number LINE_NUM is valid for generating a screen interrupt. This differs from the mask in that if this bit is not set, then the interrupt status bit for the screen interrupt will not be set. When this bit is set, then the interrupt status bit is set each time the line count is equal to the value below.
31:13	-	19'h0	Reserved.

Table 234: Interrupt Line Number (LCD_S0_INT_LINE)

- Interrupt Line Number (LCD_S0_LAYER_STATUS) – 0x0060

Bit	Name	Default	Description
7:0 (R)	LAYER_STATUS	8'h0	Bit0: Layer0 enable status Bit1: Layer1 enable status Bit6: Cursor0 enable status 1: This layer is enabled. 0: This layer is disabled. The status will be changed at the next frame after the LCD_S0_LAYER_SEL register is configured.
31:8	-	-	Reserved

Table 235: Interrupt Line Number (LCD_S0_LAYER_STATUS)

- DMA Status Register (LCD_DMA_STATUS) – 0x00F0

  Read this register to check the DMA status of each layer. If a particular bit reads 1, then this means DMA of this layer is still accessing memory. In this case users must either wait until all bits are cleared before disabling the clock of the LCD controller, or wait until the corresponding bit is cleared before reconfiguring layer settings.

Bit	Name	Default	Description
0 (R)	L0_DMA_STATUS	1'b0	0: No DMA operation. 1: DMA is running.
1 (R)	L1_DMA_STATUS	1'b0	0: No DMA operation. 1: DMA is running.
31:2	-	30'h0	Reserved.

Table 236: DMA Status Register (LCD_DMA_STATUS)

Interrupt Registers

LCD can generate interrupts from several different sources; it can generate an interrupt when a FIFO underflow or overflow occurs; it can also generate an interrupt when the vertical counter reaches a pre-determined line number (i.e. can be used as a frame interrupt). The interrupt can also be generated when the DMA operation is completed. Each interrupt source can be independently masked and cleared.

- Interrupt Mask (LCD_INT_MASK) – 0x00F4

  This register provides an interrupt mask for all LCD interrupt sources. If a mask bit is set to 1, then that interrupt will be enabled. If the interrupt status bit turns to 1, then an interrupt will be generated to RISC. If the mask bit is set to 0, then no interrupts will be generated, yet the status bit can still be set to 1 and be cleared by writing to the above register.

Bit	Name	Default	Description
0 (R/W)	L0_DMA_MASK	1'b0	One frame DMA over mask
1 (R/W)	L1_DMA_MASK	1'b0	One frame DMA over mask
5:2	-	-	Reserved
6 (R/W)	L0_OFLOW_MASK	1'b0	Screen FIFO Overflow mask
7 (R/W)	L1_OFLOW_MASK	1'b0	Screen FIFO Overflow mask
11:8	-	-	Reserved
12 (R/W)	L0_UFLOW_MASK	1'b0	Screen FIFO Underflow mask
13 (R/W)	L1_UFLOW_MASK	1'b0	Screen FIFO Underflow mask
17:14	-	-	Reserved
18 (R/W)	S0_LINE_INT_MASK	1'b0	Line Interrupt mask
31:19	-	9'h0	Reserved

Table 237: Interrupt Mask (LCD_INT_MASK)

- Current Interrupt Status and Interrupt Clear (LCD_INT_CTRL_STATUS) – 0x00F8

  Read this register to determine the source of interrupt. If a particular bit reads 1, then that interrupt will be asserted. Writing a 1 to a particular bit will clear the interrupt for that source, which is the only way to clear the interrupt. That hardware does not clear any interrupts.

NOTE – The interrupt bit for each source can be read or cleared even if the mask bit is not set. However without the mask bit, the interrupt will not be generated to RISC.

Bit	Name	Default	Description
0 (R/W)	L0_DMA_INT	1'b0	One frame DMA over Interrupt
1 (R/W)	L1_DMA_INT	1'b0	One frame DMA over Interrupt
5:2	-	-	-
6 (R/W)	L0_OFLOW_INT	1'b0	Screen FIFO Overflow Interrupt
7 (R/W)	L1_OFLOW_INT	1'b0	Screen FIFO Overflow interrupt
11:8	-	-	-
12 (R/W)	L0_UFLOW_INT	1'b0	Screen FIFO Underflow Interrupt
13 (R/W)	L1_UFLOW_INT	1'b0	Screen FIFO Underflow Interrupt
17:14	-	-	-
18 (R/W)	S0_LINE_INT	1'b0	Line interrupt
31:19	-	-	Reserved

Table 238: Current Interrupt Status and Interrupt Clear (LCD_INT_CTRL_STATUS)

Other Screen Control Registers

- Dual Display Control Register (LCD_SCR_CTRL) – 0x00FC

Bit	Name	Default	Description
0 (R/W)	SCREEN0_EN	1'b0	1: This screen is enabled. 0: This screen is disabled. When working in master mode, and after this bit is enabled, the internal counter will start counting according to the pixel clocks and horizontal sync signals.
30:1	-	-	Reserved.

Table 239: Dual Display Control Register (LCD_SCR_CTRL)

Layer Control Registers of Layer0

- Layer0 Control Register (LCD_L0_CTRL) – 0x0100

Bit	Name	Default	Description
2:0 (R/W)	BPP	3'h0	000: 18-bit per pixel, RGB666 001: 16-bit per pixel, RGB565 010: 16-bit per pixel, RGB556 011: 16-bit per pixel, RGB655 100: 32-bit per pixel, RGB888 101: 32-bit per pixel, TRGB888 110: 32-bit per pixel, ARGB8888 111: Reserved
3 (R/W)	QWORD_BLE	1'b0	Big/Little Endian selection of DWORD for QWORD data: 1: Little Endian: MSB- DWORD1, DWORD0-LSB 0: Big Endian: MSB- DWORD0, DWORD1-LSB
4 (R/W)	DWORD_BLE	1'b0	Big/Little Endian selection of byte for image data: 1: Little Endian:MSB- byte3, byte2, byte1, byte0-LSB 0: Big Endian: MSB- byte0, byte1, byte2, byte3-LSB
5 (R/W)	FIFO_RESET	1'b0	Set this bit will reset the FIFO write and read pointer.
6 (R/W)	CKEY_EN	1'b0	Enable the color key function.
7	FIFO_FKRDY	1'b0	FIFO fake ready: 1: DMA or VPP can write data to memory even if FIFO_RESET is set to 1. 0: DMA or VPP cannot write data to the memory when FIFO_RESET is set to 1.
8 (R/W)	CONFIRM	-	This register confirms all settings for this layer. This bit must be set last after all other registers have been configured. This bit will clear itself once the configuration becomes valid.
31:9	-	-	Reserved.

Table 240: Layer0 Control Register (LCD_L0_CTRL)

- Horizontal Start Position for Layer0 Screen (LCD_L0_HSTART) – 0x104

Bit	Name	Default	Description
10:0 (R/W)	L0_HSTART	11'h0	Horizontal Start Position (in pixel number). This value, along with the horizontal end position and vertical start and end positions, define the rectangle region of layer0. When working in 8-bit RGB mode: L0_HSTART - SCREEN_HSTART must be the multiple of 3. When working in 8-bit YUV mode: L0_HSTART - SCREEN_HSTART must be the multiple of 2.
31:11	-	21'h0	Reserved.

Table 241: Horizontal Start Position for Layer0 Screen (LCD_L0_HSTART)

- Vertical Start Position for Layer0 Screen (LCD_L0_VSTART) – 0x108

Bit	Name	Default	Description
10:0 (R/W)	L0_VSTART	11'h0	Vertical Start Position (in line number)
31:11	-	21'h0	Reserved

Table 242: Vertical Start Position for Layer0 Screen (LCD_L0_VSTART)

- Horizontal End Position for Layer0 Screen (LCD_L0_HEND) – 0x10C

Bit	Name	Default	Description
10:0 (R/W)	L0_HEND	11'h0	Horizontal End Position (in pixel number) In 8-bit RGB mode: L0_HSTART + (WIDTH – 1)*3 In 8-bit YUV mod: L0_HSTART + (WIDTH – 1)*2 Others: L0_HSTART + WIDTH – 1
31:11	-	21'h0	Reserved

Table 243: Horizontal End Position for Layer0 Screen (LCD_L0_HEND)

- Vertical End Position for Layer0 Screen (LCD_L0_VEND) – 0x110

Bit	Name	Default	Description
10:0 (R/W)	L0_VEND	11'h0	Vertical End Position (in line number) =L0_VSTART + DEPTH - 1
31:11	-	-	Reserved

Table 244: Vertical End Position for Layer0 Screen (LCD_L0_VEND)

- Screen Memory Base Register (LCD_L0_BASE0) – 0x114

Bit	Name	Default	Description
29:0 (R/W)	BASE0_ADDR	30'h0	This is the DMA address of the starting memory location for the screen data, this is a byte address, but the lower 3 address bits must be all zeroes, which means the DMA start address must lay on a Gword burst boundary.
31:30	-	-	Reserved.

Table 245: Screen Memory Base Register (LCD_L0_BASE0)

- Screen Memory Base Register (LCD_L0_BASE1) – 0x118

Bit	Name	Default	Description
29:0 (R/W)	BASE1_ADDR	30'h0	The DMA address of the second starting memory location for the screen data, this is a byte address, but the lower 3 address bits must be all zeroes, which means the DMA start address must lay on a Gword burst boundary.
31:30	-	-	Reserved.

Table 246: Screen Memory Base Register (LCD_L0_BASE1)

- X Size for Screen DMA (LCD_L0_XSIZE) – 0x11C

Bit	Name	Default	Description

12:0 (R/W)	L0_XSIZE	13'h0	The number of consecutive bursts per line for the screen DMA:
			If (ByteWidthOfL0ValidLine % ((DMA_UNIT+1)*8))
			L0_XSIZE = ByteWidthOfL0ValidLine /((DMA_UNIT+1)*8)
			Else
			L0_XSIZE = ByteWidthOfL0ValidLine /((DMA_UNIT+1)*8) – 1

31:13	-	19'h0	Reserved

Table 247: X Size for Screen DMA (LCD_L0_XSIZE)

- Y Size for Screen DMA (LCD_L0_YSIZE) – 0x120

Bit	Name	Default	Description
12:0 (R/W)	L0_YSIZE	13'h0	This value specifies the number of "lines" for the screen DMA. Each line designates a segment of consecutive QWORDs with a skip in between. L0_YSIZE = DEPTH – 1
31:13	-	19'h0	Reserved

Table 248: Y Size for Screen DMA (LCD_L0_YSIZE)

- Skip Value for Screen DMA (LCD_L0_SKIP) – 0x124

Bit	Name	Default	Description
12:0 (R/W)	L0_SKIP	13'h0	This value specifies the number of bytes for the DMA address generator for skipping in between lines of the screen DMA: L0_SKIP = ByteWidthOfL0FrameBuf – (L0_XSIZE*(DMA_UNIT+1)*8) The ByteWidthofL0FrameBuf must be a multiple of 8.
31:13	-	19'h0	Reserved.

Table 249: Skip Value for Screen DMA (LCD_L0_SKIP)

- Control Register for DMA (LCD_L0_DMA_CTRL) – 0x128

  If the screen size or location requires the start of a new line in the middle of a burst, then the DMA interface will write an extra QWORD to the FIFO which may cause the data to mismatch. In order to avoid this, use this register to suppress extra writes to the FIFO at the end of a line. To do this, write the number of SUPPRESS QWORD to bits <11:8>.

  For example, to write 19 QWORD per line, it will require 5 bursts (each burst with four QWORDs) from the DMA, and suppress the last QWORD, thus, SUPPRESS_QW_NUM should be set to 0x001. It is important that the frame buffer's start address lie on a DMA Qword burst boundary, and that the width of the frame buffer is a multiple of the DMA.

Bit	Name	Default	Description
0 (R/W)	START	1'h0	Set this bit will start DMA transfers; the bit will clear itself once all other DMA registers take effect.
1 (R/W)	DMA_MODE	1'b0	Continuous mode DMA: 1: When this DMA operation is completed, it will automatically generate a DMA with exactly the same setting. 0: Each DMA must be explicitly started by the software.
2 (R/W)	DMA_CHAIN_MODE	1'b0	Chain DMA mode:

Bit	Name	Default	Description
			1: Enables the DMA chain mode. 0: Disables the DMA chain mode.
3	-	-	Reserved.
7:4 (R/W)	DMA_UNIT	4'h3	Burst unit for a DMA operation which only supports the following settings: 4'h3: 4 QWORD 4'h7: 8 QWORD 4'hF: 16 QWORD
11:8 (R/W)	SUPPRESS_QW_NUM	4'h0	The number of QWORD written to the FIFO must be suppressed at the end of each DMA line.
31:12	-	19'h0	Reserved.

Table 250: Control Register for DMA (LCD_L0_DMA_CTRL)

- L0 Alpha Blending Control Register (LCD_L0_ALPHA) – 0x12C

Bit	Name	Default	Description
7:0 (R/W)	ALPHA_VAL	4'h0	The 8-bit planar alpha value that blends all pixels on the L0 layer.
31:8	-	-	Reserved.

Table 251: L0 Alpha Blending Control Register (LCD_L0_ALPHA)

- L0 Color Key Big RGB Value (LCD_L0CKEYB) – 0x130

Bit	Name	Default	Description
23:0 (R/W)	CKEYB	24'h0	The bigger value of a color key.
31:24	-	-	Reserved.

Table 252: L0 Color Key Big RGB Value (LCD_L0CKEYB)

Color Key is a color or a range of color which is defined as transparent. The 24-bit (8:8:8) color mode can be used for 16-bit (5:6:5, 5:5:6, 6:5:5), 24-bit (RGB888), and so on. Although this mode is flexible, but it still needs to calculate CKEYB and CKEYS values according to the data format. For the 16-bit data format, the 24-bit CKEYB or CKEYS are calculated by expanding Red, Green, or Blue to 8 bits, and setting the expanded LSB to 1'b0.

- L0 Color Key Small RGB Value (LCD_L0CKEYS) – 0x134

.

Bit	Name	Default	Description
23:0 (R/W)	CKEYS	24'h0	The smaller value of a color key for the L0 channel display.
31:24	-	-	Reserved.

Table 253: L0 Color Key Small RGB Value (LCD_L0CKEYS)

- Screen FIFO Control Register (LCD_L0FIFO) – 0x138

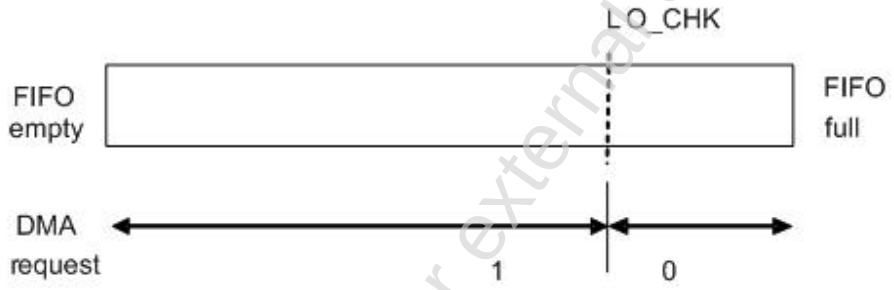


Figure 45: FIFO's Request Level (L0_REQ_SEL=0)

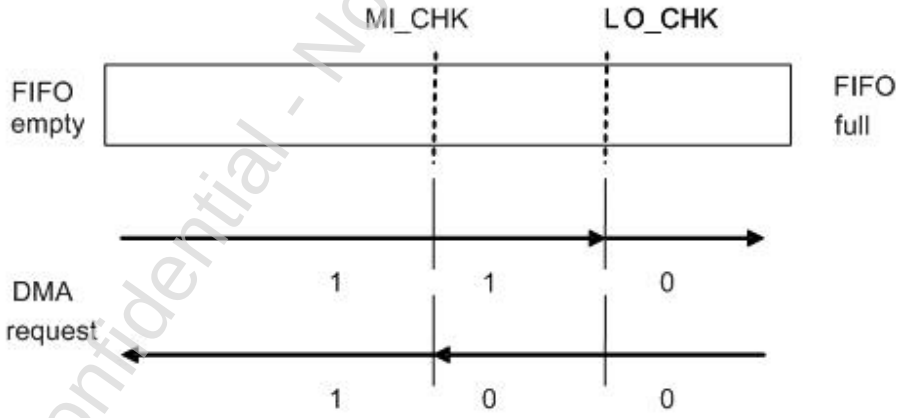


Figure 46: FIFO's Request Level (L0_REQ_SEL=1)

The DMA requests to the bus interface are generated based on the FIFO fullness counter. It provides two methods for request generation selected by bit 24 of the L0_REQ_SEL register. When this bit is set to 0, and if the FIFO fullness is less than the low water mark, then it will generate a request. In addition, when bit 24 is equal to 1, the middle and low water marks will become useful. When the FIFO fullness is less than the middle water mark, it will generate request (1), and when the FIFO fullness is more than the low water mark, the request will stop (0). The second method will lead much more continuous data dumps from the memory than the first one. It is obvious that enlarging any of the 2 watermarks will make the FIFO more aggressive.

This register must be set before the DMA register is configured.

Bit	Name	Default	Description
7:0 (R/W)	L0_LO_CHK	8'h60	This is the value for the low request watermark of the screen FIFO. If the FIFO (FIFO count) is filled beyond this mark, then the request will stop. However if the FIFO is read below this point, then the request will become a low-level request The maximum of this value is (0x100 – (dma_unit + 1)).
15:8 (R/W)	L0_MI_CHK	8'h40	This is the value for the middle request watermark of the screen FIFO. If the FIFO is filled beyond this mark, then the request will drop to a low-level request. However if the FIFO is read below this point, then the request will become a high-level request The maximum of this value is (0x100 – 2*(dma_unit + 1)).
23:16	-	-	Reserved
24 (R/W)	L0_REQ_SEL	1'b0	This register is for selecting a request generation method for L0 FIFO: 1: Middle request and low request watermark are useful. Using this mode will save the bandwidth. 0: Normal request generation.
31:8	-	24'h0	Reserved.

Table 254: Screen FIFO Control Register (LCD_L0FIFO)

- FIFO Status (LCD_L0_FIFO_STATUS) – 0x13C
  This register is for reference and testing purposes only.

Bit	Name	Default	Description
8:0 (R)	FIFO_LEN	-	If this register is read, the valid FIFO length will be given.
31:9	-	-	Reserved.

Table 255: FIFO Status (LCD_L0_FIFO_STATUS)

Layer Control Registers of L1

The layer1 register has the same definition as that of layer0 except for the base address.

Cursor0 Control Registers

- Control Register for Cursor0 (LCD_CUR0_CTRL) – 0x1000

Bit	Name	Default	Description
2:0 (R/W)	MODE	3'h0	000: 32x32x2bpp 2-color and transparency mode 001: 32x32x2bpp 4-color mode 010: 32x32x2bpp 3-color and transparency mode 011: Reserved 100: 64x64x2bpp 2-color and transparency mode 101: 64x64x2bpp 4-color mode 110: 64x64x2bpp 3-color and transparency mode
3	-	-	Reserved.
4(R/W)	DWORD_BLE	1'b0	Big/little endian selection of byte for image data. 1: Little Endian: MSB- byte3, byte2, byte1, byte0-LSB 0: Big Endian: MSB- byte0, byte1, byte2, byte3-LSB
5(R/W)	BYTE_BLE	1'b0	Big/Little Endian selection of pixel data for each byte 1: Little Endian: MSB- P3, P2, P1, P0-LSB 0: Big Endian: MSB- P0, P1, P2, P3-LSB
7:6	-	-	Reserved.
8(R/W)	SRAM_ADDRST	1'b0	Soft reset of cursor0 SRAM addresses: 1: Setting this bit to 1 will reset the read and the write address of SRAM. It can be used for recovery purposes from an abnormal operation. 0: After reset, this bit should be set back to 0 for normal operations.
15:9	-	-	Reserved.
16 (R/W)	SETTING_VALID	1'h0	This register is for confirming the new region and other settings by writing this bit with a 1.This bit will clear itself after a valid frame start is detected. This bit is used as confirmation of the setting-group.
31:17	-	-	Reserved.

Table 256: Control Register for Cursor0 (LCD_CUR0_CTRL)

- Horizontal Start Position for Layer0 Screen (LCD_CUR0_HSTART) – 0x1004

Bit	Name	Default	Description
10:0 (R/W)	CUR0_HSTART	11'h0	Horizontal Start Position (in pixel number). This value, along with the horizontal end position and vertical start and end positions, define the rectangular region of layer0. This value must be greater or equal to the ACT_HSTART.
31:11	-	21'h0	Reserved.

Table 257: Horizontal Start Position for Layer0 Screen (LCD_CUR0_HSTART)

- Vertical Start Position for Layer0 Screen (LCD_CUR0_VSTART) – 0x1008

Bit	Name	Default	Description
10:0 (R/W)	CUR0_VSTART	11'h0	Vertical start position (in line number) This value must be greater or equal to the ACT_VSTART.
31:11	-	21'h0	Reserved.

Table 258: Vertical Start Position for Layer0 Screen (LCD_CUR0_VSTART)

- Horizontal End Position for Layer0 Screen (LCD_CUR0_HEND) – 0x100C

Bit	Name	Default	Description
10:0 (R/W)	CUR0_HEND	11'h0	Horizontal end position (in pixel number) This value must be less or equal to the ACT_HEND.
31:11	-	21'h0	Reserved

Table 259: Horizontal End Position for Layer0 Screen (LCD_CUR0_HEND)

- Vertical End Position for Layer0 Screen (LCD_CUR0_VEND) – 0x1010

Bit	Name	Default	Description
10:0 (R/W)	CUR0_VEND	11'h0	Vertical end position (in line number) This value must less or equals the ACT_VEND.
31:11	-	20'h0	Reserved.

Table 260: Vertical End Position for Layer0 Screen (LCD_CUR0_VEND)

- Color0 of Cursor0 (LCD_CUR0_COLOR0) – 0x1014

Bit	Name	Default	Description
7:0 (R/W)	BLUE	8'h0	Blue value for the color0 of cursor0
15:8 (R/W)	GREEN	8'h0	Green value for the color0 of cursor0
23:16 (R/W)	RED	8'h0	Red value for the color0 of cursor0
31:24	-	8'h0	Reserved

Table 261: Color0 of Cursor0 (LCD_CUR0_COLOR0)

- Color1 of Cursor0 (LCD_CUR0_COLOR1) – 0x1018

Bit	Name	Default	Description
7:0 (R/W)	BLUE	8'h0	Blue value
15:8 (R/W)	GREEN	8'h0	Green value
23:16 (R/W)	RED	8'h0	Red value
31:24	-	8'h0	Reserved

Table 262: Color1 of Cursor0 (LCD_CUR0_COLOR1)

- Color2 of Cursor0 (LCD_CUR0_COLOR2) – 0x101C

Bit	Name	Default	Description
7:0 (R/W)	BLUE	8'h0	Blue value
15:8 (R/W)	GREEN	8'h0	Green value
23:16 (R/W)	RED	8'h0	Red value
31:24	-	8'h0	Reserved

Table 263: Color2 of Cursor0 (LCD_CUR0_COLOR2)

- Color3 of Cursor0 (LCD_CUR0_COLOR3) – 0x1020

Bit	Name	Default	Description
7:0 (R/W)	BLUE	8'h0	Blue value
15:8 (R/W)	GREEN	8'h0	Green value
23:16 (R/W)	RED	8'h0	Red value
31:24	-	8'h0	Reserved

Table 264: Color3 of Cursor0 (LCD_CUR0_COLOR3)

- CUR0 Alpha Blending Control Register (LCD_CUR0_ALPHA) – 0x1024

Bit	Name	Default	Description
7:0 (R/W)	ALPHA_VAL	8'h0	The 8-bit planar alpha value that blends all pixels on the cursor0 layer.
31:8	-	-	Reserved.

Table 265: CUR0 Alpha Blending Control Register (LCD_CUR0_ALPHA)

- CUR0 Read Pointer Register (LCD_CUR0_FIFO_RDPTR) – 0x1028
  This register is only for read and testing purposes only.

Bit	Name	Default	Description
7:0 (R)	FIFO_RDPTR	8'h0	Read pointer of SRAM
31:8	-	-	Reserved

Table 266: CUR0 Read Pointer Register (LCD_CUR0_FIFO_RDPTR)

- CUR0 XY Register (LCD_CUR0_XY) – 0x102C
  This register is used to locate the start location of the cursor when its display region is out of the screen active region.

Bit	Name	Default	Description
6:0(R/W)	CUR_X	7'h0	The X location of the cursor active region
15:7	-	-	Reserved
22:16(R/W)	CUR_Y	7'h0	The Y location of cursor active region
31:23	-	-	Reserved

Table 267: CUR0 XY Register (LCD_CUR0_XY)

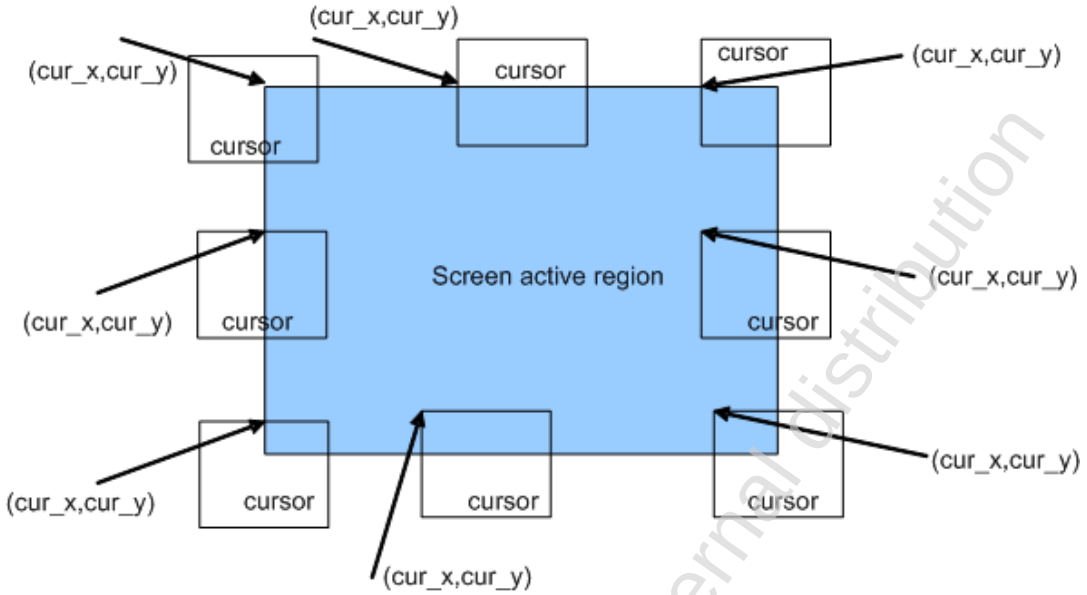


Figure 47: Cursor Position Diagram

- Read/Write FIFO Data (LCD_CUR0_FIFO) – 0x1400

Bit	Name	Default	Description
31:0 (R/W)	FIFO_DATA	-	This is the values in the cursor FIFO. This value can directly read or write anywhere in the FIFO by specifying its corresponding address.

Table 268: Read/Write FIFO Data (LCD_CUR0_FIFO)

Video Post Processor

The VPP (Video Post Processor) is used to process image data from the software video decoder.

Feature List

- Hardware de-interlacing
- Horizontal/vertical scaling
- YUV to RGB conversion; supports RGB565, RGB666 and RGB888 output data format

Limitations

- Input source should be QWORD-aligned, input width should be no less than 12 and no greater than 768.
- Source width/height should be multiple of 2.
- Output width should be multiple of 2.
- Downscaling in horizontal/vertical direction should be no less than 1/8.

Application Scenarios

The following input image types are supported:

- YUV422 format, big/little endian, Y, U, and V are placed together as one and supports four formats: YUYV, YVYU, UYVY, VYUY.

Y00	U00	Y01	V00	…	…	Y	U	Y	V
Y10	U10	Y11	V10	…	…	Y	U	Y	V
Y20	U20	Y21	V20	…	…	Y	U	Y	V
Y30	U30	Y31	V30	…	…	Y	U	Y	V
…	…	…	…	…	…	…	…	…	…
Y	U	Y	V	…	…	Y	U	Y	V
Y	U	Y	V	…	…	Y	U	Y	V
Y	U	Y	V	…	…	Y	U	Y	V
Y	U	Y	V	…	…	Y	U	Y	V

In this format, each four bytes are two pixels, and each four bytes consist of two Y's, one U and one V. Each Y goes to one of the pixels; U and V belong to both pixels. The horizontal resolution of the U and V components is half of the Y component.

- YUYV

  Bit allocation of one DWORD (YUYV, big endian):

31 24	23 16	15 8	7 0
Y0	U0	Y1	V0

  Bit allocation of one DWORD (YUYV, little endian):

31 24	23 16	15 8	7 0
Y1	U0	Y0	V0

- YVYU:

  Bit allocation of one DWORD (YVYU, big endian):

31 24	23 16	15 8	7 0
Y0	V0	Y1	U0

  Bit allocation of one DWORD (YVYU, little endian):

31 24	23 16	15 8	7 0
Y1	V0	Y0	U0

- UYVY:

  Bit allocation of one DWORD (UYVY, big endian):

31 24	23 16	15 8	7 0
U	Y0	V	Y1

  Bit allocation of one DWORD (UYVY, little endian):

31 24	23 16	15 8	7 0
U	Y1	V	Y0

– VYUY:

Bit allocation of one DWORD (VYUY, big endian):

31 24	23 16	15 8	7 0
V	Y0	U	Y1

Bit allocation of one DWORD (VYUY, little endian):

31 24	23 16	15 8	7 0
V	Y1	U	Y0

NOTE – The four YUV422 formats mentioned above may not be compatible with certain display tools, because in certain tools, YUYV format may be indicated as Y1 is the LSB in DWORD.

- The detached YUV420 format, little endian. Y, U and V are placed separately:

Y00	Y01	…	…	Y	Y
Y10	Y11	…	…	Y	Y
…	…	…	…	…	…
Y	Y	…	…	Y	Y
Y	Y	…	…	Y	Y

U00	…	U
…	…	…
U	…	U

V00	…	V
…	…	…
V	…	V

The three components are separated into three sub-images or planes. The Y plane has one byte per pixel, the U and V plane is half the width and height of the Y plane (and of the image). Each U or V belongs to four pixels, a two-by-two square of the image. For example, U00 and V00 belong to Y00, Y01, Y10, and Y11.

The bit allocation of a DWORD:

31 24	23 16	15 8	7 0
Y3	Y2	Y1	Y0

31 24	23 16	15 8	7 0
U3	U2	U1	U0

31 24	23 16	15 8	7 0
V3	V2	V1	V0

Functional Descriptions

Block Diagram

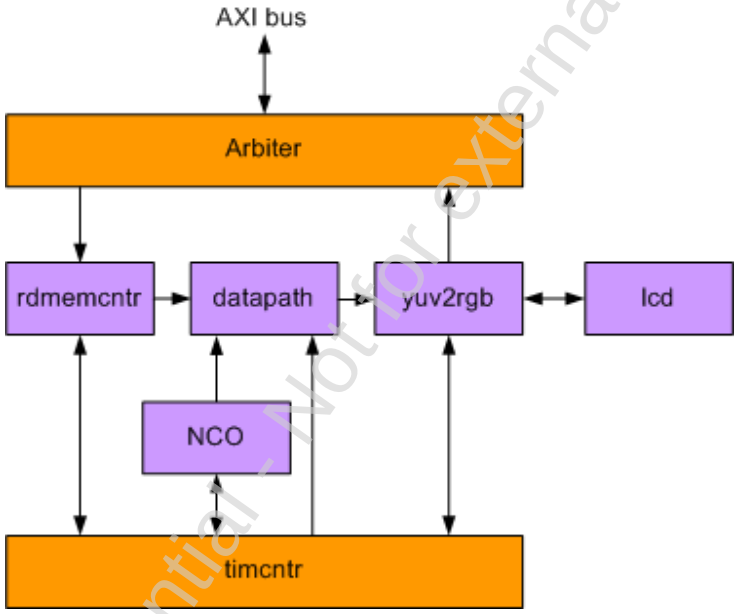


Figure 48: Video Post-Processor Block Diagram

Module Descriptions

Hardware De-interlacing

The following hardware de-interlacing modes are supported:

- Bob method

  Bob (intra-field spatial interpolation) is the process of completely discarding one of the fields in an interlaced video data. Either all the odd or all the even fields are discarded. Using the fields that remain, interpolation is performed between the lines to generate an entire non-interlaced frame for the progressive scan monitor.

The Bob method is not implemented in VPP, if users want to use intra field interpolation, it is recommended to use one field as source data, and use vertical scaling to perform interpolation instead of directly using Bob duplication or interpolation methods.

- Weave method

  Weave (merging) is the process of combining the odd and even fields of an interlaced video image to generate the entire non-interlaced frame for the progressive scan monitor.

  If the input is in YUV420 format output from video decoder, then this means both odd and even fields have been merged into the Weave format to form the frame, therefore executing weave de-interlacing in VPP is not necessary.

  If the input is YUV422 and users want to perform weave de-interlacing, then set the yuv422_weave bit of the VPP_CTRL register to have VPP_YBASE point to the base address of the odd field, and VPP_UBASE point to the base address of the even field. Set source height to be the height of the frame instead of the field.

Scalar

This module performs the horizontal scaling function with a 6-tap filter and vertical scaling function with a 4-tap filter. Hardware will automatically select one of the 16 groups to perform scaling.

The group0 coefficients will be in use if the output pixel coincides with the input one, if the output pixel does not coincide with the input pixels, for example, if there is an offset between the expected output pixel location and the input pixel location, then the offset will be quantized using 4 bits and have 16 valid offset values (0-15). If the expected output pixel situates just half between two consecutive input pixels (offset is 8), then the group8 coefficients will be used.

Nine groups of coefficients should be configured (group0 to group8), and group9 to group15 will be automatically obtained by hardware through internal symmetry of coefficients between groups.

Input pixel format: YUV420 or YUV422.

Output pixel format: YUV422.

- NCO module

  This module implements the following function in horizontal and vertical directions:

  Six coefficients of Group9 is obtained from group Group7, tap0, tap1, tap2, tap3, tap4, tap5 coefficient of group9 is the value tap5, tap4, tap3, tap2, tap1, tap0 coefficient of group7.

- Rdmemcntr module

  This module generates the input source address based on results acquired from the NCO module and stores the data input line buffer.

  This module also fetches data from the memory operation (line buffer write operation); data read from the line buffer through downstream is pipelined.

  If this module requires several lines of Y, U and V data in YUV420 mode, then the first burst will come from Y line0, Y line1, Y line2…, U line 0, U line1 …, V line0 and V line1…, and the next burst will come from Y line0, Y line1, Y line2…, U line 0, U line1 …, V line0, V line1… and so on.

- datapath module

This module fetches data from line buffers and selects pixel data from the corresponding lines (based on the center tap location of the horizontal filter) to the vertical filter. Both U and V filters share one computation unit in vertical scaling.

After vertical filtering, rounding and clipping operations are applied and the result is stored in the registers.

Horizontal scaling continues to fetch results from vertical scaling and applies rounding and clipping operations. Both U and V filters share one computation unit in horizontal scaling.

Y and UV results are combined and sent to the downstream.

- timcntr module

  This module generates signals to control the timing of other modules.

YUV to RGB Conversion

This module converts the internal YUV444 data from the horizontal scaling module to the RGB format and sends the result to the memory in the required output format.

Input pixel format: YUV422, the input format is implicitly converted to YUV422 without interpolation by UV duplication.

Output format: RGB888, RGB666 and RGB565 format.

Output data width: 64-bit data.

Data format for 16-bit RGB565:

31 27	26 21	20 16	15 11	10 5	4 0
R1[4:0]	G1[5:0]	B1[4:0]	R0[4:0]	G0[5:0]	B0[4:0]

63 59	58 53	52 48	47 43	42 37	36 32
R3[4:0]	G3[5:0]	B3[4:0]	R2[4:0]	G2[5:0]	B2[4:0]

Data format for 18-bit RGB666:

63 50	49 44	43 38	37 32	31 18	17 12	11 6	5 0
14'h0	R1[5:0]	G1[5:0]	B1[5:0]	14'h0	R0[5:0]	G0[5:0]	B0[5:0]

Data format for 24-bit RGB888:

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
8'h0	R1[7:0]	G1[7:0]	B1[7:0]	8'h0	R0[7:0]	G0[7:0]	B0[7:0]

YUV to RGB conversion formula:

R = Y*C1 + U* C2 + V* C3 – OFFSET1

G = Y*C4 - U*C5 - V* C6 + OFFSET2

B = Y*C7 + V* C8 + V* C9 – OFFSET3

The output RGB data will be rounded.

The following is a list of reference values:

- VPP_RCOEF = 0x12A00199
- VPP_GCOEF = 0x12A190D0
- VPP_BCOEF = 0x12A81000
- VPP_OFFSET1 = 0xdf20
- VPP_OFFSET2 = 0x8760
- VPP_OFFSET3 = 0x114a0

This yields the following conversion formula (eg, for YUV420 to RGB):

- d =u-128
- e = v-128
- c0 = (y0 - 16) * 298
- c1 = (y1 - 16) * 298
- c2 = (y2 - 16) * 298
- c3 = (y3 - 16) * 298
- r = e * 409 + 128
- g = 128 - 100 * d - 208 * e
- b = 516 * d + 128

The converted values for R, G and B will be as follows (i=0, 1, 2, 3):

- Ri = ci + r
- Gi = ci + g
- Bi = ci + b

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
VPP base address through RISC I/O	0x80130000

Table 269: Base Address

Register Mapping

RISC Address <15:0>	Register	Description
0x0000	VPP_CTRL	The working mode and successive BLT command number
0x0004	VPP_YBASE	The source memory address for Y
0x0008	VPP_UBASE	The source memory address for U
0x000C	VPP_VBASE	The source memory address for V
0x0010	VPP_DESBASE	The destination base memory address
0x0014	VPP_WIDTH	The window width of the source and destination
0x0018	VPP_HEIGHT	The window height
0x001c	VPP_STRIDE0	Source stride for pictures
0x0020	VPP_STRIDE1	Destination stride for pictures
0x0024	VPP_HSCA_COEF00	Filter coefficient 0 of group0 for horizontal scaling
0x0028	VPP_HSCA_COEF01	Filter coefficient 1 of group0 for horizontal scaling
0x002c	VPP_HSCA_COEF02	Filter coefficient 2 of group0 for horizontal scaling
0x0030	VPP_HSCA_COEF10	Filter coefficient 0 of group1 for horizontal scaling
0x0034	VPP_HSCA_COEF11	Filter coefficient 1 of group1 for horizontal scaling
0x0038	VPP_HSCA_COEF12	Filter coefficient 2 of group1 for horizontal scaling
0x003c	VPP_HSCA_COEF20	Filter coefficient 0 of group2 for horizontal scaling
0x0040	VPP_HSCA_COEF21	Filter coefficient 1 of group2 for horizontal scaling
0x0044	VPP_HSCA_COEF22	Filter coefficient 2 of group2 for horizontal scaling
0x0048	VPP_HSCA_COEF30	Filter coefficient 0 of group3 for horizontal scaling
0x004c	VPP_HSCA_COEF31	Filter coefficient 1 of group3 for horizontal scaling
0x0050	VPP_HSCA_COEF32	Filter coefficient 2 of group3 for horizontal scaling
0x0054	VPP_HSCA_COEF40	Filter coefficient 0 of group4 for horizontal scaling
0x0058	VPP_HSCA_COEF41	Filter coefficient 1 of group4 for horizontal scaling
0x005c	VPP_HSCA_COEF42	Filter coefficient 2 of group4 for horizontal scaling
0x0060	VPP_HSCA_COEF50	Filter coefficient 0 of group5 for horizontal scaling
0x0064	VPP_HSCA_COEF51	Filter coefficient 1 of group5 for horizontal scaling
0x0068	VPP_HSCA_COEF52	Filter coefficient 2 of group5 for horizontal scaling

RISC Address <15:0>	Register	Description
0x006c	VPP_HSCA_COEF60	Filter coefficient 0 of group6 for horizontal scaling
0x0070	VPP_HSCA_COEF61	Filter coefficient 1 of group6 for horizontal scaling
0x0074	VPP_HSCA_COEF62	Filter coefficient 2 of group6 for horizontal scaling
0x0078	VPP_HSCA_COEF70	Filter coefficient 0 of group7 for horizontal scaling
0x007c	VPP_HSCA_COEF71	Filter coefficient 1 of group7 for horizontal scaling
0x0080	VPP_HSCA_COEF72	Filter coefficient 2 of group7 for horizontal scaling
0x0084	VPP_HSCA_COEF80	Filter coefficient 0 of group8 for horizontal scaling
0x0088	VPP_HSCA_COEF81	Filter coefficient 1 of group8 for horizontal scaling
0x008c	VPP_HSCA_COEF82	Filter coefficient 2 of group8 for horizontal scaling
0x0090	VPP_VSCA_COEF00	Filter coefficient 0 of group 0 for vertical scaling
0x0094	VPP_VSCA_COEF01	Filter coefficient 1 of group 0 for vertical scaling
0x0098	VPP_VSCA_COEF10	Filter coefficient 0 of group 1 for vertical scaling
0x009c	VPP_VSCA_COEF11	Filter coefficient 1 of group 1 for vertical scaling
0x00a0	VPP_VSCA_COEF20	Filter coefficient 0 of group 2 for vertical scaling
0x00a4	VPP_VSCA_COEF21	Filter coefficient 1 of group 2 for vertical scaling
0x00a8	VPP_VSCA_COEF30	Filter coefficient 0 of group 3 for vertical scaling
0x00ac	VPP_VSCA_COEF31	Filter coefficient 1 of group 3 for vertical scaling
0x00b0	VPP_VSCA_COEF40	Filter coefficient 0 of group 4 for vertical scaling
0x00b4	VPP_VSCA_COEF41	Filter coefficient 1 of group 4 for vertical scaling
0x00b8	VPP_VSCA_COEF50	Filter coefficient 0 of group 5 for vertical scaling
0x00bc	VPP_VSCA_COEF51	Filter coefficient 1 of group 5 for vertical scaling
0x00c0	VPP_VSCA_COEF60	Filter coefficient 0 of group 6 for vertical scaling
0x00c4	VPP_VSCA_COEF61	Filter coefficient 1 of group 6 for vertical scaling
0x00c8	VPP_VSCA_COEF70	Filter coefficient 0 of group 7 for vertical scaling
0x00cc	VPP_VSCA_COEF71	Filter coefficient 1 of group 7 for vertical scaling
0x00d0	VPP_VSCA_COEF80	Filter coefficient 0 of group 8 for vertical scaling
0x00d4	VPP_VSCA_COEF81	Filter coefficient 1 of group 8 for vertical scaling
0x00d8	VPP_RCOEF	Coefficient for R
0x00dc	VPP_GCOEF	Coefficient for G
0x00e0	VPP_BCOEF	Coefficient for B
0x00e4	VPP_OFFSET1	Offset1 for YUV to RGB conversion

RISC Address <15:0>	Register	Description
0x00e8	VPP_OFFSET2	Offset2 for YUV to RGB conversion
0x00ec	VPP_OFFSET3	Offset3 for YUV to RGB conversion
0x00f0	VPP_INT_MASK	Interrupt enable
0x00f4	VPP_INT_STATUS	Interrupt status
0x00f8	VPP_ACC	ACC number
0x00fc	VIDEO_FULL_THRESH	VPP output buffer full threshold
0x0100	-	Reserved

Table 270: Address Mapping

Register Descriptions

- VPP Control Register (VPP_CTRL) – 0x0000

Bit	Name	Default	Description
0 (R/W)	PIXEL_FORMAT	1'h0	Input video format: 0: YUV422 format 1: YUV420 format
1 (R/W)	ENDIAN MODE	1'h0	Indicates the endian mode of YUV422 format: 0: Little endian (Y1U0Y0V0) 1: Big endian (Y0U0Y1V0)
3:2 (R/W)	YUV422_FORMAT	2'h0	Indicates the input YUV422 format: 2'b00: YUYV 2'b01: YVYU 2'b10: UYVY 2'b11: VYUY
5:4 (R/W)	OUT_ YUV422_FORMAT	2'h0	Indicates the output YUV422 format, if the output format is YUYV: 2'b00: YUYV 2'b01: YVYU 2'b10: UYVY 2'b11: VYUY
6 (R/W)	YUV422_WEAVE	1'h0	When the input is in YUV422 format and is interlaced. Set this bit if users want to perform weave de-interlacing followed by scaling. If this bit is true, then the VPP_YBASE register will be the odd base address, and the VPP_UBASE register will be the even base address. 0: YUV420 in, or YUV422 frame in, or YUV422

Bit	Name	Default	Description
			interlacing fields in but does not need weave de-interlacing (use scaling to perform intra field de-interlacing just like Bob de-interlacing) 1: YUV422 fields in, performs weave de-interlacing. VPP_UBASE is pointed to even base address.
7 (R/W)	DEST	1'h0	Indicates where the result will be output: 0: Output result to memory. 1: Output result to layer1 of the LCD controller to display.
9:8 (R/W)	OUT_FORMAT	2'h0	00: 16bit RGB565 01: 18bit RGB666 10: 24bit RGB888 11: YUV422, bypasses the YUV2RGB function.
10 (R/W)	OUT_ENDIAN MODE	1'h0	Indicates the endian mode of output YUV422 format: 0: Little endian (Y1U0Y0V0) 1: Big endian (Y0U0Y1V0)
11 (R/W)	CLK_OFF_ENABLE	1'h0	Indicates whether the clock should be off when VPP has completed the current frame: 0: VPP clock is on even when the current frame is completed. 1: VPP clock is off when the current frame is completed.
12 (R/W)	CON_MODE	1'h0	Indicates whether continuous mode is used: 0: Continuous mode is not used. It means that VPP will start next frame on the bit of START's setting. 1: Continuous mode is used. It means that when VPP is connected directly to LCD, VPP will start next frame automatically based on the bit of CONV_SYNC.
13 (R/W)	CON_VSYNC	1'b0	It indicates when to start new frame when continuous mode. 0: VPP start next frame when VPP finishes current frame. 1: VPP start next frame When VPP receives VSYNC from LCD. This bit should be set to 0.
14 (R/W)	Reserved	1'b0	Should be zero
28:15 (R/W)	-	14'h0	Reserved
29 (R/W)	START	1'h0	Starts VPP in single mode: 0: Not starting VPP. 1: Starting VPP after software sets this bit, hardware will clear this bit automatically.

Bit	Name	Default	Description
30 (R)	SCA_OVER	1'h0	This bit is for debugging purposes only: 1'b0: Scaling is not over 1'b1: Scaling is over, but data may not be written out.
31 (R)	BUSY_STATUS	1'h0	1'b1: VPP is running. 1'b0: Operation finished.

Table 271: VPP Control Register

- VPP Y Buffer Base Address (VPP_YBASE) – 0x0004

Bit	Name	Default	Description
29:0 (R/W)	YBASE_ADDR	30'h0	Memory address for Y buffer when working in YUV420 mode. Memory address when working in YUV422 mode. When using yuv422 weave interlacing mode, this address will be pointed to the odd field. This register must be QWORD alignment.
31:30	-	2'h0	Reserved.

Table 272: VPP Y Buffer Base Address

- VPP U Buffer Base Address (VPP_UBASE) – 0x0008

Bit	Name	Default	Description
29:0 (R/W)	UBASE_ADDR	30'h0	Memory byte address for U buffer when working inYUV420 mode. When using yuv422weave interlacing mode, this address will be pointed to the even field. This register must be QWORD alignment.
31:30	-	2'h0	Reserved

Table 273: VPP U Buffer Base Address

- VPP V Buffer Base Address (VPP_VBASE) – 0x000C

Bit	Name	Default	Description
29:0 (R/W)	VBASE_ADDR	30'h0	Memory byte address for V buffer when working inYUV420 mode. This register must be QWORD-aligned.

Bit	Name	Default	Description
31:30	-	2'h0	Reserved.

Table 274: VPP V Buffer Base Address

- VPP Destination Base Address (VPP_DESBASE) – 0x0010

Bit	Name	Default	Description
29:0 (R/W)	DESBASE_ADDR	30'h0	Memory byte address for the destination buffer. This register must be a QWORD-aligned address.
31:30	-	2'h0	Reserved.

Table 275: VPP Destination Base Address

  - If the output is RGB656, this address is 2-byte aligned.
  - If output is RGB666 or RGB888, this address is 4-byte aligned.
  - For YUV420, this address is 4-byte aligned.

- VPP Width (VPP_WIDTH) – 0x0014

Bit	Name	Default	Description
10:0 (R/W)	SRC_WIDTH	11'h0	The source window's width in pixels: Max: 768 pixels wide Min: 12 pixels wide
15:11	-	5'h0	Reserved
26:16 (R/W)	DES_WIDTH	11'h0	The destination window width in pixels: Max: 2047 pixels wide
31:27	-	5'h0	Reserved

Table 276: VPP Width

- VPP Height (VPP_HEIGHT) – 0x0018

Bit	Name	Default	Description
9:0 (R/W)	SRC_HEIGHT	10'h0	The source window height
15:10	-	6'h0	Reserved
25:16 (R/W)	DES_HEIGHT	10'h0	The destination window height
31:26	-	6'h0	Reserved

Table 277: VPP Height

If YUV422weave is used, SRC_HEIGHT should be the height of the frame.

- VPP Stride0 (VPP_STRIDE0) – 0x001c

Bit	Name	Default	Description
11:0 (R/W)	Y_STRIDE	12'h0	Line stride of Y buffer specified in bytes in YUV420 mode. Line stride of the entire buffer specified in bytes when working in YUV422 mode. The y start address of the next line should still be QWORD- aligned.
15:12	-	4'h0	Reserved.
27:16 (R/W)	U_STRIDE	12'h0	Line stride of U buffer specified in bytes in YUV420 mode. The u start address of the next line should still be QWORD-aligned.
31:28	-	4'h0	Reserved.

Table 278: VPP Stride0

- VPP Stride1 (VPP_STRIDE1) – 0x0020

Bit	Name	Default	Description
11:0 (R/W)	V_STRIDE	12'h0	Line stride of V buffer specified in bytes when working in YUV420 mode. The v start address of the next line should still be QWORD-aligned.
15:12	-	4'h0	Reserved.
28:16 (R/W)	DES_STRIDE	13'h0	Line stride of the destination buffer specified in bytes; The start address of the next destination line should meet the following criteria: RGB565: 2-byte aligned. RGB666/RGB888/YUV422: 4-byte aligned.
31:29	-	3'h0	Reserved.

Table 279: VPP Stride1

- Filter Coefficient of Horizontal Scaling Group00 (VPP_HSCA_COEF00) – 0x0024

Bit	Name	Default	Description
14:0 (R/W)	COEF00	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF01	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 280: Filter Coefficient of Horizontal Scaling Group00

Coefficient is in 2's complement <2.12>format. For example, 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group01 (VPP_HSCA_COEF01) – 0x0028

Bit	Name	Default	Description
14:0 (R/W)	COEF02	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF03	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 281: Filter Coefficient of Horizontal Scaling Group01

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group02 (VPP_HSCA_COEF02) – 0x002c

Bit	Name	Default	Description
14:0 (R/W)	COEF04	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
20:16 (R/W)	COEF05	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 282: Filter Coefficient of Horizontal Scaling Group02

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group10 (VPP_HSCA_COEF10) – 0x0030

Bit	Name	Default	Description
14:0 (R/W)	COEF10	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF11	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 283: Filter Coefficient of Horizontal Scaling Group10

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group11 (VPP_HSCA_COEF11) – 0x0034

Bit	Name	Default	Description
14:0 (R/W)	COEF12	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF13	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 284: Filter Coefficient of Horizontal Scaling Group11

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group12 (VPP_HSCA_COEF12) – 0x0038

Bit	Name	Default	Description
14:0 (R/W)	COEF14	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF15	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 285: Filter Coefficient of Horizontal Scaling Group12

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group20 (VPP_HSCA_COEF20) – 0x003c

Bit	Name	Default	Description
14:0 (R/W)	COEF20	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF21	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 286: Filter Coefficient of Horizontal Scaling Group20

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group21 (VPP_HSCA_COEF21) – 0x0040

Bit	Name	Default	Description
14:0 (R/W)	COEF22	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF23	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 287: Filter Coefficient of Horizontal Scaling Group21

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group22 (VPP_HSCA_COEF22) – 0x0044

Bit	Name	Default	Description
14:0 (R/W)	COEF24	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF25	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 288: Filter Coefficient of Horizontal Scaling Group22

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of horizontal scaling Group30 (VPP_HSCA_COEF30) – 0x0048

Bit	Name	Default	Description
14:0 (R/W)	COEF30	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF31	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 289: Filter Coefficient of Horizontal Scaling Group30

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group31 (VPP_HSCA_COEF31) – 0x004c

Bit	Name	Default	Description
14:0 (R/W)	COEF32	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF33	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 290: Filter Coefficient of Horizontal Scaling Group31

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group32 (VPP_HSCA_COEF32) – 0x0050

Bit	Name	Default	Description
14:0 (R/W)	COEF34	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF35	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 291: Filter Coefficient of Horizontal Scaling Group32

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group40 (VPP_HSCA_COEF40) – 0x0054

Bit	Name	Default	Description
14:0 (R/W)	COEF40	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF41	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 292: Filter Coefficient of Horizontal Scaling Group40

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group41 (VPP_HSCA_COEF41) – 0x0058

Bit	Name	Default	Description
14:0 (R/W)	COEF42	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF43	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 293: Filter Coefficient of Horizontal Scaling Group41

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group42 (VPP_HSCA_COEF42) – 0x005c

Bit	Name	Default	Description
14:0 (R/W)	COEF44	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF45	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 294: Filter Coefficient of Horizontal Scaling Group42

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group50 (VPP_HSCA_COEF50) – 0x0060

Bit	Name	Default	Description
14:0 (R/W)	COEF50	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF51	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 295: Filter Coefficient of Horizontal Scaling Group50

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group51 (VPP_HSCA_COEF51) – 0x0064

Bit	Name	Default	Description
14:0 (R/W)	COEF52	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF53	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 296: Filter Coefficient of Horizontal Scaling Group51

Coefficient is in 2's complement <2.12>format, i.e. 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group52 (VPP_HSCA_COEF52) – 0x0068

Bit	Name	Default	Description
14:0 (R/W)	COEF54	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF55	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 297: Filter Coefficient of Horizontal Scaling Group52

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group60 (VPP_HSCA_COEF60) – 0x006c

Bit	Name	Default	Description
14:0 (R/W)	COEF60	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF61	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 298: Filter Coefficient of Horizontal Scaling Group60

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group61 (VPP_HSCA_COEF61) – 0x0070

Bit	Name	Default	Description
14:0 (R/W)	COEF62	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF63	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 299: Filter Coefficient of Horizontal Scaling Group61

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group62 (VPP_HSCA_COEF62) – 0x0074

Bit	Name	Default	Description
14:0 (R/W)	COEF64	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF65	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 300: Filter Coefficient of Horizontal Scaling Group62

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group70 (VPP_HSCA_COEF70) – 0x0078

Bit	Name	Default	Description
14:0 (R/W)	COEF70	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF71	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 301: Filter Coefficient of Horizontal Scaling Group70

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group71 (VPP_HSCA_COEF71) – 0x007c

Bit	Name	Default	Description
14:0 (R/W)	COEF72	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF73	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 302: Filter Coefficient of Horizontal Scaling Group71

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group72 (VPP_HSCA_COEF72) – 0x0080

Bit	Name	Default	Description
14:0 (R/W)	COEF74	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF75	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 303: Filter Coefficient of Horizontal Scaling Group72

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group80 (VPP_HSCA_COEF80) – 0x0084

Bit	Name	Default	Description
14:0 (R/W)	COEF80	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF81	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 304: Filter Coefficient of Horizontal Scaling Group80

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits(COEFxx[13:12]) and one sign bit(COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group81 (VPP_HSCA_COEF81) – 0x0088

Bit	Name	Default	Description
14:0 (R/W)	COEF82	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF83	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 305: Filter Coefficient of Horizontal Scaling Group81

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Horizontal Scaling Group82 (VPP_HSCA_COEF82) – 0x008c

Bit	Name	Default	Description
14:0 (R/W)	COEF84	15'h0	Coefficient for tap4
15	-	1'h0	Reserved
30:16 (R/W)	COEF85	15'h0	Coefficient for tap5
31	-	1'h0	Reserved

Table 306: Filter Coefficient of Horizontal Scaling Group82

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group00 (VPP_VSCA_COEF00) – 0x0090

Bit	Name	Default	Description
14:0 (R/W)	COEF00	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF01	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 307: Filter Coefficient of Vertical Scaling Group00

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group01 (VPP_VSCA_COEF01) – 0x0094

Bit	Name	Default	Description
14:0 (R/W)	COEF02	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF03	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 308: Filter Coefficient of Vertical Scaling Group01

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits(COEFxx[13:12]) and one sign bit(COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group10 (VPP_VSCA_COEF10) – 0x0098

Bit	Name	Default	Description
14:0 (R/W)	COEF10	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF11	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 309: Filter Coefficient of Vertical Scaling Group10

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group11 (VPP_VSCA_COEF11) – 0x009c

Bit	Name	Default	Description
14:0 (R/W)	COEF12	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF13	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 310: Filter Coefficient of Vertical Scaling Group11

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits(COEFxx[13:12]) and one sign bit(COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group20 (VPP_VSCA_COEF20) – 0x00a0

Bit	Name	Default	Description
14:0 (R/W)	COEF20	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF21	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 311: Filter Coefficient of Vertical Scaling Group20

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits(COEFxx[13:12]) and one sign bit(COEFxx[14]).

- Filter Coefficient of vertical scaling Group21 (VPP_VSCA_COEF21) – 0x00a4

Bit	Name	Default	Description
14:0 (R/W)	COEF22	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF23	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 312: Filter Coefficient of vertical scaling Group21

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group30 (VPP_VSCA_COEF30) – 0x00a8

Bit	Name	Default	Description
14:0 (R/W)	COEF30	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF31	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 313: Filter Coefficient of Vertical Scaling Group30

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group31 (VPP_VSCA_COEF31) – 0x00ac

Bit	Name	Default	Description
14:0 (R/W)	COEF32	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF33	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 314: Filter Coefficient of Vertical Scaling Group31

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group40 (VPP_VSCA_COEF40) – 0x00b0

Bit	Name	Default	Description
14:0 (R/W)	COEF40	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF41	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 315: Filter Coefficient of Vertical Scaling Group40

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group41 (VPP_VSCA_COEF41) – 0x00b4

Bit	Name	Default	Description
14:0 (R/W)	COEF42	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF43	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 316: Filter Coefficient of Vertical Scaling Group41

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group50 (VPP_VSCA_COEF50) – 0x00b8

Bit	Name	Default	Description
14:0 (R/W)	COEF50	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF51	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 317: Filter Coefficient of Vertical Scaling Group50

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group51 (VPP_VSCA_COEF51) – 0x00bc

Bit	Name	Default	Description
14:0 (R/W)	COEF52	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF53	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 318: Filter Coefficient of Vertical Scaling Group51

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group60 (VPP_VSCA_COEF60) – 0x00c0

Bit	Name	Default	Description
14:0 (R/W)	COEF60	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF61	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 319: Filter Coefficient of Vertical Scaling Group60

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group61 (VPP_VSCA_COEF61) – 0x00c4

Bit	Name	Default	Description
14:0 (R/W)	COEF62	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF63	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 320: Filter Coefficient of Vertical Scaling Group61

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group70 (VPP_VSCA_COEF70) – 0x00c8

Bit	Name	Default	Description
14:0 (R/W)	COEF70	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF71	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 321: Filter Coefficient of Vertical Scaling Group70

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group71 (VPP_VSCA_COEF71) – 0x00cc

Bit	Name	Default	Description
14:0 (R/W)	COEF72	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF73	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 322: Filter Coefficient of Vertical Scaling Group71

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group80 (VPP_VSCA_COEF80) – 0x00d0

Bit	Name	Default	Description
14:0 (R/W)	COEF80	15'h0	Coefficient for tap0
15	-	1'h0	Reserved
30:16 (R/W)	COEF81	15'h0	Coefficient for tap1
31	-	1'h0	Reserved

Table 323: Filter Coefficient of Vertical Scaling Group80

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- Filter Coefficient of Vertical Scaling Group81 (VPP_VSCA_COEF81) – 0x00d4

Bit	Name	Default	Description
14:0 (R/W)	COEF82	15'h0	Coefficient for tap2
15	-	1'h0	Reserved
30:16 (R/W)	COEF83	15'h0	Coefficient for tap3
31	-	1'h0	Reserved

Table 324: Filter Coefficient of Vertical Scaling Group81

Coefficient is in 2's complement <2.12>format, i.e., 12 fractional data bits (COEFxx[11:0]), two integer bits (COEFxx[13:12]) and one sign bit (COEFxx[14]).

- YUV to RGB Coefficient for R (VPP_RCOEF) – 0x00d8

Bit	Name	Default	Description
9:0 (R/W)	C3	10'h0	V coefficient for R
19:10 (R/W)	C2	10'h0	U coefficient for R
29:20 (R/W)	C1	10'h0	Y coefficient for R
31:30	-	2'h0	Reserved

Table 325: YUV to RGB Coefficient for R

- YUV to RGB Coefficient for G (VPP_GCOEF) – 0x00dc

Bit	Name	Default	Description
9:0 (R/W)	C6	10'h0	V coefficient for G
19:10 (R/W)	C5	10'h0	U coefficient for G
29:20 (R/W)	C4	10'h0	Y coefficient for G
31:30	-	2'h0	Reserved

Table 326: YUV to RGB Coefficient for G

- YUV to RGB Coefficient for B (VPP_BCOEF) – 0x00e0

Bit	Name	Default	Description
9:0 (R/W)	C9	10'h0	V coefficient for B
19:10 (R/W)	C8	10'h0	U coefficient for B
29:20 (R/W)	C7	10'h0	Y coefficient for B
31:30	-	2'h0	Reserved

Table 327: YUV to RGB Coefficient for B

- Offset1 for the YUV to RGB Conversion (VPP_OFFSET1) – 0x00e4

Bit	Name	Default	Description
23:0 (R/W)	OFFSET1	24'h0	Offset coefficient for R
31:24	-	8'h0	Reserved

Table 328: Offset1 for the YUV to RGB Conversion

- Offset2 for the YUV to RGB Conversion (VPP_OFFSET2) – 0x00e8

Bit	Name	Default	Description
23:0 (R/W)	OFFSET2	24'h0	Offset coefficient for G
31:24	-	8'h0	Reserved

Table 329: Offset2 for the YUV to RGB Conversion

- Offset3 for the YUV to RGB Conversion (VPP_OFFSET3) – 0x00ec

Bit	Name	Default	Description
23:0 (R/W)	OFFSET3	24'h0	Offset coefficient for B
31:24	-	8'h0	Reserved

Table 330: Offset3 for the YUV to RGB Conversion

- Interrupt Mask (VPP_INT_MASK) – 0x00f0

Bit	Name	Default	Description
0 (R/W)	INT_SINGLE_MASK	1'h0	Interrupt enable of single mode
1 (R/W)	INT_CON_MASK	1'h0	Interrupt enable of continuous mode
2 (R/W)	INT_AB_MASK	1'h0	Interrupt enable of abnormal condition
31:3	-	29'h0	Reserved

Table 331: Interrupt Mask

- Interrupt Status (VPP_INT_STATUS) – 0x00f4

Bit	Name	Default	Description
0 (R/W)	INT_SINGLE_STATUS	1'h0	This bit will be set if the frame is completed; writing this bit will clear this interrupt.
1 (R/W)	INT_CON_STATUS	1'h0	This bit will be set if scaling is completed for last picture in continuous mode; writing this bit will clear this interrupt.
2 (R/W)	INT_AB_STATUS	1'h0	This bit will be set if either the current frame is not completed while a new start has been set in single mode, or the current frame is not completed while a new VSYNC from the LCD controller comes in continuous mode. Writing this bit will clear this interrupt.
31:3	-	29'h0	Reserved

Table 332: Interrupt Status

- ACC Number for Arbiter (VPP_ACC) – 0x00f8

Bit	Name	Default	Description
7:0 (R/W)	ACC_NUM	8'hf	The consecutive grant number of a read/write master of the arbiter in VPP.
31:8	-	24'h0	Reserved.

Table 333: ACC Number for Arbiter

- VPP Output Buffer Full Threshold (VIDEO_FULL_THRESH) – 0x00fc

Bit	Name	Default	Description
3:0 (R/W)	FIFO_FULL_THRESH	4'h7	The full level for the VPP output buffer. When VPP_CTRL[14] is set to 0, this bit must be set to 4'h7.
4	THRESH_VAILD	1'b1	If bit[3:0] is valid: 0: Setting of bit[3:0] is not valid for the hardware. 1: Setting of bit[3:0] is valid for the hardware. When VPP_CTRL14] is set to 0, this bit must be set to 1.
31:5	-	27'h0	Reserved

Table 334: VPP Output Buffer Full Threshold

PCI_COPY Data Engine

Overview

The PCI_COPY Engine is a PCI device that helps to perform single-channel DMA transfer for big block memories.

It can be configured by the RISC as a PCI slave and perform DMA operations as a PCI master.

It can help to more efficiently transfer data from/to ROM interface.

Feature List

- Supports various kinds of source and destination (such as FIFO type device)
- Supports both I/O and MEMORY transfer mode for different devices
- Supports 2D DMA operations
- Supports data switches

Register Definition

Register Address Mapping

The base address of the PCI_COPY register can be accessed at 0x57900000.

- PCI Bridge RISC I/O Registers

RISC Address <11:0>	Register	Description
0x000	COPY_CH0_STATUS	The copy mode control of CH0
0x004	COPY_CH0_X_NUM	The horizontal copy number of CH0
0x008	COPY_CH0_Y_NUM	The vertical copy number of CH0
0x00C	COPY_CH0_SRC	The address of the first source data of CH0
0x010	COPY_CH0_DST	The address of the first destination data of CH0
0x014	COPY_CH0_2D_STEP	The 2D step of CH0
0x018	COPY_CH0_RETRY_NUM	The retry number of CH0
0x01C~0x05c	-	Reserved
0x060	COPY_INT_STATUS	Interrupt status register
0x064~0x05	COPY_INT_EN	Interrupt enable register
0x068~0x06C	-	Reserved
0x070	COPY_CH0_EN	CH0 enable register

Table 335: PCI_COPY Register Mapping

- PCI_COPY Channel 0 Status Register (COPY_CH0_STATUS) – 0x0
  This register is used to configure the copy mode, including:
  - Source/destination burst control
  - Source/destination FIFO control
  - Source/destination I/O or memory control
  - Switch control of copied data
  - Source/destination width control

NOTE – When programming the source or destination is in burst mode, it should be programmed in DWORD. The source or destination cannot be a FIFO.

Bit	Name	Default	Description
0 (R/W)	DST_BURST	1'b0	The destination burst control. 1: The destination should be accessed in burst mode. 0: The destination should be accessed in single mode.
1 (R/W)	SRC_BURST	1'b0	The source burst control. 1: The source should be accessed in burst mode. 0: The source should be accessed in single mode.
2 (R/W)	DST_FIFO	1'b0	The destination FIFO control. 1: The destination is a FIFO. 0: The destination is not a FIFO.
3 (R/W)	SRC_FIFO	1'b0	The source FIFO control. 1: The source is a FIFO. 0: The source is not a FIFO.
4 (R/W)	DST_IO_MEM	1'b0	The destination memory and I/O control. 1: The destination is memory. 0: The destination is I/O.
5 (R/W)	SRC_IO_MEM	1'b0	The source memory and I/O control. 1: The source is memory. 0: The source is I/O.
7:6 (R/W)	DATA_SWITCH	2'h0	The data switch control to change the byte sequence in a DWORD. 00: No switches 01: 1234 → 3412 10: 1234 → 2143 11: 1234 → 4321
9:8 (R/W)	SRC_WIDTH	2'h0	The source width mode. 00: DWORD 01: Word 10: Byte 11: Reserved
11:10 (R/W)	DST_WIDTH	2'h0	The destination width mode. 00: DWORD 01: Word 10: Byte 11: Reserved
31:12	-	20'h0	Reserved.

Table 336: PCI_COPY Channel 0 Status Register (COPY_CH0_STATUS)

- PCI_COPY CH0X Number Register (COPY_CH0_X_NUM) – 0x4

  This register is used to configure the horizontal data amount that to be copied; the amount of horizontal data is based on bytes.

NOTE – Set X_NUM according to the values of SRC_WIDTH and DST_WIDTH (whichever is bigger) in the COPY_CH0_STATUS register. If WIDTH is in DWORD, X_NUM must also be in DWORD. If WIDTH is in WORD, X_NUM must also be in WORD.

Bit	Name	Default	Description
15:0 (R/W)	X_NUM	16'h0	The horizontal amount of the 2D copy data. Unit: Byte
31:16	-	16'h0	Reserved.

Table 337: PCI_COPY CH0X Number Register (COPY_CH0_X_NUM)

- PCI_COPY CH0 Y Number Register (COPY_CH0_Y_NUM) – 0x8

  This register is used to configure the vertical amount of the 2D copy data. If this register is large than 1, then the process will become 2D data copy.

Bit	Name	Default	Description
9:0 (R/W)	Y_NUM	10'h001	The vertical amount of the 2D copy data
31:10	-	22'h0	Reserved

Table 338: PCI_COPY CH0 Y Number Register (COPY_CH0_Y_NUM)

- PCI_COPY CH0Source Register (COPY_CH0_SRC) – 0xC

  This register is used to configure the address of the first source data:

Bit	Name	Default	Description
31:0 (R/W)	SRC_ADDR	32'h0	The address of the first source data

Table 339: PCI_COPY CH0Source Register (COPY_CH0_SRC)

- PCI_COPY CH0Destination Register (COPY_CH0_DST) – 0x10

  This register is used to configure the address of the first destination data.

Bit	Name	Default	Description
31:0 (R/W)	DST_ADDR	32'h0	The address of the first destination data

Table 340: PCI_COPY CH0Destination Register (COPY_CH0_DST)

- PCI_COPY CH0 2D Step Register (COPY_CH0_2D_STEP) – 0x14

Bit	Name	Default	Description
15:0 (R/W)	2D_STEP	16'h0	2D data step, it is the offset between two adjoining lines' start positions; it shall be greater /equal to than the X_NUM.
31:16	-	16'h0	Reserved.

Table 341: PCI_COPY CH0 2D Step Register (COPY_CH0_2D_STEP)

- PCI_COPY CH0 Retry Number Register (COPY_CH0_RETRY_NUM) – 0x18
  This register is used to configure the PCI operation retry number.

Bit	Name	Default	Description
7:0 (R/W)	RETRY_NUM	8'h07	Retry number
31:8	-	24'h0	Reserved

Table 342: PCI_COPY CH0 Retry Number Register (COPY_CH0_RETRY_NUM)

- PCI_COPY Interrupt Status Register (COPY_INT_STATUS) – 0x60
  After the block copy process is done, the corresponding bit will be set to high. It will be reset either after writing 1'b1 to it or after the corresponding channel is enabled.

Bit	Name	Default	Description
0 (R)	CH0_INT	1'b0	The status of the COPY CH0 interrupt. It will be set to 1'b1 when the CH0 transfer finishes and will be cleared to 1'b0 when CH0 is enabled.
31:1	-	31'h0	Reserved.

Table 343: PCI_COPY Interrupt Status Register (COPY_INT_STATUS)

- PCI_COPY Interrupt Enable Register (COPY_INT_EN) – 0x64
  If this register is 0, then PCI_COPY will not generate an interrupt signal when the data transfer is finished.

Bit	Name	Default	Description
0 (R/W)	CH0_INT_EN	1'b0	The CH0 interrupt enable (active high)
31:1	-	31'h0	Reserved

Table 344: PCI_COPY Interrupt Enable Register (COPY_INT_EN)

- PCI_COPY CH0Enable Register (COPY_CH0_EN) – 0x70

  After it is set to high, the corresponding channel will begin to work, and it will be reset automatically after the transfer is finished.

Bit	Name	Default	Description
0 (R/W)	CH_EN	1'b0	The Channel enable
31:1	-	31'h0	Reserved

Table 345: PCI_COPY CH0Enable Register (COPY_CH0_EN)

PERIPHERAL CONTROL MODULES

PCI Bridge (SYS2PCI)

Overview

There is a bridge between the System Bus and PCI Bus (we call this bridge SYS2PCI), which coverts the I/O and memory accesses between the System Bus and PCI Bus. For I/O accesses, the PCI Bridge will convert the RISC/DSP I/O accesses into PCI data transactions and passes to the corresponding PCI devices. For memory access, the PCI Bridge is both a master and a slave on the System Bus. In most cases when PCI devices transfer data to or from the memory Controller, the PCI Bridge acts as the bus master on the System Bus.

Feature List

- Supports the internal AXI bus protocol as a master
- Supports the internal PCI bus protocol as master and slave
- Supports different ratio of system clock and I/O clock
- Supports 64bit AXI to 32-bit PCI, I/O to PCI and 32-bit PCI to 64-bit AXI transforms

Functional Descriptions

Block Diagram

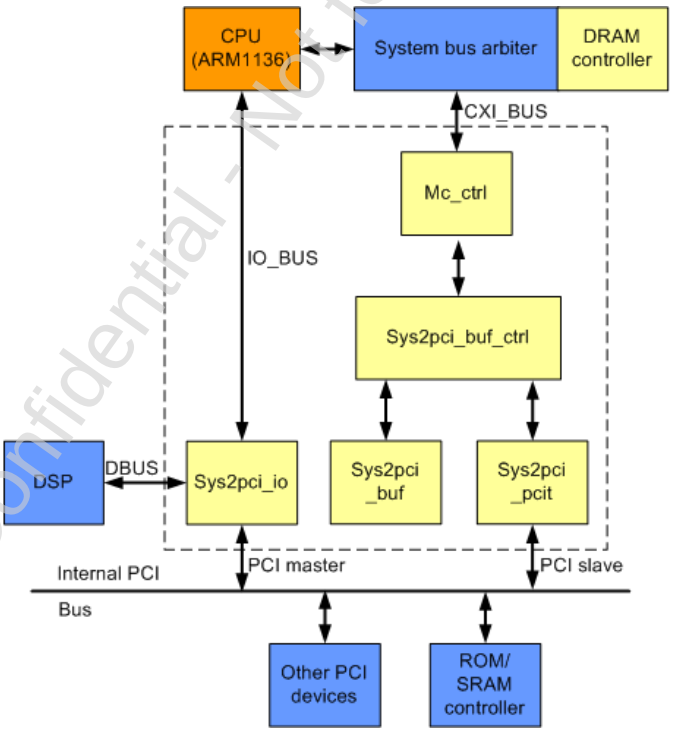


Figure 49: SYS2PCI Block Diagram

Clock Topology

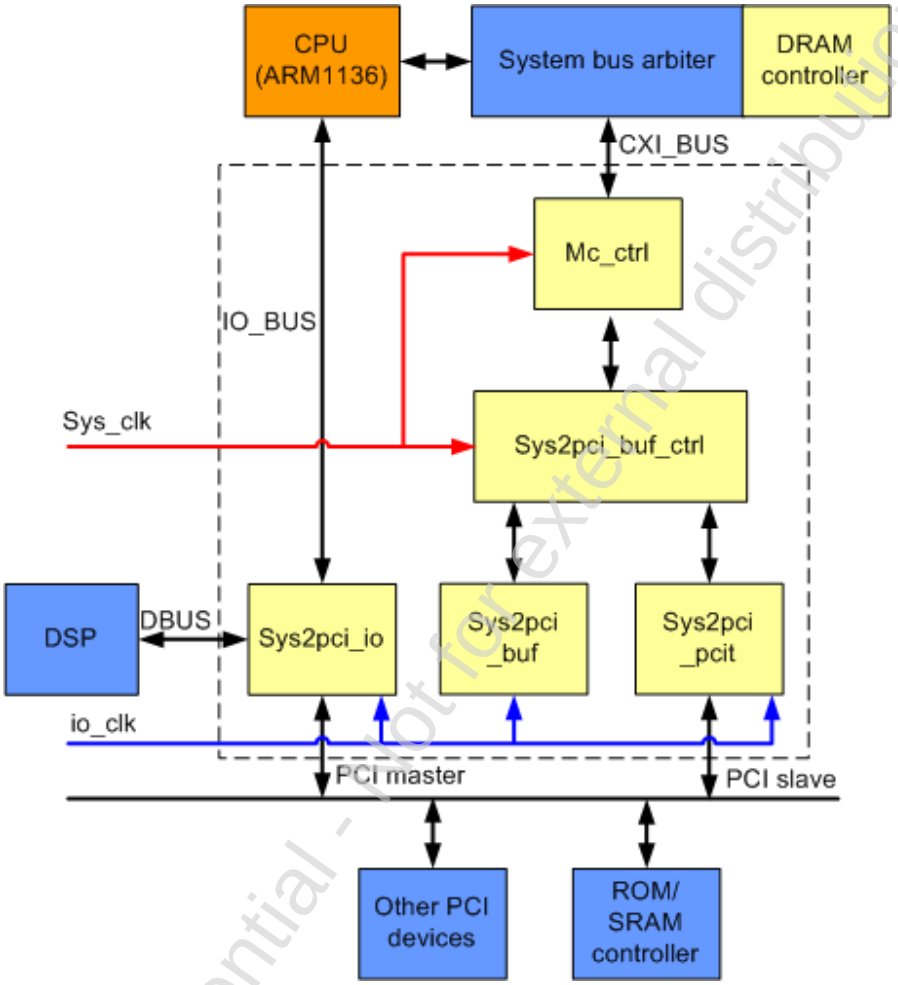


Figure 50: SYS2PCI Clock Topology Diagram

Register Definitions

Register Address Mapping

The base address of the SYS2PCI register can be accessed through RISC I/O. The base address is 0x800c0000.

Both RISC and DSP are able to access System to PCI Bridge registers.

- PCI Bridge RISC I/O Registers

RISC Address <11:0>	Register	Description
0x0000	SYS2PCI_RETRY	System to PCI Bridge retry value
0x0004	SYS2PCI_INT_EN	System to PCI Bridge interrupt enable
0x0008	SYS2PCI_INT_STATUS	System to PCI Bridge interrupt status
0x000C	SYS2PCI_WRITEBUFFER	System to PCI Bridge write buffer set
0x0010	SYS2PCI_TIMEOUT	System to PCI Bridge timeout
Others	-	Reserved

Table 346: PCI Bridge RISC I/O Register Mapping

- PCI Bridge DSP I/O Registers

  To access PCI from the DSP side, System to PCI Bridge contains specific registers to convert the 16-bit access on DSP to 32-bit access on PCI.

DSP Byte Address <5:0>	Register	Description
0x0000	SYS2PCI_DSP_OPERATE	System to PCI Bridge DSP Operation
0x0002	SYS2PCI_DSP_ADDL	System to PCI Bridge DSP address low
0x0004	SYS2PCI_DSP_ADDH	System to PCI Bridge DSP address high
0x0006	SYS2PCI_DSP_DATAL	System to PCI Bridge DSP data low
0x0008	SYS2PCI_DSP_DATAH	System to PCI Bridge DSP data high
0x000a	SYS2PCI_DSP_STATUS	System to PCI Bridge DSP status
Others	-	Reserved

Table 347: PCI Bridge DSP I/O Register Mapping

To perform a read operation, DSP needs to:

- Write SYS2PCI_DSP_ADDL, SYS2PCI_DSP_ADDH for correct PCI addresses
- Write SYS2PCI_DSP_OPERATE with correct BE, RW=1
- Poll SYS2PCI_DSP_OPERATE until RDY=1
- Read data from SYS2PCI_DSP_DATAL, SYS2PCI_DSP_DATAH

To perform a write operation, DSP needs to:

- Write SYS2PCI_DSP_ADDL, SYS2PCI_DSP_ADDH for correct PCI addresses
- Write SYS2PCI_DSP_DATAL, SYS2PCI_DSP_DATAH for write data on PCI

- Write SYS2PCI_DSP_OPERATE with correct BE, RW=0
- Poll SYS2PCI_DSP_OPERATE until RDY=1

Register Descriptions (RISC)

- PCI Bridge Retry Value Register (SYS2PCI_RETRY) – 0x0000

Bit	Name	Default	Description
5:0 (R/W)	RETRY<5:0>	6'h8	System to PCI Bridge retry value
31:6	-	-	Reserved

Table 348: PCI Bridge Retry Value Register (SYS2PCI_RETRY)

SYS2PCI_RETRY is a 6-bit register that controls the retry time on the PCI side. When PCI access is terminated with retry, the System to PCI Bridge will wait for SYS2PCI_RETRY *T_{PCI_CLK} before it retries the same PCI access.

- PCI Bridge Interrupt Enable Register (SYS2PCI_INT_EN) – 0x0004

Bit	Name	Default	Description
0 (R/W)	Reserved	1'b0	Need to be written with 0.
1 (R/W)	MABT	1'b0	System to PCI Bridge Master abort interrupt enable. 1: Enable the interrupt 0: Disable the interrupt
2 (R/W)	TIMOUT	1'b0	System to PCI Bridge Time out interrupt enable. 1: Enable the interrupt 0: Disable the interrupt
31:3	-	-	Reserved.

Table 349: PCI Bridge Interrupt Enable Register (SYS2PCI_INT_EN)

System to PCI Bridge Master abort interrupt enable: When there is master abort on the PCI bus, the current access will be cancelled. A master abort interrupt will be generated.

System to PCI Bridge Time out interrupt enable: When an access on the PCI bus is terminated with retry X (X>TIMOUT_VALUE) times, the current access will be cancelled and a time-out abort interrupt will be generated.

- PCI Bridge Interrupt Status Register (SYS2PCI_INT_STATUS) – 0x0008

Bit	Name	Default	Description
0 (R/W)	-	1'b0	Reserved (need to be written with 0)
1 (R/W)	MABT	1'b0	System to PCI Bridge Master abort interrupt Read: 1: Interrupt pending 0: No interrupt pending Write: 1: Clear the interrupt pending 0: No effects
2 (R/W)	TIMOUT	1'b0	System to PCI Bridge Time out interrupt Read: 1: Interrupt pending 0: No interrupt pending Write: 1: Clear the interrupt pending 0: No effects
31:3	-	-	Reserved

Table 350: PCI Bridge Interrupt Status Register (SYS2PCI_INT_STATUS)

System to PCI Bridge Master abort interrupt: When there is master abort on the PCI bus, the current access will be cancelled and a master abort interrupt will be generated.

System to PCI Bridge Time out interrupt: When an access on PCI bus is terminated with retry X(X> TIMEOUT) times, the current access will be cancelled and a time-out abort interrupt will be generated.

- PCI Bridge Write Buffer Enable Register (SYS2PCI_WRITEBUFFER) – 0x000C

Bit	Name	Default	Description
0 (R/W)	WB	1'b0	System to PCI Bridge write buffer enable 1: Enable write buffer 0: Disable write buffer When the write buffer is enabled, RISC I/O write to PCI device will be finished immediately when System to PCI Bridge FIFO is ready to hold the command. Then System to PCI Bridge will transfer the command into the FIFO on the PCI bus. But the write operation will only be truly finished when the command is issued to the PCI bus. It is recommended to set the write buffer to enable, which can speed up the write operation. However the user needs to perform a read operation to the same address to guarantee

Bit	Name	Default	Description
			that the write operation is really finished on the device.
1 (R/W)	RM	1'b1	System to PCI Bridge read mode 1: Read will wait for current transfer to system memory finish. 0: I/O read has no relations with transfer to system memory.
31:2	-	-	Reserved

Table 351: PCI Bridge Write Buffer Enable Register (SYS2PCI_WRITEBUFFER)

- System to PCI Bridge Time Out Register (SYS2PCI_TIMEOUT) – 0x0010

Bit	Name	Default	Description
15:0 (R/W)	TIMEOUT<15:0>	16'hffff	System to PCI Bridge timeout value
31:6	-	-	Reserved

Table 352: System to PCI Bridge Time Out Register (SYS2PCI_TIMEOUT)

System to PCI Bridge timeout value is a 16-bit register that controls the retry number on the PCI side. If a read/write from RISC is retried for more than the time specified by TIMEOUT, then this command will be discarded. An interrupt will occur if users set the interrupt enable bit. If this value is set to 0, then the retry number time-out will never occur.

Register Descriptions (DSP)

- PCI Bridge DSP Operation Register (SYS2PCI_DSP_OPERATE) – 0x0000

Bit	Name	Default	Description
0 (R/W)	RW	1'b1	1: DSP read PCI 0: DSP write PCI
1 (R)	RDY	1'b1	Read: 1: DSP operation finished 0: DSP operation not finished Write: 1: Reset to 0 0: Reset to 0
3:2	-	-	Reserved
7:4 (R/W)	BE<3:0>	4'h0	Byte enable for PCI read/write BE<0> - SYS2PCI_DSP_DATAL[7:0] BE<1> - SYS2PCI_DSP_DATAL[15:8] BE<2> - SYS2PCI_DSP_DATAH[7:0] BE<3> - SYS2PCI_DSP_DATAH[15:8]

Bit	Name	Default	Description
			1: Enable byte 0: Disable the byte
15:8	-	-	Reserved

Table 353: PCI Bridge DSP Operation Register (SYS2PCI_DSP_OPERATE)

- PCI Bridge DSP Address Low Register (SYS2PCI_DSP_ADDL) – 0x0002

Bit	Name	Default	Description
15:0 (R/W)	ADDL	16'h0	PCI address low for DSP I/O access on PCI bus

Table 354: PCI Bridge DSP Address Low Register (SYS2PCI_DSP_ADDL)

- PCI Bridge DSP Address Low Register (SYS2PCI_DSP_ADDL) – 0x0002

Bit	Name	Default	Description
15:0 (R/W)	ADDL	16'h0	PCI address low for DSP I/O access on PCI bus

Table 355: PCI Bridge DSP Address Low Register (SYS2PCI_DSP_ADDL)

- PCI Bridge DSP Address High Register (SYS2PCI_DSP_ADDH) – 0x0004

Bit	Name	Default	Description
15:0 (R/W)	ADDH	16'h0	PCI address high for DSP I/O access on PCI bus

Table 356: PCI Bridge DSP Address High Register (SYS2PCI_DSP_ADDH)

- PCI Bridge DSP Data Low Register (SYS2PCI_DSP_DATAL) – 0x0006

Bit	Name	Default	Description
15:0 (R/W)	DATAL	16'h0	PCI data low for DSP I/O access on PCI bus. It contains data for read/write.

Table 357: PCI Bridge DSP Data Low Register (SYS2PCI_DSP_DATAL)

- PCI Bridge DSP Data High Register (SYS2PCI_DSP_DATAH) – 0x0008

Bit	Name	Default	Description
15:0 (R/W)	DATAH	16'h0	PCI data high for DSP IO access on PCI bus. It contains data for read/write.

Table 358: PCI Bridge DSP Data High Register (SYS2PCI_DSP_DATAH)

- PCI Bridge DSP Status Register (SYS2PCI_DSP_STATUS) – 0x000A

Bit	Name	Default	Description
0 (R)	-	1'b0	Reserved
1 (R)	MABT	1'b0	System to PCI Bridge Master abort interrupt Read: 1: Interrupt pending 0: No interrupt pending Write: No effects
2 (R)	TIMEOUT	1'b0	System to PCI Bridge Time out interrupt Read: 1: Interrupt pending 0: No interrupt pending Write: No effects
31:3	-	-	Reserved

Table 359: PCI Bridge DSP Status Register (SYS2PCI_DSP_STATUS)

This register is read-only about interrupt status information for DSP access. It requires RISC access to clear the interrupt status.

PCI Arbiter

Overview

SiRFatlasV implements a high-speed internal PCI bus that can run up to 125MHz. The PCI Bus Arbiter supports up to three PCI devices: ROM/SRAM Controller, SDIO Host Controller, PCI_COPY. The arbitration works in round-robin fashion.

All the PCI device registers in PCI Subsystem share the same 256MB segment of the RISC address space (0x5000_0000~5FFF_FFFF).

RISC Address Range	Usage	Resource Size
0x57E0_0000~57EF_FFFF	Reserved	1MB
0x57D0_0000~57DF_FFFF	Reserved	1MB
0x57C0_0000~57CF_FFFF	Reserved	1MB
0x57B0_0000~57BF_FFFF	Reserved	1MB
0x57A0_0000~57AF_FFFF	PCI_ROM	1MB
0x5790_0000~579F_FFFF	PCI_COPY	1MB
0x5780_0000~578F_FFFF	Reserved	1MB
0x5770_0000~577F_FFFF	Reserved	1MB

Table 360: PCI Device Registers Mapping

Function Descriptions

Block Diagram

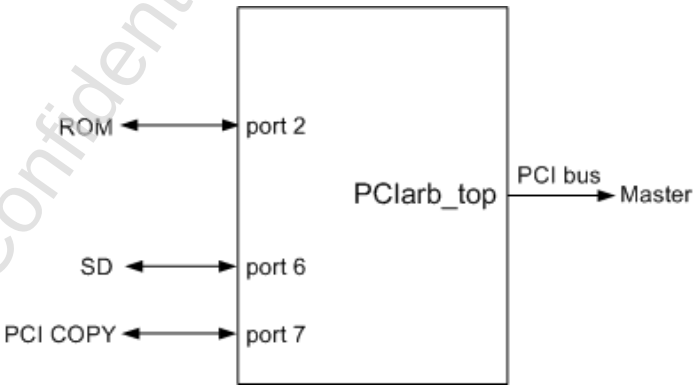


Figure 51: PCI_ARB Block Diagram

USB Controller

Overview

The USB is standard USB interface (D+/D-, ID), integrated PHY. The USB-HS core is implementing a USB interface in compliance with the USB 2.0 specification. The USB-HS core is able to act as peripheral controller, or host controller or a dual role OTG controller that is able to negotiate the host or peripheral role on the bus in compliance with the On-The-Go (OTG) supplement to the USB specification.

The USB-HS core is designed to make efficient use of the system resources in a SoC design. The 32-bit system bus interface contains a chaining Direct Memory Access (DMA) engine that reduces the interrupt load on the application processor, and reduces the total system bus bandwidth that must be dedicated to servicing the USB interface requirements. By transferring the data to system memory at wire rates, the buffer memory requirement within the core is minimized. The USB-HS also makes strategic use of the processor for tasks that do not require timing critical responses to reduce the amount of special purpose logic.

Feature List

- USB 2.0 high-speed OTG (On-The-Go) dual-role USB host controller or USB device controller operation using the same hardware
- Intel™ EHCI host controller
- Direct support for connecting with USB host or device with (VBUS, D+, D-, ID, GND)
- Dual-port RAM buffers that isolate memory latency on the system bus based on USB timing requirements

Pin Descriptions

External Pin Descriptions

USB has an integrated transceiver on the chip; you can directly connect USB devices with USB through it.

Pin Name	Direction	Description
X_USB_VBUS	I/O	USB power signal, 5V
X_USB_DP	I/O	Positive output channel connected to the serial USB cable
X_USB_DN	I/O	Negative output channel connected to the serial USB cable
X_USB_ID	I/O	USB ID pin of mini-AB receptacle

Table 361: USB External Pin Description

Functional Descriptions

Block Diagram

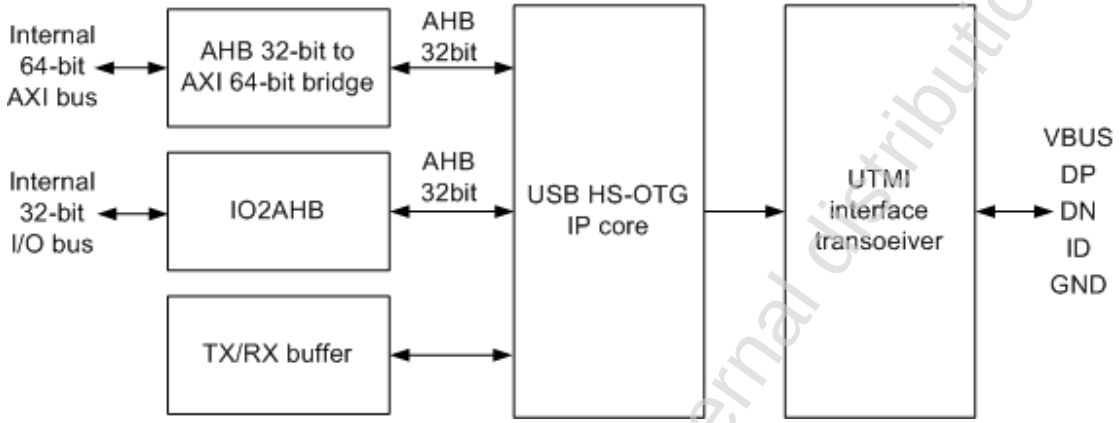


Figure 52: USB Block Diagram

Module Descriptions

Host and Device Data Structure

For details of the data structure, contact your SiRF FAE.

AHB 32Bit to AXI 64Bit Bridge and IO2AHB Bridge

The USB-OTG core supports an AHB interface as well as a general bus interface. This interface goes on top of the original bus interface and conforms to the AHB protocol as defined in the AMBA specification. The USB-OTG core contains both an AHB master and slave.

- The master interface only uses Single or unspecified INCR burst types
- The slave interface only supports single accesses
- Master can handle split, retry and error transfers
- Slave does not handle error and retry responses. The slave transfers are simple and will always be completed

To connect the USB-OTG core to the internal AXI bus, the AHB to AXI conversion is needed, it performs the following conversions:

- Converts the AHB master signal to the AXI master.
- Converts the I/O Bus signal to the AHB slave interface.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O interface	0x99000000

Table 362: USB Base Address

Register Mapping

RISC Address<15:0>	Register	Description
0x0000~0x13C	Reserved	-
0x0140	USBOTG_USBCMD	USB command
0x0144	USBOTG_USBSTS	USB status
0x0148	USBOTG_USBINTR	USB interrupt enable
0x014C	USBOTG_FRINDEX	USB frame index
0x0150	Reserved	-
0x0154	USBOTG_PERIODICLISTBASE_DEVICEADDR	Frame list base address (host) USB device address (device)
0x0158	USBOTG_ASYNCLISTADDR_EPLISTADDR	Next asynchronous list address (host) address at endpoint list in memory
0x015C	USBOTG_TTCTRL	TT status and control
0x0160	USBOTG_BURSTSIZE	Programmable burst size
0x0164	USBOTG_TXFILLTUNING	Host transmit pre-buffer packet tuning
0x0168~0x016C	Reserved	-
0x0170	USBOTG_ULPIVIEWPORT	Reserved
0x0174	Reserved	-
0x0178	USBOTG_ENDPTNAK	Endpoint NAK
0x017C	USBOTG_ENDPTNAKEN	Endpoint NAK enable
0x0180	Reserved	-
0x0184	USBOTG_PORTSC1	Port status control 1
0x0188~0x1A0	Reserved	-
0x01A4	USBOTG_OTGSC	OTG status and control

RISC Address<15:0>	Register	Description
0x01A8	USBOTG_USBMODE	USB device mode
0x01AC	USBOTG_ENDPTSETUPSTAT	Endpoint setup status
0x01B0	USBOTG_ENDPTPRIME	Endpoint initialization
0x01B4	USBOTG_ENDPTFLUSH	Endpoint de-initialization
0x01B8	USBOTG_ENDPTSTATUS	Endpoint status
0x01BC	USBOTG_ENDPTCOMPLETE	Endpoint complete
0x01C0~0x01CC	USBOTG_ENDPTCTRL0~3	Endpoint control0~3
0x0200	USBOTG_PHY_PROGRAM	PHY program register

Table 363Address Mapping

Register Definitions

- USB-OTG USBCOMD Register (USBOTG_USBCMD) – 0x0140

Bit	Name	Default	Description
0 (R/W)	RS	1'b0	Run/Stop (RS) Default 0b. 1=Run. 0=Stop. Host Controller: When set to a 1, the Host Controller proceeds with the execution of the schedule. The Host Controller continues execution as long as this bit is set to a one. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Host Controller has finished the transaction and has entered the stopped state. Software should not write a one to this field unless the host controller is in the Halted state (i.e. HCHalted in the USBSTS register is a one). Device Controller: Writing a one to this bit will cause the device controller to enable a pull-up on D+ and initiate an attach event. This control bit is not directly connected to the pull-up enable, as the pull-up will become disabled upon transitioning into high-speed mode. Software should use this bit to prevent an attach event before the device controller has been properly initialized. Writing a 0 to this will cause a detach event.
1 (R/W)	RST	1'b0	RST Controller Reset (RESET) Software uses this bit to reset the controller. This bit is set to 0 by the Host/Device Controller when the reset process is complete. Software cannot terminate the reset process early by writing a 0 to this register. Host Controller: When software writes a one to this bit, the Host Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. Software should not set this bit to a one when the HCHalted bit in the USBSTS

Bit	Name	Default	Description
			register is a 0. Attempting to reset an actively running host controller will result in undefined behavior. Device Controller: When software writes a one to this bit, the Device Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Writing a one to this bit when the device is in the attached state is not recommended, since the effect on an attached host is undefined. In order to ensure that the device is not in an attached state before initiating a device controller reset, all primed endpoints should be flushed and the USBCMD Run/Stop bit should be set to 0.
2 (R/W)	FS0	1'b0	Frame List Size Default 000b. This field is Read/Write only if Programmable Frame List Flag in the HCCPARAMS registers is set to one. This field specifies the size of the frame list that controls which bits in the Frame Index Register should be used for the Frame List Current index. Note that this field is made up from USBCMD bits 15, 3, and 2. Value description: 000: 1024 elements (4096 bytes) Default value 001: 512 elements (2048 bytes) 010: 256 elements (1024 bytes) 011: 128 elements (512 bytes) 100: 64 elements (256 bytes) 101: 32 elements (128 bytes) 110: 16 elements (64 bytes) 111: 8 elements (32 bytes) Only the host controller uses this field.
3 (R/W)	FS1	1'b0	See bit 2
4 (R/W)	PSE	1'b0	Periodic Schedule Enable Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule. Values meaning 0: Do not process the Periodic Schedule 1: Use the PERIODICLISTBASE register to access the Periodic Schedule. Only the host controller uses this bit.
5 (R/W)	ASE	1'b0	Asynchronous Schedule Enable Read/Write. Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule. Value description: 0: Do not process the Asynchronous Schedule. 1: Use the ASYNCLISTADDR register to access the Asynchronous Schedule. Only the host controller uses this bit.
6 (R/W)	IAA	1'b0	Interrupt on Async Advance Doorbell Read/Write. This bit is used as a

Bit	Name	Default	Description
			doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule states, it sets the Interrupt on Async Advance status bit in the USBSTS register. If the Interrupt on Sync Advance Enable bit in the USBINTR register is one, then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to zero after it has set the Interrupt on Sync Advance status bit in the USBSTS register to one. Software should not write a one to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results. This bit is only used in host mode. Writing a one to this bit when device mode is selected will have undefined results.
7 (R)	LR	1'b0	Host/Device Controller Reset (OPTIONAL). Read only. Not Implemented. This field will always be 0.
8 (R/W)	ASP0		Asynchronous Schedule Park Mode Count (OPTIONAL) Read/Write. If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this field defaults to 3h and is R/W. Otherwise it defaults to zero and is RO. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a zero to this bit when Park Mode Enable is a one as this will result in undefined behavior. This field is set to 3h in this implementation.
9	ASP1		See bit 8
10			Reserved
11 (R/W)	ASPE		Asynchronous Schedule Park Mode Enable (optional) Read/Write. If the Asynchronous Park Capability bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled. This field is set to 1 in this implementation.
12 (R/W)	ATDTW	1'b0	Add dTD TripWire. Read/write. Device mode only. This bit is used as a semaphore to ensure the proper addition of a new dTD to an active (primed) endpoint's linked list. This bit is set and cleared by software. This bit shall also be cleared by hardware when is state machine is hazard region for which adding a dTD to a primed endpoint may go unrecognized.
13 (R/W)	SUTW	1'b0	Setup TripWire. Read/write. Device mode only. This bit is used as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a QH by the DCD without being corrupted. If the setup lockout mode is off (See

Bit	Name	Default	Description
			USBMODE) then there exists a hazard when new setup data arrives while the DCD is copying the setup data payload from the QH for a previous setup packet. This bit is set and cleared by software and will be cleared by hardware when a hazard exists.
14			Reserved
15	FS2	1'b0	See bit 2
23:16 (R/W)	ITC[7:0]	8'h8	Interrupt Threshold Control. Read/write. The system software uses this field to set the maximum rate at which the host/device controller will issue interrupts. ITC contains the maximum interrupt interval measured in micro-frames. Value: Maximum Interrupt Interval 00h: Immediate (no threshold) 01h: 1 micro-frame 02h: 2 micro-frames 04h: 4 micro-frames 08h: 8 micro-frames 10h: 16 micro-frames 20h: 32 micro-frames 40h: 64 micro-frames
31:24	-	8'h0	Reserved

Table 364: USB-OTG USBCOMD Register

- USB-OTG USBSTS Register (USBOTG_USBSTS) – 0x0144

Bit	Name	Default	Description
0 (R/W)	UI	1'b0	Interrupt (USBINT). R/WC. This bit is set by the Host/Device Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set. This bit is also set by the Host/Device Controller when a short packet is detected. A short packet is when the actual number of bytes received was less than the expected number of bytes.
1 (R/W)	UEI	1'b0	Error Interrupt (USBERRINT). R/WC. When completion of a USB transaction results in an error condition, this bit is set by the Host/Device Controller. This bit is set along with the USBINT bit, if the TD on which the error interrupt occurred also had its interrupt on complete (IOC) bit set The device controller detects resume signaling only.
2 (R/W)	PCI	1'b0	Port Change Detect. R/WC. The Host Controller sets this bit to a one when on any port a Connect Status occurs, a Port Enable/Disable Change occurs, or the Force Port Resume bit is set as the result of a J-K

Bit	Name	Default	Description
			transition on the suspended port. The Device Controller sets this bit to a one when the port controller enters the full or high-speed operational state. When the port controller exits the full or high-speed operation states due to Reset or Suspend events, the notification mechanisms are the USB Reset Received bit and the DCSuspend bits respectively. This bit is not EHCI compatible.
3 (R/W)	FRI	1'b0	Frame List Rollover. R/WC. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example, if the frame list size (as programmed in the Frame List Size field of the USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX [1 3] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FHINDEX [12] toggles. Only used by the host controller.
4 (R/W)	SEI	1'b0	System Error R/WC. This bit is not used in this implementation and will always be set to 0.
5 (R/W)	AAI	1'b0	Interrupt on Async Advance. R/WC. Default: 0. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the USBCMD register. This status bit indicates the assertion of that interrupt source. Only used by the host controller.
6 (R/W)	URI	1'b0	USB Reset Received. R/WC. Default: 0 When the device controller detects a USB Reset and enters the default state, this bit will be set to a one. Software can write a 1 to this bit to clear the USB Reset Received status bit. Only used by the device controller.
7 (R/W)	SRI	1'b0	SOF Received. R/WC. Default: 0 When the device controller detects a Start Of (micro) Frame, this bit will be set to a one. When a SOF is extremely late, the device controller will automatically set this bit to indicate that an SOF was expected. Therefore, this bit will be set roughly every 1ms in device FS mode and every 125ms in HS mode and will be synchronized to the actual SOF that is received. Since the device controller is initialized to FS before connect, this bit will be set at an interval of 1ms during the prelude to connect and chirp. In host mode, this bit will be set every 125us and can be used by host controller driver as a time base. Software writes a 1 to this bit to clear it. This is a non-EHCI status bit.
8 (R/W)	SLI	1'b0	DCSuspend. R/WC. Default: 0 When a device controller enters a suspend state from an active state,

Bit	Name	Default	Description
			this bit will be set to a one. The device controller clears the bit upon exiting from a suspend state. Only used by the device controller.
9			Reserved
10 (R/W)	ULPII	1'b0	ULPI Interrupt. R/WC. Default:0 When the ULPI Viewport is present in the design, an event completion will set this interrupt. Used by both host & device controller. Only present in designs where configuration constant VUSB_HS_PHY_ULPI = 1. **NOTE**: AtlasV has no ULPI, so please ignore it.
11			Reserved
12 (R)	HCH	1'b1	HCHalted. Read only. Default: 1 This bit is a zero whenever the Run/Stop bit is a one. The Host Controller sets this bit to one after it has stopped executing because of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (e.g. internal error). Only used by the host controller.
13 (R)	RCL	1'b0	Reclamation. Read only. Default: 0 This is a read-only status bit used to detect an empty asynchronous schedule. Only used by the host controller.
14 (R)	PS	1'b0	Periodic Schedule Status. Read only. Default: 0 This bit reports the current real status of the Periodic Schedule. When set to zero the periodic schedule is disabled, and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). Only used by the host controller.
15 (R)	AS	1'b0	Asynchronous Schedule Status. Read only. Default: 0 This bit reports the current real status of the Asynchronous Schedule. When set to zero the asynchronous schedule status is disabled and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). Only used by the host controller.
16 (R)	NAKI	1'b0	NAK Interrupt Bit. Read only. This bit is read-only. It is set by hardware when for a particular endpoint both the TX/RX Endpoint NAK bit and the corresponding TX/RX

Bit	Name	Default	Description
			Endpoint NAK Enable bit are set. This bit is automatically cleared by hardware when all the enabled TX/RX Endpoint NAK bits are cleared.
17			Reserved
18 (R/W)	UAI	1'b0	USB Host Asynchronous Interrupt (USBHSTASYNCINT). R/WC. This bit is set by the Host Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set AND the TD was from the asynchronous schedule. This bit is also set by the Host when a short packet is detected AND the packet is on the asynchronous schedule. A short packet is when the actual number of bytes received is less than expected. This bit is not used by the device controller and is always zero.
19 (R/W)	UPI		USB Host Periodic Interrupt (USBHSTPERINT). R/WC. This bit is set by the Host Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set and the TD was from the periodic schedule. This bit is also set by the Host Controller when a short packet is detected AND the packet is on the periodic schedule. A short packet is when the actual number of bytes received was less than the expected number of bytes. This bit is not used by the device controller and should always be zero.
23:20			Reserved
24 (R/W)	TI0	1'b0	General Purpose Timer Interrupt 0 (GPTINT0). R/WC. This bit is set when the counter in the GPTIMER0CTRL (Non-EHCI) register transitions to zero. Writing a one to this bit will clear it.
25 (R/W)	TI1	1'b0	General Purpose Timer Interrupt 1 (GPTINT1). R/WC. This bit is set when the counter in the GPTIMER1CTRL (Non-EHCI) register transitions to zero. Writing a 1 to this bit will clear it.
31:26	-	29'h0	Reserved.

Table 365: USB-OTG USBSTS Register

- USB-OTG USBINTR Register (USBOTG_USBINTR) – 0x0148

Bit	Name	Default	Description
0 (R/W)	UE	1'b0	USB Interrupt Enable When this bit is a one, and the USBINT bit in the USBSTS register is a one, the host/device controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software by clearing the USBINT bit.
1 (R/W)	UEE	1'b0	USB Error Interrupt Enable

Bit	Name	Default	Description
			When this bit is a one, and the USBERRINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software by clearing the USBERRINT bit in the USBSTS register.
2 (R/W)	PCE	1'b0	Port Change Detect Enable When this bit is 1, and the Port Change Detect bit in the USBSTS register is 1, the host/device controller will issue an interrupt. The interrupt is acknowledged by software by clearing the Port Change Detect bit.
3 (R/W)	FRE	1'b0	Frame List Rollover Enable When this bit is 1, and the Frame List Rollover bit in the USBSTS register is 1, the host controller will issue an interrupt. The interrupt is acknowledged by software by clearing the Frame List Rollover bit. Only used by the host controller.
4 (R/W)	SEE	1'b0	System Error Enable When this bit is a one, and the System Error bit in the USBSTS register is a one, the host/device controller will issue an interrupt. The interrupt is acknowledged by software clearing the System Error bit.
5 (R/W)	AAE	1'b0	Interrupt on Async. Advance Enable When this bit is a one, and the Interrupt on Async Advance bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the Interrupt on Async Advance bit. Only used by the host controller.
6 (R/W)	URE	1'b0	USB Reset Enable When this bit is a one, and the USB Reset Received bit in the USBSTS register is a one, the device controller will issue an interrupt. The interrupt is acknowledged by software clearing the USB Reset Received bit. Only used by the device controller.
7 (R/W)	SRE	1'b0	SOF Received Enable When this bit is a one, and the SOF Received bit in the USBSTS register is a one, the device controller will issue an interrupt. The interrupt is acknowledged by software clearing the SOF Received bit.
8 (R/W)	SLE	1'b0	Sleep Enable When this bit is a one, and the DCSuspend bit in the USBSTS register transitions, the device controller will issue an interrupt. The interrupt is acknowledged by software writing a one to the DCSuspend bit. Only used by the device controller.
9	-	1'b0	Reserved.
10 (R/W)	ULPIE	1'b0	ULPI Enable When this bit is a one, and the ULPI Interrupt bit in the USBSTS register transitions, the controller will issue and interrupt. The interrupt is acknowledged by software writing a one to the ULPI Interrupt bit. Used by both host and device controller. Only present in designs where

Bit	Name	Default	Description
			configuration constant VUSB_HS_PHY_ULPI = 1. **NOTE**: SiRFatlasV has no ULPI, ignore it.
15:11		5'h0	Reserved
16 (R/W)	NAKE	1'b0	NAK Interrupt Enable This bit is set by software if it wants to enable the hardware interrupt for the NAK Interrupt bit. If both this bit and the corresponding NAK Interrupt bit are set, a hardware interrupt is generated.
17	-	1'b0	Reserved
18 (R/W)	UAIE	1'b0	USB Host Async. Interrupt Enable When this bit is a one, and the USBHSTASYNCINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBHSTASYNCINT bit.
19 (R/W)	UPIE	1'b0	USB Host Periodic Interrupt Enable When this bit is a one, and the USBHSTPERINT bit in the USBSTS register is a one, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the USBHSTPERINT bit.
23:20	-	4'h0	Reserved
24 (R/W)	TIE0	1'b0	General Purpose Timer interrupt Enable 0 When this bit is a one, and the GPTINT0 bit in the USBSTS register is a one, the controller will issue an interrupt. The interrupt is acknowledged by software clearing the GPTINT0 bit.
25 (R/W)	TIE1	1'b0	General Purpose Timer interrupt Enable 1 When this bit is a one, and the GPTINT1 bit in the USBSTS register is a one, the controller will issue an interrupt. The interrupt is acknowledged by software clearing the GPTINT1 bit.
31:26	-	6'h0	Reserved

Table 366: USB-OTG USBINTR Register

- USB-OTG FRINDX Register (USBOTG_FRINDX) – 0x014C

Bit	Name	Default	Description
13:0 (R/W)	FRINDEX	14'h0	Frame index The value, in this register, increments at the end of each time frame (e.g. micro-frame). Bits [N: 3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the

Bit	Name	Default	Description
			value of the Frame List Size field in the USBCMD register, when used in host mode. USBCMD [Frame List Size] Number Elements N 000b (1024) 12 001b (512) 11 010b (256) 10 011b (128) 9 100b (64) 8 101b (32) 7 110b (16) 6 111b (8) 5 In device mode the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode bits 2:0 indicate the current microframe.
31:14	-	18'h0	Reserved

Table 367: USB-OTG FRINDX Register

- USB-OTG PERIODICLISTERBASE / DEVICEADDR Register (USBOTG_PD) **– 0x0154**
  - Host controller

Bit	Name	Default	Description
11:0	-	24'h0	Reserved
31:12 (R/W)	PERBASE	8'h0	Base address (low). These bits correspond to memory address signals [31:12], respectively. Only used by the host controller.

Table 368: USB-OTG PERIODICLISTERBASE Register

  - Device controller

Bit	Name	Default	Description
23:0 (R)	-	24'h0	Reserved
24 (R/W)	USBADRA	1'b0	Device Address Advance. Default: 0. When this bit is 0, any writes to USBADR are instantaneous. When this bit is written to 1 at the same time or before USBADR is written, the write to the USBADR field is staged and held in a hidden register. After an IN occurs on endpoint 0 and is ACKed, USBADR will be loaded from the holding register.

Bit	Name	Default	Description
			Hardware will automatically clear this bit on the following conditions: • IN is ACKed to endpoint 0. (USBADR is updated from staging register). • OUT/SETUP occurs to endpoint 0. (USBADR is not updated). • Device Reset occurs (USBADR is reset to 0). **NOTE** - After the status phase of the SET_ADDRESS descriptor, the DCD has 2ms to program the USBADR field. This mechanism will ensure this specification is met when the DCD can not write of the device address within 2ms from the SET_ADDRESS status phase. If the DCD writes USBADR with USBADRA=1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR will be programmed instantly at the correct time and meet the 2ms USB requirement.
31:25	DEVICE ADDR	8'h0	Device Address. These bits correspond to the USB device address

Table 369: USB-OTG DEVICEADDR Register

- USB-OTG ASYNCLISTADDR / ENDPOINTLISTADDR Register (USBOTG_PD) – 0x0158
  - Host controller

Bit	Name	Default	Description
4:0	-	-	Reserved
31:5 (R/W)	ASYBASE	8'h0	Link pointer low (LPL). These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). Only used by the host controller.

Table 370: USB-OTG ASYNCLISTADDR Register

  - Device controller

Bit	Name	Default	Description
10:0	-		Reserved
31:11 (R/W)	EPBASE	8'h0	Endpoint List Pointer (Low). These bits correspond to memory address signals [31:11], respectively. This field will reference a list of up to 32 Queue Heads (QH). (i.e. one queue head per endpoint and direction.)

Table 371: USB-OTG ENDPOINTLISTADDR Register

- USB-OTG TTCTRL Register (USBOTG_TTCTRL) – 0x015C

Bit	Name	Default	Description
23:0 (R)	-	24'h0	Reserved
30:24	TTHA	7'h0	-
31	-	1'b0	Reserved

Table 372: USB-OTG TTCTRL Register

- USB-OTG BURSTSIZE Register (USBOTG_BUSTSIZE) – 0x0160

Bit	Name	Default	Description
7:0 (R/W)	RXPBURST	8'h4	Programmable TX burst length
15:8 (R/W)	TXPBURST	8'h4	Programmable RX burst length
31:16	-	16'h0	Reserved

Table 373: USB-OTG BURSTSIZE Register

- USB-OTG TXFILLTUNING Register (USBOTG_TXFILLTUNING) – 0x0164

Bit	Name	Default	Description
7:0 (R/W)	TXSCHOH	8'h0	Scheduler Overhead. Read/write. Default: 0 This register adds an additional fixed offset to the schedule time estimator described above as Tff. As an approximation, the value chosen for this register should limit the number of back-off events captured in the TXSCHHEALTH to less than 10 per second in a highly utilized bus. Choosing a value that is too high for this register is not desired as it can needlessly reduce USB utilization. The time unit represented in this register is 1.267us when a device is connected in High-Speed Mode for OTG & SPH. The time unit represented in this register is 6.333us when a device is connected in Low/Full Speed Mode for OTG & SPH.
12:8 (R/W)	TXSCHEALTH	5'h0	Scheduler Health Counter. Read/write to clear. Default: 0 This register increments when the host controller fails to fill the TX latency FIFO to the level programmed by TXFIFOTHRES before running out of time to send the packet before the next Start-Of-Frame . This health counter measures the number of times this occurs to provide feedback to selecting a proper TXSCHOH. Writing to this register will clear the counter. The maximum of this

Bit	Name	Default	Description
			counter is 31.
15:13	-	3'h0	Reserved
21:16 (R/W)	TXFIFOTHRES	6'h0	FIFO Burst Threshold. Read/write. Default: 0 This register controls the number of data bursts that are posted to the TX latency FIFO in host mode before the packet begins on to the bus. The minimum value is 2 and this value should be a low as possible to maximize USB performance. A higher value can be used in systems with unpredicable latency and/or insufficient bandwidth where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from system memory. This value is ignored if the Stream Disable bit in USBMODE register is set.
31:22	-	-	Reserved

Table 374: USB-OTG TXFILLTUNING Register

- USB-OTG ULPIVIEWPORT Register (USBOTG_ULPIVIEWPORT) – 0x0170
  SiRFatlasV has no ULPI interface, so this register is reserved.

Bit	Name	Default	Description
7:0 (R/W)	ULPIDATWR	8'h0	ULPI Data Write. Read/write. When a write operation is commanded, the data to be sent is written to this field.
15:8 (R)	ULPIDATRD	8'h0	ULPI Data Read. Read only. After a read operation completes, the result is placed in this field.
23:16 (R/W)	ULPIADDR	8'h0	ULPI Data Address. Read/write. When a read or write operation is commanded, the address of the operation is written to this field.
26:24 (R/W)	ULPIPORT	3'b000	ULPI Port Number. Read/write. For the wakeup or read/write operation to be executed, this value selects the port number the ULPI PHY is attached to in the multi-port host. Value range: 0 to 7 It should always be 0 for non-multi port products.
27 (R)	ULPISS	1'b1	ULPI Sync State. Read only. 1: Normal Sync. State. 0: In another state such as carkit, serial, and low power This bit represents the state of the ULPI interface. Before reading this bit, the ULPIPORT field should be set accordingly if used with the multi-port host. Otherwise, this field should always remain 0.

Bit	Name	Default	Description
28	-	-	Reserved.
29 (R/W)	ULPIRW	1'b0	ULPI Read/Write Control. Read/write. 0: Read 1: Write This bit selects a read or writes operation.
30 (R/W)	ULPIRUN	1'b0	ULPI Read/Write Run. Read/write. Writing 1 to this bit will begin the read/write operation. The bit will change to 0 when the read/write is complete. Once this bit is set, the driver can not set it to 0. The driver must never execute a wakeup and a read/write operation at the same time.
31 (R/W)	ULPIWU	1'b0	ULPI Wakeup. Read/write. Writing 1 to this bit will begin the wakeup operation. The bit will change to 0 when the wakeup is complete. Once this bit is set, the driver can not set it back to 0. The driver must never execute a wakeup and a read/write operation at the same time.

Table 375: USB-OTG ULPIVIEWPORT Register

- USB-OTG ENDPTNAK Register (USBOTG_ ENDPTNAK) – 0x0178

Bit	Name	Default	Description
15:0 (R/WC)	EPRN	16'h0	RX Endpoint NAK ¨C R/WC. Each RX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received OUT or PING token for the corresponding endpoint. Bit 15 Endpoint #15 Bit 1 Endpoint #1 Bit 0 Endpoint #0
31:16 (R/WC)	EPTN	16'h0	TX Endpoint NAK. Each TX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received IN token for the corresponding endpoint. Bit 15 Endpoint #15 Bit 1 Endpoint #1 Bit 0 Endpoint #0

Table 376: USB-OTG ENDPTNAK Register

- USB-OTG ENDPTNAKEN Register (USBOTG_ ENDPTNAKEN) – 0x017C

Bit	Name	Default	Description
15:0 (R/W)	EPRNE	16'h0	RX Endpoint NAK Enable. Each bit is an enable bit for the corresponding RX Endpoint NAK bit. If this bit is set and the corresponding RX Endpoint NAK bit is set, the NAK interrupt bit is set. Bit 15 Endpoint #15 Bit 1 Endpoint #1 Bit 0 Endpoint #0
31:16 (R/W)	EPTNE	16'h0	TX Endpoint NAK Enable. Each bit is an enable bit for the corresponding TX Endpoint NAK bit. If this bit is set and the corresponding TX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit 15 Endpoint #15 Bit 1 Endpoint #1 Bit 0 Endpoint #0

Table 377: USB-OTG ENDPTNAKEN Register

- USB-OTG PORTSC1 Register (USBOTG_ PORTSC1) – 0x0184

Bit	Name	Default	Description
0 (R)	CCS	1'b0	Current Connect Status Read Only. In Host Mode: 1: Device is present on port. 0: No device is present. Default: 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set. This field is zero if Port Power (PP) is zero in host mode. In Device Mode: 1: Attached. 0: Not Attached. Default. 1 indicates that the device successfully attached and is operating in either high speed or full speed as indicated by the High Speed Port bit in this register. 0 indicates that the device did not attach successfully or was forcibly disconnected by software by writing 0 to the Run bit in the USBCMD register. It does not state the device being disconnected or suspended.
1 (R/WC)	CSC	1'b0	Connect Status Change. R/WC.

Bit	Name	Default	Description
			1: Change in Current Connect Status. 0: No change. Default: 0. In Host Mode: Indicates a change has occurred in the port's Current Connect Status. The host/device controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be setting a ready-set bit (the bit remains set). Software clears this bit by writing a one to it. This field is 0 if Port Power (PP) is 0 in host mode. In Device Mode: This bit is undefined in device controller mode.
2 (R/W)	PE	1'b0	Port Enabled/Disabled. Read/write. 1: Enable. 0: Disable. Default: 0. In Host Mode: Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the host software. **NOTE** - The bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, (0b) downstream propagation of data is blocked except for reset. This field is 0 if Port Power (PP) is 0 in host mode. In Device Mode: The device port is always enabled. (This bit will be 1.)
3 (R/WC)	PEC	1'b0	Port Enable/Disable Change R/WC. 1=Port enabled/disabled status has changed. 0=No change. Default = 0. In Host Mode: For the root hub, this bit gets set to a one only when a port is disabled due to disconnect on the port or due to the appropriate conditions existing at the EOF2 point. Software clears this by writing 1 to it. This field is 0 if Port Power (PP) is 0. In Device mode: The device port is always enabled. (This bit will be 0.)
4 (R)	OCA	1'b0	Over-current Active. Read only. Default 0. 1: This port currently has an over-current condition. 0: This port does not have an over-current condition. This bit will

Bit	Name	Default	Description
			automatically switch from 1 to 0 when the over current condition is removed. For host/OTG implementations the user can provide over-current detection to the vbus_pwr_fault input for this condition. For device-only implementations this bit shall always be 0.
5 (R/WC)	OCC	1'b0	Over-current Change. R/WC. Default=0. 1: This bit is set to one when there is a change to Over-current Active. Software clears this bit by writing a one to this bit position. For host/OTG implementations the user can provide over-current detection to the vbus_pwr_fault input for this condition. For device-only implementations this bit shall always be 0.
6 (R/W)	FPR	1'b0	Force Port Resume. Read/write. 1: Resume detected/driven on port. 0: No resume (K-state) detected/driven on port. Default. In host mode: Software sets this bit to one to drive resume signaling. The Host Controller sets this bit to one if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the Port Change Detect bit in the USBSTS register is also set to 1. This bit will automatically change to 0 after the resume sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a 0 after the rsume duration is timed in the driver. **NOTE -** When the Host controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a 1. This bit will remain a one until the port has switched to the high-speed idle. Writing a 0 has no affect because the port controller will time the resume operation clear the bit the port control state switches to HS or FS idle. This field is 0 if Port Power (PP) is 0 in host mode. This bit is not-EHCI compatible. In device mode: After the device has been in Suspend State for 5ms or more, software must set this bit to one to drive resume signaling before clearing. The Device Controller will set this bit to one if a J-to-K transition is detected while the port is in the Suspend state. The bit will be cleared when the device returns to normal operation. Also, when this bit transitions to a one because a J-to-K transition detected, the Port Change Detect bit in the USBSTS register is also set to one.
7 (R/W)	SUSP	1'b0	Suspend In host mode: Read/write. 1: Port in suspend state. 0: Port not in suspend state.

Bit	Name	Default	Description
			Default: 0. Port Enabled Bit and Suspend bit of this register define the port states as follows: Bits [Port Enabled, Suspend] Port State 0x: Disable 10: Enable 11: Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. The host controller will unconditionally set this bit to 0 when software sets the Force Port Resume bit to 0. The host controller ignores a write of 0 to this bit. If host software sets this bit to a 1 when the port is not enabled (i.e. Port enabled bit is a 0) the results are undefined. This field is 0 if Port Power (PP) is 0 in host mode. In Device Mode: Read only. 1: Port in suspended state. 0: Port not in suspended state. Default. In device mode this bit is a read only status bit.
8 (R/W)	PR	1'b0	Port Reset This field is 0 if Port Power (PP) is 0. In Host Mode: Read/write. 1: Port is in Reset. 0: Port is not in Reset. Default. When software writes a 1 to this bit the bus-reset sequence as defined in the USB Specification Revision 2.0 is started. This bit will automatically change to 0 after the reset sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a 0 after the reset duration is timed in the driver. In Device Mode: read only. Device reset from the USB bus is also indicated in the USBSTS register.
9 (R)	HSP	1'b0	High-Speed Port. Read only. Default: 0b. When the bit is one, the host/device connected to the port is in high-speed mode and if set to 0, the host/device connected to the port is not in a high-speed mode. **NOTE** - HSP is redundant with PSPD (27:26) but will remain in the design for compatibility.

Bit	Name	Default	Description
			This bit is not defined in the EHCI specification.
11:10 (R)	LS	2'b00	Line Status Read Only. These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. The encoding of the bits are: Bits [11:10] Meaning 00b SE0 10b J-state 01b K-state 11b Undefined In host mode, the use of linestate by the host controller driver is not necessary (unlike EHCI), because the port controller state machine and the port routing manage the connection of LS and FS. In device mode, the use of linestate by the device controller driver is not necessary.
12 (R/W)	PP	1'b0	Port Power (PP). Read/write or read only. The function of this bit depends on the value of the Port Power Switching (PPC) field in the HCSPARAMS register. The behavior is as follows: PPC: PP Operation 0b: 0b read only. A device controller with no OTG capability does not have port power control switches. 1b: 1b/0b read/write. Host/OTG controller requires port power control switches. This bit represents the current setting of the switch (0=off, 1=on). When power is not available on a port (PP = 0), the port is non-functional and will not report attaches, detaches, etc. When an over-current condition is detected on a powered port and PPC is a 1, the PP bit in each affected port may be transitioned by the host controller driver from a one to a 0 (removing power from the port). This feature is implemented in the host/OTG controller (PPC = 1). In a device only implementation port power control is not necessary, thus PPC and PP = 0.
13 (R)	PO	1'b0	Port Owner Read Only. Port owner handoff is not implemented in this design, therefore this bit will always read back as 0. The EHCI definition is include here for reference: Default: 0 This bit unconditionally goes to a 0 when the configured bit in the CONFIGFLAG register makes a 0 to 1 transition. This bit unconditionally goes to 1 whenever the Configured bit is zero System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that an internal companion controller owns and controls the port.
15:14 (R/W)	PIC	2'b00	Port Indicator Control. Read/write. Default: 00b. Writing to this field has no effect if the P_INDICATOR bit in the HCSPARAMS register is 0. If P_INDICATOR bit is 1, then the bit is:

Bit	Name	Default	Description
			00b: Port indicators are off 01b: Amber 10b: Green 11b: Undefined Refer to the USB Specification Revision 2.0 [3] for a description on how these bits are to be used. This field is output from the controller as signals port_ind_ctl_1 & port_ind_ctl_0 for use by an external led driving circuit.
19:16 (R/W)	PTC	4'b0000	Port Test Control Read/Write. Default = 0000b. A non-zero value indicates that the port is operating in test mode. 0000b: TEST_MODE_DISABLE 0001b: J_ STATE 0010b: K_STATE 0011b: SE0 (host) / NAK (device) 0100b: Packet 0101b: FORCE_ENABLE_HS 0110b: FORCE_ENABLE_FS 0111b: FORCE_ENABLE_LS 1000b to 1111b: Reserved The FORCE_ENABLE_FS and FORCE ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the PTC field to any of the FORCE_ENABLE_{HS/FS/LS} values will force the port into the connected and enabled state at the selected speed. Writing the PTC field back to TEST_MODE_DISABLE will allow the port state machines to progress normally from that point. **NOTE** - Low speed operations are not supported as a peripheral device.
20(R/W)	WKCN	1'b0	Wake on Connect Enable (WKCNNT_E). Read/write. Default: 0b. Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. This field is 0 if Port Power (PP) is 0 or in device mode. This bit is output from the controller as signal pwrctl_wake_dscnnt_en (OTG/host core only) for use by an external power control circuit.
21 (R/W)	WKDC	1'b0	Wake on Disconnect Enable (WKDSCNNT_E) ¡ª read/write. Default: 0b Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is 0 if Port Power (PP) is 0 or in device mode. This bit is output from the controller as signal pwrctl_wake_dscnnt_en (OTG/host core only) for use by an external power control circuit.
22 (R/W)	WKOC	1'b0	Wake on Over-current Enable (WKOC_E) Read/Write. Default: 0b

Bit	Name	Default	Description
			Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events. This field is 0 if Port Power (PP) is 0. This bit is output from the controller as signal pwrctl_wake_ovrcurr_en (OTG/host core only) for use by an external power control circuit.
23 (R/W)	PHCD	1'b0	PHY Low Power Suspend - Clock Disable (PLPSCD) Read/Write. Default: 0b Writing this bit to a 1b will disable the PHY clock. Writing a 0b enables it. Reading this bit will indicate the status of the PHY clock. NOTE: The PHY clock cannot be disabled if it is being used as the system clock. In device mode, The PHY can be put into Low Power Suspend - Clock Disable when the device is not running (USBCMD Run/Stop=0b) or the host has signaled suspend (PORTSC SUSPEND=1b). Low power suspend will be cleared automatically when the host has signaled resume if using a circuit similar to that in 10. Before forcing a resume from the device, the device controller driver must clear this bit. In host mode, the PHY can be put into Low Power Suspend - Clock Disable when the downstream device has been put into suspend mode or when no downstream device is connected. Low power suspend is completely under the control of software. This bit is not defined in the EHCI specification.
24 (R/W)	PFSC	1'b0	Port Force Full Speed Connect Read/Write. Default = 0b. Writing this bit to a 1b will force the port to only connect at Full Speed. It disables the chirp sequence that allows the port to identify itself as High Speed. This is useful for testing FS configurations with a HS host, hub or device. This bit is not defined in the EHCI specification. This bit is for debugging purposes.
25			Reserved
27:26 (R)	PSPD	2'b11	Port Speed Read Only. This register field indicates the speed at which the port is operating. For HS mode operation in the host controller and HS/FS operation in the device controller the port routing steers data to the Protocol engine. For FS and LS mode operation in the host controller, the port routing steers data to the Protocol Engine w/ Embedded Transaction Translator. 00 Full speed 01: Low speed 10: High speed This bit is not defined in the EHCI specification.
28 (R)	PTW	1'b0	Parallel Transceiver Width Read/Write. This register bit is used in conjunction with the configuration constant VUSB_HS_PHY16_8 to control whether the data bus width of the UTMI transceiver interface. If VUSB_HS_PHY16_8 is set for 0 or 1 then this bit is read only. If VUSB_HS_PHY16_8 is 2 or 3 then this bit is read/write. This bit is reset to 1 if VUSB_HS_PHY16_8 selects a

Bit	Name	Default	Description
			default UTMI interface width of 16-bits else it is reset to 0. Writing this bit to 0 selects the 8-bit [60MHz] UTMI interface. Writing this bit to 1 selects the 16-bit [30MHz] UTMI interface. This bit has no effect if the Serial interface is selected. This bit is not defined in the EHCI specification.
29 (R)	STS	1'b0	Serial Transceiver Select. This register bit is used in conjunction with the configuration constant VUSB_HS_PHY_SERIAL to control whether the parallel or serial transceiver interface is selected for FS and LS operation. The Serial Interface Engine can be used in combination with the UTMI+ physical interface to provide FS/LS signaling instead of the parallel interface. If VUSB_HS_PHY_SERIAL is set for 0 or 1 then this bit is read only. If VUSB_HS_PHY_SERIAL is 3 or 4 then this bit is read/write. This bit has no effect unless Parallal Transc eiver Select is s et to UTMI+. The Serial/1.1 physical interface will use the Serial Interface Engine for FS/LS signaling regardless of this bit value. **NOTE**: This bit is reserved for future operation and is a placeholder adding dynamic use of the serial engine in accord with UMTI+ characterization logic. This bit is not defined in the EHCI specification.
31:30 (R)	PTS	2'b10	Parallel Transceiver Select. This register bit pair is used in conjunction with the configuration constant VUSB_HS_PHY_TYPE to control which parallel transceiver interface is selected. This bit is not defined in the EHCI specification.

Table 378: USB-OTG PORTSC1 Register

- USB-OTG OTGSC Register (USBOTG_ OTGSC) – 0x01A4

Bit	Name	Default	Description
0 (R/W)	VD	1'b0	VBUS_Discharge. Setting this bit causes VBus to discharge through a resistor.
1(R/W)	VC	1'b0	VBUS Charge Read/Write. Setting this bit causes the VBus line to be charged. This is used for VBus pulsing during SRP.
2(R/W)	HAAR	1'b0	Hardware Assist Auto-Reset Read/Write. 0: Disabled. 1: Enable automatic reset after connect on host port.
3(R/W)	OT	1'b0	OTG Termination Read/Write. This bit must be set when the OTG device is in device mode, this controls the pull-down on DM.
4(R/W)	DP	1'b0	Data Pulsing Read/Write. Setting this bit causes the pull-up on DP to be asserted for data pulsing during SRP.
5(R/W)	IDPU	1'b1	ID Pullup Read/Write This bit controls the ID pull-up resister. 0: Off 1: On (default). When this bit is 0, the ID input will not be sampled.
6(R/W)	HADP	1'b0	HADP Hardware Assist Data-Pulse Write to Set. 1: Start Data Pulse Sequence.
7(R/W)	HABA	1'b0	HABA Hardware Assist B-Disconnect to A-connect Read/Write. 0: Disabled. 1: Enable automatic B-disconnect to A-connect sequence.
8(R)	ID	1'b1	USB ID Read Only. 0: A device 1: B device
9(R)	AVV	1'b0	A VBus Valid Read Only. Indicates VBus is above the A VBus valid threshold.
10(R)	ASV	1'b0	A Session Valid Read Only. Indicates VBus is above the A session valid threshold.
11(R)	BSV	1'b0	B Session Valid. Read only. Indicates VBus is above the B session valid threshold.
12(R)	BSE	1'b0	B Session End. Read only. Indicates VBus is below the B session end threshold.

Bit	Name	Default	Description
13(R)	1msT	1'b0	1 milisecond timer toggle. Read only. This bit toggles once per millisecond.
14(R)	DPS	1'b0	Data Bus Pulsing Status. Read only. 1 indicates data bus pulsing is being detected on the port.
15			Reserved
16(R/W)	IDIS	1'b0	USB ID Interrupt Status. Read/write. This bit is set when a change on the ID input has been detected. Software must write a one to clear this bit.
17 (R/WC)	AVVIS	1'b0	A VBus Valid Interrupt Status. Read/write to clear. This bit is set when VBus has either risen above or fallen below the VBus valid threshold (4.4 VDC) on an A device. Software must write a one to clear this bit.
18 (R/WC)	ASVIS	1'b0	A Session Valid Interrupt Status. Read/write This bit is set when VBus has either risen above or fallen below the A session valid threshold (0.8 VDC). Software must write a one to clear this bit.
19 (R/WC)	BSVIS	1'b0	B Session Valid Interrupt Status. Read/write to clear. This bit is set when VBus has either risen above or fallen below the B session valid threshold (0.8 VDC). Software must write a one to clear this bit.
20 (R/WC)	BSEIS	1'b0	B Session End Interrupt Status. Read/write to clear. This bit is set when VBus has fallen below the B session end threshold. Software must write a one to clear this bit
21 (R/WC)	1msS	1'b0	1 milisecond timer Interrupt Status. Read/write to clear. This bit is set once every millisecond. Software must write a one to clear this bit.
22 (R/W)	DPIS	1'b0	Data Pulse Interrupt Status. Read/Write to Clear. This bit is set when data bus pulsing occurs on DP or DM. Data bus pulsing is only detected when USBMODE.CM = Host (11) and PORTSC(0). PortPower = Off (0). Software must write a one to clear this bit.
23			Reserved
24 (R/W)	IDIE	1'b0	USB ID Interrupt Enable Read/Write. Setting this bit enables the USB ID interrupt.
25 (R/W)	AVVIE	1'b0	A VBus Valid Interrupt Enable Read/Write. Setting this bit enables the A VBus valid interrupt.
26 (R/W)	ASVIE	1'b0	A Session Valid Interrupt Enable Read/Write. Setting this bit enables the A session valid interrupt.

Bit	Name	Default	Description
27 (R/W)	BSVIE	1'b0	B Session Valid Interrupt Enable Read/Write. Setting this bit enables the B session valid interrupt.
28 (R/W)	BSEIE	1'b0	B Session End Interrupt Enable Read/Write. Setting this bit enables the B session end interrupt.
29 (R/W)	1msE	1'b0	1 milisecond timer Interrupt Enable Read/Write
30 (R/W)	DPIE	1'b0	Data Pulse Interrupt Enable
31			Reserved

Table 379: USB-OTG OTGSC Register

- USB-OTG USBMODE Register (USBOTG_ USBMODE) – 0x01A8

Bit	Name	Default	Description
1:0 (R/W)	CM	2'b00	Controller Mode. Controller mode is defaulted to the proper mode for host only and device only implementations. For those designs that contain both host and device capability, the controller will default to an idle state and will need to be initialized to the desired operating mode after reset. For combination host/device controllers, this register can only be written once after reset. If it is necessary to switch modes, software must reset the controller by writing to the RESET bit in the USBCMD register before reprogramming this register. 00: Idle (default for combination host/device) 01: Reserved 10: Device controller (default for device only controller) 11: Host controller (default for host only controller)
2(R/W)	ES	1'b0	Endian Select. Read/Write. This bit can change the byte ordering of the transfer buffers to match the host microprocessor bus architecture. The bit fields in the microprocessor interface and the DMA data structures (including the setup buffer within the device QH) are unaffected by the value of this bit, because they are based upon 32-bit words. 0: Little Endian (default) first byte referenced in least significant byte of 32-bit word. 1: Big Endian - first byte referenced in most significant byte of 32-bit word.
3(R/W)	SLOM	1'b0	Setup Lockout Mode. In device mode, this bit controls behavior of the setup lock mechanism. 0: Setup Lockouts On (default) 1: Setup Lockouts Off (DCD requires use of Setup Data Buffer Tripwire in USBCMD)

Bit	Name	Default	Description
4(R/W)	SDIS	1'b0	Stream disable mode. (0: Inactive [default]; 1: Active) Device mode: Setting to 1 disables double priming on both RX and TX for low bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet that the standared double buffering scheme is disabled to prevent overruns/underruns in bandwidth limited systems. **Note –** In High Speed Mode, all packets received will be responded to with a NYET handshake when stream disable is active. Host mode: Setting to 1 ensures that overruns/underruns of the latency FIFO are eliminated for low bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also has the effect of ensuring the the TX latency is filled to capacity before the packet is lanched onto the USB. Time duration to pre-fill the FIFO becomes significant when stream disable is active. See TXFILLTUNING to characterize the adjustments needed for the scheduler when using this feature. This feature may lower the overall USB performance.
5(R/W)	VBPS	1'b0	Vbus power select 0: Output is 0. 1: Output is 1. This bit is connected to the vbus_pwr_select output and can be used for any generic control but is named to be used by logic that selects between an on-chip Vbus power source (charge pump) and an off-chip source in systems when both are available.
31:6		26'h0	Reserved

Table 380: USB-OTG USBMODE Register

- USB-OTG ENDPTSETUPSTAT Register (USBOTG_ ENDPTSETUPSTAT) – 0x01AC

Bit	Name	Default	Description
15:0 (R/WC)	ENDPSETUPSTAT	16'h0	Setup endpoint status. For every received setup transaction, a corresponding bit in this register is set to 1. Software must clear or acknowledge the setup transfer by writing 1 to that bit after it has read the setup data from Queue head. The response to a setup packet as in the order of operations and total response time is crucial to limit bus time outs while the setup lock our mechanism is engaged. See Managing Endpoints in the Device Operational Model. This register is only used in device mode.
31:16		16'h0	Reserved

Table 381: USB-OTG ENDPTSETUPSTAT Register

- USB-OTG ENDPTPRIME Register (USBOTG_ ENDPTPRIME) – 0x01B0

Bit	Name	Default	Description
15:0 (R/W)	PERB	16'h0	SiRFatlasV endpoint receive buffer - R/WS. For each endpoint, a corresponding bit is used to request a buffer prepare for a receive operation for when a USB host initiates a USB OUT transaction. Software should write a one to the corresponding bit whenever posting a new transfer descriptor to an endpoint. Hardware will automatically use this bit to begin parsing for a new transfer descriptor from the queue head and prepare a receive buffer. Hardware will clear this bit when the associated endpoint is successfully primed. These bits will be momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. Bit 15 Endpoint #15 Bit 1 Endpoint #1 Bit 0 Endpoint #0
31:16 (R/W)	PETB	16'h0	SiRFatlasV endpoint transmit buffer -R/WS. For each endpoint a corresponding bit is used to request that a buffer prepared for a transmit operation in order to respond to a USB IN/INTERRUPT transaction. Software should write a one to the corresponding bit when posting a new transfer descriptor to an endpoint. Hardware will automatically use this bit to begin parsing for a new transfer descriptor from the queue head and prepare a transmit buffer. Hardware will clear this bit when the associated endpoint is successfully primed. These bits will be momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PETB[15] Endpoint #15 PETB[1] Endpoint #1 PETB[0] Endpoint #0

Table 382: USB-OTG ENDPTPRIME Register

- USB-OTG ENDPTFLUSH Register (USBOTG_ ENDPTFLUSH) – 0x01B4

Bit	Name	Default	Description
15:0 (R/B)	FERB	16'h0	Flush endpoint receive buffer. C R/WS. Writing 1 to a bit will cause the associated endpoint to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer will continue until completion. Hardware will clear this register after the endpoint flush operation is successful. Bit 15 Endpoint #15 Bit 1 Endpoint #1 Bit 0 Endpoint #0

Bit	Name	Default	Description
31:16 (R/B)	FETB	16'h0	Flush endpoint transmit buffer. R/WS. Writing 1 to a bit in this register will cause the associated endpoint to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer will continue until completion. Hardware will clear this register after the endpoint flush operation is successful. FETB[15] Endpoint #15 FETB[1] Endpoint #1 FETB[0] Endpoint #0

Table 383: USB-OTG ENDPTFLUSH Register

- USB-OTG ENDPTSTAT Register (USBOTG_ ENDPTSTAT) – 0x01B8

Bit	Name	Default	Description
15:0 (R)	ERBR	16'h0	Endpoint receive buffer ready. Read only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to 1 by the hardware as a response to receiving a command from a corresponding bit in the ENDPTPRIME register. There is always a delay between setting a bit in the ENDPTPRIME register and endpoint indicating ready. This delay time varies with the current USB traffic and the number of bits set in the ENDPTPRIME register. Buffer ready can be cleared by USB reset, USB DMA system, or the ENDPTFLUSH register. These bits will be cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ERBR[15] Endpoint #15 ERBR[1] Endpoint #1 ERBR[0] Endpoint #0
31:16 (R)	ETBR	16'h0	Endpoint transmit buffer ready. Read only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to 1 by the hardware as a response to receiving a command from a bit in the ENDPTPRIME register. There is always a delay between setting a bit in the ENDPTPRIME register and endpoint indicating ready. This delay time varies with the current USB traffic and the number of bits set in the ENDPTPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. These bits will be cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ETBR[15] Endpoint #15 ETBR[1] Endpoint #1 ETBR[0] Endpoint #0

Table 384: USB-OTG ENDPTSTAT Register

- USB-OTG ENDPTCOMPLETE Register (USBOTG_ ENDPTCOMPLETE) – 0x01BC

Bit	Name	Default	Description
15:0 (R/WC)	ERCE	16'h0	Endpoint receive complete event RW/C. Each bit indicates a received event (OUT/SETUP) occurred and software should read the endpoint queue to determine the transfer status. If the IOC bit is set in Transfer Descriptor, this bit will be set simultaneously with the USBINT. Write 1 to clear the corresponding bit in this register. ERCE[15] Endpoint #15 ERCE[1] Endpoint #1 ERCE[0] Endpoint #0
31:16 (R/WC)	ETCE	16'h0	Endpoint transmit complete event R/WC. Each bit indicates a transmit event (IN/INTERRUPT) occurred and software should read the endpoint queue to determine the endpoint status. If the IOC bit is set in Transfer Descriptor, this bit will be set simultaneously with the USBINT. Write 1 to clear the corresponding bit in this register. ETCE[15] Endpoint #15 ETCE[1] Endpoint #1 ETCE[0] Endpoint #0

Table 385: USB-OTG ENDPTCOMPLETE Register

- USB-OTG ENDPTCTRL0 Register (USBOTG_ ENDPTCTRL0) – 0x01C0

Bit	Name	Default	Description
0 (R/W)	RXS	1'b0	RXS RX endpoint stall 0: End point OK (default) 1: End point stalled Software can write a one to this bit to force the endpoint to return a stall handshake to the Host. It will continue returning stall until the bit is cleared by software or it will automatically be cleared upon receipt of a new setup request. After receving a setup request, this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. There is a short delay (up to 50 clocks) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the dcd software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: Continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
1		1'b0	Reserved
3:2(R)	RXT	2'b00	RX endpoint type 00: Control

Bit	Name	Default	Description
			Endpoint0 is fixed as a Control End Point.
6:4		1'b0	Reserved
7(R)	RXE	1'b1	RX endpoint enable 1: Enabled Endpoint0 is always enabled.
15:8		8'h0	Reserved
16(R/W)	TXS	1'b0	TX endpoint stall. Read/write 0: End point OK (default) 1: End point stalled Software can write a one to this bit to force the endpoint to return a stall handshake to the Host. It will continue returning stall until the bit is cleared by software or it will automatically be cleared upon receipt of a new setup request. After receving a setup request, this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. There is a short delay (up to 50 clocks) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the dcd software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: Continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
17		1'b0	Reserved
19:18(R)	TXT	2'b00	TX endpoint type 00: Control Endpoint0 is fixed as a Control End Point.
22:20		3'b000	Reserved
23(R)	TXE	1'b1	TX endpoint enable 1: Enabled Endpoint0 is always enabled.
31:24		8'h0	Reserved

Table 386: USB-OTG ENDPTCTRL0 Register

- USB-OTG ENDPTCTRL1~3 Register (USBOTG_ ENDPTCTRL1~3) – 0x01C4~0x1CC

Bit	Name	Default	Description
0 (R/W)	RXS	1'b0	RX endpoint stall. Read/write 0: End point OK (default) 1: End point stalled This bit will be cleared automatically upon receipt of a setup request if this endpoint is configured as a control endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a stall handshake to the host. This control will continue to stall until this bit is either cleared by software or automatically cleared as above for control endpoints. For control endpoint types, there is a short delay (up to 50 clocks) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the dcd software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: Continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
1 (R/W)	RXD	1'b0	RX endpoint data sink 0: Dual Port Memory Buffer/DMA Engine (default) Should always be written to 0.
3:2 (R/W)	RXT	2'b00	RX endpoint type 00: Control 01: Isochronous 10: Bulk 11: Reserved
4		1'b0	Reserved
5 (R/W)	RXI	1'b0	RX data toggle inhibit 0: Disabled (default) 1: Enabled This bit is only used for test and should always be written to 0. Writing 1 to this bit will cause this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
6 (R/W)	RXR	1'b0	RX data toggle reset (WS) Write 1: Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write 1 to this bit in order to synchronize the data PID's between the host and device.
7 (R/W)	RXE	1'b0	RX endpoint enable 0: Disabled (default)

Bit	Name	Default	Description
			1: Enabled An Endpoint should be enabled only after it has been configured.
15:8		8'h0	Reserved
16 (R/W)	TXS	1'b0	TX endpoint stall 0: End point OK 1: End point stalled This bit will be cleared automatically upon receipt of a setup request if this endpoint is configured as a control endpoint and this bit will continue to be cleared by hardware until the associated ENDPTSETUPSTAT bit is cleared. Software can write a one to this bit to force the endpoint to return a stall handshake to the host. This control will continue to stall until this bit is either cleared by software or automatically cleared as above for control endpoints. For control endpoint types: There is a short delay (up to 50 clocks) between the endptsetupstat being cleared and hardware continuing to clear this bit. In most systems it is unlikely the dcd software will observe this delay. However, should the dcd observe that the stall bit is not set after writing a one to it then follow this procedure: Continually write this stall bit until it is set or until a new setup has been received by checking the associated endptsetupstat bit.
17 (R/W)	TXD	1'b0	TX endpoint data source 0: dual port memory buffer/DMA engine (default) Should always be written as 0.
19:18 (R/W)	TXT	2'b00	TX endpoint type 00: Control 01: Isochronous 10: Bulk 11: Interrupt
20		1'b0	Reserved
21 (R/W)	TXI	1'b0	TX data toggle inhibit 0: PID sequencing enabled. (default) 1: PID sequencing disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit will cause this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
22 (R/W)	TXR	1'b0	TX data toggle reset (WS) Write 1: Reset PID sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
23	TXE	1'b0	TX endpoint enable

Bit	Name	Default	Description
(R/W)			0: Disabled (default) 1: Enabled An Endpoint should be enabled only after it has been configured.
31:24		8'h0	Reserved

Table 387: USB-OTG ENDPTCTRL1~3 Register

- USBOTG PHY PROGRAM Register (USBOTG_PHY_PROGRAM) – 0x0200

Bit	Name	Default	Description
31 (R/W)	comp_always_on	1'b1	Enables or disables the VBUS Valid comparator. When OTGDISABLE is set to 1'b0 and comp_always_on is asserted, the Bandgap circuitry and VBUS Valid comparator are powered, even in Suspend mode: • 1'b0: The VBUS Valid comparator is enabled • 1'b1: The VBUS Valid comparator is disabled.
30:27		4'b0	Reserved
26:25 (R/W)	refclksel	2'b01	Reference clock select for PHY PLL block • 2'b11, 2'b10: The PLL uses CLKCORE as reference. • 2'b01: The PLL uses an external clock supplied on the XO pin. • 2'b00: Reversed
24 (R/W)	commononn	1'h1	Common block power-down control Control the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 nanoPHY is suspended. • 1: The XO, Bias, and PLL blocks are powered down in Suspend mode. • 0: The XO, Bias, and PLL blocks remain powered in Suspend mode. This signal is a strapping option that must be set prior to a power-on reset and remain static during normal operation.
23 (R/W)	otgdisable	1'h0	OTG block disable Power down the OTG block, including the VBUS Valid comparator. If the application does not use OTG functionality, you can set this input high to save power. • 1: The OTG block is powered down. • 0: The OTG block is not powered down.
22 (R/W)	atebisterror	1'h1	Tx jitter adjustment. Adjust tx jitter. • 1 : TX Jitter Fix enable (design default) • 0 : TX Jitter Fix disable

Bit	Name	Default	Description
21:20 (R/W)	txhsxvtune	2'h01	Transmitter high-speed crossover adjustment. Adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode • 11: The crossover voltage is increased by 15mV • 10: The crossover voltage is increased by 30mV • 01: Default setting • 00: Reserved
19:16 (R/W)	txvreftune	4'h8	HS DC voltage level adjustment. Adjust the voltage to which the high speed DC level is tuned. • 1111: + 8.75% • 1110: + 7.5% • 1101: + 6.25% • 1100: + 5% • 1011: + 3.75% • 1010: + 2.5% • 1001: + 1.25% • 1000: Design default • 0111: – 1.25% • 0110: – 2.5% • 0101: – 3.75% • 0100: – 5% • 0011: – 6.25% • 0010: – 7.5% • 0001: – 8.75% • 0000: – 10%
15 (R/W)	txrisetune	1'h0	HS transmitter rise/fall time adjustment. Adjust the rise/fall times of the high speed waveform. • 1: – 8% • 0: Design default
14 (R/W)	txpreemphasistune	1'h0	HS transmitter pre-emphasis enable. Enable or disable the pre-emphasis for a J-K or K-J state transition in HS mode. • 1: The HS transmitter pre-emphasis is enabled. • 0 (design default): The HS Transmitter pre-emphasis is disabled.
13:10 (R/W)	txfslstune	4'h7	FS/LS pull-up resistance adjustment. Adjust the low- and full-speed pull-up resistance, based on nominal power, voltage, and temperature. • 1111: – 2.5% • 0111: Design default • 0011: + 2.5%

Bit	Name	Default	Description
			• 0001: + 5% • 0000: + 7.5%
9:7 (R/W)	sqrtune	3'h3	Squelch threshold tune. Adjust the voltage level for the threshold used to detect valid high-speed data. • 111: – 20% • 110: – 15% • 101: – 10 • 100: – 5% • 011: Design default • 010: + 5% • 001: + 10% • 000: + 15%
6:4 (R/W)	otgtune	3'h4	VBUS valid threshold adjustment. Adjust the voltage level for the VBUS valid threshold. • 111: + 9% • 110: + 6% • 101: + 3% • 100: + Design default • 011: – 3% • 010: – 6% • 001: – 9% • 000: – 12%
3:1 (R/W)	compdistune	3'h3	Disconnect threshold adjustment. Adjust the voltage level for the threshold used to detect a disconnect event at the host. • 111: + 6% • 110: + 4.5% • 101: + 3% • 100: +1.5 • 011: Design default • 010: – 1.5% • 001: – 3% • 000: – 4.5%
0 (R/W)	-	1'h0	Reserved.

Table 388: USBOTG PHY PROGRAM Register

ROM Interface

Overview

The ROM interface is one of the PCI devices on the internal PCI bus. The only purpose is to support CPU interface LCD panel through either RISC I/O access or RISC PCI I/O access. The port only has one bit address.

Feature List

- Access CPU interface LCD panel through either RISC I/O bus or RISC PCI bus
- Only 1-bit address,

Pin Descriptions

The following table shows the pins used in ROM interface and their function. The ROM/SRAM interface pins are multiplexed with other devices.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_L_PCLK	O	ROM/ LCD	LCD	Output high	Chip select pins. Active low.
X_L_DE	O	ROM/ LCD/GP IO	GPIO	Output	Address pin
X_LDD<15:0>	Bi	ROM LCD	LCD	Input	Data pins of high 16 bits
X_L_FCK	O	ROM LCD	LCD	Output	Data output enable. Active low.
X_L_LCK	O	ROM LCD	LCD	Output	Data write enable. Active low.

Table 389: ROM External Pin Descriptions

Functional Descriptions

Block Diagram

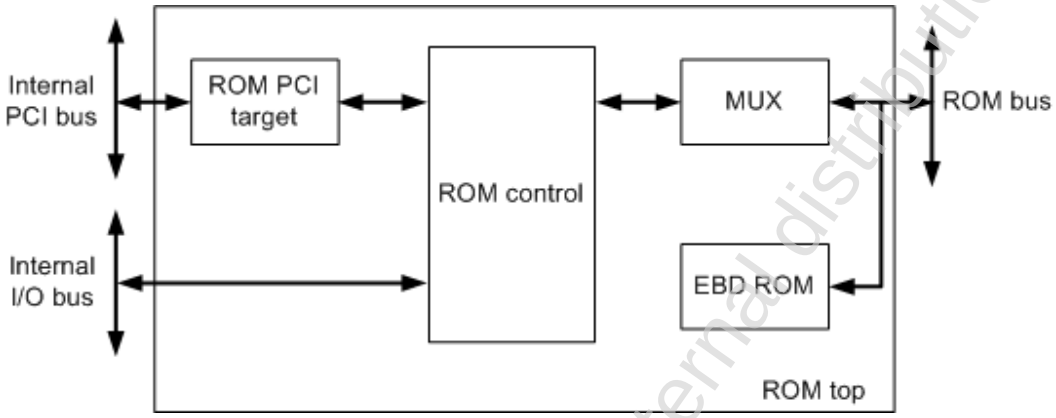


Figure 53: ROM Interface Block Diagram

Address Mapping

RISC IO access address[27:0]	PCI IO access address [31:0]	FA[1]	FCE_B<1>
0xC000000 ~ 0xFFFFFFF	0x6C000000~ 0x6FFFFFFF	0/1	X_DF_CS_B<1> active

Table 390: RISC I/O Access/PCI I/O Access Address Mapping

NOTE – It is recommended to use PCI I/O access mode.

Timing Diagram

Fixed Latency

- Single read
  ROM interface transfers one data every time X_FCE_B asserted.

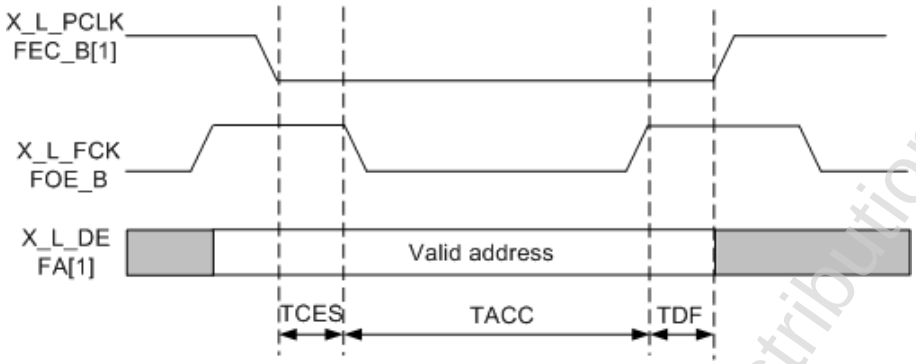


Figure 54: Fixed Latency Single Read

- Burst read
  ROM interface transfers more than one data every time X_FCE_B asserted.

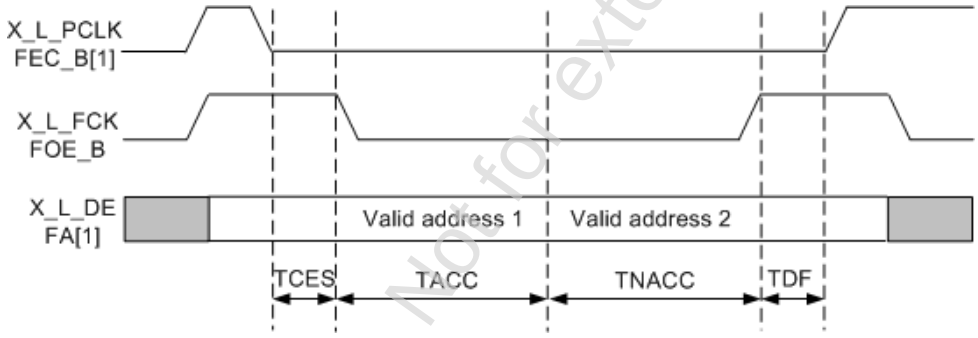


Figure 55: Fixed Latency Burst Read

- Single write
  ROM interface transfers one data every time X_FCE_B asserted.

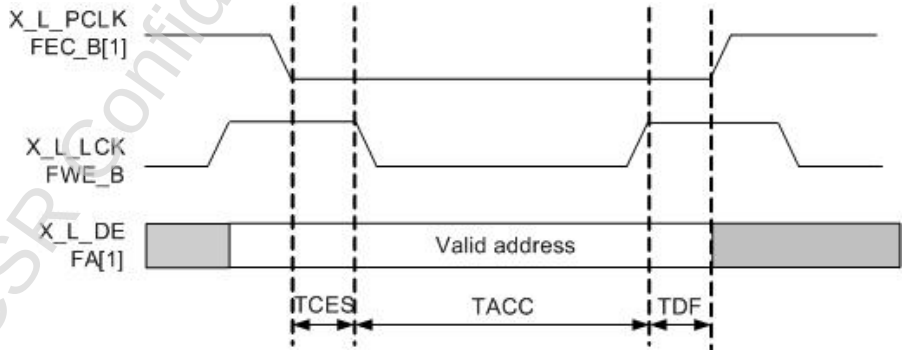


Figure 56: Fixed Latency Single Write

- Burst write
  ROM interface transfers one data every time X_FCE_B asserted.

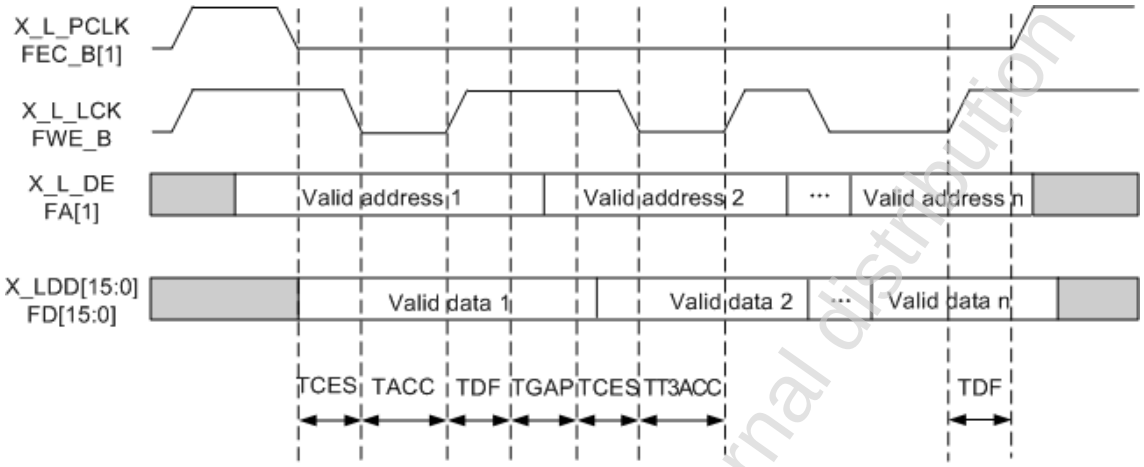


Figure 57: Fixed Latency Burst Write

Register Definition

Register Address Mapping

Base Address

Access Type	Address Mapping
ROM internal register base address through RISC I/O	0xA0010000
ROM internal register base address through PCI	0x57A00000
ROM device base address through RISC I/O	0x10000000
ROM device base address through PCI	0x60000000

Table 391: ROM Interface Address Mapping

Register Mapping

The ROM interface registers include configure registers CS0 to CS3.

RISC Address <11:0>	Register	Description
0x0004	ROM_CFG_CS1	ROM configure register for CS1
0x0014	ROM_SEL_EBD	Embedded ROM enable register
Others	-	Reserved

Table 392: ROM RISC I/O Register Mapping

Register Descriptions

- ROM CS1 Configure Register (ROM_CFG_CS1) – 0x0004
  ROM_CFG_CS1 control the timing and access mode for ROM port CS1

Bit	Name	Default	Description
6:0 (R/W)	RWGAP<6:0>	6'h0	Read/write interval (RWGAP x Tclk_pci) between each two read or two write.
7 (R/W)	PCI_FIFO_MODE	1'b0	Set 1'b1 or 1'b0 depending on the ROM device FIFO mode. Do not use FIFO when doing PCI_COPY.
13:8 (R/W)	TACC<5:0>	6'h0	Access time ((TACC+1) x Tclk_pci) is the active time for read/write signals.
15:14 (R/W)	TCES<1:0>	2'h0	CS active time before read/write signal is active. (TCES+1) x Tclk_pci for write, (TCES+2) x Tclk_pci for read
21:16 (R/W)	TNACC<5:0>	6'h0	Read burst access time. For burst read, the access time following the first access time can be smaller.
23:22 (R/W)	TDF<1:0>	2'h0	Read/write data hold time.
24 (R/W)	DWORD_ACC	1'b0	Enables DWORD access.
25 (R/W)	BURST_READ	1'b0	Enables burst read.
26 (R/W)	BURST_WRITE	1'b0	Enables burst write.
27 (R/W)	-	1'b0	Must be 0
28 (R/W)	BE_ACTIVER	1'b0	Read byte enabled.
29 (R/W)	WRITE_EN	1'b0	Enables write. 1: Writes to the ROM address will be written to the ROM port. 0: Writes to the ROM address will be masked. No writes will occur on the ROM port.
31:30 (R/W)	BUS_WIDTH	2'b00	Defines the bus width of ROM port. 00: Bus width is 8 bits. 01: Bus width is 16 bits. 10: Reserved 11: Reserved.

Table 393: ROM CS1 Configure Register

- Embedded ROM Enable Register (ROM_SEL_EBD) – 0x0014

Bit	Name	Default	Description
14:0 (R)		15'h0	Reserved
15 (R/W)	SEL_EBD	X_TEST_MODE<3>	0: External ROM 1: Embedded ROM
31:16 (R)		16'h0	Reserved

Table 394: Embedded ROM Enable Register

NAND Flash Controller

Overview

The NAND Flash controller provides the most cost-effective solution for the solid-state mass storage market, supporting up to 5-cycle address and 8-bit data width bus NAND Flash and SLC/MLC type NAND Flash.

Feature List

- The NAND Flash can be used as bootloader during start-up, or as a simple storage device. After a successful boot-up, the remaining block of NAND Flash can be used for data storage.
- The NAND Flash controller supports most of current NAND Flash types on market, such as 512+16bytes/page or 2048+64bytes/page or 4096+128/218bytes/page or 8192+436bytes/page, which can be configured by SM_PAGE_SIZE register. It also provides an independent memory FIFO and a DMA channel for data transfer to/from DRAM.
- The NAND Flash controller supports five types of data transfer:
  - DMA read data from FIFO
  - DMA write data to FIFO
  - RISC read data from FIFO
  - RISC write data to FIFO
  - RISC direct address read data from NAND Flash (like read ROM)
- In normal read/write mode, RISC writes commands or addresses to the NAND Flash interface to start a read/write cycle, data can be exchanged either between the DMA controller and the FIFO of the NAND flash interface, or between RISC and the FIFO of the NAND flash interface.
- In direct read mode (used when booting from the NAND flash), RISC will send an address to the I/O address bus and select a NAND flash simultaneously, then the NAND Flash interface will generate a read command and an address sequence to the NAND Flash. When data is ready on the I/O data bus, the NAND flash interface will assert a data-ready signal to IOCLK.
- The NAND flash controller supports 256 bytes or 512 bytes hamming ECC encoded for 8-bit NAND flash. Software can use encode to check and correct bit errors.
- The NAND flash controller supports 2048/4096/8192 bytes BCH ECC encode/decode for the 8bit NAND flash. Software can use the function to check and correct up to 12-bit or 24-bit/1024 bytes of errors.
  - For 2048+64 bytes, can support correct 12*2 bits error limited.
  - For 4096+128 bytes, can support correct 12*4 bits error limited.
  - For 4096+218 bytes, can support correct 24*4 bits error limited.
  - For 8192+436 bytes, can support correct 24*8 bits error limited.

- If the system boots from the NAND flash, then the NAND flash will be mapped to the bottom of the system address starting from 0x0000 0000. Usually a bootloader code will be put into block 0 of the NAND flash. RISC first executes the bootloader in which it configures the DRAM and initializes the hardware, then goes into a loop to copy the system code page by page.
- Multiple CS enable should be avoided, otherwise it will cause errors and prevent the hardware from being protected and recovered.

Pin Description

External Pin Descriptions

The following table shows the pins used in the NAND interface and their functions. The NAND interface pins are multiplexed with other devices.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_DF_WE_B	O	SD slot2 bus interface	Decided by boot mode	Output high	Write enable, active low
X_DF_RE_B	O	SD slot2 bus interface	Same as above	Output high	Read enable, active low
X_DF_WP_B	O	SD slot0 interface	Same as above	Output high	Read enable, active low.
X_DF_CLE	O	SD slot0 interface	Same as above	Output Low	command latch
X_DF_ALE	O	SD slot0 interface	Same as above	Output low	address latch
X_DF_CS_B<0>	O	Internal nand bank0 and bank3	Decided by boot mode	Output high	NAND Flash chip select pins. Active low
X_DF_CS_B<1>	O	-	NAND Flash	Output high	NAND Flash chip select pins, active low
X_DF_RY_BY	I	-	NAND Flash	Input pull high	NAND Flash ready/busy
X_DF_AD[0]	Bi	SD slot0 data bus and SD slot2 data bus interface	Decided by boot mode	Input	Low 8-bit NAND flash data/address bus signal d0
X_DF_AD[1]	Bi	SD slot0 data bus and SD slot2 data bus interface	Decided by boot mode	Input	Low 8-bit NAND flash data/address bus signal d1
X_DF_AD[2]	Bi	SD slot0 data bus and SD slot2 data bus interface	Decided by boot mode	Input	Low 8-bit NAND flash data/address bus signal d2
X_DF_AD[3]	Bi	SD slot0 data bus and SD slot2 data	Decided by boot mode	Input	Low 8-bit NAND flash data/address

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
		bus interface			bus signal d3
X_DF_AD[4]	Bi	SD slot0 data bus and SD slot2 data bus interface	Decided by boot mode	Input	Low 8-bit NAND flash data/address bus signal d4
X_DF_AD[5]	Bi	SD slot0 data bus and SD slot2 data bus interface	Decided by boot mode	Input	Low 8-bit NAND flash data/address bus signal d5
X_DF_AD[6]	Bi	SD slot0 data bus and SD slot2 data bus interface	Decided by boot mode	Input	Low 8-bit NAND flash data/address bus signal d6
X_DF_AD[7]	Bi	SD slot0 data bus and SD slot2 data bus interface	Decided by boot mode	Input	Low 8-bit NAND flash data/address bus signal d7

Table 395: NAND External Pin Descriptions

If the boot mode is NAND boot or Embedded ROM NAND boot, pin X_DF_WE_B and X_DF_CS_B<0> do NAND Flash Interface function; if the boot mode is Embedded ROM SD boot, this pin does SD Slot0 or slot2 bus function.

For pin X_DF_CS_B<0>, if the boot mode is NAND boot, internal bank0 will be selected. If boot mode is Embeded NAND boot, internal bank2 will be selected.

External NAND CS0/CS1

- Embed ROM (SLC, MLC NAND) boot mode, nand controller bank2 must connects with external nand cs0 pin, then it always be used
- In NAND direct boot mode, nand controller bank0 connects with external nand cs0 pin, then it always be used
- NAND controller bank3 always connects with external nand cs1 pin.
- NAND controller bank1 didn't connect with any external pin.
- In emmbed ROM (SD) boot mode, external NAND CS0 and CS1 cannot be used.

Functional Descriptions

Block Diagram

A program operation can typically program the 528-byte page in approximately 200us, and an erase operation can be performed in typically 2ms on a 16K-byte block. Data in the page can be read out at 50ns cycle time per byte. The I/O pins serve as ports for address and data input/output as well as command inputs. The on-chip write controller automates all program and erase functions including pulse repetition (where required), internal verification and margining of data.

Even write-intensive systems can take advantage of the extended 100K program/erase cycles reliability of the NAND Flash by providing two types of error-correction code (ECC) with a real-time map-out algorithm. NAND Flash is an optimum solution for large non-volatile storage applications such as solid-state storage, digital voice recorder, digital still camera and other portable applications.

The features are:

- Low cost: Compatible cost to memory, no overhead
- Good reliability: Two types of ECC built into the data format
  - Hamming algorithmic ECC encode
  - BCH code ECC encode/decode
- High performance: NAND Flash provides the highest speed
  - DMA mode (read/write)
  - I/O mode (read/write)
  - Direct read mode (such as ROM, read-only)
- Easy interface

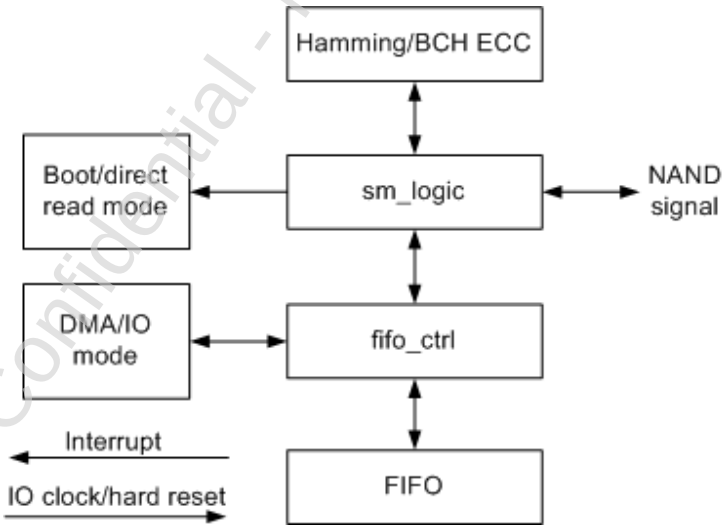


Figure 58: NAND Block Diagram

ECC Data in NAND Format

There are two types of ECC:

- Hamming encode: supports 512+16 bytes /page and 2048+64 bytes/page
- BCH code ECC: supports 2048+64 bytes/page, 4096+128/218 bytes/page and 8192+436 bytes/page

When writing ECC result into the spare area of the NAND Flash using hardware ECC, some data format used in different page size must be used, such as:

- For Hammig encode
  - 512 bytes/page
  - 2048 bytes/page

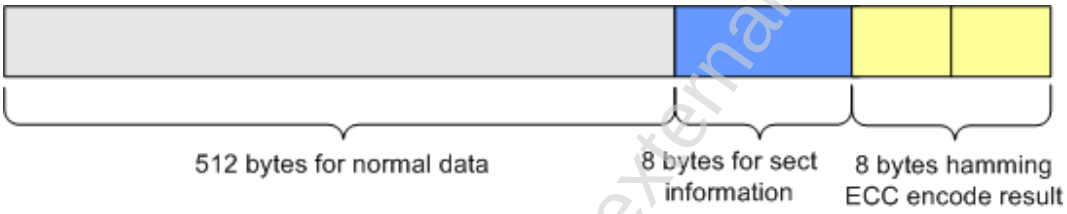


Figure 59: 512+16 Bytes/Page with Hamming ECC Data Format

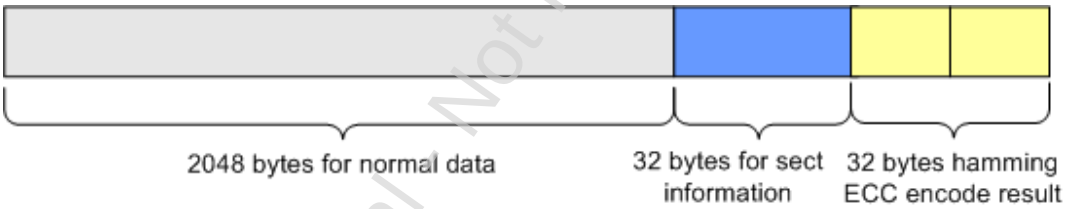


Figure 60: 2048+64 Bytes/Page with Hamming ECC Data Format

- For BCH encode
  - 2048 bytes/page
  - 4096 bytes/page
  - 8192 bytes/page

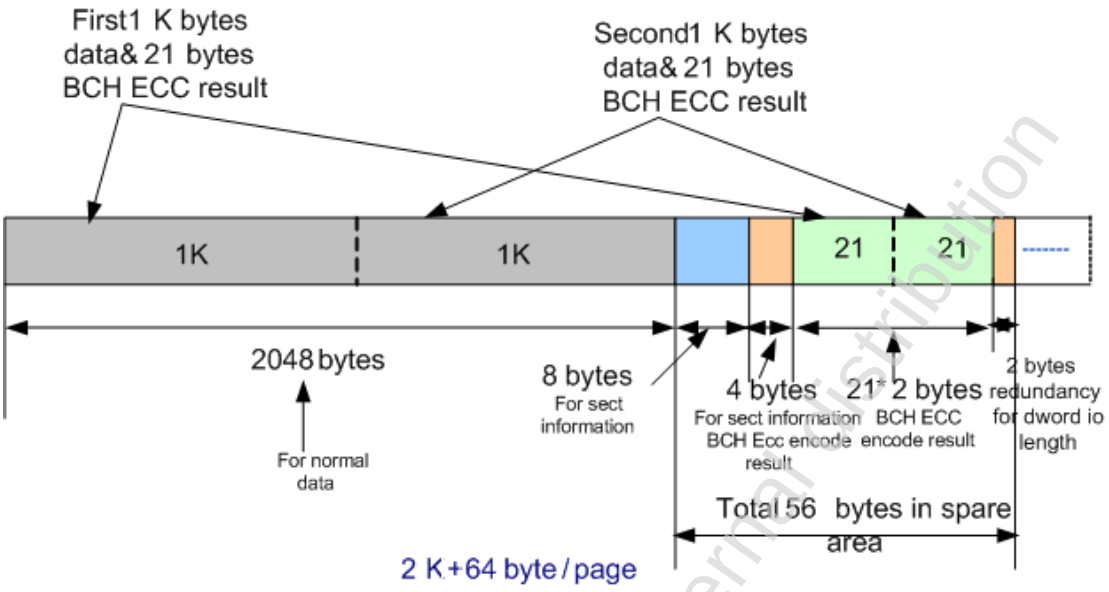


Figure 61: 2048+64 Bytes/Page with 12-bit BCH ECC Data Format

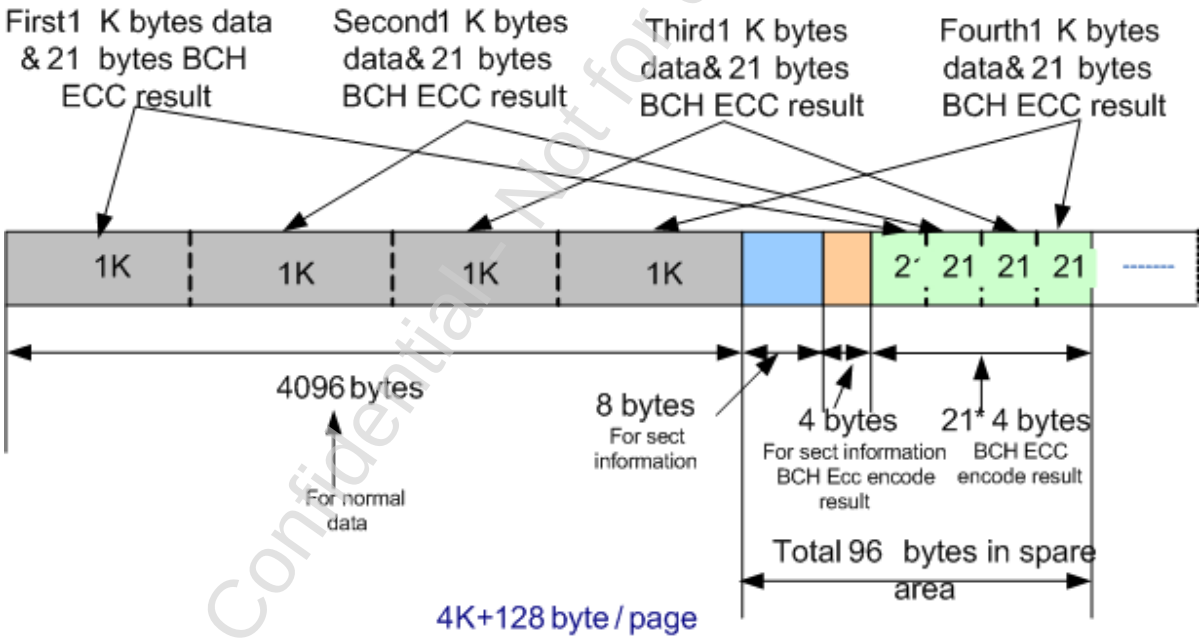


Figure 62: 4096+128 Bytes/Page with 12-bit BCH ECC Data Format

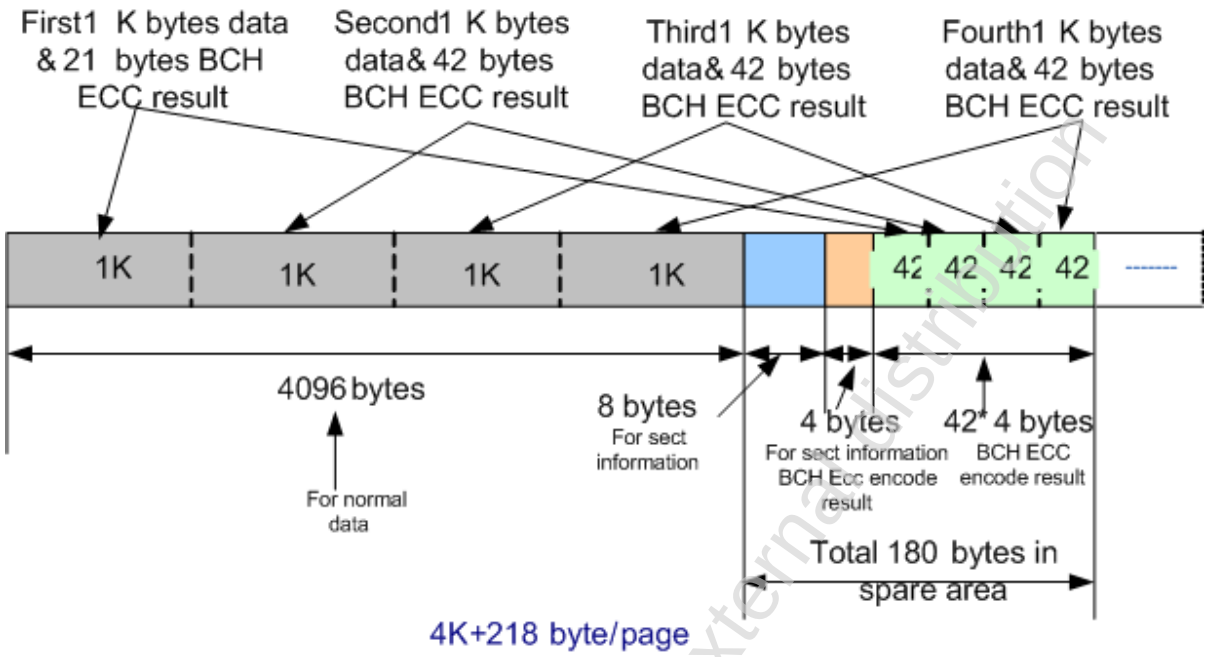


Figure 63: 4096+218 Bytes/Page with 24-bit BCH ECC Data Format

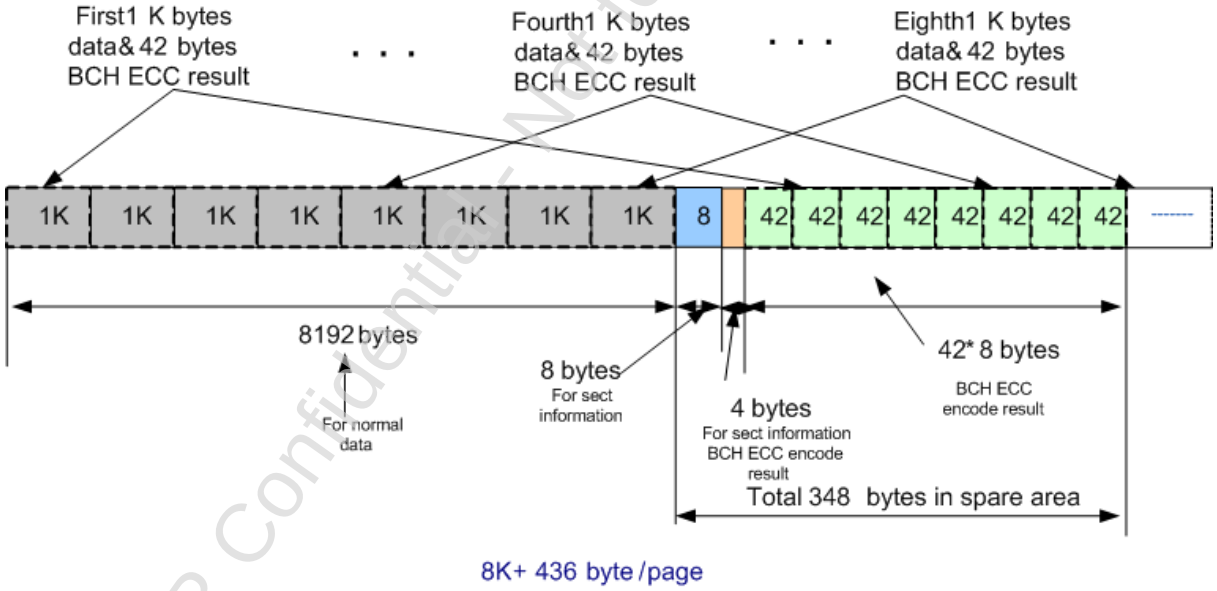


Figure 64: 8192+436 Bytes/Page with 24-Bit BCH ECC Data Format

Functional Timing

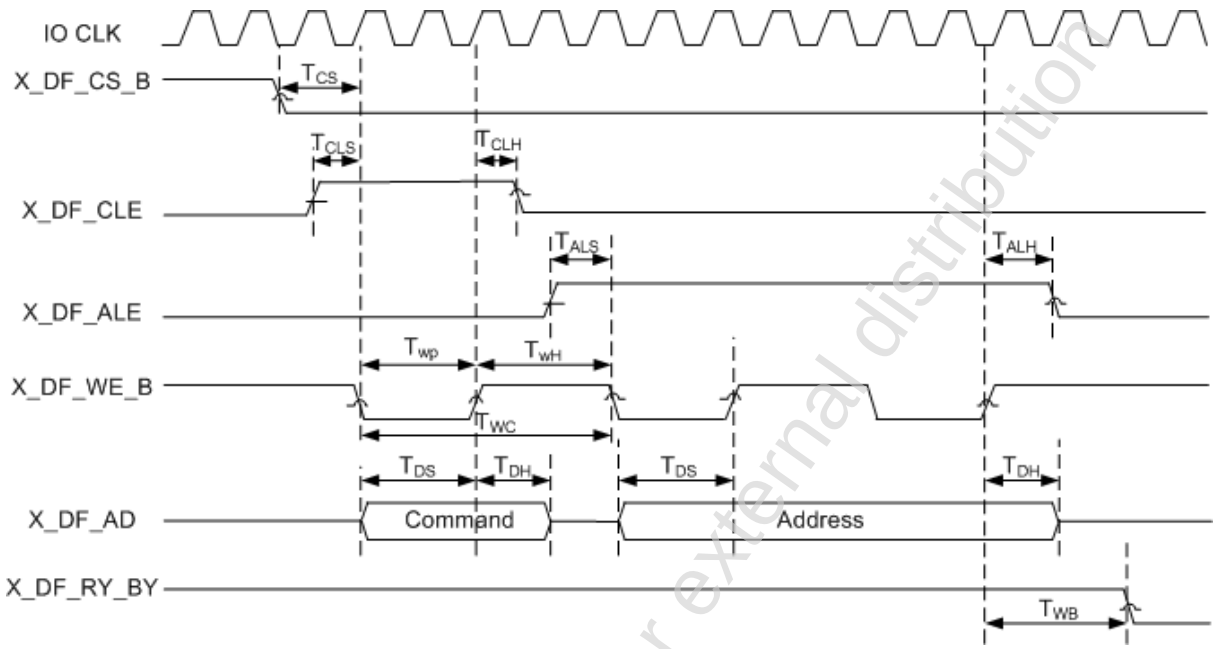


Figure 65: Command and Address Timing Diagram

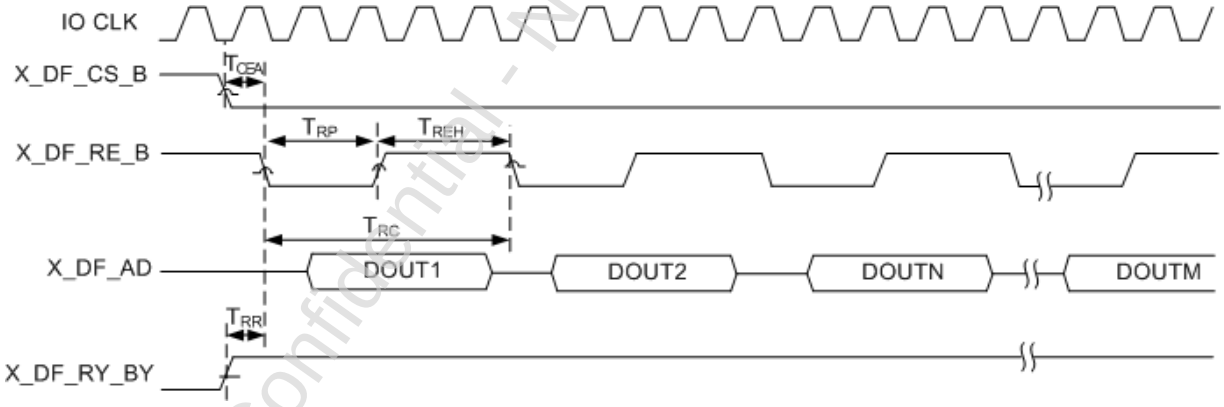


Figure 66: Read Data Timing Diagram

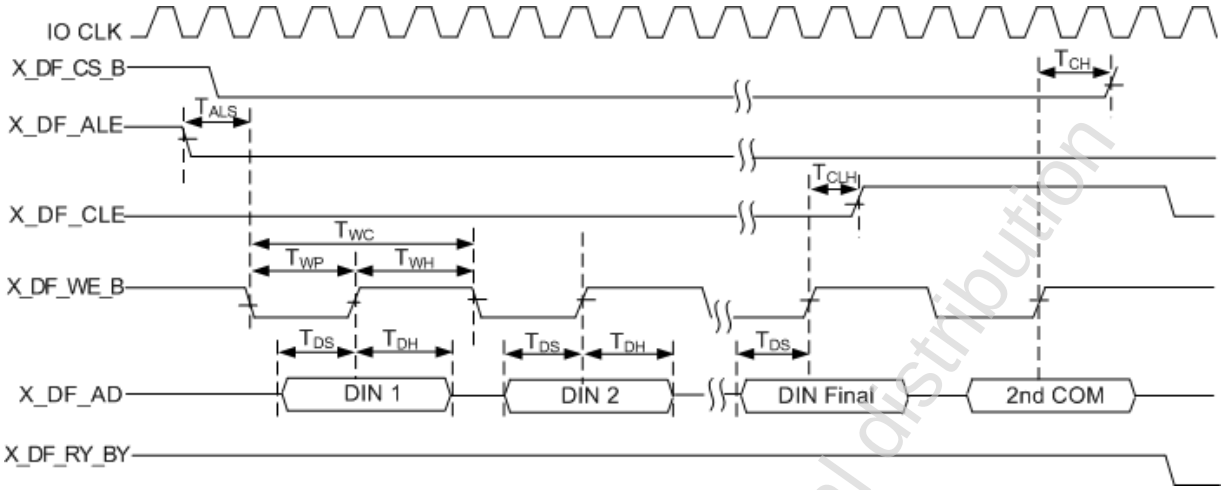


Figure 67: Write Data Timing Diagram

Timing Value	I/O CLK Number and SM_WAIT Register
T_{CS}	2
T_{CLS}	1
T_{ALS}	1
T_{CLH}	1+RD_WT_HI
T_{ALH}	1+RD_WT_HI
T_{WP}	WT_PULSE
T_{WH}	1+RD_WT_HI
T_{DS}	WT_PULSE
T_{DH}	RD_WT_HI
T_{WB}	WAIT
T_{CEA}	1
T_{RP}	1+RD_PULSE
T_{REH}	1+RD_WT_HI
T_{RR}	10

Table 396: Read Data Timing Values

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
The NAND controller internal register base address	0x80070000

Table 397: NAND Interface Address Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x00	BOOT_MODE_ALL	Boot mode register
0x04	SM_WAIT	Wait state for read/write NAND flash
0x08	SM_BANK_SEL	Chip select
0x0C	SM_ADD_NUM	Bytes number of NAND flash address
0x10	SM_CMD	Command to NAND flash
0x14	SM_LOW_ADDR	Low 4 bytes of address to NAND flash
0x18	SM_HIGH_ADDR	High bytes of address to the NAND flash (needed only by the NAND flash with more than 256 MB of memory size)
0x1C	SM_PAGE_SIZE	Page size of NAND flash
0x20	SM_INT_EN	Interrupt mask register
0x24	SM_INT_STATUS	Interrupt status register
0x28	SM_CTRL	Control register of NAND flash
0x2C	SM_ECC_SET	Hardware ECC set register
0x30	SM_ECC_AREA1	Hardware ECC result store register
0x34	SM_ECC_AREA2	Hardware ECC result store register
0x38	SM_ECC_AREA3	Hardware ECC result store register
0x3C	SM_ECC_AREA4	Hardware ECC result store register
0x40	SM_ECC_AREA5	Hardware ECC result store register
0x44	SM_ECC_AREA6	Hardware ECC result store register
0x48	SM_ECC_AREA7	Hardware ECC result store register

RISC Address <11:0>	Register	Description
0x4C	SM_ECC_AREA8	Hardware ECC result store register
0x50	SM_WP_EN	NAND flash read-only zone index
0x54	SM_WP_CS	NAND flash read-only zone chip enable
0x58	-	Reserved
0x5C	-	Reserved
0x60~0x7C	SM_WP_START_ADDR0~7	Read-only page start address value
0x80~0x9C	SM_WP_END_ADDR0~7	Read-only page end address value
0xA0~0xFC	-	Reserved
0x100	SM_ECC_ERR_STATUS0	Indicates no errors or has no errors to indicate
0x104	SM_ECC_ERR_NUMS0	Indicates error number for the first to fourth 1Kbyte
0x108	SM_ECC_ERR_NUMS1	Indicates error number for the fifth to eighth 1Kbyte
0x10C	-	Reserved
0x110	SM_FIRST_ERR_INDEX0	First bank error index0
0x114	SM_FIRST_ERR_INDEX1	First bank error index1
0x118	SM_FIRST_ERR_INDEX2	First bank error index2
0x11C	SM_FIRST_ERR_INDEX3	First bank error index3
0x120	SM_FIRST_ERR_INDEX4	First bank error index4
0x124	SM_FIRST_ERR_INDEX5	First bank error index5
0x128	SM_FIRST_ERR_INDEX6	First bank error index6
0x12C	SM_FIRST_ERR_INDEX7	First bank error index7
0x130	SM_FIRST_ERR_INDEX8	First bank error index8
0x134	SM_FIRST_ERR_INDEX9	First bank error index9
0x138	SM_FIRST_ERR_INDEX10	First bank error index10
0x13C	SM_FIRST_ERR_INDEX11	First bank error index11
0x140	SM_FIRST_ERR_INDEX12	First bank error index12
0x144	SM_FIRST_ERR_INDEX13	First bank error index13
0x148	SM_FIRST_ERR_INDEX14	First bank error index14

RISC Address <11:0>	Register	Description
0x14C	SM_FIRST_ERR_INDEX15	First bank error index15
0x150	SM_FIRST_ERR_INDEX16	First bank error index16
0x154	SM_FIRST_ERR_INDEX17	First bank error index17
0x158	SM_FIRST_ERR_INDEX18	First bank error index18
0x15C	SM_FIRST_ERR_INDEX19	First bank error index19
0x160	SM_FIRST_ERR_INDEX20	First bank error index20
0x164	SM_FIRST_ERR_INDEX21	First bank error index21
0x168	SM_FIRST_ERR_INDEX22	First bank error index22
0x16C	SM_FIRST_ERR_INDEX23	First bank error index23
0x170	SM_SECOND_ERR_INDEX0	Second bank error index0
0x174	SM_SECOND_ERR_INDEX1	Second bank error index1
0x178	SM_SECOND_ERR_INDEX2	Second bank error index2
0x17C	SM_SECOND_ERR_INDEX3	Second bank error index3
0x180	SM_SECOND_ERR_INDEX4	Second bank error index4
0x184	SM_SECOND_ERR_INDEX5	Second bank error index5
0x188	SM_SECOND_ERR_INDEX6	Second bank error index6
0x18C	SM_SECOND_ERR_INDEX7	Second bank error index7
0x190	SM_SECOND_ERR_INDEX8	Second bank error index8
0x194	SM_SECOND_ERR_INDEX9	Second bank error index9
0x198	SM_SECOND_ERR_INDEX10	Second bank error index10
0x19C	SM_SECOND_ERR_INDEX11	Second bank error index11
0x1A0	SM_SECOND_ERR_INDEX12	Second bank error index12
0x1A4	SM_SECOND_ERR_INDEX13	Second bank error index13
0x1A8	SM_SECOND_ERR_INDEX14	Second bank error index14
0x1AC	SM_SECOND_ERR_INDEX15	Second bank error index15
0x1B0	SM_SECOND_ERR_INDEX16	Second bank error index16
0x1B4	SM_SECOND_ERR_INDEX17	Second bank error index17
0x1B8	SM_SECOND_ERR_INDEX18	Second bank error index18
0x1BC	SM_SECOND_ERR_INDEX19	Second bank error index19

RISC Address <11:0>	Register	Description
0x1C0	SM_SECOND_ERR_INDEX20	Second bank error index20
0x1C4	SM_SECOND_ERR_INDEX21	Second bank error index21
0x1C8	SM_SECOND_ERR_INDEX22	Second bank error index22
0x1CC	SM_SECOND_ERR_INDEX23	Second bank error index23
0x1D0	SM_THIRD_ERR_INDEX0	Third bank error index0
0x1D4	SM_THIRD_ERR_INDEX1	Third bank error index1
0x1D8	SM_THIRD_ERR_INDEX2	Third bank error index2
0x1DC	SM_THIRD_ERR_INDEX3	Third bank error index3
0x1E0	SM_THIRD_ERR_INDEX4	Third bank error index4
0x1E4	SM_THIRD_ERR_INDEX5	Third bank error index5
0x1E8	SM_THIRD_ERR_INDEX6	Third bank error index6
0x1EC	SM_THIRD_ERR_INDEX7	Third bank error index7
0x1F0	SM_THIRD_ERR_INDEX8	Third bank error index8
0x1F4	SM_THIRD_ERR_INDEX9	Third bank error index9
0x1F8	SM_THIRD_ERR_INDEX10	Third bank error index10
0x1FC	SM_THIRD_ERR_INDEX11	Third bank error index11
0x200	SM_THIRD_ERR_INDEX12	Third bank error index12
0x204	SM_THIRD_ERR_INDEX13	Third bank error index13
0x208	SM_THIRD_ERR_INDEX14	Third bank error index14
0x20C	SM_THIRD_ERR_INDEX15	Third bank error index15
0x210	SM_THIRD_ERR_INDEX16	Third bank error index16
0x214	SM_THIRD_ERR_INDEX17	Third bank error index17
0x218	SM_THIRD_ERR_INDEX18	Third bank error index18
0x21C	SM_THIRD_ERR_INDEX19	Third bank error index19
0x220	SM_THIRD_ERR_INDEX20	Third bank error index20
0x224	SM_THIRD_ERR_INDEX21	Third bank error index21
0x228	SM_THIRD_ERR_INDEX22	Third bank error index22
0x22C	SM_THIRD_ERR_INDEX23	Third bank error index23
0x230	SM_FOURTH_ERR_INDEX0	Fourth bank error index0

RISC Address <11:0>	Register	Description
0x234	SM_FOURTH_ERR_INDEX1	Fourth bank error index1
0x238	SM_FOURTH_ERR_INDEX2	Fourth bank error index2
0x23C	SM_FOURTH_ERR_INDEX3	Fourth bank error index3
0x240	SM_FOURTH_ERR_INDEX4	Fourth bank error index4
0x244	SM_FOURTH_ERR_INDEX5	Fourth bank error index5
0x248	SM_FOURTH_ERR_INDEX6	Fourth bank error index6
0x24C	SM_FOURTH_ERR_INDEX7	Fourth bank error index7
0x250	SM_FOURTH_ERR_INDEX8	Fourth bank error index8
0x254	SM_FOURTH_ERR_INDEX9	Fourth bank error index9
0x258	SM_FOURTH_ERR_INDEX10	Fourth bank error index10
0x25C	SM_FOURTH_ERR_INDEX11	Fourth bank error index11
0x260	SM_FOURTH_ERR_INDEX12	Fourth bank error index12
0x264	SM_FOURTH_ERR_INDEX13	Fourth bank error index13
0x268	SM_FOURTH_ERR_INDEX14	Fourth bank error index14
0x26C	SM_FOURTH_ERR_INDEX15	Fourth bank error index15
0x270	SM_FOURTH_ERR_INDEX16	Fourth bank error index16
0x274	SM_FOURTH_ERR_INDEX17	Fourth bank error index17
0x278	SM_FOURTH_ERR_INDEX18	Fourth bank error index18
0x27C	SM_FOURTH_ERR_INDEX19	Fourth bank error index19
0x280	SM_FOURTH_ERR_INDEX20	Fourth bank error index20
0x284	SM_FOURTH_ERR_INDEX21	Fourth bank error index21
0x288	SM_FOURTH_ERR_INDEX22	Fourth bank error index22
0x28C	SM_FOURTH_ERR_INDEX23	Fourth bank error index23
0x290	SM_FIFTH_ERR_INDEX0	Fifth bank error index0
0x294	SM_FIFTH_ERR_INDEX1	Fifth bank error index1
0x298	SM_FIFTH_ERR_INDEX2	Fifth bank error index2
0x29C	SM_FIFTH_ERR_INDEX3	Fifth bank error index3
0x2A0	SM_FIFTH_ERR_INDEX4	Fifth bank error index4
0x2A4	SM_FIFTH_ERR_INDEX5	Fifth bank error index5

RISC Address <11:0>	Register	Description
0x2A8	SM_FIFTH_ERR_INDEX6	Fifth bank error index6
0x2AC	SM_FIFTH_ERR_INDEX7	Fifth bank error index7
0x2B0	SM_FIFTH_ERR_INDEX8	Fifth bank error index8
0x2B4	SM_FIFTH_ERR_INDEX9	Fifth bank error index9
0x2B8	SM_FIFTH_ERR_INDEX10	Fifth bank error index10
0x2BC	SM_FIFTH_ERR_INDEX11	Fifth bank error index11
0x2C0	SM_FIFTH_ERR_INDEX12	Fifth bank error index12
0x2C4	SM_FIFTH_ERR_INDEX13	Fifth bank error index13
0x2C8	SM_FIFTH_ERR_INDEX14	Fifth bank error index14
0x2CC	SM_FIFTH_ERR_INDEX15	Fifth bank error index15
0x2D0	SM_FIFTH_ERR_INDEX16	Fifth bank error index16
0x2D4	SM_FIFTH_ERR_INDEX17	Fifth bank error index17
0x2D8	SM_FIFTH_ERR_INDEX18	Fifth bank error index18
0x2DC	SM_FIFTH_ERR_INDEX19	Fifth bank error index19
0x2E0	SM_FIFTH_ERR_INDEX20	Fifth bank error index20
0x2E4	SM_FIFTH_ERR_INDEX21	Fifth bank error index21
0x2E8	SM_FIFTH_ERR_INDEX22	Fifth bank error index22
0x2EC	SM_FIFTH_ERR_INDEX23	Fifth bank error index23
0x2F0	SM_SIXTH_ERR_INDEX0	Sixth bank error index0
0x2F4	SM_SIXTH_ERR_INDEX1	Sixth bank error index1
0x2F8	SM_SIXTH_ERR_INDEX2	Sixth bank error index2
0x2FC	SM_SIXTH_ERR_INDEX3	Sixth bank error index3
0x300	SM_SIXTH_ERR_INDEX4	Sixth bank error index4
0x304	SM_SIXTH_ERR_INDEX5	Sixth bank error index5
0x308	SM_SIXTH_ERR_INDEX6	Sixth bank error index6
0x30C	SM_SIXTH_ERR_INDEX7	Sixth bank error index7
0x310	SM_SIXTH_ERR_INDEX8	Sixth bank error index8
0x314	SM_SIXTH_ERR_INDEX9	Sixth bank error index9
0x318	SM_SIXTH_ERR_INDEX10	Sixth bank error index10

RISC Address <11:0>	Register	Description
0x31C	SM_SIXTH_ERR_INDEX11	Sixth bank error index11
0x320	SM_SIXTH_ERR_INDEX12	Sixth bank error index12
0x324	SM_SIXTH_ERR_INDEX13	Sixth bank error index13
0x328	SM_SIXTH_ERR_INDEX14	Sixth bank error index14
0x32C	SM_SIXTH_ERR_INDEX15	Sixth bank error index15
0x330	SM_SIXTH_ERR_INDEX16	Sixth bank error index16
0x334	SM_SIXTH_ERR_INDEX17	Sixth bank error index17
0x338	SM_SIXTH_ERR_INDEX18	Sixth bank error index18
0x33C	SM_SIXTH_ERR_INDEX19	Sixth bank error index19
0x340	SM_SIXTH_ERR_INDEX20	Sixth bank error index20
0x344	SM_SIXTH_ERR_INDEX21	Sixth bank error index21
0x348	SM_SIXTH_ERR_INDEX22	Sixth bank error index22
0x34C	SM_SIXTH_ERR_INDEX23	Sixth bank error index23
0x350	SM_SEVENTH_ERR_INDEX0	Seventh bank error index0
0x354	SM_SEVENTH_ERR_INDEX1	Seventh bank error index1
0x358	SM_SEVENTH_ERR_INDEX2	Seventh bank error index2
0x35C	SM_SEVENTH_ERR_INDEX3	Seventh bank error index3
0x360	SM_SEVENTH_ERR_INDEX4	Seventh bank error index4
0x364	SM_SEVENTH_ERR_INDEX5	Seventh bank error index5
0x368	SM_SEVENTH_ERR_INDEX6	Seventh bank error index6
0x36C	SM_SEVENTH_ERR_INDEX7	Seventh bank error index7
0x370	SM_SEVENTH_ERR_INDEX8	Seventh bank error index8
0x374	SM_SEVENTH_ERR_INDEX9	Seventh bank error index9
0x378	SM_SEVENTH_ERR_INDEX10	Seventh bank error index10
0x37C	SM_SEVENTH_ERR_INDEX11	Seventh bank error index11
0x380	SM_SEVENTH_ERR_INDEX12	Seventh bank error index12
0x384	SM_SEVENTH_ERR_INDEX13	Seventh bank error index13
0x388	SM_SEVENTH_ERR_INDEX14	Seventh bank error index14
0x38C	SM_SEVENTH_ERR_INDEX15	Seventh bank error index15

RISC Address <11:0>	Register	Description
0x390	SM_SEVENTH_ERR_INDEX16	Seventh bank error index16
0x394	SM_SEVENTH_ERR_INDEX17	Seventh bank error index17
0x398	SM_SEVENTH_ERR_INDEX18	Seventh bank error index18
0x39C	SM_SEVENTH_ERR_INDEX19	Seventh bank error index19
0x3A0	SM_SEVENTH_ERR_INDEX20	Seventh bank error index20
0x3A4	SM_SEVENTH_ERR_INDEX21	Seventh bank error index21
0x3A8	SM_SEVENTH_ERR_INDEX22	Seventh bank error index22
0x3AC	SM_SEVENTH_ERR_INDEX23	Seventh bank error index23
0x3B0	SM_EIGHTH_ERR_INDEX0	Eighth bank error index0
0x3B4	SM_EIGHTH_ERR_INDEX1	Eighth bank error index1
0x3B8	SM_EIGHTH_ERR_INDEX2	Eighth bank error index2
0x3BC	SM_EIGHTH_ERR_INDEX3	Eighth bank error index3
0x3C0	SM_EIGHTH_ERR_INDEX4	Eighth bank error index4
0x3C4	SM_EIGHTH_ERR_INDEX5	Eighth bank error index5
0x3C8	SM_EIGHTH_ERR_INDEX6	Eighth bank error index6
0x3CC	SM_EIGHTH_ERR_INDEX7	Eighth bank error index7
0x3D0	SM_EIGHTH_ERR_INDEX8	Eighth bank error index8
0x3D4	SM_EIGHTH_ERR_INDEX9	Eighth bank error index9
0x3D8	SM_EIGHTH_ERR_INDEX10	Eighth bank error index10
0x3DC	SM_EIGHTH_ERR_INDEX11	Eighth bank error index11
0x3E0	SM_EIGHTH_ERR_INDEX12	Eighth bank error index12
0x3E4	SM_EIGHTH_ERR_INDEX13	Eighth bank error index13
0x3E8	SM_EIGHTH_ERR_INDEX14	Eighth bank error index14
0x3EC	SM_EIGHTH_ERR_INDEX15	Eighth bank error index15
0x3F0	SM_EIGHTH_ERR_INDEX16	Eighth bank error index16
0x3F4	SM_EIGHTH_ERR_INDEX17	Eighth bank error index17
0x3F8	SM_EIGHTH_ERR_INDEX18	Eighth bank error index18
0x3FC	SM_EIGHTH_ERR_INDEX19	Eighth bank error index19
0x400	SM_EIGHTH_ERR_INDEX20	Eighth bank error index20

RISC Address <11:0>	Register	Description
0x404	SM_EIGHTH_ERR_INDEX21	Eighth bank error index21
0x408	SM_EIGHTH_ERR_INDEX22	Eighth bank error index22
0x40C	SM_EIGHTH_ERR_INDEX23	Eighth bank error index23
0x510	SM_2K_ECC_AREA0	Data BCH encode result storage
0x514	SM_2K_ECC_AREA1	Data BCH encode result storage
0x518	SM_2K_ECC_AREA2	Data BCH encode result storage
0x51C	SM_2K_ECC_AREA3	Data BCH encode result storage
0x520	SM_2K_ECC_AREA4	Data BCH encode result storage
0x524	SM_2K_ECC_AREA5	Data BCH encode result storage
0x528	SM_2K_ECC_AREA6	Data BCH encode result storage
0x52C	SM_2K_ECC_AREA7	Data BCH encode result storage
0x530	SM_2K_ECC_AREA8	Data BCH encode result storage
0x534	SM_2K_ECC_AREA9	Data BCH encode result storage
0x538	SM_2K_ECC_AREA10	Data BCH encode result storage
0x53C	SM_2K_ECC_AREA11	Data BCH encode result storage
0x540	SM_2K_ECC_AREA12	Data BCH encode result storage
0x544	SM_2K_ECC_AREA13	Data BCH encode result storage
0x548	SM_2K_ECC_AREA14	Data BCH encode result storage
0x54C	SM_2K_ECC_AREA15	Data BCH encode result storage
0x550	SM_2K_ECC_AREA16	Data BCH encode result storage
0x554	SM_2K_ECC_AREA17	Data BCH encode result storage
0x558	SM_2K_ECC_AREA18	Data BCH encode result storage
0x55C	SM_2K_ECC_AREA19	Data BCH encode result storage
0x560	SM_2K_ECC_AREA20	Data BCH encode result storage
0x564	SM_4K_ECC_AREA21	Data BCH encode result storage
0x568	SM_4K_ECC_AREA22	Data BCH encode result storage
0x56C	SM_4K_ECC_AREA23	Data BCH encode result storage
0x570	SM_4K_ECC_AREA24	Data BCH encode result storage
0x574	SM_4K_ECC_AREA25	Data BCH encode result storage

RISC Address <11:0>	Register	Description
0x578	SM_4K_ECC_AREA26	Data BCH encode result storage
0x57C	SM_4K_ECC_AREA27	Data BCH encode result storage
0x580	SM_4K_ECC_AREA28	Data BCH encode result storage
0x584	SM_4K_ECC_AREA29	Data BCH encode result storage
0x588	SM_4K_ECC_AREA30	Data BCH encode result storage
0x58C	SM_4K_ECC_AREA31	Data BCH encode result storage
0x590	SM_4K_ECC_AREA32	Data BCH encode result storage
0x594	SM_4K_ECC_AREA33	Data BCH encode result storage
0x598	SM_4K_ECC_AREA34	Data BCH encode result storage
0x59C	SM_4K_ECC_AREA35	Data BCH encode result storage
0x5A0	SM_4K_ECC_AREA36	Data BCH encode result storage
0x5A4	SM_4K_ECC_AREA37	Data BCH encode result storage
0x5A8	SM_4K_ECC_AREA38	Data BCH encode result storage
0x5AC	SM_4K_ECC_AREA39	Data BCH encode result storage
0x5B0	SM_4K_ECC_AREA40	Data BCH encode result storage
0x5B4	SM_4K_ECC_AREA41	Data BCH encode result storage
0x5B8	SM_8K_ECC_AREA42	Data BCH encode result storage
0x5BC	SM_8K_ECC_AREA43	Data BCH encode result storage
0x5C0	SM_8K_ECC_AREA44	Data BCH encode result storage
0x5C4	SM_8K_ECC_AREA45	Data BCH encode result storage
0x5C8	SM_8K_ECC_AREA46	Data BCH encode result storage
0x5CC	SM_8K_ECC_AREA47	Data BCH encode result storage
0x5D0	SM_8K_ECC_AREA48	Data BCH encode result storage
0x5D4	SM_8K_ECC_AREA49	Data BCH encode result storage
0x5D8	SM_8K_ECC_AREA50	Data BCH encode result storage
0x5DC	SM_8K_ECC_AREA51	Data BCH encode result storage
0x5E0	SM_8K_ECC_AREA52	Data BCH encode result storage
0x5E4	SM_8K_ECC_AREA53	Data BCH encode result storage
0x5E8	SM_8K_ECC_AREA54	Data BCH encode result storage

RISC Address <11:0>	Register	Description
0x5EC	SM_8K_ECC_AREA55	Data BCH encode result storage
0x5F0	SM_8K_ECC_AREA56	Data BCH encode result storage
0x5F4	SM_8K_ECC_AREA57	Data BCH encode result storage
0x5F8	SM_8K_ECC_AREA58	Data BCH encode result storage
0x5FC	SM_8K_ECC_AREA59	Data BCH encode result storage
0x600	SM_8K_ECC_AREA60	Data BCH encode result storage
0x604	SM_8K_ECC_AREA61	Data BCH encode result storage
0x608	SM_8K_ECC_AREA62	Data BCH encode result storage
0x60C	SM_8K_ECC_AREA63	Data BCH encode result storage
0x610	SM_8K_ECC_AREA64	Data BCH encode result storage
0x614	SM_8K_ECC_AREA65	Data BCH encode result storage
0x618	SM_8K_ECC_AREA66	Data BCH encode result storage
0x61C	SM_8K_ECC_AREA67	Data BCH encode result storage
0x620	SM_8K_ECC_AREA68	Data BCH encode result storage
0x624	SM_8K_ECC_AREA69	Data BCH encode result storage
0x628	SM_8K_ECC_AREA70	Data BCH encode result storage
0x62C	SM_8K_ECC_AREA71	Data BCH encode result storage
0x630	SM_8K_ECC_AREA72	Data BCH encode result storage
0x634	SM_8K_ECC_AREA73	Data BCH encode result storage
0x638	SM_8K_ECC_AREA74	Data BCH encode result storage
0x63C	SM_8K_ECC_AREA75	Data BCH encode result storage
0x640	SM_8K_ECC_AREA76	Data BCH encode result storage
0x644	SM_8K_ECC_AREA77	Data BCH encode result storage
0x648	SM_8K_ECC_AREA78	Data BCH encode result storage
0x64C	SM_8K_ECC_AREA79	Data BCH encode result storage
0x650	SM_8K_ECC_AREA80	Data BCH encode result storage
0x654	SM_8K_ECC_AREA81	Data BCH encode result storage
0x658	SM_8K_ECC_AREA82	Data BCH encode result storage
0x65C	SM_8K_ECC_AREA83	Data BCH encode result storage

RISC Address <11:0>	Register	Description
0x660~0x6FC	-	Reserved
0x700	SECT_BCH_ECC_AREA	Sector info BCH encode result storage
0x704	SECT_ECC_ERR_FLAG	Sector infor BCH ECC error number flag
0x708	SECT_ECC_ERR_INDEX	Sector error number index
0x70C~0xEFC	-	Reserved
0xF00	SM_DMA_IO_CTRL	Control bit for read/write NAND flash
0xF04	SM_DMA_IO_LEN	DMA length
0xF08	SM_FIFO_CTRL	Control bit for FIFO
0xF0C	SM_FIFO_LEVEL_CHK	FIFO level check
0xF10	SM_FIFO_OP	Reset and start bit for FIFO
0xF14	SM_FIFO_STATUS	FIFO data status
0xF18	SM_FIFO_DATA	FIFO data content

Table 398: NAND Flash Interface Register Mapping

Register Descriptions

- BOOT_MODE_ALL - 0x00

Bit	Name	Default	Description
5:0 (R)	boot_mode<5:0>	5'bxxxx	Chip boot mode define, it's the same value. Different boot modes will render different values.
6(R)	-	1'b1	This bit must be 1.
31:7	-	25'h0	Reserved

Table 399: Boot Mode Status Register

- SM_WAIT - 0x04

  Refer to the timing request in product datasheet to set the proper value. Value >= timing request (ns) / iocik (ns)

Bit	Name	Default	Description
3:0 (R/W)	RD_PULSE	4'b0001	The low pulse width of NAND Flash read signal (tRP) 0: One IOCLK

Bit	Name	Default	Description
			1: Two IOCLK **NOTE**: Set it to (fix value+1) for margin.
7:4 (R/W)	WT_PULSE	4'b0001	The low pulse width of write signal of NAND flash (tWP) 0: One IOCLK 1: Two IOCLK
15:8 (R/W)	WAIT	8'b1100	The width between WE high to Busy (tWB) 0: One IOCLK 1: Two IOCLK
19:16 (R/W)	RD_WT_HI	4'b0000	The write or read signal high hold time (tWH) or (tREH) these are the same register: 0: One IOCLK 1: Two IOCLK
30:20	-	11'h0	Reserved
31 (R/W)	WRITE_PROTECT_B	1'b0	Write protect pin set, 0: Output 0, protect ,cannot write 1: No output, write enable Read back is outside pin status, not register value.

Table 400: SM_WAIT Register

- SM_BANK_SEL - 0x08

Bit	Name	Default	Description
3:0 (R/W)	BANK_SEL<3:0>	4'b1110	Select NAND flash bank. Hardware does not protect this case. Set the bit to 0 will enable the corresponding NAND flash Set the bit to 1 will disable the corresponding NAND flash Cannot enable multi_cs active at any time. This will cause errors. For direct read the base address will be the following: bank 0: 0x10000000 bank 1: 0x14000000 bank 2: 0x18000000 bank 3: 0x1c000000 Set this register must follow the External NAND CS0, CS1 Descriptions 4 rules.
31:4	-	28'h0	Reserved

Table 401: SM_BANK_SEL Register

- SM_ADD_NUM - 0x0C

Bit	Name	Default	Description
3:0 (R/W)	ADD_MODE<3:0>	4'b0101	Number of address bytes 0000: 0 bytes address 0001: 1 bytes address 0101: 5 bytes address 1000: 8 bytes address
31:4	-	28'h0	Reserved

Table 402: SM_ADD_NUM Register

- SM_CMD - 0x10

  Write this register will start FSM (Finite State Machine). If there is no second command, then bit[15:8] must be 8'h0, or it will cause the FSM to run into the wrong state.

Bit	Name	Default	Description
7:0 (R/W)	CMD<7:0>	8'h0	Command to the NAND flash
15:8 (R/W)	CMD2<7:0>	8'h0	Second command to write to the NAND flash (used in the "command---address---command" format), this is to reduce delay time
31:16	-	16'h0	Reserved

Table 403: SM_CMD Register

- SM_LOW_ADDR - 0x14

Bit	Name	Default	Description
31:0 (R/W)	LO_ADD<31:0>	32'h0	Start address to NAND flash, 4 bytes.

Table 404: SM_LOW_ADDR Register

- SM_HIGH_ADDR - 0x18

Bit	Name	Default	Description
31:0 (R/W)	HI_ADD<31:0>	8'h0	Start address to NAND flash, the fifth to eighth bytes.

Table 405: SM_HIGH_ADDR Register

- SM_PAGE_SIZE – 0x1C
  The PAGE_SIZE register default value is 528 bytes.

Bit	Name	Default	Description
15:0 (R/W)	PAGE_SIZE	16'h210	528 bytes per page, it is used during DMA transfer. If auto hardware ECC decode is needed, the value must include the spare area. This value must be DWORD-aligned.
31:12	-	20'h0	Reserved

Table 406: SM_PAGE_SIZE Register

- SM_INT_EN - 0x20
  This is an interrupt mask register. Default value is 0x00, set the bit to 1 will enable the interrupt, but the corresponding bit in INT_STATUS_REG will be set regardless of whether or not the interrupt is enabled.

Bit	Name	Default	Description
0	-	-	Reserved
1 (R/W)	WT_ADD_DONE_EN	1'b0	Write address to the NAND flash done interrupt enable
2 (R/W)	IO_DMA_DONE_EN	1'b0	IO or DMA transfer done Interrupt enable
3 (R/W)	FIFO_THRESHOLD_EN	1'b0	Data in the FIFO reaches the threshold interrupt enable
4 (R/W)	WRIT_ZONE_ERR_EN	1'b0	Write protect zone error occur interrupt enable
5 (R/W)	ECC_ENC_OVER_EN	1'b0	Data ECC encode complete interrupt enable
6 (R/W)	ECC_DEC_OVER_EN	1'b0	Data ECC decode complete interrupt enable
7 (R/W)	SECT_ECC_ENC_OVER_EN	1'b0	SECT ECC encode complete interrupt enable
8 (R/W)	SECT_ECC_DEC_OVER_EN	1'b0	SECT ECC decode complete interrupt enable
31:9	-	27'h0	Reserved

Table 407: SM_INT_EN Register

- SM_INT_STATUS - 0x24

  When an interrupt occurs, set the bit to 1 (even when the interrupt enable register bit is not enabled). Write 1'b1 will clear the interrupt bit.

Bit	Name	Default	Description
0	-	-	Reserved
1 (R/W)	WT_ADD_DONE	1'b0	Write address to the NAND flash done interrupt
2 (R/W)	IO_DMA_DONE	1'b0	IO or DMA transfer done interrupt
3 (R/W)	FIFO_THRESHOLD	1'b0	Data in the FIFO reaches the threshold Interrupt
4 (R/W)	WRIT_ZONE_ERR	1'b0	Write protect zone error happen interrupt
5 (R/W)	ECC_ENC_OVER	1'b0	ECC encode complete interrupt
6 (R/W)	ECC_DEC_OVER	1'b0	ECC decode complete interrupt
7 (R/W)	SECT_ECC_ENC_OVER	1'b0	SECT INFO ECC encode complete interrupt
8 (R/W)	SECT_ECC_DEC_OVER	1'b0	SECT INFO ECC decode complete interrupt
31:9	-	23'h0	Reserved

Table 408: SM_INT_STATUS Register

- SM_CTRL - 0x28

  This register is used to control the NAND Flash.

Bit	Name	Default	Description
0 (R/W)	Direct read	1'b1	Set 1 will enable I/O directly read NAND flash Set 0 will disable direct read
1 (R/W)	Data width	1'b0	Set 0 will enable the 8-bit NAND flash Set 1 not support. Must not set 1;
2	-	-	Reserved
3	NAND page sel0	-	Use with bit5 to choose page size in direct read
4 (R/W)	Auto write ECC	1'b0	Set 1 will enable hardware write ECC auto, in this case, DMA io write length must be equal to the page size, including the spare area (for example, (512+16),(2048+64)) Set 0 to disable
5 (R/W)	NAND page sel1	1'bx	Use with bit3 to choose page size in direct read: Set {sel1,sel0} 00: 512page size Set {sel1,sel0} 01:2k page size Set {sel1,sel0} 10: 4k page size

Bit	Name	Default	Description
			Set {sel1,sel0} 11: 8k page size
6 (R/W)	READ_SPARE	1'b0	Only used in direct read mode Set 1 to continue to read the spare area; Clear this page to read the next pate.
31:7	-	23'h0	Reserved

Table 409: SM_CTRL Register

- SM_ECC_SET - 0x2C

  Hardware only supports one-page ECC generation, before using, software must reset ECC to clear ECC store registers.

  SiRFatlasV supports only one type of auto write ECC (auto encode). When enable hardware write ECC auto (SM_CTRL bit4), ECC encode auto write into nand flash spear area. When program one-page with auto write ECC, input data (program NAND) will be written firstly,and then followed by the encode ECC result.

  SiRFatlasV supports only two types of read data with auto hard BCH ECC (decode):

  - If the 2-BCH (date_bch and sector_bch) module is enabled, then it will read the full page (including spare area) from address0 to page end.
  - If only SECT BCH ECC module is enabled, then it will only read the spare area from spare area address 0 to page end

NOTE – Other multi-time page write (program) is not supported in MLC NAND flash and by BCH ECC.

Bit	Name	Default	Description
0 (R/W)	Hardware ECC	1'b0	Set 1: Enable hamming hardware ECC. Set 0: Bypass ECC.
1 (R/W)	Hardware ECC reset	1'b0	Set 1 to reset the ECC register of the Hamming hardware.
2 (R/W)	Half page	1'b1	For 8-bit NAND flash: Set 1 to enable 256-byte ECC, Set 0 to enable 512-byte ECC.
3(R/W)	BCH_ECC_mode_sel	1'b0	Set 0: 12-bit ECC per 1Kbyte. Set 1: 24-bit ECC per 1Kbyte.
4 (R/W)	data_BCH_ECC_enable	1'b0	Set 1 to enable data hardware BCH ECC.
5 (R/W)	data_BCH_ECC_reset	1'b0	Set 1 to reset BCH hardware ECC register.
7:6 (R/W)	2k_4k_8k_page	2'b00	2048/4096/8192 bytes BCH ECC select: 2'b00: 2048 bytes ECC 2'b01: 4096 bytes ECC

Bit	Name	Default	Description
			2'b10: 8192 bytes ECC
8 (R/W)	sector_BCH_ECC_enable	1'b0	Set 1 to enable Sector information hardware BCH ECC.
9 (R/W)	sector_BCH_ECC_reset	1'b0	Set 1 to reset Sector information BCH hardware ECC register.
10 (R/W)	sector_read_only	1'b0	0: Decode when full page data are read 1: Decode if only the sector information is read. It will be enabled when the software only reads from the spare area.
31:11	-	21'h0	Reserved

Table 410: SM_ECC_SET Register

- SM_ECC_AREA1~8 - 0x30~0x4c
  - For 8-bit 512+16-Byte page size, if it is set to enable 256-byte ECC, then register 1 will store the first half page of ECC, register 2 on the other hand, stores the second half page of ECC, other registers are not used.
  - For 8-bit 2048+64-Byte page size, if it's set to enable 256-byte ECC, then register 1~8 will store ECC.

Bit	Name	Default	Description
32:0 (R)	ECC_AREA(x)	32'h0	The hardware ECC result store register.

Table 411: SM_ECC_AREA Register

- SM_WP_EN - 0x50

  This register is used to enable the read-only zone of the NAND Flash. Bit [15:0] is used to enable which zone, the block start address stored in register SM_WP_START_ADDR0~7 are read-only. A total of 8 zones can be set to read-only for all NAND Flash chips. If all 8 zones are set to read-only in chip 0, other chips cannot set read-only zones.

Bit	Name	Default	Description
7:0 (R/W)	ZONE_EN	8'h0	The read-only zone enable, 1 bit for each zone bit[7] zone 7 bit[6] zone 6 ..... bit[1] zone 1 bit[0] zone 0
31:8	-	24'h0	Reserved

Table 412: SM_WP_EN Register

- SM_WP_CS - 0x54

Bit	Name	Default	Description
15:0 (R/W)	CHIP_SET	15'h0	Each 2 bits represent the chip number value of zone 0 to zone 7: CHIP_SET =0, chip 0 CHIP_SET =1, chip1 CHIP_SET =2, chip 2 CHIP_SET =3, chip3 bit[15:14] zone 7 bit[13:12] zone 6 ... bit[7:6] zone 3 bit[1:0] zone 0

Table 413: SM_WP_CS Register

- SM_WP_START_ADDR0~7 - 0x60~0X7C
  This register is used to store the start address value of the read-only block.

Bit	Name	Default	Description
31:0 (R/W)	START_ADDR_VAL	32'hFFFFFFFF	The read-only block start address

Table 414: SM_WP_START_ADDR Register

- SM_WP_END_ADDR0~7 - 0x80~0X9C
  This register is used to store the end address value of the read-only page.

Bit	Name	Default	Description
31:0 (R/W)	END_ADDR_VAL	32'hFFFFFFFF	The read-only block end address

Table 415: SM_WP_END_ADDR Register

- SM_ECC_ERR_STATUS0 - 0x100

Bit	Name	Default	Description
0 (R)	ECC_ERR_FLAG	1'b0	When ECC decoding is completed, error flag interrupt will be indicated as follows: 0: No errors 1: There are errors, SM_ECC_ERR_INDEX needs to be

Bit	Name	Default	Description
			checked, then the error flag will be read
1 (R)	ECC_FAIL_FLAG	1‘b0	1: At least one bank data failed (errors are more than 12/24 bits) and there is not correctable in all bank data. Setting BCH_ECC_reset will clear this bit. If the ECC_ERR_FLAG is 1’b1, then this bit will be disabled.
7:2		6’h0	Reserved.
8 (R)	ECC_ERR0	1’b1	1: There are errors (< 12/24 bits) in the first 1K bytes after decoding. SM_ECC_ERR_NUMS is used to find detail error numbers in the 1Kbyte. 0: There are no errors.
9 (R)	ECC_ERR1	1’b1	1: There are errors in the second 1Kbytes (< 12/24 bits). 0: There are no errors.
10 (R)	ECC_ERR2	1’b1	1: There are errors in the third 1Kbytes (< 12/24 bits), ECC decode needs to be checked. 0: There are no errors.
11 (R)	ECC_ERR3	1’b1	1: There are errors in the fourth 1Kbytes (< 12/24 bits), ECC decode needs to be checked. 0: There are no errors.
12 (R)	ECC_ERR4	1’b1	1: There are errors in the fifth 1K bytes (< 12/24 bits), ECC decode needs to be checked. 0: There are no errors.
13 (R)	ECC_ERR5	1’b1	1: There are errors in the sixth 1K bytes (< 12/24 bits), ECC decode needs to be checked. 0: There are no errors.
14 (R)	ECC_ERR6	1’b1	1: There are errors in the seventh 1K bytes (< 12/24 bits), ECC decode needs to be checked. 0: There are no errors.
15 (R)	ECC_ERR7	1’b1	1: There are errors in the eighth 1K bytes (< 12/24 bits), ECC decode needs to be checked. 0: There are no errors.
16 (R)	ECC_FAIL0	1’b0	Block 0 ECC fail signal 1: There are more than 12/24 bits errors in the 1st 1K bytes. 0: Block 0 ECC does not fail.
17 (R)	ECC_FAIL1	1’b0	Block 1 ECC fail signal 1: There are more than 12/24bits errors in the 2nd 1K bytes.

Bit	Name	Default	Description
			0: Block1 ECC does not fail.
18 (R)	ECC_FAIL2	1'b0	Block 2 ECC fail signal 1: There are more than 12/24bits errors in the 3rd 1K bytes. 0: Block 2 ECC does not fail.
19 (R)	ECC_FAIL3	1'b0	Block 3 ECC fail signal 1: There are more than 12/24bits errors in the 4th 1K bytes 0: Block 3 ECC does not fail.
20 (R)	ECC_FAIL4	1'b0	Block 4 ECC fail signal 1: There are more than 12/24 bits errors in the 5th 1K bytes. 0: Block 4 ECC does not fail.
21 (R)	ECC_FAIL5	1'b0	Block 5 ECC fail signal 1: There are more than 12/24 bits errors in the 6th 1K bytes. 0: Block 5 ECC does not fail.
22	ECC_FAIL6	1'b0	Block 6 ECC fail signal 1: There are more than 12/24 bits errors in the 7th 1K bytes. 0: Block 6 ECC does not fail.
23	ECC_FAIL7	1'b0	Block 7 ECC fail signal 1: There are more than 12/24 bits errors in the first 1K bytes. 0: Block 7 ECC does not fail.
31:24	-	16'h0	Reserved

Table 416: SM_ECC_ERR_STATUS0 Register

- SM_ECC_ERR_NUMS0 - 0x104

Bit	Name	Default	Description
7:0 (R)	ERROR0_BIT_NUM	8'b0	Number of error bits in the 1st 1K bytes
15:8 (R)	ERROR1_BIT_ NUM	8'b0	Number of error bits in the 2nd 1K bytes
23:16 (R)	ERROR2_BIT_ NUM	8'b0	Number of error bits in the 3rd 1Kbytes
31:24 (R)	ERROR3_BIT_ NUM	8'b0	Number of error bits in the 4th 1K bytes

Table 417: SM_ECC_ERR_NUMS0 Register

- SM_ECC_ERR_NUMS1 - 0x108

Bit	Name	Default	Description
7:0 (R)	ERROR4_BIT_NUM	8'b0	Number of error bits in the 5th 1K bytes
15:8 (R)	ERROR5_BIT_ NUM	8'b0	Number of error bits in the 6th 1K bytes
23:16 (R)	ERROR6_BIT_ NUM	8'b0	Number of error bits in the 7th 1Kbytes
31:24 (R)	ERROR7_BIT_ NUM	8'b0	Number of error bits in the 8th 1K bytes

Table 418: SM_ECC_ERR_NUMS1 Register

- SM_FIRST_ERR_INDEX(0~23) - 0x110~0x16C
  Indicate the addresses of errors in the first 1Kbyte.

Bit	Name	Default	Description
(2:0) (R)	CORR_BIT_ADD	3'b0	Correct bit address in a byte: 0: bit0 ... 2: bit2 ... 7: bit7
13:3 (R)	CORR_BYTE_ADD	11'b0	Correct byte address in a page: 0: byte0 3: byte 3 ... 1023: byte 1023
31:14	-	18'h00	Reserved

Table 419: SM_FIRST_ERR_INDEX Register

- SM_(i)_ERR_INDEX(0~23) - 0x170~0x40C
  I = Second, third, fourth, fifth, sixth, seventh, eighth

Bit	Name	Default	Description
(2:0) (R)	CORR_BIT_ADD	3'b0	Correct bit address in a byte: 0: bit0 ... 2: bit2

Bit	Name	Default	Description
			... 7: bit7
13:3 (R)	CORR_BYTE_ADD	11'b0	Correct byte address in a page: 0: byte0 3: byte 3 ... 1023: byte 1023
31:14	-	18'h0	Reserved

Table 420: SM_(i)_ERR_INDEX Register

- SM_2K_ECC_AREA(0~20) -0x510~0x560
  The first, second 1K bytes of the ECC code, total of 21 DWORD.

Bit	Name	Default	Description
31:0 (R)	BCH_ECC_AREA(x)	32'h0	The ECC result store register of the BCH Hardware for 2K page

Table 421: SM_2K_ECC_AREA(0~20) Register

- SM_4K_ECC_AREA(21~41) -0x564~0x5B4
  The third, fourth 1K bytes of the ECC code, total of 42 DWORD including SM_2K_ECC_AREA.

Bit	Name	Default	Description
31:0 (R)	BCH_ECC_AREA(x)	32'h0	The ECC result store register of the BCH Hardware continues for 4K page including SM_2K_ECC_AREA (0~20)

Table 422: SM_4K_ECC_AREA(21~41) Register

- SM_8K_ECC_AREA(42~83) -0x5B8~0x65C
  The fifth, sixth, seventh and eighth 1K bytes of the ECC code, a total of 84 DWORD including SM_2K_ECC_AREA and SM_4K_ECC_AREA.

Bit	Name	Default	Description
31:0 (R)	BCH_ECC_AREA(x)	32'h0	The ECC result store register of the BCH Hardware continues for 8K page including SM_2K_ECC_AREA and SM_4K_ECC_AREA (0~41)

Table 423: SM_8K_ECC_AREA(42~83) Register

- SM_SECT_ECC_AREA -0X700
  The sector info 4 bytes ECC code.

Bit	Name	Default	Description
31:0 (R)	SECT_BCH_ECC_AREA	32'h0	BCH Hard ware ECC result store register

Table 424: SM_SECT_ECC_AREA Register

- SM_SECT_ECC_FLAG –0X704

Bit	Name	Default	Description
0 (R)	SECT_ERR_FLAG	1'b0	1: The sector information area contains errors (including 4-bit errors). Set sector_BCH_ECC_reset will clear the bit.
1 (R)	SECT_ERR_FAULT	1'b0	1: The sector information error correct failed (more than 4 bits). Set sector_BCH_ECC_reset will clear the bit.
4:2 (R)	SECT_ERR_BIT	3'b0	Number bits of errors. 1: One bit error 2: Two bit errors 3: Third bit errors 4: Four bit errors
31:5 (R)	-	27'h0	Reserved

Table 425: SM_SECT_ECC_FLAG Register

- SM_SECT_ECC_ERR_INDEX –0X708

Bit	Name	Default	Description
6:0 (R)	SECT_ERR1_INDEX	1'b0	Correct the first bit address in a byte+byte address in a total of 12 bytes. Bytes add[6:3]+bit add[2:0] next is the same.
7 (R)	-	1'b0	Reserved.
14:8 (R)	SECT_ERR2_INDEX	1'b0	Correct the second bit address in a byte+byte address in a total of 12 bytes.
15 (R)	-	1'b0	Reserved.
22:16 (R)	SECT_ERR3_INDEX	1'b0	Correct the third bit address in a byte a byte+byte address in a total of 12 bytes.

Bit	Name	Default	Description
23 (R)	-	1'b0	Reserved.
30:24 (R)	SECT_ERR4 _INDEX	1'b0	Correct the fourth bit address in a byte a byte+byte address in a total of 12 bytes.
31 (R)	-	1'b0	Reserved.

Table 426: SM_SECT_ECC_ERR_INDEX Register

- SM_DMA_IO_CTRL – 0xF00

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b1	1 for I/O mode, 0 for DMA mode
1 (R/W)	IO_DMA_RW	1'b1	1: Read from peripheral. 0: Write to peripheral.
2 (R/W)	DMA_FLUSH	1'b0	Flushes the DMA receive FIFO in case data_length set on the peripheral side does not match the DWORD size set in DMA control.
3 (R/W)	RW_ENDIAN	1'b0	1: Big endian write/read. 0: Little endian write/read.
4 (R/W)	NO_RDWT	1'b0	1: Writes the command and the address to the NAND Flash but not data.
31:5 (R/W)	-	-	Reserved.

Table 427: SM_DMA_IO_CTRL Register

- SM_DMA_IO_LEN – 0xF04
  The smallest data size is DWORD, i.e. bit1 and bit 0 of this register will be ignored.

Bit	Name	Default	Description
15:0 (R/W)	DATA_LEN	16'h0	The byte length of a DMA or an I/O transfer. If it is set to 0, the I/O or DMA transfer will work continuously until it is stopped.
31:16	-	16'h0	Reserved.

Table 428: SM_DMA_IO_LEN Register

- SM_FIFO_CTRL - 0xF08
  In read mode when FIFO_STATUS_REG [5:2]>=FIFO_CTRL_REG [7:4], then it will cause an interrupt. However in write mode interrupts occur when FIFO_STATUS_REG [5:2]<=FIFO_CTRL_REG [7:4].

Bit	Name	Default	Description
1:0 (R/W)	FIFO_WIDTH	2'b00	Reserved.
7:2 (R/W)	FIFO_THRESHOLD	6'h0	A threshold in byte that triggers an interrupt. An interrupt is triggered when data count in the FIFO reaches the threshold.
31:8	-	24'h0	Reserved.

Table 429: SM_FIFO_CTRL Register

- SM_FIFO_LEVEL_CHK - 0xF0C

Bit	Name	Default	Description
3:0 (R/W)	FIFO_SC	4'h0	FIFO stop check in DWORD length
9:4	-	6'h0	Reserved
13:10 (R/W)	FIFO_LC	4'h0	FIFO low check in DWORD length
19:14	-	6'h0	Reserved
23:20 (R/W)	FIFO_HC	4'h0	FIFO high check in DWORD length
31:24	-	8'h0	Reserved

Table 430: SM_FIFO_LEVEL_CHK Register

- SM_FIFO_OP - 0xF10

Bit	Name	Default	Description
0 (R/W)	FIFO_START	1'b1	It starts the read/write transfer when this bit is declared.
1 (R/W)	FIFO_RESET	1'b0	Set to 1 to stop the FIFO and reset the FIFO internal status, including its relevant interrupt status. Set to 0 in normal operations.
31:2	-	30'h0	Reserved.

Table 431: SM_FIFO_OP Register

- SM_FIFO_STATUS - 0xF14

Bit	Name	Default	Description
5:0 (R)	FIFO_LEVEL	6'h0	The byte count of the valid data in the FIFO. In case FIFO is full, the value of this register is 0, thus FIFO_FULL bit must be concatenated with this value in order to determine the actual data count in the FIFO.
6 (R)	FIFO_FULL	1'b0	FIFO full status. The FIFO is full when read out as 1. This bit is concatenated with FIFO_LEVEL to the actual FIFO data count.
7 (R)	FIFO_EMPTY	1'b1	FIFO empty status. (FIFO_FULL or FIFO_LEVEL) == 0
31: 8	-	24'h0	Reserved

Table 432: SM_FIFO_STATUS Register

- SM_FIFO_DATA - 0xF18

Bit	Name	Default	Description
32:0 (R)	FIFO_DATA	32'h0	The FIFO read/write data register

Table 433: SM_FIFO_DATA Register

SD/MMC/MMC+

Overview

The SiRFatlasV SD/SDIO/MMC4+ Host Controller is a host controller with a 32-bit internal PCI bus interface. This module conforms to the SD Host Controller Standard Specification Draft Version 1.0 and supports SD memory cards, MMC Plus, MMC Mobile cards, and SDIO cards.

Feature List

- Meets SD Host Controller Standard Specification Draft Version 1.0
- Meets SDIO Card Specification Version 1.0.
- Meets SD Memory Card Security Specification version 2.0 and compatible to new SDv2.1 standard
- Meets MMC Specification Version 3.31, 4.0, and 4.2 and compatible to new MMC4.3 and 4.4 standard
- Works simultaneously between two slots
- Supports MMC Plus and MMC Mobile media
- Provides host clock rate between 0 and 50 MHz
- Supports 1-bit and 4-bit SD modes

- Supports 1-bit, 4-bit, and 8-bit MMC modes
- Supports both DMA and Non-DMA operating modes
- Supports 4 slots, each can work independently (excluding slot0 and slot2 share with 8-bit data bus)
  - Slot0 supports 1-bit, 4-bit, and 8-bit modes (8-bit data bus shares with slot2, and NAND flash interface)
  - Slot1 supports 1-bit and 4-bit modes
  - Slot2 supports 1-bit, 4-bit, and 8-bit modes (8-bit data bus shares with slot0, and NAND flash interface)
  - Slot3 supports 1-bit and 4-bit modes

Internal Feature List

- Interrupt controller

  The SD/SDIO host controller generates interrupt to the RISC.

- SDIO/SD host controller

  The SDIO/SD Host Controller is comprised of:

  - A host interface
  - SD/SDIO controller registers
  - A bus monitor
  - A clock generator
  - A CRC generator
  - A checker (CRC7 and CRC16).

  The SD/SDIO controller registers are programmed by the host driver. Interrupts are generated to the host based on the values set in the Interrupt Status Enable (SD_INT_STA_ENA) and Interrupt Signal Enable registers (SD_INT_SIG_EN). The bus monitor checks for violations that occur in the SD bus and timeout conditions. The clock generator generates the SD clock based on the value programmed by the host driver in the Host Control Register (SD_CLK_TIMEOUT_SRST_CTRL). The CRC7 and CRC16 generators calculate the CRC for command and data respectively for sending the CRC to the SD/SDIO card. The CRC7 and CRC16 checkers check for any CRC errors in the Response and Data register sent by the SD/ SDIO card.

- DATA FIFO

  The SD/SDIO Host Controller uses two 512-byte dual-port RAMs for performing data transactions (One for read transactions and one for write transactions). Each slot requires two 512-byte buffers. If the host controller cannot accept data coming from the SD card, it should stop the clock in order to prevent data coming from the card. This is why the block length can only be set to 512 bytes.

- DAT[0-7] control logic

  The DAT[0-7] control logic block transmits data through the data line during the write transaction and during read transaction in which data is being received through the data line. The interrupt generated from the SD/SDIO card is detected.

- Command control logic

  The command control logic block sends the command using the command line and receives responses from an SD/SDIO card.

- Power control

The SD/SDIO host controller supplies the SD bus power based on the value programmed in the power control register. The PCI Host Driver is responsible for turning on and off the SD bus power supply according to the card OCR. If the SD bus power is set to 1 in the Power Control Register, then the host controller should supply power to the card.

Pin Description

External Pin Descriptions

The following table lists the pins used in the SD interface and their functions. The SD interface pins are multiplexed with other devices.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_DF_WP_B	Output	NADN Flash Interface	decided by boot mode[2]	Output low	SD_slot0 card power supply control
X_DF_CLE	Output	NADN Flash Interface	decided by boot mode	Output low	SD_slot0 bus clock
X_DF_ALE	Bi-directional	NADN Flash Interface	decided by boot mode	Input	SD_slot0 bus command signal
X_DF_AD[0]	Bi-directional	NADN Flash Interface and SD slot2 data bus	decided by boot mode	Input	SD_slot0 bus data signal d0
X_DF_AD[1]	Bi-directional	NADN Flash Interface and SD slot2 data bus	decided by boot mode	Input	SD_slot0 bus data signal d1
X_DF_AD[2]	Bi-directional	NADN Flash Interface and SD slot2 data bus	decided by boot mode	Input	SD_slot0 bus data signal d2
X_DF_AD[3]	Bi-directional	NADN Flash Interface and SD slot2 data bus	decided by boot mode	Input	SD_slot0 bus data signal d3
X_DF_AD[4]	Bi-directional	NADN Flash Interface and SD slot2 data bus	decided by boot mode	Input	SD_slot0 bus data signal d4
X_DF_AD[5]	Bi-directional	NADN Flash Interface and SD slot2 data bus	decided by boot mode	Input	SD_slot0 bus data signal d5
X_DF_AD[6]	Bi-directional	NADN Flash Interface and SD slot2 data bus	decided by boot mode	Input	SD_slot0 bus data signal d6
X_DF_AD[7]	Bi-directional	NADN Flash Interface and SD	decided by boot mode	Input	SD_slot0 bus data signal d7

[2] In SD slot0 slot2 bus, if the boot mode select to Embedded ROM SD BOOT, this pin will be select SD Slot0 bus Function. If the boot mode select to.NAND Boot or Embedded ROM Nand Boot , this pin is selected to NAND Flash Interface function.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
		slot2 data bus			
X_SD_CD_B_1	Input	GPIO Group1[7]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 card detected pin
X_SD_VCC_ON_1	Output	GPIO Group1[8]	GPIO	Input, pull-down (pull function only for GPIO)	SD_slot1 card power supply control
X_SD_WP_B_1	Input	GPIO Group1[9]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 card write detected pin
X_SD_CLK_1	Bi-directional	GPIO Group1[10]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 bus clock
X_SD_CMD_1	Bi-directional	GPIO Group1[11]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 bus command signal
X_SD_DAT_1[0]	Bi-directional	GPIO Group1[12]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 bus data signal d0
X_SD_DAT_1[1]	Bi-directional	GPIO Group1[13]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 bus data signal d1
X_SD_DAT_1[2]	Bi-directional	GPIO Group1[14]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 bus data signal d2
X_SD_DAT_1[3]	Bi-directional	GPIO Group1[15]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot1 bus data signal d3
X_DF_WE_B	Output	NADN Flash Interface	decided by boot mode	Output Low	SD_slot2 bus clock
X_DF_RE_B	Bi-directional	NADN Flash Interface	decided by boot mode	Input	SD_slot2 bus command signal
X_DF_AD[0]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode[3]	Input	SD_slot2 bus data signal d0
X_DF_AD[1]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode	Input	SD_slot2 bus data signal d1

[3] In SD slot0,slot2 data[7:0] ,if the boot mode select by Embedded ROM SD BOOT, this pin will be select SD Slot0 bus Function not SD slot2 function. if you need select to SD slot2 function, software need set the rRSC_PIN_MUX register in chip.

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_DF_AD[2]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode	Input	SD_slot2 bus data signal d2
X_DF_AD[3]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode	Input	SD_slot2 bus data signal d3
X_DF_AD[4]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode	Input	SD_slot2 bus data signal d4
X_DF_AD[5]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode	Input	SD_slot2 bus data signal d5
X_DF_AD[6]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode	Input	SD_slot2 bus data signal d6
X_DF_AD[7]	Bi-directional	NADN Flash Interface and SD slot0 data bus	decided by boot mode	Input	SD_slot2 bus data signal d7
X_SD_CLK_3	Bi-directional	GPIO Group1[2]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot3 bus clock
X_SD_CMD_3	Bi-directional	GPIO Group1[3]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot3 bus command signal
X_SD_DAT_3[0]	Bi-directional	spi_en_0 and GPIO Group1[28]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot3 bus data signal d0
X_SD_DAT_3[1]	Bi-directional	spi_clk_0 and GPIO Group1[29]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot3 bus data signal d1
X_SD_DAT_3[2]	Bi-directional	spi_din_0 and GPIO Group1[0]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot3 bus data signal d2
X_SD_DAT_3[3]	Bi-directional	spi_dout_0 and GPIO Group1[1]	GPIO	Input, pull-up (pull function only for GPIO)	SD_slot3 bus data signal d3

Table 434: SDIO External Pin Description

Functional Descriptions

Block Diagram

The SiRFatlasV SD/SDIO/MMC4+ Host Controller handles SDIO/SD Protocol at the transmit level. It packs data, adds Cyclic Redundancy Check (CRC), starts or terminates bits, and checks for the correctness of the transaction format. The SD Mode wide bus width is also supported.

The SiRFatlasV SD/SDIO/MMC4+ Host Controller provides both programmed I/O and DMA data transfer methods. In the programmed I/O, the RISC transfer data uses the Buffer Data Port Register. The host controller support for DMA can be determined by checking the DMA support in the Capabilities register. The DMA allows a peripheral to read or write memory without CPU intervention. The system address register points to the first data address, then the data will be accessed sequentially from that address.

This module conforms to SDIO Specification and SD memory card physical layer specifications.

Power consumption for the system can be maintained to the minimum using gated clock control.

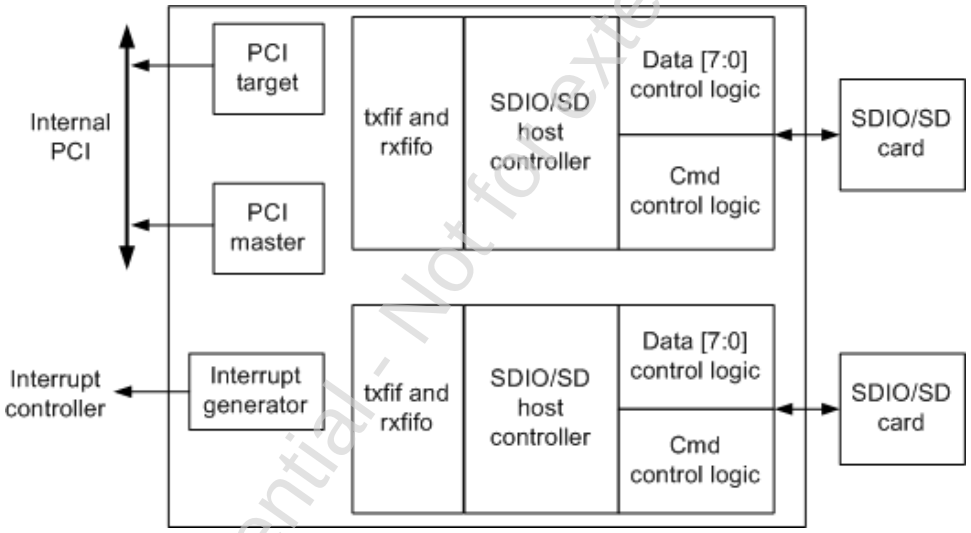


Figure 68: SD Host Block Diagram

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
BASE address of SD_solt0	0x5600_0000
BASE address of SD_solt1	0x5610_0000
BASE address of SD_solt2	0x5620_0000
BASE address of SD_solt3	0x5630_0000

Access Type	Address Mapping
SD_PCI_CONFIGE_BASE01	0x5770_0000
SD_PCI_CONFIGE_BASE23	0x5780_0000

Table 435: SD Interface Address Mapping

Register Mapping

RISC Address <11:0>	Bit	Register	Default	Description
0x010c	15:8 (R/W)	SD_PCI_LT_TIME	8'hff	The SD master handles the PCI bus maximum clock after getting a grant (test only).

Table 436: SD_PCI_CONFIGE_REG

NOTE – 0x00FC (SD_SLOT_INT_STAUTS) is the same register in both slot0/slot1 and slot2/slot3. All registers access of the SD host controller by RISC is memory access.

RISC Address <11:0>	Register	Description
0x0000	SD_SYS_ADDR	System memory address
0x0004	SD_BLK_PARAMETER	Block size/count register
0x0008	SD_ARGUMENT	Command argument register
0x000C	SD_CMD_TRAN_MODE	Transfer control register
0x0010	SD_REPONSE_REG0	SD card response register 0
0x0014	SD_REPONSE_REG1	SD card response register 1
0x0018	SD_REPONSE_REG2	SD card response register 2
0x001C	SD_REPONSE_REG3	SD card response register 3
0x0020	SD_BUF_DATA	Transfer data buffer register
0x0024	SD_PRESENT_STATE	SD host controller state register
0x0028	SD_HOST_PWR_BCG_WKP_CTRL	SD host controller control register 0
0x002C	SD_CLK_TIMEOUT_SRST_CTRL	SD host controller control register 1
0x0030	SD_INT_STATUS	SD interrupt status register
0x0034	SD_INT_STA_ENA	SD interrupt status enable register
0x0038	SD_INT_SIG_ENA	SD interrupt signal enable register
0x003C	SD_AUTO_CMD12_ERR_STA	Automatically command 12 error status register

RISC Address <11:0>	Register	Description
0x0040	SD_CAPABILITY_REG	SD host capabilities register
0x0044	-	Reserved
0x0048	SD_MAX_CUR_CAPABILITY	Maximum current capability
0x004C	SD_CLK_DELAY_SETTING	SD bus clock delay register
0x0050~0xF8	-	Reserved
0x00FC	SD_SLOT_INT_STAUTS	Slot interrupt status and version register
Others	-	Reserved

Table 437: SDIO Host Controller SLOT0/SLOT1/SLOT2/SLOT3 Register Mapping

There is a standard register defined for each slot which is independently controlled. This enables support for combinations of high-speed, full-speed and low-speed cards in regards to SD clock frequencies.

The host driver must determine the number of slots and base pointers to each slot's standard register set using the system bus or vendor-specific methods. Offsets from 0F0h to 0FFh are reserved for the common register area that defines information for slot control and common status. The common register area is accessible from any slot's register set. This allows software to independently control each slot, since it has access to the Slot Interrupt Status register and the Host Controller Version register from each register set.

The register set is 256 bytes in size. For slot0/slot1, slot2/slot3 controllers, one register set is assigned per slot, but the registers at offsets 0F0h - 0FFh are assigned as a common area. These registers contain the same values for each slot's register set.

- Interrupt signal for each slot:

  These status bits indicate the logical OR of the interrupt signal and the walk-up signal for each slot. Two bits can be defined. If one interrupt signal is associated with multiple slots, then the host driver will be able to know which interrupt is being generated by reading these status bits. By a power-on reset or by setting software reset for all, the interrupt signal will be de-asserted and its status will read 00h.

  - Bit 00 slot0/slot2
  - Bit 01 slot1/slot3
  - Others reserved
- Card detection:

  Only slot1 support Card detetion. The slot0, slot2 and slot3 have not the card detection pin.

  When a card is inserted or removed in/from slot1, the corresponding interrupt will be set in the interrupt status register. This will go as an interrupt to the host driver. This is also reflected in bit1 of the slot interrupt status register (bit1 = 0). Upon receiving the interrupt, the host reads the slot interrupt status register to locate the interrupt source (slot1), then it reads the corresponding Interrupt status register (slot1 interrupt status register) to find the interrupt source (card inserted, card removal). But for slot0, slot2 and slot3, the card status is fixed to 1b'1, no matter whether the card is in the slot or not.

Register Descriptions

- SDIO System Address Register (SD_SYS_ADDR) – 0x0000

Bit	Name	Default	Description
31:0 (R/W)	DMA_MEM_ADDR	32'h0	This register contains the system memory address for a DMA transfer. When the Host Controller (HC) stops a DMA transfer, it points to the system address of the next contiguous data position. It can be accessed only if no transaction is being executed (i.e. after a transaction has stopped). Read operations during transfer will return an invalid value. The Host Driver (HD) will initialize this register before starting another DMA transaction. After DMA is stopped, the next system address of the next contiguous data position will be read from this register. The DMA transfer waits at the boundary specified by HOST_DMA_BUF_SIZE in the SD_BLK_PARAMETER register. The HC generates DMA_INT to request an update to this register. The HD then sets the next system address of the next data position to this register. When most upper bytes of this register (0x003) are set, HC will restart the DMA transfer. When restarting DMA by the resume command or by setting the Continue Request in the Block Gap Control register, the HC starts at the next contiguous address stored in this register.

Table 438: SDIO System Address Register

- SDIO Block Size/Count Register (SD_BLK_PARAMETER) – 0x0004

Bit	Name	Default	Description
{15,11:0} (R/W)	BLK_SIZE	13'h0	This register specifies the block size for block data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53. It can be accessed only if no transaction is being executed (i.e. after a transaction has stopped). Read operations during transfer will return an invalid value and write operations will be ignored. 0000h: No data transfer 0001h: 1 byte 0002h: 2 bytes 0003h: 3 bytes ..... 01FFh: 511 bytes 0200h: 512 bytes ..... 0800h: 2048 Bytes 1000h: 4096 Bytes Data FIFO only supports up to 512 bytes of memory space.

Bit	Name	Default	Description
14:12 (R/W)	HOST_DMA_BUF_SIZE	3'h0	To perform long DMA transfers, the System Address register shall be updated at every system boundary during the transfer. These bits specify the size of the contiguous buffer in the system memory. The DMA transfer will wait at the boundary specified by these fields and the HC will then generate DMA_INT to request HD to update the System Address register. These bits shall be supported when DMA_SUP (DMA Support) in the Capabilities register is set to 1. This function will be active when DMA_EN in the Transfer Control register is set to 1. 000b: 4KB (detects A11 carry out) 001b: 8KB (detects A12 carry out) 010b: 16KB (detects A13 carry out) 011b: 32KB (detects A14 carry out) 100b: 64KB (detects A15 carry out) 101b: 128KB (detects A16 carry out) 110b: 256KB (detects A17 carry out) 111b: 512KB (detects A18 carry out)
31:16 (R/W)	BLK_CNT	16'h0	This bit is enabled when BLK_CNT_EN in the Transfer Control register is set to 1 and is valid only for multiple block transfers. The HC decrements the block count after each block transfer and stops when the count reaches zero. It can be accessed only if no transaction is being executed (i.e. after a transaction has stopped). Read operations during transfer will return an invalid value and write operations shall be ignored. When saving transfer context as a result of Suspend command, the number of blocks to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, the HD will restore the previously save block count. 0x0000: Stop Count 0x0001: 1 block 0x0002: 2 blocks ..... 0xFFFF – 65535 blocks

Table 439: SDIO Block Size/Count Register

- SDIO Command Argument Register (SD_ARGUMENT) – 0x0008

Bit	Name	Default	Description
31:0 (R/W)	CMD_ARG	32'h0	The SD Command Argument is specified as bit<39:8> of Command-Format.

Table 440: SDIO Command Argument Register

- SDIO Transfer Control Register (SD_CMD_TRAN_MODE) – 0x000C

Bit	Name	Default	Description
0 (R/W)	DMA_EN	1'b0	DMA can be enabled only if DMA_SUP (DMA Support) bit in the Capabilities register is set. If this bit is set to 1, then a DMA operation shall begin when the HD writes to the upper byte of the Command register (00Fh). 0: Disable 1: Enable
1 (R/W)	BLK_CNT_EN	1'b0	This bit is used to enable the BLK_CNT in the SD_BLK_PARAMETER register, which is only relevant for multiple block transfers. When this bit is set to 0, the Block Count register will be disabled, which is useful for executing an infinite transfer. 0: Disable 1: Enable
2 (R/W)	AUTO_CMD12_EN	1'b0	Multiple block transfers for memory that requires CMD12 to stop the transaction. When this bit is set to 1, HC shall then issue CMD12 automatically when the last block transfer is completed. HD shall not set this bit to issue commands that do not require CMD12 to stop data transfer. 0: Disable 1: Enable
3	-	-	Reserved
4 (R/W)	DAT_TRAN_DIR_SEL	1'b0	This bit defines the direction of DAT line data transfers. 0: Write (Host to Card) 1: Read (Card to Host)
5 (R/W)	MULT_BLK_SEL	1'h0	This bit enables multiple block DAT line data transfers. 0: Single Block 1: Multiple Block
6 (R/W)	-	1'h0	Reserved must set 1'b0;
15:7	-	-	Reserved
17:16 (R/W)	RES_TYPE_SEL	2'h0	Response Type Select: 00: No Response 01: Response length 136 10: Response length 48 11: Response length 48 check busy after response
18	-	1'b0	Reserved
19	CMD_CRC_CHK_EN	1'b0	If this bit is set to 1, then HC shall check the CRC field in the response. If an error is detected, then it will be

Bit	Name	Default	Description
(R/W)			reported as a Command CRC error. If this bit is set to 0, then the CRC field will not be checked. 0: Disable 1: Enable
20 (R/W)	CMD_IND_CHK_EN	1'b0	If this bit is set to 1, then HC shall check the index field in the response to see if it contains the same value as the command index. If it doesn't, then it will be reported as a Command Index error. If this bit is set to 0, then the Index field will not be checked. 0: Disable 1: Enable
21 (R/W)	DAT_PRE_SEL	1'h0	This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. This bit should be set to 0 for the following reasons: • Commands using only CMD line (ex. CMD52) • Commands with no data transfer but using busy signal on DAT<0> line (R1b or R5b ex. CMD38) • Resume Command 0: No Data Present 1: Data Present
23:22 (R/W)	CMD_TYPE	2'h0	There are three types of special commands: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands. **Suspend Command**: If the Suspend command is executed successfully, then HC shall assume the SD bus has been released and that it is possible to issue the next command that uses the DAT line. HC shall then de-assert Read Wait for read transactions and stop checking busy for write transactions. Then the interrupt cycle will start in 4-bit mode. If the Suspend command fails, then HC will maintain its current state and restart the transfer by setting Continue Request in the Block Gap Control register. **Resume Command:** HD re-starts the data transfer by restoring the registers in the range of 000-00Dh. HC shall check for busy signals before starting the write transfers. **Abort Command:** If this command is set when executing a read transfer, then HC shall stop reading to the buffer. If this command is set when executing a write transfer, then HC shall stop driving the DAT line. After issuing an Abort command, HD should issue a software reset: 00b: Normal

Bit	Name	Default	Description
			01b: Suspend 10b: Resume 11b: Abort
29:24 (R/W)	CMD_INDEX	6'h0	This bit should be set to the command number (CMD0-63, ACMD0-63
31:30	-	2'h0	Reserved

Table 441: SDIO Transfer Control Register

Multi/Single Block Select	Block Count Enable	Block Count Function	Transfer Type
0	Don't Care	Don't Care	Single transfer
1	0	Don't Care	Infinite transfer
1	1	Not 0	Multiple transfer
1	1	0	Stop multiple transfer

Table 442: Definition of Transfer Type

Response type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R6, R5
11	1	1	R1b, R5b

Table 443: Definition of Response Type

- SDIO Card Response Register 0~3 (SD_CARD_RESP_x) – 0x0010~0x001C

Bit	Name	Default	Description
127:0 (R)	RESP<127:0>	128'h0	The following table describes the mapping of command responses from the SD bus to this register for each response type. In the table, R<> refers to a bit range within the response data as transmitted on the SD bus, whereas RESP<> refers to a bit range within the Response register

Table 444: SDIO Card Response Register

Response Type	Description	Response Field	Response Register
R1, R1b (normal response)	Card Status	R<39:8>	RESP<31:0>
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R<39:8>	RESP<127:96>
R2 (CID, CSD Register)	CID or CSD register included	R<127:8>	RESP<119:0>
R3 (OCR Register)	OCR Register for memory	R<39:8>	RESP<31:0>
R4 (OCR Register)	OCR Register for I/O	R<39:8>	RESP<31:0>
R5, R5b	SDIO Response	R<39:8>	RESP<31:0>
R6 (Published RCA response)	New published RCA[31:16]	R<39:8>	RESP<31:0>

Table 445: Response Bit Definition for Each Response Type

- SDIO Buffer Data Port Register (SD_BUF_DATA) – 0x0020

Bit	Name	Default	Description
31:0 (R/W)	BUF_DATA_PORT	32'h0	The Host Controller buffer can be accessed through this 32-bit data port register

Table 446: SDIO Buffer Data Port Register

- SDIO Current State Register (SD_PRESENT_STATE) – 0x0024

Bit	Name	Default	Description
0 (R)	CMD_INHABIT (CMD)	1'b0	If this bit is 0, then this means the command line is not in use and HC can issue an SD command using the CMD line. This bit will be set immediately after the SD_CMD_TRAN_MODE (0x00C) is written. This bit is cleared when the command response is received. Even if the CMD_INHABIT (DAT) is set to 1, commands using only the CMD line can still be issued when this bit is 0. Changing from 1 to 0 will generate a CMD_END interrupt in the SD_INT_STATUS register. If HC cannot issue the command because of a command conflict error or AUTO_CMD12_ERR, then this bit shall remain 1 and CMD_END will not be set. Status issuing Auto CMD12 is not read from this bit.
1 (R)	CMD_INHABIT (DAT)	1'b0	This status bit is generated if either the DAT_LINE_ACTIVE or the RD_TRAN_ACTIVE register is set to 1. When this bit is 0, it indicates that HC can issue the next SD command. Commands with busy signals belong to CMD_INHIBIT (DAT) (ex. R1b, R5b type). Changing from 1 to 0 will generate a TRAN_END interrupt in the SD_INT_STATUS

Bit	Name	Default	Description
			register. The SD host driver can save registers in the range of 000-00Dh for a suspend transaction after this bit has changed from 1 to 0. 1: Cannot issue commands which use the DAT line 0: Can issue commands which use the DAT line
2 (R)	DAT_LINE_ACTIVE	1'b0	This bit indicates whether one of the DAT lines on the SD bus is in use: 1: DAT line is active 0: DAT line is inactive
7:3 (R)	-	-	Reserved
8 (R)	WT_TRAN_ACTIVE	1'b0	This status indicates whether a write transfer is active. If this bit is set to 0, then it means no valid write data exists in the HC. This bit is set in either of the following cases: • After the end bit of a write command. • When writing a 1 to CONTINUE_REQ in the SD_HOST_PWR_BCG_WKP_CTRL register to restart a write transfer. This bit is cleared in either of the following cases: • After getting the CRC status of the last data block specified by the transfer count (Single or Multiple). • After getting a CRC status of any block where data transmission is about to be stopped by STOP_AT_BLK_GAP_REQ. During a write transaction, a BLK_GAP_EVT interrupt will be generated when this bit is changed to 0, as a result of the STOP_AT_BLK_GAP_REQ being set. This status is useful for the HD in determining when to issue commands during write busy: 1: Transferring data 0: No valid data

Bit	Name	Default	Description
9 (R)	RD_TRAN_ACTIVE	1'b0	This status is used for detecting completion of a read transfer. This bit is set to 1 for either of the following conditions: • After the end bit of the read command. • When writing a 1 to CONTINUE_REQ in the SD_HOST_PWR_BCG_WKP_CTRL register to restart a read transfer. This bit is cleared to 0 for either of the following conditions: • When the last data block specified by the block length is transferred to the system. • When all valid data blocks have been transferred to the system and no current block transfers are being sent. As a result of the STOP_AT_BLK_GAP_REQ set to 1, a TRAN_END interrupt will be generated when this bit changes to 0. 1: Transferring data 0: No valid data
10 (R)	BUFF_WT_EN	1'b0	This status is used for I/O write transfers. This read-only flag indicates whether space is available for write data. If this bit is 1, then data can be written to the buffer. A change of this bit from 1 to 0 will occur when all block data is written to the buffer. Whereas a change of this bit from 0 to 1 will occur when the top of the block data can be written to the buffer and generate the BUFF_WT_RDY Interrupt. 0: Write Disable 1: Write Enable
11 (R)	BUFF_RD_EN	1'b0	This status is used for I/O read transfers. This read-only flag indicates that valid data exists in the host side buffer status. If this bit is 1, then it means a readable data exists in the buffer. A change of this bit from 1 to 0 will occur when all block data is read from the buffer. Whereas a change of this bit from 0 to 1 will occur when all block data is ready in the buffer and generate the BUFF_RD_RDY Interrupt. 0: Read Disable 1: Read Enable
15:12 (R)	-	-	Reserved
16 (R)	CARD_INSERTED	1'b0	This bit indicates whether a card has been inserted. Changing from 0 to 1 will generate a CARD_INSERT interrupt in the Normal Interrupt Status register and changing from 1 to 0 will generate a CARD_REMOVE Interrupt in the Normal Interrupt Status register. The SOFT_RST_ALL in the SD_CLK_TIMEOUT_SRST_CTRL register will not affect this bit. If a Card is removed while its power is on and its clock is still oscillating, then HC will clear SD_BUS_PWR and SD_CLK_EN in the

Bit	Name	Default	Description
			SD_HOST_PWR_BCG_WKP_CTRL register. In addition, HD will clear HC using the SOFT_RST_ALL register. The card detect is active regardless of the SD bus power. 0: Reset or debouching or no cards detected 1: Card is inserted
17 (R)	CARD_STATE_STABLE	1'b0	This bit is used for testing purposes. If it is 0, the CARD_DETECT_PIN_LEVEL will not be stable. If this bit is set to 1, then it means the Card Detect Pin Level is stable. The SOFT_RST_ALL in the SD_CLK_TIMEOUT_SRST_CTRL Register will not affect this bit. 0: Reset of debouching 1: No cards detected or inserted
18 (R)	CARD_DETECT_PIN_LEVEL	1'b0	This bit reflects the inverse value of the SD Card Detect pin (SDCD#): 0: No Cards present (SDCD# = 1) 1: Card is present (SDCD# = 0)
19 (R)	Write Protect Switch Pin Level	1'b0	The Write Protect Switch is supported for memory and combo cards. This bit reflects the SDWP# pin. 0: Write protected (SDWP# = 1) 1: Write enabled (SDWP# = 0)
23:20 (R)	DAT_LINE_LEVEL	4'b0	This status is used to check DAT line level to recover from errors and for debugging. This is especially useful for detecting the busy signal level from DAT<0>. D23 – DAT<3> D22 – DAT<2> D21 – DAT<1> D20 – DAT<0>
24 (R)	CMD_LINE_LEVEL	1'b0	This status is used to check command line level to recover from errors and for debugging
28:25 (R)	DAT_LINE_LEVEL	4'b0	This status is used to check DAT line level to recover from errors and for debugging. This is especially useful in detecting the busy signal level from DAT<0>. D28 – DAT<7> D27 – DAT<6> D26 – DAT<5> D25 – DAT<4>
31:29	-	-	Reserved

Table 447: SDIO Current State Register

NOTE – DAT line active indicates whether one of the DAT lines on the SD bus is in use.

In read transactions this register indicates whether a read transfer is executing on the SD bus. Changing this value from 1 to 0 between data blocks will generate a BLK_GAP_EVT interrupt in the SD_INT_STATUS register.

This bit shall be set in either of the following cases:

- After the end bit of the read command
- When writing a 1 to CONTINUE_REQ in the SD_HOST_PWR_BCG_WKP_CTRL register to restart a read transfer

This bit shall be cleared in either of the following cases:

- When the end bit of the last data block is sent from the SD bus to the HC
- When beginning a wait read transfer at a stop of the block gap initiated by a STOP_AT_BLK_GAP_REQ.

HC should wait at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), then HC can wait for the current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the suspending/resuming function.

In write transactions, this register indicates that a write transfer is executing on the SD bus. Changing this value from 1 to 0 will generate a TRAN_END interrupt in the SD_INT_STATUS register.

This bit shall be set in either of the following cases:

- After the end of the write command
- When writing 1 to CONTINUE_REQ in the SD_HOST_PWR_BCG_WKP_CTRL register to continue a write transfer

This bit shall be cleared in either of the following cases:

- When the SD card releases write busy of the last data block, HC should also detect if the output is not busy. If SD card does not drive busy signal for eight SD clocks, HC will then consider the card driven as “Not Busy”
- When the SD card releases write busy prior to waiting for write transfer as a result of STOP_AT_BLK_GAP_REQ.

NOTE – HD can issue cmd0, cmd12, cmd13 (for memory) and cmd52 (for SDIO) when the DAT lines are busy during data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands should also be issued when Command Inhibit (DAT) is set to zero.

- SDIO Host Control Register 0 (SD_HOST_PWR_BCG_WKP_CTRL) – 0x0028

Bit	Name	Default	Description
0 (R/W)	RESEVED	1'b0	Must be 0.
1 (R/W)	DAT_TRAN_WIDTH	1'b0	This bit selects the data width of the HC. The HD should select it to match the data width of the SD card: 1: 4-bit mode 0: 1-bit mode Bit[1] and bit[3] cannot be both 1 simultaneously.
2 (R/W)	HIGH_SPEED_EN	1'b0	This bit is optional. Before setting this bit, HD should check HIGH_SPEED_SUPPORT in the Capabilities register. If this bit is set to 0 (default), then HC will output CMD and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, then HC will output CMD and DAT lines at the rising edge of the SD clock (up to 50 MHz). 1: High speed mode 0: Normal speed mode
3 (R/w)	-	1'b0	1: 8-bit mode 0: 1-bit mode Bit[1] and bit[3] cannot be both 1 simultaneously.
7:4 (R)	-	-	Reserved
8 (R/W)	SD_BUS_PWR	1'b0	If HC detects No Card State, then this bit will be cleared. 1: Power on 0: Power off Set this bit only as a controller signal. It needs an external power supply for the SD card.
11:9 (R/W)	SD_BUS_VOL_SEL	3'b000	Reserved (not supported)
15:12 (R)	-	-	Reserved
16 (R/W)	STOP_AT_BLK_GAP_REQ	1'b0	This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1 which indicates a transfer completion the HD, should this bit be left to 1. Clearing both the STOP_AT_BLK_GAP_REQ and CONTINUE_REQ registers will not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. HC should honor STOP_AT_BLK_GAP_REQ for write transfers, but for read transfers, it will require the SD card to support Read Wait. Therefore the HD shall not set this bit during read transfers unless the SD card supports Read Wait and has set RD_WAIT_CTRL to 1. In the case of write

Bit	Name	Default	Description
			transfers in which the HD writes data to the SD_BUF_DATA register, HD shall set this bit after all block data is written. If this bit is set to 1, then HD will not write data to the buffer data port register. This bit affects RD_TRAN_ACTIVE, WT_TRAN_ACTIVE, DAT_LINE_ACTIVE and CMD_INHIBIT (DAT) in the SD_PRESENT_STATE register. 1: Stop 0: Transfer
17 (R/W)	CONTINUE_REQ	1'b0	This bit is used to restart a transaction that was stopped after using STOP_AT_BLK_GAP_REQ. To cancel stop at the block gap, set STOP_AT_BLK_GAP_REQ to 0 and set it to restart the transfer. HC will automatically clear this bit in either of the following cases: • In a read transaction, DAT_LINE_ACTIVE will change from 0 to 1 when the read transaction restarts. • In a write transaction, WR_TRAN_ACTIVE will change from 0 to 1 when the write transaction restarts. It is unnecessary for the host driver to set this bit to 0. If STOP_AT_BLK_GAP_REQ is set to 1, any write attempts to this bit will be ignored. 1: Restart 0: Ignored
18 (R/W)	RD_WAIT_CTRL	1'b0	The read wait function is optional for SDIO cards. If the card supports read wait, then set this bit to enable use of the read wait protocol in order to stop read data using DAT<2> line. Otherwise HC will have to stop the SD clock to hold read data, which restricts commands generation. When the HD detects an SD card insertion, it will then set this bit according to the CCCR of the SDIO card. If the card does not support read wait, then this bit will never be set to 1, otherwise DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported. 1: Enable Read Wait Control 0: Disable Read Wait Control
19 (R/W)	INT_AT_BLK_GAP	1'b0	This bit is valid only in 4-bit mode of the SDIO card; it selects a sample point in the interrupt cycle. Setting to 1 will enable interrupt detection at the block gap for multiple block transfers. If the SD card cannot signal an interrupt during a multiple block transfer, then this bit will be set to 0. When HD detects an SD card insertion, it will then set this bit according to the CCCR of the SDIO card.
23:20 (R)	-	-	Reserved
24 (R/W)	-	1'b0	Reserved
25 (R/W)	-	1'b0	Reserved

Bit	Name	Default	Description
26 (R/W)	-	1'b0	Reserved
31:27 (R)	-	-	Reserved

Table 448: SDIO Host Control Register 0

There are three situations in which the transfer needs to be restarted after it stops at the block gap. Whichever case is appropriate depends on whether the HC issues a Suspend command or the SD card accepts the Suspend command:

- If HD does not issue a Suspend command, then CONTINUE_REQ should be used to restart the transfer
- If HD issues a Suspend command and the SD card accepts it, then a Resume command should be used to restart the transfer
- If HD issues a Suspend command and the SD card does not accept it, then CONTINUE_REQ should be used to restart the transfer

Any time STOP_AT_BLK_GAP_REQ stops the data transfer, HD will wait for TRAN_END (in the SD_INT_STATUS register) before attempting to restart the transfer. When restarting data transfer by CONTINUE_REQ, HD will clear STOP_AT_BLK_GAP_REQ prior to or in parallel of the data transfer.

NOTE – The hardware driver should maintain its current voltage on the SD bus by setting SD_BUS_PWR to 1 in this register when the wakeup event via card interrupt is desired.

- SDIO Host Control Register 1 (SD_CLK_TIMEOUT_SRST_CTRL) – 0x002C

  Upon HC initialization, HD should set SDCLK_FREQ_SEL based on the Capabilities register.

Bit	Name	Default	Description
0 (R/W)	INT_CLK_EN	1'b0	This bit is set to 0 when the HD is not using HC or when HC is waiting for a wake-up event. HC will stop its internal clock to go into very low power state. In this case, registers should still be able to be read and written. Clock starts to oscillate when this bit is set to 1. When clock oscillation is stable, HC will set INT_CLK_STABLE in this register to 1. This bit shall not affect card detection. 1: Oscillate 0: Stop
1 (R/W)	INT_CLK_STABLE	1'b0	This bit is set to 1 when the SD clock is stable after writing 1 to INT_CLK_EN in this register. The SD host driver should wait to set SDCLK_EN until this bit is set to 1. This is useful when using PLL for a clock oscillator that requires setup time. 1: Ready 0: Not ready

Bit	Name	Default	Description
2 (R/W)	SDCLK_EN	1'b0	HC will stop SDCLK when writing this bit to 0. SDCLK_FREQ_SEL can be changed when this bit is 0. HC will maintain the same clock frequency until SDCLK is stopped (stop at SDCLK = 0). If HC detects the No Card state, this bit will be cleared. 1: Enable 0: Disable
7:3	-	-	Reserved
15:8 (R/W)	SDCLK_FREQ_SEL	8'h0	This register is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather it holds the divider of the BASE_CLK_FREQ for the SD clock in the Capabilities register. Only the following settings are allowed: 0x80: Base clock divided by 256 0x40: Base clock divided by 128 0x20: Base clock divided by 64 0x10: Base clock divided by 32 0x08: Base clock divided by 16 0x04: Base clock divided by 8 0x02: Base clock divided by 4 0x01: Base clock divided by 2 0x00: Base clock (10MHz-63MHz) Setting 0x00 specifies the highest frequency of the SD Clock. When setting multiple bits, the most significant bit will be used as the divider. However it is not recommended to set multiple bits. The two default divider values can be calculated by the frequency defined by BASE_CLK_FREQ for the SD clock in the Capabilities register. • 25 MHz divider value • 400 KHz divider value The frequency of the SDCLK is set by the following formula: Clock Frequency = (Base clock) / divider. Thus choosing the smallest possible divider will result in clock frequency which is less than or equal to the target frequency.
19:16 (R/W)	DAT_TIMEOUT_VAL	4'h0	This value determines the interval by which DAT line timeouts are detected. Refer to DAT_TIMEOUT_ERR in the SD_INT_STATUS register for information on factors that determine timeout generation. Timeout clock frequency will be generated by dividing the base clock TMCLK with this value. When setting this register, make sure to prevent inadvertent timeout events by clearing DAT_TIMEOUT_ERR_EN (in the SD_INT_EN

Bit	Name	Default	Description
			register) 1111: Reserved 1110: TMCLK * 2^27 …… 0001: TMCLK * 2^14 0000: TMCLK * 2^13
23:20	-	-	Reserved
24 (R/W)	SOFT_RST_ALL	1'b0	This reset affects the entire HC except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared to 0. During its initialization, HD will set this bit to 1 to reset the HC. The HC will then reset this bit to 0 when the Capabilities registers are valid and are readable by HD. Additional use of SOFT_RST_ALL will not affect the value of the Capabilities registers. If this bit is set to 1, then the SD card will reset itself and it must be reinitialized by the HD. 1: Reset 0: Work
25 (R/W)	SOFT_RST_CMD	1'b0	Only part of the command circuit will be reset. The following registers and bits are cleared by this bit: SD_PRESENT_STATE register CMD_INHIBIT (CMD) SD_INT_STATUS register CMD_END 1: Reset 0: Work
26 (R/W)	SOFT_RST_DAT	1'b0	Only part of the data circuit will be reset. The DMA circuit will also be reset. The following registers and bits are cleared by this bit: SD_BUF_DATA register is cleared and Initialized SD_PRESENT_STATE register BUFF_RD_EN BUFF_WT_EN RD_TRAN_ACTIVE WT_TRAN_ACTIVE DAT_LINE_ACTIVE CMD_INHIBIT (DAT) SD_HOST_PWR_BCG_WKP_CTRL register CONTINUE_REQ STOP_AT_BLK_GAP_REQ SD_INT_STATUS register

Bit	Name	Default	Description
			BUFF_RD_READY BUFF_WT_READY BLK_GAP_EVT TRAN_END 1: Reset 0: Work
31:27	-	-	Reserved

Table 449: SDIO Host Control Register 1

NOTE – Upon HC initialization, HD will set DAT_TIMEOUT_VAL according to the Capabilities register.

NOTE – A reset pulse is generated when writing 1 to each register bit. After completing the reset, HC will clear each bit. The SD host driver will confirm that these bits are 0 as it may take some time for software to completely reset the bit.

- SDIO Interrupt Status Register (SD_INT_STATUS) – 0x0030

  The interrupt status register includes two sections: low word for normal interrupt status and high word for error interrupt status.

  The Interrupt Status Enable affects read of this register, but Interrupt Signal Enable does not affect these reads. An Interrupt will be generated when the Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits except ERR_INT, writing 1 to this bit will clear it.

Bit	Name	Default	Description
0 (R/W)	CMD_END	1'b0	This bit is set when getting the end bit of the command response (Except Auto CMD12). CMD_TIMEOUT_ERR has higher priority over Command Complete. If both are set to 1, then it can be considered the response was not received correctly. 0: No Command Complete 1: Command Complete
1 (R/W)	TRAN_END	1'b0	This bit is set when a read / write transaction is completed. **For read transactions:** This bit is set at the falling edge of RD_TRAN_ACTIVE status. There are two cases in which the Interrupt is generated. The first is when a data transfer is completed as specified by the data length (After the last data has been read to the host system), the other one is when data has stopped at the block gap and completed data transfer by setting STOP_AT_BLK_GAP_REQ in the SD_HOST_PWR_BCG_WKP_CTRL Register (After a valid data has been read to the host system).

Bit	Name	Default	Description
			For write transactions: This bit is set at the falling edge of the DAT_LINE_ACTIVE status. There are two cases in which the interrupt is generated. The first is when the last data is written to the card as specified by data length and the busy signal is released. The second is when data transfers are stopped at the block gap by setting STOP_AT_BLK_GAP_REQ in the SD_HOST_PWR_BCG_WKP_CTRL register and data transfers are completed. (After a valid data is written to the SD card and the busy signal is released). TRAN_END has higher priority over DAT_TIMEOUT_ERR. If both bits are set to 1, then the data transfer will be considered completed 0: No Data Transfer Complete 1: Data Transfer Complete
2 (R/W)	BLK_GAP_EVT	1'b0	If STOP_AT_BLK_GAP_REQ in SD_HOST_PWR_BCG_WKP_CTRL is set, then this bit will be set. **For read transactions:** This bit is set at the falling edge of the DAT_LINE_ACTIVE status (When the transaction is stopped at SD Bus timing, Read Wait must be supported in order to use this function). **For write transaction:** This bit is set at the falling edge of WT_TRAN_ACTIVE status (After getting CRC status at SD Bus timing). 0: No BLK_GAP_EVT 1: Transaction stopped at block gap
3 (R/W)	DMA_INT	1'b0	This status is set if the HC detects the host DMA buffer boundary in the Block Size register: 0: No DMA interrupts 1: DMA interrupt generated
4 (R/W)	BUFF_WT_RDY	1b0	This status is set if the BUFF_WT_EN changes from 0 to 1: 0: Not ready to write buffer 1: Ready to write buffer
5 (R/W)	BUFF_RD_RDY	1'b0	This status is set if the BUFF_RD_EN changes from 0 to 1: 0: Not ready to read buffer 1: Ready to read buffer
6 (R/W)	CARD_INSERT	1'b0	This status is set if CARD_INSERTED in the SD_PRESENT_STATE registers changes from 0 to 1. When HD writes this bit to 1, this status will be cleared. The status for CARD_INSERTED in the SD_CUR_STA register should be confirmed. Because the card detect may possibly be changed when HD clears this bit, in this case, an interrupt event may not be generated.

Bit	Name	Default	Description
			0: Card state stable or debouncing 1: Card inserted
7 (R/W)	CARD_REMOVE	1'b0	This status is set if CARD_INSERTED in the SD_PRESENT_STATE register changes from 1 to 0. When the HD writes this bit to 1, this status will be cleared. The status of CARD_INSERTED in the SD_CUR_STA register should be confirmed. Because the card detect may possibly be changed when HD clears this bit, in this case, an interrupt event may not be generated. 0: Card state stable or debouncing 1: Card removed
8 (R/W)	CARD_INT	1'b0	Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, HC will detect CARD_INT without the SD clock to support wake-up events. In the 4-bit mode, the card interrupt signal will be sampled during the interrupt cycle, so there will be some sample delays between the interrupt signal from the card and the interrupt to the host system. When this status has been set and HD needs to start this interrupt service, CARD_INT_EN in the SD_INT_EN register will be set to 0 in order to clear the card interrupt status latched in the HC and stop driving the host system. Upon completion of the card interrupt service (the reset factor in the SD card and the interrupt signal may not be asserted), set CARD_INT_EN to 1 and start sampling the interrupt signal again. 0: No card interrupt 1: Generate card interrupt
14:9	-	-	Reserved
15 (R/W)	ERR_INT	1'b0	If any of the bits in the upper word of SD_INT_STATUS are set and the corresponding status enable bits are also set, this bit will be set. Software can test for errors by checking this bit first. 0: No error 1: Error
16 (R/W)	CMD_TIMOUT_ERR	1'b0	Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If HC detects a command line conflict, CMD_CRC_ERR should also be set. This bit should be set without waiting for 64 SDCLK cycles because the command will be aborted by the HC. 0: No error 1: Timeout
17 (R/W)	CMD_CRC_ERR	1'b0	CMD_CRC_ERR is generated in two cases. • If a response is returned and the CMD_TIMEOUT_ERR is set to 0, then this bit will be set to 1 when detecting a CRC error in the command response. • The HC detects a command line conflict by monitoring the

Bit	Name	Default	Description
			command line when a command is issued. If HC drives the command line to 1 level, but detects 0 levels at the next SDCLK edge, then HC will abort the command (Stop driving the command line) and set this bit to 1. The Command Timeout Error will also be set to 1 to distinguish command line conflicts. 0: No error 1: CRC error generated
18 (R/W)	CMD_ENDBIT_ERR	1'b0	Occurs when detecting the end bit of a command response is 0. 0: No error 1: End bit error generated
19 (R/W)	CMD_INDEX_ERR	1'b0	Occurs if a Command Index error occurs in Command Response. 0: No Error 1: Error
20 (R/W)	DAT_TIMEOUT_ERR	1'b0	Occurs when detecting one of the following timeout conditions: • Busy timeout for R1b, R5b type • Busy timeout after write CRC status • Write CRC status timeout • Read data timeout 0: No error 1: Timeout
21 (R/W)	DAT_CRC_ERR	1'b0	Occurs when detecting CRC error while transferring read data that uses the DAT line or when detecting the Write CRC status with a value other than 010. 0: No error 1: Error
22 (R/W)	DAT_ENDBIT_ERR	1'b0	Occurs when detecting 0 at the end bit position of read data that uses the DAT line or the end bit position of the CRC status. 0: No error 1: Error
23 (R/W)	CURRENT_LIM_ERR	1'b0	By setting the SD_BUS_PWR bit in the SD_HOST_PWR_BCG_WKP_CTRL register; HC is requested to supply power to the SD bus. If HC supports the Current Limit function, then it can be protected from an Illegal card by stopping power supply to the card, in which case this bit will indicate failure. Reading 1 means the HC is not supplying power to the SD card due to failure. Reading 0 means HC is supplying power and no error has occurred. This bit should always be set to 0 if HC does not support this function. 0: No error 1: Power failure

Bit	Name	Default	Description
24 (R/W)	AUTO_CMD12 _ERR	1'b0	Occurs when detecting one of the bits in the AUTO_CMD12_ERR_STATUS register has changed from 0 to 1. This bit is set to 1 when Auto CMD12 is not executed due to the previous command error. 0: No error 1: Error
31:25	-	-	Reserved

Table 450: SDIO Interrupt Status Register

Transfer Complete	Data Timeout Error	Meaning of the Status
0	0	Interrupted by another factor
0	1	Timeout occurs during transfer
1	Don't care	Data transfer completed

Table 451: Relation between Transfer Complete and Data Timeout Error

Command Complete	Command Timeout Error	Meaning of the Status
0	0	Interrupted by another factor
Neglected	1	Response not received within 64 SDCLK cycles
1	0	Response Received

Table 452: Relation between Command Complete and Command Timeout Error

Command CRC Error	Command Timeout Error	Kinds of Error
0	0	No Errors
0	1	Response timeout error
1	0	Response CRC error
1	1	Command line conflict

Table 453: Relation between Command CRC Error and Command Timeout Error

- SDIO Interrupt Status Enable Register (SD_INT_STA_ENA) – 0x0034

Bit	Name	Default	Description
0 (R/W)	CMD_END_INT_STA_EN	1'b0	0: Masked 1: Enabled

Bit	Name	Default	Description
1 (R/W)	TRAN_END_INT_STA_EN	1'b0	As above
2 (R/W)	BLK_GAP_EVT_INT_STA_EN	1'b0	As above
3 (R/W)	DMA_INT_STA_EN	1'b0	As above
4 (R/W)	BUFF_WT_RDY_INT_STA_EN	1'b0	As above
5 (R/W)	BUFF_RD_RDY_INT_STA_EN	1'b0	As above
6 (R/W)	CARD_INSERT_INT_STA_EN	1'b0	As above
7 (R/W)	CARD_REMV_INT_STA_EN	1'b0	As above
8 (R/W)	CARD_INT_STA_EN	1'b0	As above
15:9	-	-	Reserved
16 (R/W)	CMD_TIMOUT_ERR_INT_STA_EN	1'b0	0: Masked 1: Enabled
17 (R/W)	CMD_CRC_ERR_INT_STA_EN	1'b0	As above
18 (R/W)	CMD_ENDBIT_ERR_INT_STA_EN	1'b0	As above
19 (R/W)	CMD_INDEX_ERR_INT_STA_EN	1'b0	As above
20 (R/W)	DAT_TIMEOUT_ERR_INT_STA_EN	1'b0	As above
21 (R/W)	DAT_CRC_ERR_INT_STA_EN	1'b0	As above
22 (R/W)	DAT_ENDBIT_ERR_INT_STA_EN	1'b0	As above
23 (R/W)	CURRENT_LIM_ERR_INT_STA_EN	1'b0	As above
24 (R/W)	AUTO_CMD12_ERR_INT_STA_EN	1'b0	As above
31:25	-	-	Reserved

Table 454: SDIO Interrupt Status Enable Register

NOTE – Setting to 1 will enable the interrupt status.

HC may sample the card interrupt signals during the interrupt period and may hold its value in the flip–flop. If Card Interrupt Status Enable is set to 0, then HC will clear all internal signals related to card interrupts.

To detect command line conflicts, HD must set both the CMD_TIMEOUT_ERR_INT_EN and the CMD_CRC_ERR_INT_EN bits to 1.

- SDIO Interrupt Signal Enable Register (SD_INT_SIG_ENA) – 0x0034

Bit	Name	Default	Description
0 (R/W)	CMD_END_INT_SIG_EN	1'h0	0: Masked 1: Enabled
1 (R/W)	TRAN_END_INT_SIG_EN	1'h0	0: Masked 1: Enabled
2 (R/W)	BLK_GAP_EVT_INT_SIG_EN	1'h0	0: Masked 1: Enabled
3 (R/W)	DMA_INT_SIG_EN	1'h0	0: Masked 1: Enabled
4 (R/W)	BUFF_WT_RDY_INT_SIG_EN	1'h0	0: Masked 1: Enabled
5 (R/W)	BUFF_RD_RDY_INT_SIG_EN	1'h0	0: Masked 1: Enabled
6 (R/W)	CARD_INSERT_INT_SIG_EN	1'h0	0: Masked 1: Enabled
7 (R/W)	CARD_REMV_INT_SIG_EN	1'h0	0: Masked 1: Enabled
8 (R/W)	CARD_INT_SIG_EN	1'h0	0: Masked 1: Enabled
15:9	-	-	Reserved
16 (R/W)	CMD_TIMOUT_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
17 (R/W)	CMD_CRC_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
18 (R/W)	CMD_ENDBIT_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
19 (R/W)	CMD_INDEX_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
20 (R/W)	DAT_TIMEOUT_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
21 (R/W)	DAT_CRC_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
22 (R/W)	DAT_ENDBIT_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled

Bit	Name	Default	Description
23 (R/W)	CURRENT_LIM_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
24 (R/W)	AUTO_CMD12_ERR_INT_SIG_EN	1'h0	0: Masked 1: Enabled
31:25	-	-	Reserved

Table 455: SDIO Interrupt Signal Enable Register

This register is used to select which interrupt status is indicated to the host system as interrupt. These status bits share the sample 1-bit interrupt line. Setting any of these bits to 1'b1 will enable interrupt generation.

- SDIO Auto CMD12 Error Status Register (SD_AUTO_CMD12_ERR_STATUS)

  When Auto CMD12 Error Status is set, HD will check this register to identify what kind of error Auto CMD12 indicates. This register is valid only when the Auto CMD12 Error is set.

Bit	Name	Default	Description
0 (R)	AUTO_CMD12_NOT_EXEC	1'b0	If memory multiple block data transfer is not started due to command errors, then this bit will not be set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means HC cannot issue Auto CMD12 to stop memory multiple block transfers caused by command errors. If this bit is set to 1, then other error status bits (D04 – D01) will become meaningless. 0: Executed 1: Not Executed
1 (R)	AUTO_CMD12_TIMEOUT_ERR	1'b0	Occurs if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, other error status bits (D04 – D02) will be meaningless. 0: No error 1: Timeout
2 (R)	AUTO_CMD12_CRC_ERR	1'b0	Occurs when detecting a CRC error in the command response. 0: No error 1: CRC error generated
3 (R)	AUTO_CMD12_ENDBIT_ERR	1'b0	Occurs when detecting the end bit of the command response is 0. 0: No error 1: End bit error generated
4 (R)	AUTO_CMD12_INDEX_ERR	1'b0	Occurs if the Command Index error occurs in response to a command. 0: No error

Bit	Name	Default	Description
			1: Error
6:5	-	-	Reserved
7 (R)	CMD_NOT_ISSURED_BY_CMD12_ERR	1'b0	Setting this bit to 1 means CMD_WO_DAT is not executed due to an Auto CMD12 error (D04 – D01) in this register. 0: No error 1: Not issued
31:8	-	-	Reserved

Table 456: SDIO Auto CMD12 Error Status Register

Auto Cmd12 CRC Error	Auto CMD12 Timeout Error	Kinds of Error
0	0	No errors
0	1	Response timeout error

Table 457: Relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error

Auto Cmd12 CRC Error	Auto CMD12 Timeout Error	Kinds of Error
1	0	Response CRC error
1	1	Command line conflict

Table 458: Relation between Auto CMD12 CRC Error and Auto CMD12 Timeout Error

The timing for changing Auto CMD12 Error Status can be classified in three scenarios:

- When HC is going to issue Auto CMD12
  - Δ Set bit 0 to 1'b1 if Auto CMD12 cannot be issued due to an error in the previous command.
  - Δ Set bit 0 to 1'b0 if Auto CMD12 is issued.

- At the end bit of Auto CMD12 response
  - Δ Check received responses by checking the error bits 1~4.
  - Δ Set to 1'b1 if Error is detected.
  - Δ Set to 1'b0 if Error is not detected.

- Before reading Auto CMD12 Error Status bit 7
  - Δ Set bit 7 to 1'b1 when there is a command that cannot be issued.
  - Δ Set bit 7 to 1'b0 when there is no command to issue.

Timing for generating the Auto CMD12 error and writing to the SD_CMD_TRAN_MODE register is Asynchronous. Bit 7 is usually sampled when the host driver never writes to the SD_CMD_TRAN_MODE register. Therefore the best time to set the bit 7 status bit is before the AUTO_CMD12_ERR_STATUS register is read.

- SDIO Capabilities Register (SD_CAPABILITY_REG) – 0x0040

Bit	Name	Default	Description
5:0 (R)	-	6'h30	Reserved
6	-	-	Reserved
7 (R)	TIMEOUT_UNIT	1'b1	This bit shows the unit of base clock frequency used to detect DAT_TIMEOUT_ERR. 0: KHz 1: MHz
13:8 (R)	-	6'h30	Reserved
15:14	-	-	Reserved
17:16 (R)	-	2'b10	Reserved
20:18	-	-	Reserved
21 (R)	HIGH_SPEED_SUP	1'b1	This bit indicates whether HC and the host system supports high-speed mode and whether they can supply SD clock frequency from 25MHz to 50 MHz. 0: High speed is not supported. 1: High speed is supported.
22 (R)	DMA_SUP	1'b1	It indicates whether HC is able to use DMA for directly transferring data between the system memory and HC. 0: DMA is not supported. 1: DMA is supported.
23 (R)	SUS_RESM_SUP	1'b1	This bit indicates whether HC supports Suspend / Resume functionality. If this bit is 0, then the Suspend and Resume mechanism will not be supported and HD will not issue either Suspend or Resume commands. 0: Not supported. 1: Supported.
24 (R)	-	-	Reserved
25 (R)	-	-	Reserved
26 (R)	-	-	Reserved (not supported)
31:27	-	-	Reserved

Table 459: SDIO Capabilities Register

NOTE – The host system should support at least one of the voltages listed in the table above. HD sets the SD_BUS_VOL_SEL in SD_HOST_PWR_BCG_WKP_CTRL register according to these support bits. If multiple voltages are supported, then select a usable lower voltage by comparing the OCR value from the card.

- SDIO Maximum Current Capabilities Register (SD_MAX_CUR_CAPABILITY) – 0x0048

  These registers indicate the maximum current capability for each voltage. The value is useful only if the corresponding Voltage Support bit is set in the Capabilities register.

Bit	Name	Default	Description
7:0 (R)	MAX_CUR_33V	8'hff	Reserved (needs external power supply)
15:8 (R)	MAX_CUR_30V	8'hff	Reserved (needs external power supply)
23:16 (R)	Reserved	8'hff	Reserved
31:24 (R)	-	-	Reserved

Table 460: SDIO Maximum Current Capabilities Register

Register Value	Current Value
0	Get information via another method
1	4 mA
2	8 mA
3	16 mA
……	……
255	1020 mA

Table 461: Maximum Current Value Definition

- SD Bus Clock Delay Register (SD_CLK_DELAY_SETTING) – 0x004C

Bit	Name	Default	Description
5:0 (R/W)	CLK_DELAY	6'h00	The register is used to adjust to the clock delay only during micro-adjust clock phase. The maximum change value is approximately 4ns.
31:6 (R/W)	-	26'h0	Reserved.

Table 462: SD Bus Clock Delay Register

- SDIO Slot Interrupt Status and Version Register (SD_SLOT_INT_STAUTS) – 0x00FC

Bit	Name	Default	Description
1:0 (R)	INT_SIG_FOR_SLOT	2'h0	These status bits indicate the logical OR of the interrupt and the wake-up signal for each slot. A maximum of 8 slots can be defined. If one of the interrupt signals is associated with multiple slots, then HD will be able to know which interrupt is being generated by reading these status bits. Using a power-on reset or the SOFT_RST_ALL register will de-assert the interrupt signal and cause its status to read 00h. Bit 00: Slot 0 Bit 01: Slot 1
15:3	-	-	Reserved.
23:16 (R/W)	SPEC_VER	8'h0	This status indicates the Host Controller Spec Version. The upper and lower 4 bits indicate the version: 00: SD Host Specification version 1.0 Others: Reserved
31:24 (R/W)	VENDOR_VER	8'hff	Reserved for vendor version number. HD should not use it.

Table 463: SDIO Slot Interrupt Status and Version Register

I/O Bridge

Overview

The I/O Bridge interfaces with the DMA bus and the AXI system bus. It converts read/write commands from the DMA bus into AXI bus operations. In the read direction, the I/O Bridge will read data from the memory via the AXI bus and store data in the data FIFO for the DMA controller to read. In the write direction, the I/O Bridge will read data from the DMA bus and store it in the internal data FIFO. It then reads out data and outputs to AXI bus. Therefore the I/O Bridge will convert the DMA bus timing into AXI bus timing and AXI bus timing to DMA bus timing. It is a bridge between the I/O DMA bus and the AXI system bus.

Feature List

- Main features of the block
  - The I/O Bridge is a protocol converter between the DMA controller bus and the AXI system bus.
  - If the current operation is the burst write operation and the write address has not been granted by the memory controller, another burst write request is issued. Addresses of the two write operations are continuous. Two burst write operations can be combined to form one write operation in the AXI bus with doubled burst length. This mode is considered the performance optimization mode.

- Operation modes of the block
  - It supports single mode and burst mode DMA
  - It supports performance optimization mode

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O Interface	0xb0000000

Table 464: I/O Bridge Base Address

Register Mapping

Address <15:0>	Register	Description
0x0800	IOBG_SCMD_EN	IOBG enable second command
0x0804	IOBG_FLUSH	IOGB flush
0x0814	IOBG_RESP_CNT	IOBG response counter

Table 465: I/O Bridge Address Mapping

Register Descriptions

- Enable Second Command Register (IOBG_SCMD_EN) – RISC: 0x0800

Bit	Name	Default	Description
0 (R/W)	Enable	1'h0	I/O Bridge second command enable bit: 1: I/O Bridge can convert two continuous DMA write operations into one AXI burst transfer; performance optimization mode enabled. 0: Performance optimization mode disabled.
31:1	-	31'h0	Reserved

Table 466: Enable Second Command Register

- I/O Bridge Flush Register (IOBG_FLUSH) – 0x0804

  After all data stored in the internal data FIFO of I/O Bridge has been sent to the destination (memory or I/O peripheral), FLASH will be set to indicate the data FIFO is empty.

Bit	Name	Default	Description
0 (R/W)	FLUSH	1'h0	I/O bridge flush bit: 0: The data FIFO in I/O bridge is empty. The I/O Bridge has sent all data to the destination. 1: The data FIFO in I/O bridge is not empty. Some data still remains in the I/O Bridge. I/O Bridge will clear this register when the I/O Bridge is in an idle state or a transaction in the I/O Bridge is completed. In order to determine whether the current I/O Bridge operation is finished, user should read this register until it becomes 0.
31:1	-	31'h0	Reserved

Table 467: I/O Bridge Flush Register

- I/O Bridge Response Counter Register (IOBG_RESP_CNT) – 0x0814

Bit	Name	Default	Description
0 (R/W)	START_CNT	1'h0	1: Starts to count the responses from memory controller which indicate the AXI data has been sent out. When this bit is set, the counter will be cleared and will start to count bvalid signals from the memory controller. This bit can be auto-cleared by hardware. 0: Counter is not started; the counter will hold the old value. Reading this bit will always get 1.
5:1 (R)	RESP_CNT	5'h0	The counter of the bvalid signal from the memory controller.
31:6	-	26'h0	Reserved.

Table 468: I/O Bridge Response Counter Register

DMA Controller

Overview

SiRFatlasV contains some peripherals such as audio codec, UART and USP which may at times be slow in speed. When these peripherals request memory access, RISC will have to respond to them one by one and process them one byte at a time. It therefore will create heavy overhead for RISC and causing it to be interrupted.

The DMA controller is intended to relieve the processor of the interrupt overhead while servicing these peripherals via a programmed I/O. The DMA controller can convert sparse memory access requests from slow interfaces into burst transfer requests on the system bus.

Feature List

The DMA controller consists of 16 independent DMA channels. Each channel is allocated to a different function described in the figure below except for three channels are reserved.

Channel	Function	Priority
		High
0	USP0 input	
1	USP0 output	
2	Reserved	
12	SPI0 input	
13	SPI0 output	
14	USP1 input	
15	USP1 output	
10	UART0 input	
11	UART0 output	
6	AC97/I^2S input	
7	AC97/I^2S output	
8	AC97 Aux input	
3	TSC stream mode input	
9	Reserved	
4	NAND Flash input/output	
5	Reserved	Low

Figure 69: DMA Controller Predetermined Priority

In the above figure:

- Channels 6 and 7 are shared between:
  - AC97 (input/output)
  - I^2S (input/output)

Each peripheral has its own FIFO for DMA. The user can set up the FIFO request control register to control when the FIFO generates requests to the DMA controller. For example, the value in the request control register can be half the FIFO size. Once the FIFO is half-full/empty, it will generate a request signal to the DMA controller to start or stop the DMA. If the peripheral needs service sooner, the user can program the register to a higher/lower value, depending on the direction of the DMA.

If the peripheral is a full-duplex device, then it will need to have two FIFO, one for each direction. If the peripheral is half-duplex, then one FIFO would be good enough. Yet the peripheral designer will need to be careful when determining the size of the FIFO in order to prevent FIFO overflow or underflow. If an overflow or underflow occurs, it will generate an interrupt.

The DMA controller is intended to relieve the processor of the interrupt overhead while servicing these peripherals via a programmed I/O. But if desired, any or all peripherals can be serviced by the programmed I/O instead of the DMA. Each peripheral is capable of requesting processor services through its own interrupt line.

The DMA controller supports 1-D, 2-D and loop DMA. DMA controller supports single (1-DWORD) and burst (4-DWORD) mode transfers.

Functional Descriptions

DMA Channel Arbitration

The DMA controller serves multiple blocks and arbitrates between different blocks that need to access the external memory. Each DMA channel has a 2-bit request level which is determined by the internal FIFO status automatically. In addition, they all have an inherent priority (as shown above); so that when two channels have the same request level, the channel with the higher priority will win. The priority level can be programmed by the RISC or determined by the fullness of FIFO, depending on the channel.

Arbitration is a 3-step process:

1. Requests for each level are determined by the request signals and whether a DMA is set up for that channel.
2. Each level goes to a priority encoder in the pre-determined order to indicate whether there is a request for that channel and a channel to service.
3. The channel for the highest valid request will be serviced.

When the DMA controller detects a request with a higher arbitration priority than the current one (based on the above protocol), it will interrupt the current DMA, store the location of the last access (i.e. where the stop occurs), and proceed to service the new request. The interrupted request can resume from where it last stopped after its arbitration priority becomes the highest again. This scheme can be repeated for all the channels active at the same time.

The FIFO is a block of SRAM on-chip with a read pointer chasing a write pointer (one may never pass the other one). DMA transfer is implemented via the data transfer between the external memory and FIFOs. Each peripheral has its own FIFO. The FIFO size is peripheral dependent. Lower-bandwidth peripherals do not need large FIFO size. The FIFO must provide status flags and interrupts to the host (RISC or DSP) if there is an underflow or overflow.

The fullness of the FIFO is determined by the difference of the two pointers. Since most FIFOs in this chip are bi-directional, one must be very careful with both FIFO overflow and underflow. Make sure that the request levels to the memory controller are set properly so that it can start or stop in time.

The FIFO will interface with both a data producing/consuming peripheral and the DMA controller. A FIFO will send a 2-bit request level to the DMA controller. If that FIFO is serviced, then the DMA controller will send it a data valid signal. In the following cycle, the FIFO will either write the 32-bit data into the data bus (if it is writing to the memory), or read the 32-bit data from the data bus (if it is reading from the memory).

The request level is determined by two sets of registers (one for reading and one for writing) setting three checkpoints (stop, low, high) that will trigger different request levels.

FIFO Requests	FIFO Write to External Memory	FIFO Read from External Memory
2'b00	FIFO is between Empty and Stop.	FIFO is between Full and Stop

FIFO Requests	FIFO Write to External Memory	FIFO Read from External Memory
2'b01	FIFO is between Stop and Low.	FIFO is between Stop and Low
2'b10	FIFO is between Low and High.	FIFO is between Low and High
2'b11	FIFO is between High and Full.	FIFO is between High and Empty

Table 469: FIFO Requests Levels

There are two FIFOs for the peripheral that perform bi-directional data transfer, and it will be one FIFO for each direction, thereby limiting both the DMA controller and the peripheral block to write the same FIFO.

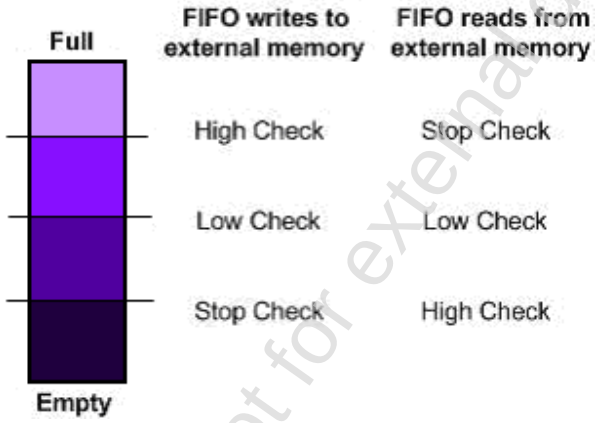


Figure 70: FIFO Request Level Checkpoints

Some peripherals are relatively fast and may need to transfer several words of data at one time. Then it can be programmed in burst DMA mode. In burst DMA mode, the DMA will transfer four 32-bit words of data at one time instead of one in normal mode. The stop checkpoint of the input FIFO in burst mode should be set greater than or equal to 4.

It is necessary to configure a flush-bit to the input FIFO so that the last few words (under stop checkpoint) of the input will not be stuck in the FIFO at the end of DMA. When the flush bit is set, the request level will maintain at 2'b01 even the FIFO is below the stop level. This ensures the FIFO is emptied even if new data is not coming in.

Each DMA has its own interrupt flag and enable/disable bit. If the disable bit is asserted, then DMA will be stopped. If a DMA is finished, then an interrupt to the RISC or DSP will be generated. The interrupt can either be enabled or disabled.

NOTE – FIFO and its corresponding DMA request generation mechanism are implemented in their respective peripherals, instead of the DMA controller. DMA controller receives the 3-level DMA requests and arbitrates them. The highest level request will get granted. If several requests have the same level, then the pre-determined priority order will apply.

The DMA_arbiter module implements the arbitration function. It generates the current_channel signal after arbitration to indicate which channel will be granted.

Function Modes

- Single and Burst DMA

As we specified before, the DMA controller has two data transfer modes, the single transfer mode and the burst transfer mode. In single transfer mode, the DMA controller executes one 32-bit word read/write at a time; while in burst mode, it executes four 32-bit words a time. The user can select the transfer mode by programming one of the register bits in the DMA controller.

One thing to note is that in burst mode, the DMA address is better to be in 4-DWORD boundary. Otherwise, the transfer will be split into multiple single-word transfers in the system bus. This is because the system bus does not allow the non-aligned burst transfer.

- 1-D and 2-D DMA

The DMA controller supports both 1-D and 2-D DMA. In 2-D DMA, the system memory space is interpreted as a 2-D layout instead of a linear 1-D layout. More specifically, the system memory can be considered as multiple data lines. The length of the data line is determined by the user-selected DMA_WIDTH register. The user can specify a data window that the user wants to access using four parameters:

  - Start address
  - X length
  - Y length
  - Width

The idea of a 2-D DMA is shown in the following diagram.

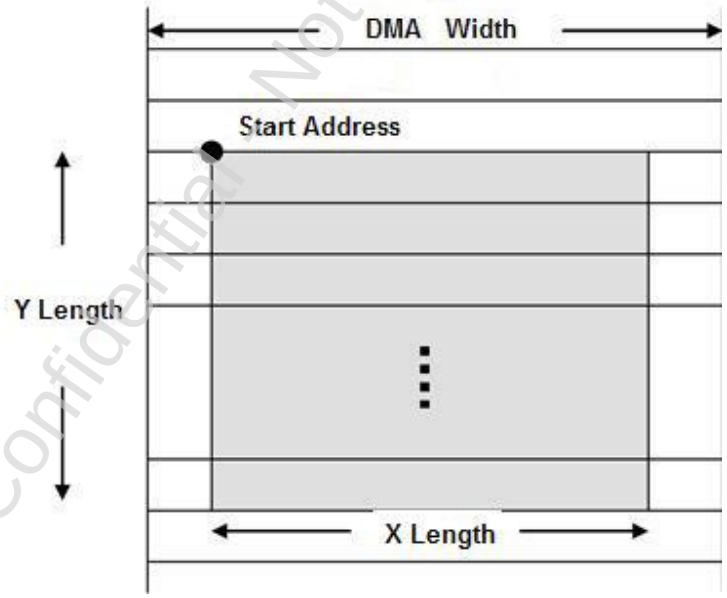


Figure 71: 2-D DMA Parameters

If the user specifies the Y length as 0 or the X length equals to the DMA width, then this 2-D DMA will reduce to 1-D.

If the user configures the X length greater than the DMA width, then the extra data will be wrapped around to the next data line, this may corrupt the DMA transfer for multiple-line 2-D DMA. If this is a 1-D

DMA, then there will not be an issue. The following diagram shows the wrap-around of the extra data in case the X length greater than DMA width.

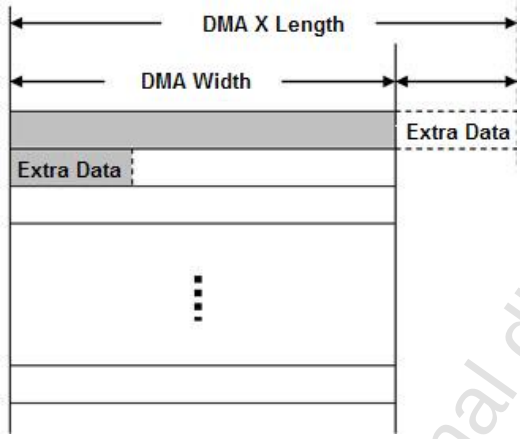


Figure 72: 2-D DMA Wrap Around (X-Length > DMA Width)

- Loop DMA

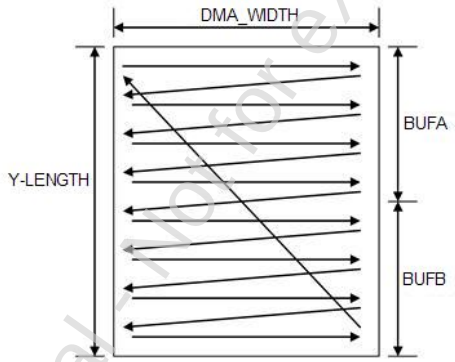


Figure 73: DMA Address Change If X_LEN=0

If the X-length is set to 0, then it would be considered a special DMA mode, the loop mode. In loop mode, DMA will never finish until user forces it to stop (by writing 1'b0 to the BUFA_VALID or BUFB_VALID register). The DMA will keep looping as illustrated in the above figure.

As shown in the figure above, the DMA address will keep increasing until reaching the end of a loop area whose size is defined by (DMA_WIDTH x (Y_LENGTH + 1)). Then the DMA address will go back to the beginning of this area. If both Y_LENGTH and DMA_WIDTH are 0, then the DMA address will not change at all, in this case, DMA will continue to transfer data to the same DMA address until user forces it to stop.

In loop mode, the DMA data region is divided into two parts, BUFA and BUFB (such as "Buffer A and Buffer B"). DMA controller will generate interrupts twice in each loop: when the DMA address reaches the end of BUFA and when the DMA address reaches the end of the BUFB. Of course the interrupt can only be generated when the corresponding interrupt enable (DMA_INT_EN) bit is set.

Each buffer (BUFA or BUFB) has its own buffer valid register bit, which can be programmed by the user. The loop DMA will not be started until the current buffer valid register bit is asserted. For example, if DMA

goes to the end of BUFA and the valid bit of BUFB is not set, DMA will stop at the end of BUFA until the valid bit of BUFB is set.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
DSP I/O interface	0x100
RISC I/O Interface	0xb0000000

Table 470: DMA Controller Base Address

Register Mapping

RISC Address <11:0>	DSP I/O Address <7:0>	Register	Description
0x0000	0x00~0x01	DMA_CH0_ADDR	DMA channel 0 address register
0x0004	0x02	DMA_CH0_XLEN	DMA channel 0 X-length register
0x0008	0x04	DMA_CH0_YLEN	DMA channel 0 Y-length register
0x000C	0x06	DMA_CH0_CTRL	DMA channel 0 control register
0x0010	0x08~0x09	DMA_CH1_ADDR	DMA channel 1 address register
0x0014	0x0A	DMA_CH1_XLEN	DMA channel 1 X-length register
0x0018	0x0C	DMA_CH1_YLEN	DMA channel 1 Y-length register
0x001C	0x0E	DMA_CH1_CTRL	DMA channel 1 control register
0x0020	0x10~0x11	DMA_CH2_ADDR	DMA channel 2 address register
0x0024	0x12	DMA_CH2_XLEN	DMA channel 2 X-length register
0x0028	0x14	DMA_CH2_YLEN	DMA channel 2 Y-length register
0x002C	0x16	DMA_CH2_CTRL	DMA channel 2 control register
0x0030	0x18~0x19	DMA_CH3_ADDR	DMA channel 3 address register
0x0034	0x1A	DMA_CH3_XLEN	DMA channel 3 X-length register
0x0038	0x1C	DMA_CH3_YLEN	DMA channel 3 Y-length register
0x003C	0x1E	DMA_CH3_CTRL	DMA channel 3 control register
0x0040	0x20~0x21	DMA_CH4_ADDR	DMA channel 4 address register
0x0044	0x22	DMA_CH4_XLEN	DMA channel 4 X-length register
0x0048	0x24	DMA_CH4_YLEN	DMA channel 4 Y-length register

RISC Address <11:0>	DSP I/O Address <7:0>	Register	Description
0x004C	0x26	DMA_CH4_CTRL	DMA channel 4 control register
0x0050	0x28~0x29	DMA_CH5_ADDR	DMA channel 5 address register
0x0054	0x2A	DMA_CH5_XLEN	DMA channel 5 X-length register
0x0058	0x2C	DMA_CH5_YLEN	DMA channel 5 Y-length register
0x005C	0x2E	DMA_CH5_CTRL	DMA channel 5 control register
0x0060	0x30~0x31	DMA_CH6_ADDR	DMA channel 6 address register
0x0064	0x32	DMA_CH6_XLEN	DMA channel 6 X-length register
0x0068	0x34	DMA_CH6_YLEN	DMA channel 6 Y-length register
0x006C	0x36	DMA_CH6_CTRL	DMA channel 6 control register
0x0070	0x38~0x39	DMA_CH7_ADDR	DMA channel 7 address register
0x0074	0x3A	DMA_CH7_XLEN	DMA channel 7 X-length register
0x0078	0x3C	DMA_CH7_YLEN	DMA channel 7 Y-length register
0x007C	0x3E	DMA_CH7_CTRL	DMA channel 7 control register
0x0080	0x40~0x41	DMA_CH8_ADDR	DMA channel 8 address register
0x0084	0x42	DMA_CH8_XLEN	DMA channel 8 X-length register
0x0088	0x44	DMA_CH8_YLEN	DMA channel 8 Y-length register
0x008C	0x46	DMA_CH8_CTRL	DMA channel 8 control register
0x0090	0x48~0x49	DMA_CH9_ADDR	DMA channel 9 address register
0x0094	0x4A	DMA_CH9_XLEN	DMA channel 9 X-length register
0x0098	0x4C	DMA_CH9_YLEN	DMA channel 9 Y-length register
0x009C	0x4E	DMA_CH9_CTRL	DMA channel 9 control register
0x00A0	0x50~0x51	DMA_CH10_ADDR	DMA channel 10 address register
0x00A4	0x52	DMA_CH10_XLEN	DMA channel 10 X-length register
0x00A8	0x54	DMA_CH10_YLEN	DMA channel 10 Y-length register
0x00AC	0x56	DMA_CH10_CTRL	DMA channel 10 control register
0x00B0	0x58~0x59	DMA_CH11_ADDR	DMA channel 11 address register
0x00B4	0x5A	DMA_CH11_XLEN	DMA channel 11 X-length register
0x00B8	0x5C	DMA_CH11_YLEN	DMA channel 11 Y-length register
0x00BC	0x5E	DMA_CH11_CTRL	DMA channel 11 control register
0x00C0	0x60~0x61	DMA_CH12_ADDR	DMA channel 12 address register

RISC Address <11:0>	DSP I/O Address <7:0>	Register	Description
0x00C4	0x62	DMA_CH12_XLEN	DMA channel 12 X-length register
0x00C8	0x64	DMA_CH12_YLEN	DMA channel 12 Y-length register
0x00CC	0x66	DMA_CH12_CTRL	DMA channel 12 control register
0x00D0	0x68~0x69	DMA_CH13_ADDR	DMA channel 13 address register
0x00D4	0x6A	DMA_CH13_XLEN	DMA channel 13 X-length register
0x00D8	0x6C	DMA_CH13_YLEN	DMA channel 13 Y-length register
0x00DC	0x6E	DMA_CH13_CTRL	DMA channel 13 control register
0x00E0	0x70~0x71	DMA_CH14_ADDR	DMA channel 14 address register
0x00E4	0x72	DMA_CH14_XLEN	DMA channel 14 X-length register
0x00E8	0x74	DMA_CH14_YLEN	DMA channel 14 Y-length register
0x00EC	0x76	DMA_CH14_CTRL	DMA channel 14 control register
0x00F0	0x78~0x79	DMA_CH15_ADDR	DMA channel 15 address register
0x00F4	0x7A	DMA_CH15_XLEN	DMA channel 15 X-length register
0x00F8	0x7C	DMA_CH15_YLEN	DMA channel 15 Y-length register
0x00FC	0x7E	DMA_CH15_CTRL	DMA channel 15 control register
0x0100	0x80	DMA_WIDTH0	DMA width 0 register
0x0104	0x81	DMA_WIDTH1	DMA width 1 register
0x0108	0x82	DMA_WIDTH2	DMA width 2 register
0x010C	0x83	DMA_WIDTH3	DMA width 3 register
0x0110	0x84	DMA_WIDTH4	DMA width 4 register
0x0114	0x85	DMA_WIDTH5	DMA width 5 register
0x0118	0x86	DMA_WIDTH6	DMA width 6 register
0x011C	0x87	DMA_WIDTH7	DMA width 7 register
0x0120	0x88	DMA_CH_VALID	DMA channel valid register
0x0124	0x89	DMA_CH_INT	DMA channel interrupt register
0x0128	0x8A	DMA_INT_EN	DMA interrupt enable register
0x012C	-	DMA_CH_DSP_CTRL	DMA channel DSP control register
0x0130	0x8C~8D	DMA_CH_LOOP_CTRL	DMA channel loop control register
Others	-	-	Reserved

Table 471: DMA Controller Address Mapping

NOTE – The DMA interface must occupy DSP 256x16-bit WORD I/O space.

Register Descriptions

- DMA Channel <11:0> Address Register (DMA_CH<11:0>_ADDR)

  The DMA channel start address is in the 32-bit D-word boundary. This must be the last register to be set among all the DMA configuration registers. Setting this register will start DMA.

Bit	Name	Default	Description
24:0 (R/W)	A<24:0>	25'h0	The DMA start address in 32-bit D-word boundary
31:25	-	7'h0	Reserved

Table 472: DMA Channel Address Register

NOTE – For burst-mode DMA, please set the start address to be within 16-byte boundary. Otherwise, the burst will be cut into multiple single transfers and the efficiency is much lower. For loop-mode DMA, setting the start address will not start DMA at once. The user needs to set DMA_CH_LOOP_CTRL to start DMA. Before starting a new DMA, user also need to configure this start address register to clear the internal status of the last DMA.

- DMA Channel <11:0> X-Length Register (DMA_CH<11:0>_XLEN)

  The DMA X-length is in the 32-bit D-word boundary. This value specifies the number of D-words transferred in each line. The value should be less than the value of DMA width register.

  In burst mode, X-length should be in the 16-byte boundary, but it is allowed to be set to any value. For example, if X-length = 5, the DMA controller will execute 2 bursts. However, in the last burst, the DMA controller will only transfer one DWord data to or from the peripheral.

  For 1-D DMA, this is not important because the last data transferred in the entire DMA will not affect the peripheral FIFO. But in 2-D DMA, when the X-length is not in the 16-byte boundary, then the last burst of every line will only execute a 1 DWORD data transfer. Thus, the DMA controller will not execute extra data transfer and affect the data sequence in the peripheral FIFO.

NOTE – From the system memory's point of view, there are still 2 bursts in each line. For example, the five DWORD data transfers of each line will take 8-DWORD's memory space.)

Bit	Name	Default	Description
11:0 (R/W)	XL<11:0>	12'h0	DMA X-length in 32-bit DWORD boundary
31:12	-	20'h0	Reserved

Table 473: DMA Channel X-Length Register

- DMA Channel <11:0> Y-Length Register (DMA_CH<11:0>_YLEN)

The DMA Y-length specifies the number of lines in DMA transfers. The number of the lines in DMA transfer is Y-length + 1. The maximum number of line can be 2048. Setting Y-length to 0 produces the same effect as 1-D DMA.

Bit	Name	Default	Description
11:0 (R/W)	YL<11:0>	12'h0	DMA Y-length
31:12	-	20'h0	Reserved

Table 474: DMA Channel Y-Length Register

- DMA Channel <11:0> Control Register (DMA_CH<11:0>_CTRL)

Bit	Name	Default	Description
2:0 (R/W)	WS	3'h0	DMA width register select signal: 000: Select DMA_WIDTH0. 001: Select DMA_WIDTH1. 010: Select DMA_WIDTH2. 011: Select DMA_WIDTH3. 100: Select DMA_WIDTH4. 101: Select DMA_WIDTH5. 110: Select DMA_WIDTH6. 111: Select DMA_WIDTH7.
3 (R/W)	BURST	1'b0	DMA transfer mode: 1: DMA is in burst transfer mode. 0: DMA is in single transfer mode.
4 (R/W)	DIR	1'b0	DMA direction: 1: DMA from memory to peripheral FIFO. 0: DMA from peripheral FIFO to memory.
31:5	-	27'h0	Reserved.

Table 475: DMA Channel Control Register

- DMA Width Registers (DMA_WIDTH0~7) – RISC: 0x100~0x11C, DSP: 0x80~0x87

  There are two separate sets of DMA Width registers, one for RISC and one for DSP. Accessing to which set of register depends on the setting of the DMA_CH_DSP_EN. Each set has eight Width registers. Each DMA channel can choose to use any one of these eight Width registers.

  To enable a correct 2-D DMA, the DMA Width register must be correctly set. The value of the DMA Width register must be greater than or equal to the X-length, otherwise, the data will be overlapped.

  NOTE –DMA_WIDTH should not be set to zero.

Bit	Name	Default	Description
11:0 (R/W)	WIDTH	12'h0	DMA width
31:12	-	20'h0	Reserved

Table 476: DMA Width Registers

NOTE – RISC can read or write its bank of registers either in RISC mode or DSP mode. However DSP can only write its bank of registers in DSP mode and read either in DSP mode or RISC mode.

- DMA Channel Valid Register (DMA_CH_VALID) – RISC: 0x120, DSP: 0x88

  If this register is 1, it means that the corresponding DMA channel is active. In mode other than the loop DMA, when user configures DMA_CH_ADDR, the corresponding VL bit will be set; when the DMA operation is finished, the corresponding VL bit will be cleared. In loop DMA, when DMA_CH_XLEN is set to 0, VL bit will be 0 until BUFA_VALID becomes valid, and when the address reaches the end of BUFA, VL will be 1 if BUFB_VALID is valid. In mode other than the loop DMA, the user can write a 1 to the register bit to force the corresponding DMA channel to stop. In loop DMA, user cannot set this register to stop loop DMA. In loop DMA, user can invalidate BUFA_VALID or BUFB_VALID to stop it.

Bit	Name	Default	Description
15:0 (R/W)	VL	16'h0	DMA channel valid bits. This bit will be automatically set to 1 if the START_ADDR register is written. Write a 1 to each bit will stop that channel's DMA.
31:16	-	16'h0	Reserved

Table 477: DMA Channel Valid Register

- DMA Channel Interrupt Register (DMA_CH_INT) – RISC: 0x124, DSP: 0x89

  After each DMA is finished, it will generate an interrupt bit in the corresponding bit of this register. RISC or DSP can read this register to check which DMA is finished. If either RISC or DSP writes 1 to that bit, then the interrupt bit will be cleared. If the DMA interrupt occurs, then it will only indicate that the DMA controller has sent all commands to the I/O Bridge. This doesn't mean that the data has been written to the memory or the peripheral.

NOTE – If the corresponding bits if DMA_INT_EN is not set, then the interrupt will not be active, the bits of DMA_CH_INT will also not be active.

Bit	Name	Default	Description
15:0 (R/W)	INT	16'h0	DMA interrupt bits
31:16	-	16'h0	Reserved

Table 478: DMA Channel Interrupt Register

- DMA Interrupt Enable Register (DMA_INT_EN) – RISC: 0x128, DSP: 0x8A

  There are two separate interrupt enable registers, one for RISC and the other for DSP. The DMA channel can only generate interrupts when the corresponding bits in this register and DMA_CH_DSP_EN are set correctly.

Bit	Name	Default	Description
15:0 (R/W)	EN	16'h0	DMA interrupt enable bits: 1: The DMA interrupt is enabled. 0: The DMA interrupt is disabled.
31:16	-	16'h0	Reserved

Table 479: DMA Interrupt Enable Register

NOTE – RISC can read or write its bank of registers either in DSP mode or RISC mode. DSP can write its bank of register only in DSP mode and can read either in DSP mode or RISC mode.

- DMA Channel DSP Control Register (DMA_CH_DSP_CTRL) – RISC: 0x12C

  This register is accessible to RISC only. This signal only indicates the DMA channel is accessed by DSP or RISC and cannot be the interrupt mask. For example, if DSP_EN [15:0] is 16'h1, then DMA channel0 will be accessed by DSP while other channels are accessed by RISC. Yet interrupts of channel1-channel15 can still be sent to DSP, because interrupts of all channels are ORed and there are no registers to flag the interrupt source.

Bit	Name	Default	Description
15:0 (R/W)	DSP_EN	16'h0	DMA channel DSP control enable bits: 1: The DMA channel is controlled by DSP. 0: The DMA channel is controlled by RISC.
31:16	-	16'h0	Reserved

Table 480: DMA Channel DSP Control Register

- DMA Channel Loop Control Register (DMA_CH_LOOP_CTRL) – RISC: 0x130, DSP: 0x8C~8D

  Similar to DMA_WIDTH registers and the DMA_INT_EN register, there are two separate DMA_CH_LOOP_CTRL registers, one for RISC and one for DSP, depending on the current channel which is controlled by RISC or DSP, in order to decide which DMA_CH_LOOP_CTRL register is being used.

Bit	Name	Default	Description
15:0 (R/W)	BUFA_VALID	16'h0	1: The first section of the loop area (BUFA) is valid. 0: The first section of the loop area (BUFA) is invalid.
31:16 (R/W)	BUFB_VALID	16'h0	1: The second section of the loop area (BUFB) is valid. 0: The second section of the loop area (BUFB) is invalid.

Table 481: DMA Channel Loop Control Register

NOTE – RISC can read or write its bank of register either in DSP mode or RISC mode, but DSP can only write its bank of register in DSP mode and read either in DSP mode or RISC mode.

USP Controller

Overview

Universal Serial Ports (USP) are used for serial communication in which only 1 bit is transmitted at a time. The advantage of a serial port is that it requires relatively few pins so it is often more cost-effective than parallel ports (especially in long-range communications). The serial port is a general purpose interface that transmits or receives data one bit at a time and can be used for almost any device types. There are 3 main USP device categories:

- PC peripherals: modem, mouse, printer and so on
- Communication devices: Cable modem, ISDN, and xDSL
- Embedded systems: A/D, D/A converters, RF modules, and serial EEPROM

The SiRFatlasV USP (Universal Serial Port) is a multifunctional serial interface that communicates with many common serial ports. "Universal" does not mean that it can interface with any types of serial devices, but only those that are USP-compliant can be connected via the USP interface. Compared to existing designs, the SiRFatlasV USP offers better expendability, configurability and flexibility.

For configuring the right frame format, users can set parameters such as data length and transmitting data length for the transfer frame. Depending on connection protocols, the frame format can be configured as either a full-duplex asynchronous system that communicates with peripheral devices such as CRT terminals and modems, or as a half-duplex synchronous system that communicates with peripheral devices such as A/D or D/A integrated circuits, serial EEPROMs, or touch panel controllers.

There are a total of two USPs, identified as USP0 and USP1 on the SiRFatlasV processor. These USPs are identical to each other.

Feature List

Depending on the clock used, the SiRFatlasV USP can be configured to the following modes:

- Asynchronous (no clock is needed.)
  - UART, IrDA: Maximum data rate = 3 Mbps
- Synchronous
  - DSP and PCM for both Master and Slave mode
  - SPI for Master mode
  - I^2S for Slave mode
  - SIM Card Host timing

    When USP works in Master mode, the clock is generated by USP, and maximum clock frequency = 1/6 of the I/O clock.

  - When USP works in Slave mode, the clock is generated by the external device, and maximum clock frequency = 1/20 the I/O clock.

Depending on the type of the serial bus used, the SiRFatlasV USP may be able to support the following types of serial ports:

- ASYNC (UART and IrDA)
- SPI bus (such as serial EEPROM 25C040 and Philsar PH2401)
- I^2S bus (such as WM8978)

The SiRFatlasV USP provides 128-byte data FIFO which supports DMA and I/O modes. It also supports RISC control pin (GPIO mode).

Pin Descriptions

External Pin Descriptions

The following table shows the pins used in the USP controller and their functions. The USP controller pins are multiplexed with other devices. Refer to the section for Pin Sequence for more information.

Pin Name	I/O Type	Pin MUX	Default Function	Default Status	Description
X_USCLK_0	Bi	USP0 GPIO	GPIO	Input, pull-down	USP0 Serial Clock/Smart Card Clock output
X_UTXD_0	Bi	USP0 GPIO	GPIO	Input, pull-down	USP0 Transmit Data/Smart Card bidirectional serial data input
X_URXD_0	Bi	USP0 GPIO	GPIO	Input, pull-down	USP0 Receive Data
X_UTFS_0	Bi	USP0 GPIO	GPIO	Input, pull-down	USP0 Transmit Frame Sync/Smart Card RST output
X_URFS_0	Bi	USP0 I^2S GPIO	GPIO	Input, pull-down	USP0 Receive Frame Sync
X_USCLK_1	Bi	USP1 GPIO	GPIO	Input, pull-down	USP1 Serial Clock/Smart Card Clock output
X_UTXD_1	Bi	USP1 GPIO	GPIO	Input, pull-down	USP1 Transmit Data/Smart Card bidirectional serial data inout
X_URXD_1	Bi	USP1 GPIO	GPIO	Input, pull-down	USP1 Receive Data
X_UTFS_1	Bi	USP1 GPIO	GPIO	Input, pull-down	USP1 Transmit Frame Sync/Smart Card RST output
X_URFS_1	Bi	USP1 I^2S GPIO	GPIO	Input, pull-down	USP1 Receive Frame Sync

Table 482: USP External Pin Description

NOTE – Each USP pin can be configured to USP or I/O functions. Firstly you should configure pins of USP to be USP function. If only one of the five pins is used for the serial interface, then the rest can be set to I/O mode through register USP_MODE_REG1.You can read or write the USP_PIN_IO_DATA register if you configure it to be INPUT or OUTPUT mode.

Functional Descriptions

Block Diagram

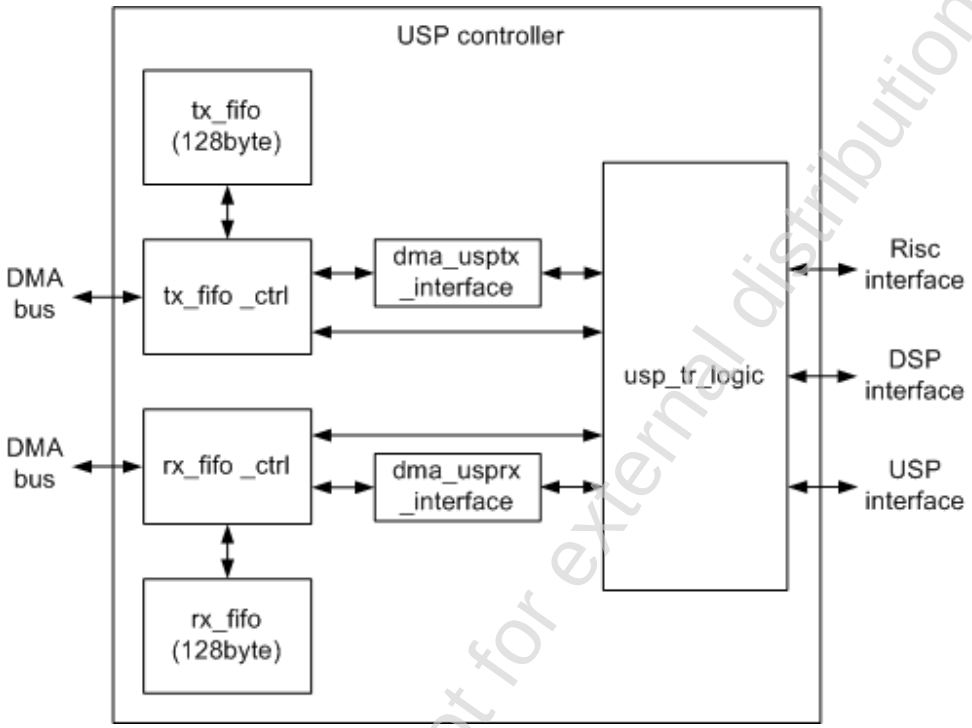


Figure 74: Block Diagram of the USP

Module Descriptions

Top Level Description

There are two USPs in SiRFatlasV (USP0~1), each USP provides three interfaces:

- RISC
- DSP
- DMA

The SiRFatlasV USP can be accessed through either the I/O or DMA interface. In I/O mode, either RISC or DSP can access USP. In DMA mode however, it is controlled by the DMA controller in the I/O Bridge. Each USP can be independently configured to the DMA interface.

Each USP have two DMA channels. Please refer to the section of “DMA Controller” for more details.

All registers can be accessed in I/O mode by the RISC or DSP, but not simultaneously. The register USP_RISC_DSP_MODE can only be written by the RISC; it determines whether the USP can be accessed and controlled by the RISC or DSP. If the bit is configured to 0, then RISC will access USP only. Because the DSP data bus is 16-bit, if DSP accesses the USP register, then it must access it two times in order to completely read or write a 32-bit register.

There are three methods for SiRFatlasV to access the TX_FIFO and RX_FIFO of USP:

- I/O mode with Interrupt
- I/O mode without Interrupt
- DMA mode

The TX_FIFO and the RX_FIFO can be accessed in I/O mode. Both TX_FIFO and RX_FIFO have the interrupt state registers that reflect their state. The operation in the I/O mode usually uses these interrupts to decide whether to write or read. If the DMA accesses TX_FIFO and the RX_FIFO, then they cannot be accessed by RISC or DSP.

NOTE – In all three modes, before the transmitting operation begins, the TX_FIFO must be reset by setting the FIFO_RESET bit of the TX_FIFO_OP register to 1, and then it would be started by clearing FIFO_RESET and by setting the FIFO_START bit of the TX_FIFO_OP register. The same procedure can be applied to RX_FIFO before it is operated.

- I/O mode with interrupt

  The I/O mode with interrupt is suitable for those time-sensitive, small-data cases. There are four interrupts for the transmit operation, TX_DONE, TXFIFO_SERVE, TXFIFO_EMPTY and TX_UFLOW.

  After data is put from TX_FIFO to tx_shifter and the transmitting operation starts, TX_DONE interrupt will occur. In addition, if the number of data in TX_FIFO is less than the parameter defined in the FIFO_THD bits of register USP_TX_FIFO_CTRL, then the TXFIFO_SERVE interrupt will take place. The TXFIFO_EMTPY interrupt will occur when TX_FIFO is empty. The TX_UFLOW interrupt is used in case of underflow while the transmitting operation is in progress.

  After enabling these interrupts, RISC/DSP can monitor them while transmitting data. If data is only transmitted one at a time, then the TX_DONE interrupt should be enough to handle this operation. However if several data needs to be transmitted, then the TXFIFO_SERVE interrupt or the TXFIFO_EMPTY interrupt would be useful to improve efficiency. When the TXFIFO_EMPTY interrupt takes place, software can write several new data into TX_FIFO.

  There are four interrupts for the receiving operation, RX_DONE, RXFIFO_SERVE, RXFIFO_FULL and RX_OFLOW. The receiving operation is similar to the transmitting. After an interrupt occurs, RISC/DSP will be able to read data out from RX_FIFO. Both the RXFIFO_SERVE and RXFIFO_FULL interrupts are useful for reading several data from the RX_FIFO at a time.

- I/O mode without interrupt

  This mode is suitable for time-sensitive, small-data cases. RISC/DSP can poll the USP_TXFIFO_STATUS and USP_RXFIFO_STATUS registers before the transmitting or receiving operation. If TX_FIFO is not full or RX_FIFO is not empty, then RISC/DSP can write or read new data from the TX_FIFO or the RX_FIFO.

- DMA mode

  This mode is suitable for large-data cases. If the transmitting/receiving FIFO is set to the DMA mode, then RISC/DSP will not be able to access the data. For the transmitting operation, first RISC/DSP will set the number of data to be transmitted in USP_TX_DMA_IO_LEN, then resets and starts the TX_FIFO, DMA can be started. The DMA controller will transfer data from the memory to TX_FIFO until all data has been transmitted. The DMA receiving operation is similar to the transmitting operation. The number of data to be received is stored in the

USP_RX_DMA_IO_LEN. DMA controller will read data from the RX_FIFO register to the memory until all data has been received.

TX and RX FIFO

TX and RX FIFO are the data FIFOs between the USP interface and the DMA interface or between the USP interface and the I/O interface (including both RISC I/O and DSP I/O). All data must be stored to these two FIFOs first, and then fetched by the RISC I/O interface, DSP I/O interface, DMA interface or the USP interface. The FIFO size is 128 bytes.

TX and RX DMA Interface

Dma_usptx_interface and dma_usprx_interface are the bridges between usp_tr_logic (the USP main logic block) and the FIFO controller. They transform the data protocol between the usp_tr_logic data interface and the FIFO controller data interface. There are no FIFOs in these two blocks.

USP TR Logic

Usp_tr_logic is the main logic block in the USP controller which includes four blocks: uart_bound, uart_reg_file, uart_tx and uart_rx.

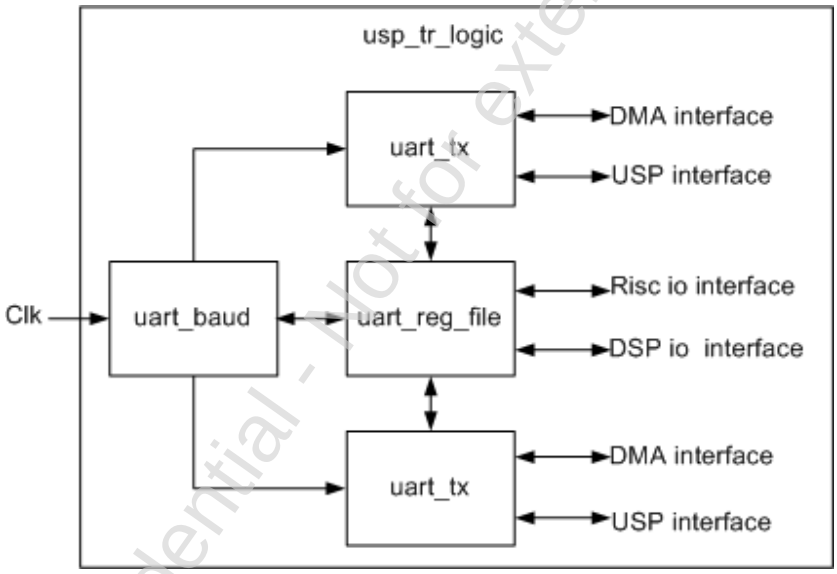


Figure 75: usp_tr_logic Block Diagram

- uart_baud:

  The uart_baud block is the baud rate controller block; it generates different baud rate sync signals to the uart_tx and uart_rx block through different modes and baud rate configuration. The mode and configuration are inputs from the uart_reg_file block and set by either RISC or DSP.

- uart_reg_file:

  This block is the USP registers module; it includes all USP registers that can be accessed by RISC and DSP. This block outputs all USP control signals and inputs USP status signals.

- uart_rx:

  The uart_rx block is the USP receiving logic block. It receives data input from the USP port, then decodes the format, transforms serial data to parallel and transmits it to the USP RX FIFO.

- uart_tx:
  The uart_tx is the USP transmit logic block. It gets data from USP TX FIFO and sends them to the USP port. The transmit data format is set by the RISC or DSP.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
USP0 DSP I/O interface	0x300
USP0 RISC I/O interface	0x80010000
USP1 DSP I/O interface	0x380
USP1 RISC I/O interface	0x80020000

Table 483: USP Base Address

Register Mapping

Each USP in SiRFatlasV has the same set of registers, therefore the offset addresses are the same.

The actual address of the USP register is equal to the USP base address plus the offset address:

- USP0 base address 0x80010000
- USP1 base address 0x80020000

The following table shows all USP registers and their offset addresses:

RISC Address <11:0>	DSP I/O Address <7:0>	Register	Description
0x0	0x00~0x01	USP_MODE1	USP mode setting register 1
0x4	0x02~0x03	USP_MODE2	USP mode setting register 2
0x8	0x04~0x05	USP_TX_FRAME_CTRL	USP transmit frame control register
0xC	0x06~0x07	USP_RX_FRAME_CTRL	USP receive frame control register
0x10	0x08	USP_TX_RX_ENABLE	USP transmit and receive enable register
0x14	0x0A(0x0B)	USP_INT_ENABLE	USP interrupt enable register
0x18	0x0C(0x0D)	USP_INT_STATUS	USP interrupt register
0x1C	0x0E	USP_PIN_IO_DATA	USP pin I/O data register
0x20	0x10	USP_RISC_DSP_MODE	USP accessing select register

RISC Address <11:0>	DSP I/O Address <7:0>	Register	Description
0x24	0x12~0x13	USP_AYSNC_PARAM_REG	USP ASYNC parameter
0x28	0x14	USP_IRDA_X_MODE_DIV	USP IrDA X_Mode clock divider
0x2C	0x16	USP_SM_CFG	USP smart card configure
0x30	0x18	USP_TRX_ LEN_HI	USP Tx/Rx length high register
0x34	0x1A	USP_TX_SYNC_LEN_HI	USP TX SYNC length high register
0x38~FC	-	-	Reserved
0x100	0x40	USP_TX_DMA_IO_CTRL	USP TXFIFO DMA/IO register
0x104	0x42~0x43	USP_TX_DMA_IO_LEN	USP transmit data length register
0x108	0x44	USP_TXFIFO_CTRL	USP TXFIFO control register
0x10C	0x46~0x47	USP_TXFIFO_LEVEL_CHK	USP TXFIFO check level register
0x110	0x48	USP_TXFIFO_OP	USP TXFIFO operation register
0x114	0x4A	USP_TXFIFO_STATUS	USP TXFIFO status register
0x118	0x4C	USP_TXFIFO_DATA	USP TXFIFO bottom
0x11C	-	-	Reserved
0x120	0x50	USP_RX_DMA_IO_CTRL	USP RXFIFO DMA/IO register
0x124	0x52~0x53	USP_RX_DMA_IO_LEN	USP receive length register
0x128	0x54	USP_RXFIFO_CTRL	USP RXFIFO control register
0x12C	0x56~x57	USP_RXFIFO_LEVEL_CHK	USP RXFIFO check level register
0x130	0x58	USP_RXFIFO_OP	USP RXFIFO operation register
0x134	0x5A	USP_RXFIFO_STATUS	USP RXFIFO status register
0x138	0x5C	USP_RXFIFO_DATA	USP RXFIFO bottom
0x13C	-	-	Reserved

Table 484: Address Mapping

Register Descriptions

- USP Mode Register 1 (USP_MODE1) – RISC: 0x0, DSP: 0x0~0x1

Bit	Name	Default	Description
0 (R/W)	SYNC_MODE	1'b0	USP operation mode: 0: Asynchronous mode 1: Synchronous mode
1 (R/W)	CLOCK_MODE	1'b0	USP clock mode: 0: Master mode If USP is in asynchronous mode, it must be set to 1. 1: Slave mode, clock is input from other device
2 (R/W)	LOOP_BACK_EN	1'b0	USP transmit data loop back mode: 0: No loop back 1: Loop back (The receiving data comes from the transmit data.)
3 (R/W)	HPSIR_EN	1'b0	IrDA function (SYNC_MODE must be set to 0): 0: Disabled 1: Enabled
4 (R/W)	ENDIAN_CTRL	1'b0	Transmit/receive data endian mode: 0: Big endian (MSB first) 1: Little endian (LSB first)
5 (R/W)	USP_EN	1'b0	USP operation enable signal: 0: Reset USP and its transmitted and received control registers 1: The reset of all above registers will be released, and the USP can operate.
6 (R/W)	RXD_ACT_EDGE	1'b0	Receive data is driven at: 0: SCLK rising edge 1: SCLK falling edge
7 (R/W)	TXD_ACT_EDGE	1'b0	Transmit data is driven at: 0: SCLK rising edge 1: SCLK falling edge
8 (R/W)	RFS_ACT_LEVEL	1'b0	Receive sync signal (RFS)/async start bit valid level: 0: Logic 0 1: Logic 1
9 (R/W)	TFS_ACT_LEVEL	1'b0	Transmit sync signal (TFS) valid level: 0: Logic 0 1: Logic 1
10	SCLK_IDLE_MODE	1'b0	In frame idle state, SCLK mode:

Bit	Name	Default	Description
(R/W)			0: Stops and keeps the sclk idle level 1: Continues to toggle
11 (R/W)	SCLK_IDLE_LEVEL	1'b0	In frame idle state, SCLK will stop at level: 0: Logic 0 1: Logic 1
12 (R/W)	SCLK_PIN_MODE	1'b0	SCLK pin operation mode: 0: USP mode (SCLK) 1: I/O mode
13 (R/W)	RFS_PIN_MODE	1'b0	RFS pin operation mode: 0: USP mode (RFS) 1: I/O mode
14 (R/W)	TFS_PIN_MODE	1'b0	TFS pin operation mode: 0: USP mode (TFS) 1: I/O mode
15 (R/W)	RXD_PIN_MODE	1'b0	RXD pin operation mode: 0: USP mode (RXD) 1: I/O mode
16 (R/W)	TXD_PIN_MODE	1'b0	TXD pin operation mode: 0: USP mode (TXD) 1: I/O mode
17 (R/W)	SCLK_IO_MODE	1'b0	SCLK pin input/output mode (when SCLK_PIN_MODE=1): 0: Output mode 1: Input mode
18 (R/W)	RFS_IO_MODE	1'b0	RFS pin input/output mode (when RFS_PIN_MODE =1): 0: Output mode 1: Input mode
19 (R/W)	TFS_IO_MODE	1'b0	TFS pin input/output mode (when TFS_PIN_MODE =1) 0: Output mode 1: Input mode
20 (R/W)	RXD_IO_MODE	1'b0	RXD pin input/output mode (when RXD_PIN_MODE =1): 0: Output mode 1: Input mode
21 (R/W)	TXD_IO_MODE	1'b0	TXD pin input/output mode (when TXD_PIN_MODE =1): 0: Output mode 1: Input mode

Bit	Name	Default	Description
29:22 (R/W)	IRDA_WIDTH_DIV	8'h0	The IrDA data pulse width register. If users want to transmit/receive 1.6us width data by setting the IRDA_DATA_WIDTH bit in USP_MODE2 register, then this register must be set according to the I/O clock frequency. IRDA_WIDTH_DIV=1.6μs*f_{ioclk} + 1
30 (R/W)	IrDA_IDLE_LEVEL	1'b0	IrDA received data IDLE level: 0: Low level for idle state 1: High level for idle state
31 (R/W)	TX_UFLOW_REPEAT	1'bO	TX underflow repeat mode: 0: Repeat the last transmitted data 1: Repeat 0

Table 485: USP Mode Register 1

- USP Mode Register 2 (USP_MODE2) – RISC: 0x4, DSP: 0x2~0x3

Bit	Name	Default	Description
7:0 (R/W)	RXD_DELAY_LEN	8'h0	Delay clock number between FS and the first receive data. It should be set to actual_delay_length in both the slave and master mode of RFS. The actual_delay_length should be at least 1.
15:8 (R/W)	TXD_DELAY_LEN	8'h0	Delay clock number between FS and the first transmit data: When TFS is slave, it must be set to actual_delay_length - 1 When TFS is master, it must be set to actual_delay_length The actual_delay_length should be at least 1.
16 (R/W)	ENA_CTRL_MODE	1'b0	USP_TX_RX_ENABLE register operation mode: 1'b0: TX_ENA/RX_ENA will not be cleared after the current operation is finished. The TX_ENA/RX_ENA bit only needs to be set once before multi-time transmit/receive operation begins. 1'b1: The TX_ENA/RX_ENA bit in the USP_TX_RX_ENABLE register will be cleared automatically after the previous operation finishes, and TX_ENA/RX_ENA bit must be set before a new transmit/receive operation begins.
17 (R/W)	FRAME_CTRL_MODE	1'b0	USP transmit and receive operation control mode: 0: New frame transmit/receive will begin only with new data in TX_FIFO. 1: After the USP operation starts, frame

Bit	Name	Default	Description
			transmit/receive will continue to send data in USP_TX_DATA regardless of whether there is new data received.
18 (R/W)	TFS_SOURCE_MODE	1'b0	The sync signal source mode: 0: TFS is generated by hardware 1: TFS is generated by software from the upper 16-bit of the 32-bit data in TX_FIFO.
19 (R/W)	RFS_MS_MODE	1'b0	Receive sync signal source mode: 0: RFS is in master mode and output 1: RFS is in slave mode and input from an external device.
20 (R/W)	TFS_MS_MODE	1'b0	Transmit sync signal source mode: 0: TFS is master mode and output 1: TFS is slave mode and input from the external device
30:21 (R/W)	USP_CLK_DIVISOR	10'h0	USP serial clock divider. For ASYNC mode(General): USP_CLK_DIVISOR = FLOOR ((2*fio_clock/ (baud * (ASYNC_DIV2+1)) + 1)/2–1) For Smart Card Mode USP_CLK_DIVISOR= FLOOR (((2*USP_SM_CLK_DIV*F_div_D)/ (ASYNC_DIV2+1) +1)/2 -1) **NOTE** - F_div_D comes from the software parameter, default is 372 For SYNC mode: USP_CLK_DIVISOR = fio_clock/ (fsclk * 2) – 1 Here 10bits are the lowest bits. The [11:10] bits are in the USP_TX_FRAME_CTRL register The highest four bits are in the USP_RX_FRAME_CTRL register
31 (R/W)	IRDA_DATA_WIDTH	1'b0	IrDA logic 1 pulse width select (ASYNC_DIV2 must be set to 16 in IrDA mode): 0: Data pulse width is 3/16 of a bit time wide. 1: Data pulse width is 1.6 µs.

Table 486: USP Mode Register 2

- USP Transmit Frame Control Register (USP_TX_FRAME_CTRL) – RISC: 0x8, DSP: 0x4~0x5

Bit	Name	Default	Description
7:0 (R/W)	TX_DATA_LEN[7:0]	8'h0	Low byte of data length in one frame: Must be set to actual_tx_data_length -1
15:8 (R/W)	TX_SYNC_LEN[7:0]	8'h0	Valid length of transmitted sync signal (TFS) in one frame: Must be set to actual_tfs_valid_length -1
23:16 (R/W)	TX_FRAME_LEN[7:0]	8'h0	Low byte of transmit frame length includes active state and idle state: Must be set to actual_tx_frame_length -1
28:24 (R/W)	TX_SHIFTER_LEN	5'h0	Data length in the transmit shifter in one transmit operation: Must be set to actual_tx_shifter_length -1
29 (R/W)	SLAVE_CLK_SAMPLE	1'b0	When sampling slave SCLK by I/O clock, setting this bit can avoid the glitch of the slave SCLK. 0: No filters for glitch of slave clock and the fmax of the slave SCLK is 1/8 of I/O clock. 1: Filter for glitch of slave clock and the fmax of slave SCLK is 1/10 of I/O clock.
31:30 (R/W)	USP_CLK_DIVISOR	2'h0	This is the two bit [11:10] of USP_CLK_DIVISOR in USP_MODE2.

Table 487: USP Transmit Frame Control Register

In the ASYNC mode (RS232 or IrDA), this register has different meanings:

TX_DATA_LEN = Data bit number – 1

TX_FRAME_LEN = Start bit number + Data bit number + Stop bit number – 1

TX_SHIFTER_LEN = Data bit number – 1

The Parity bit is not supported in USP but is supported in the Smart Card mode. The Stop bit number does not support 1.5 bit and should be set to TXD_DELAY_LEN in the USP_MODE2 register.

NOTE – In Smart Card mode, the transmit data is always 8 bits wide with 1bit even parity bit:

```
TX_DATA_LEN = 0x07                    //8 bits data
TX_SYNC_LEN=0x09                      //1 start + 8 data + 1 even parity
TX_FRAME_LEN = 0x09 + idle_bit_num //1 Start bit + 8 Data Bit + 1
                                      //parity + >=1 (guard + idle)
TX_SHIFTER_LEN=0x08                   //8 data bit + 1 even parity
```

- USP Receive Frame Control Register (USP_RX_FRAME_CTRL) – RISC: 0xC, DSP: 0x6~0x7

Bit	Name	Default	Description
7:0 (R/W)	RX_DATA_LEN[7:0]	8'h0	Low byte of received data length in one frame: Must be set to actual_rx_data_length -1
15:8 (R/W)	RX_FRAME_LEN[7:0]	8'h0	Low byte of received frame length includes active and idle state: Must be set to actual_rx_frame_length -1.
20:16 (R/W)	RX_SHIFTER_LEN	5'h0	Data length in the receive shifter in one transmit operation: Must be set to actual_rx_shifter_length -1.
21(R/W)	I2S_SYNC_CHG	1'b0	1'b0: USP will not change the input RFS signal. 1'b1: USP has an I^2S mode device and is working in the slave mode, it will change the input RFS signal into a DSP mode sync signal. Every frame I^2S signal will be changed into two DSP mode sync signals
22	SINGLE_SYNC_MODE	1'b0	Fame sync pin source select for RX: 0: From RFS 1: From TFS
23	START_EDGE_MODE	1'b0	Start bit edge dectect mode: 0: dectect level for start bit 1: dectect edge for start bit This bit should be configured with RFS_ACT_LEVEL to decide which level/edge of the start bit should be. (START_EDGE_MODE, RFS_ACT_LEVEL) : (0,0): Low level (0,1): High level (1,0): Falling edge (1,1): Rising edge
27:24 (R/W)	USP_CLK_DIVISOR	4'h0	This is the four highest bit [15:12] of USP_CLK_DIVISOR in the USP_MODE2
31:28	-	4'h0	Reserved

Table 488: USP Receive Frame Control Register

In ASYNC mode (RS232 or IrDA), this register has different meanings:

RX_DATA_LEN = Data bit number – 1

RX_FRAME_LEN = Start bit number + Data bit number + Stop bit number – 1

RX_SHIFTER_LEN = Data bit number – 1

The Parity bit is not supported in USP but is supported in Smart Card mode. The Stop bit number does not support 1.5 bit and should be set to TXD_DELAY_LEN in the USP_MODE2 register.

NOTE – For Smart Card Mode, the received data is always 8-bit width with 1-bit even parity bit.

```
RX_DATA_LEN = 0x07    //8 bit data
RX_FRAME_LEN = 0x0a   //1 Start bit + 8 Data Bit + 1 parity +
                      //1 (guard //time)
RX_SHIFTER_LEN=0x08   //8 data bit + 1 even parity
```

- USP Transmit/Receive Enable Register (USP_TX_RX_ENABLE) – RISC: 0x10, DSP: 0x8

  If ENA_CTRL_MODE=0 (USP_MODE_REG2), the TX_ENA/RX_ENA bits will not be cleared automatically after Tx/Rx operation finishes; these bits only need to set one time before first Tx/Rx operation.

  If ENA_CTRL_MODE =1, the TX_ENA/RX_ENA bits will be cleared automatically after Tx/Rx operation finishes, they need to be set again before next transmit/receive operation.

  RX_ENA/TX_ENA can be set according to the requirements of the USP that interfaces with external devices.

Bit	Name	Default	Description
0 (R/W)	RX_ENA	1'b0	Receive enable bit: 0: Disabled 1: Enabled
1 (R/W)	TX_ENA	1'b0	Transmit enable bit: 0: Disabled 1: Enabled
31:2	-	30'h0	Reserved

Table 489: USP Transmit/Receive Enable Register

- USP Interrupt Enable Register (USP_INT_ENABLE) – RISC: 0x14, DSP: 0xA~0xB

Bit	Name	Default	Description
0 (R/W)	RX_DONE_INT_EN	1'b0	Receive done interrupt enable: 0: Disabled (same for the bits below) 1: Enabled (same for the bits below)
1 (R/W)	TX_DONE_INT_EN	1'b0	Transmit done interrupt enable
2 (R/W)	RX_OFLOW_INT_EN	1'b0	Receive overflow interrupt enable
3 (R/W)	TX_UFLOW_INT_EN	1'b0	Transmit underflow interrupt enable
4 (R/W)	RX_IO_DMA_ INT_EN	1'b0	IO/DMA receive interrupt enable

Bit	Name	Default	Description
5 (R/W)	TX_ IO_DMA_ INT_EN	1'b0	IO/DMA transmit interrupt enable
6 (R/W)	RXFIFO_FULL_INT_EN	1'b0	Receive FIFO full interrupt enable
7 (R/W)	TXFIFO_EMPTY_INT_EN	1'b0	Transmit FIFO empty interrupt enable
8 (R/W)	RXFIFO_THD_INT_EN	1'b0	Receive FIFO threshold interrupt enable
9 (R/W)	TXFIFO_THD_INT_EN	1'b0	Transmit FIFO threshold interrupt enable
10 (R/W)	UART_FRM_ERR_INT_EN	1'b0	UART error frame interrupt enable
11 (R/W)	USP_RX_TIMEOUT_INT_EN	1'b0	USP receive timeout interrupt enable
12 (R/W)	USP_TX_ALLOUT_INT_EN	1'b0	USP all transmit out interrupt enable
13	-	1'b0	Reserved
14 (R/W)	USP_RX_PLUG_INT_EN	1'b0	USP RXD Rising edge when IDLE interrupt enable
15 (R/W)	USP_RX_BREAK_INT_EN	1'b0	USP RX break interrupt enable
<31:16>	-	16'h0	Reserved, read as 16'h0

Table 490: USP Interrupt Enable Register

- USP Interrupt Status Register (USP_INT_STATUS) – RISC: 0x18, DSP: 0xC~0xD
  RISC/DSP writes a 1'b1 to the bit will clear that interrupt.

Bit	Name	Default	Description
0 (R/W)	RX_DONE	1'b0	Valid data has been received in the RXFIFO interrupt (valid data length is define by RXFIFO_WIDTH): 0: Invalid 1: Valid
1 (R/W)	TX_DONE	1'b0	Valid data has been transmitted from the TXFIFO interrupt (valid data length is define by TXFIFO_WIDTH): 0: Invalid 1: Valid
2 (R/W)	RX_OFLOW	1'b0	RXFIFO overflow Interrupt: 0: Invalid 1: Valid
3 (R/W)	TX_UFLOW	1'b0	TXFIFO underflow interrupt: 0: Invalid 1: Valid
4 (R/W)	DMA_IO_RX_ DONE	1'b0	RXFIFO has received all data of the data package, the size of which is defined by USP_RX_DMA_IO_LEN_REG: 0: Invalid

Bit	Name	Default	Description
			1: Valid
5 (R/W)	DMA_IO_TX_ DONE	1'b0	TXFIFO has transmitted all data of the data package, the size of which is defined by USP_TX_DMA_IO_LEN_REG: 0: Invalid 1: Valid
6 (R/W)	RXFIFO_FULL	1'b0	RXFIFO full interrupt: 0: Invalid 1: Valid
7 (R/W)	TXFIFO_EMP TY	1'b0	TXFIFO empty interrupt (This will be asserted after power up): 0: Invalid 1: Valid
8 (R/W)	RXFIFO_THD _REACH	1'b0	It is the time to read USP_ RXFIFO when the number of data in USP_ RXFIFO reaches the threshold: 0: Invalid 1: Valid
9 (R/W)	TXFIFO_THD _REACH	1'b0	It is the time to write USP_TXFIFO when the number of data in the USP_TXFIFO reaches the threshold. (This will be asserted after power up.) 0: Invalid 1: Valid
10 (R/W)	UART_FRM_E RR	1'b0	Receive an error UART frame interrupt: 0: Invalid 1: Valid
11 (R/W)	USP_RX_TIM EOUT	1'b0	If this bit is set to 1, there is no more new data received for the time specified by the timeout bit number in USP_AYSNC_PARAM_REG.
12 (R/W)	USP_TX_ALL OUT	1'b0	If this bit is become 1, the data in both tx_fifo and tx shifter has been transmitted out: 0: Disabled 1: Enabled
13	-	1'b0	Reserved
14 (R/W)	USP_PLUG_I N	1'b0	USP Plug-in valid status, if USP RX is in IDLE state and dectect a rising edge for rxd, this status will be set. 0: Invalid 1: Valid
15 (R/W)	USP_RX_BRE AK	1'b0	USP Rx break status If this bit becomes 1, the RXD line has become low level for at least 10 bits and the receiving action must be stopped. To correct this, disable the RX_ENA bit in the USP_TX_RX_ENABLE register, then wait for some time to

Bit	Name	Default	Description
			double check this bit. If no new break occurs, then enable RX_ENA and wait for some time to reset and restart RXFIFO, this will flush invalid data and prepare to accept new data. 0: Invalid 1: Valid
<31:16>	-	16'h0	Reserved, read as 16'h0

Table 491: USP Interrupt Status Register

- USP Pin I/O Data Register (USP_PIN_IO_DATA) – RISC: 0x1C, DSP: 0xE

  These bits can only be accessed by RISC or DSP when the corresponding USP pins are in I/O mode.

Bit	Name	Default	Description
0 (R/W)	RFS_PIN_VALUE	1'b0	Pin value of GPIO function (RFS)
1 (R/W)	TFS_PIN_VALUE	1'b0	Pin value of GPIO function (TFS)
2 (R/W)	RXD_PIN_VALUE	1'b0	Pin value of GPIO function (RXD)
3 (R/W)	TXD_PIN_VALUE	1'b0	Pin value of GPIO function (TXD)
4 (R/W)	SCLK_PIN_VALUE	1'b0	Pin value of GPIO function (SCLK)
31:5	-	27'h0	Reserved

Table 492: USP Pin I/O Data Register

- USP RISC/DSP Mode Register (USP_RISC_DSP_MODE) – RISC: 0x20, DSP: 0x10

  This register can only be written by RISC. For DSP, it is read-only.

Bit	Name	Default	Description
0 (R/W)	RISC_DSP_SEL	1'b0	0: USP registers are accessed by RISC. 1: USP registers are accessed by DSP.
31:1	-	31'h0	Reserved

Table 493: USP RISC/DSP Mode Register

- USP ASYNC PARAMETER Register (USP_AYSNC_PARAM_REG) – RISC: 0x24, DSP: 0x12

  This register can only be written by RISC. For DSP, it is read-only.

Bit	Name	Default	Description
15:0 (R/W)	Aysnc_timeout_num	16'hffff	This parameter specifies the TIMEOUT bit number for the receive operation. The RX timeout interrupt will be triggered if no more data is received from the timeout bit number time since the last bit is received.

Bit	Name	Default	Description
21:16 (R/W)	ASYNC_DIV2	6'b0	The parameter is used to cooperate with the USP_CLK_DIVISOR in the USP_MODE2 register to generate the right baud rate for ASYNC mode when IOCLK is in low frequency.
31:22	-	10'h0	Reserved.

Table 494: USP ASYNC PARAMETER Register

- USP IRDA X MODE Divider Register (USP_IRDA_X_MODE_DIV) – RISC: 0x28, DSP: 0x14

Bit	Name	Default	Description
7:0 (R/W)	USP_IRDA_X_MODE_DIV	8'haa	This register is used for the USP working in the IrDA mode which receiving data in X_mode.
31:8	-	24'h0	Reserved.

Table 495: USP IRDA X MODE Divider Register

- USP Smart Card Configure Register(USP_SM_CFG) – RISC: 0x2C, DSP: 0x16

  If the USP is configured as a Smart Card controller, then this register must be configured. This register is not used in other USP modes.

Bit	Name	Default	Description
6:0 (R/W)	USP_SM_Clk_div	7'h18	This is active when USP_SM_EN is active; otherwise default value will be always kept. USP Smart Card Clock Generation divider, the output clock frequency is fio_cloc / (USP_SM_CLK_DIV+1)
7(R/W)	USP_SM_EN	1'b0	USP smart card mode enable 1: Enabled 0: Disabled When this bit is enabled: X_USCLK_* will output the clock for Smart Card X_TXD will be configured to bi-directional pin PAD X_TXD is working with external pull up (about 20K) Character repetition will be enabled automatically and the even parity will also be added automatically.
8(R/W)	USP_SM_CLK_STP	1'b0	This is for clock stop mode when the USP is configured to a smart card interface device: 1: Stop clock 0: Enable clock (default) When the clock is stopped, the default output value of CLK will come from SCLK_PIN_VALUE (bit 4 of

Bit	Name	Default	Description
			USP_PIN_IO_DATA), in this case, SCLK pin should be configured as GPIO mode.
31:9	-	24'h0	Reserved

Table 496: USP Smart Card Configure Register

- USP TX/RX Length High Register(USP_TRX_LEN_HI)-RISC: 0x30, DSP:0x18

Bit	Name	Default	Description
7:0 (R/W)	TX_DATA_LEN[15:8]	8'h0	High byte of data length in one frame: Must be set to actual_tx_data_length -1
15:8 (R/W)	TX_FRAME_LEN[15:8]	8'h0	High byte of transmit frame length includes active state and idle state: Must be set to actual_tx_frame_length -1
23:16 (R/W)	RX_DATA_LEN[15:8]	8'h0	High byte of received data length in one frame: Must be set to actual_rx_data_length -1
31:24 (R/W)	RX_FRAME_LEN[15:8]	8'h0	High byte of received frame length includes active and idle state: Must be set to actual_rx_frame_length -1

Table 497: USP TX/RX Length High Register

- USP TX SYNC Length High Register(USP_TX_SYNC_LEN_HI)-RISC: 0x34, DSP:0x1a

Bit	Name	Default	Description
7:0 (R/W)	TX_SYNC_LEN[15:8]	8'h0	High byte of valid length of transmitted sync signal (TFS) in one frame: Must be set to actual_tfs_valid_length -1

Table 498: USP TX SYNC Length High Register

- USP TX DMA I/O MODE Register (USP_TX_DMA_IO_CTRL) –RISC: 0x100, DSP: 0x40

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b0	1 for I/O mode, 0 for DMA mode
3:1	-	3'b0	Reserved
5:4 (R/W)	TX_ENDIAN_MODE	2'b0	Reserved
31:6	-	26'h0	Reserved

Table 499: USP TX DMA I/O MODE Register

- USP TX DMA I/O Length Register (USP_TX_DMA_IO_LEN) –RISC: 0x104, DSP: 0x42~0x43

Bit	Name	Default	Description
31:0 (R/W)	DATA_LEN	32'hffffffff	The total byte number of the data that can be transmitted by USP after it is enabled. 0: No limit, USP can transmit data continuously until it is diabled 0x1~32'hffffffff: only the given number of data can be transmitted. If there is more data to be transmitted, restarted the USP. It is recommended to set this field to 0.

Table 500: USP TX DMA I/O Length Register

- USP TX FIFO Control Register (USP_TX_FIFO_CTRL) –RISC: 0x108, DSP: 0x44

Bit	Name	Default	Description
1:0 (R/W)	FIFO_WIDTH<1:0>	2'h0	Data width of FIFO 0: Byte 1: Word 2: DWORD
8:2 (R/W)	FIFO_THD<6:0>	7'h0	A threshold in byte that triggers an interrupt. An interrupt is triggered when data count in the FIFO reaches the threshold.
31:9	-	23'h0	Reserved

Table 501: USP TX FIFO Control Register

- USP TX FIFO Level Check Register (USP_TX_FIFO_LEVEL_CHK) –RISC: 0x10C, DSP: 0x46~0X47

Bit	Name	Default	Description
4:0 (R/W)	FIFO_SC<4:0>	5'h0	Stop checking in DWORD
9:5	-	5'h0	Reserved
14:10 (R/W)	FIFO_LC<4:0>	5'h0	Low check in DWROD
19:15	-	5'h0	Reserved
24:20 (R/W)	FIFO_HC<4:0>	5'h0	High check in DWORD
31:25	-	7'h0	Reserved

Table 502: USP TX FIFO Level Check Register

- USP TX FIFO Operation Register (USP_TX_FIFO_OP) –RISC: 0x110, DSP: 0x48
  This register is different from other peripheral operation registers and the reset bit is bit 0.

Bit	Name	Default	Description
0 (R/W)	FIFO_RESET	1'b0	Set to 1 to stop FIFO and to reset the FIFO internal status, including interrupt status. Set to 0 in normal operation.
1 (R/W)	FIFO_START	1'b0	Start the read/write transfer when this bit is declared.
31:2	-	30'h0	Reserved.

Table 503: USP TX FIFO Operation Register

- USP TX FIFO Status Register (USP_TX_FIFO_STATUS) –RISC: 0x114, DSP: 0x4a

Bit	Name	Default	Description
6:0 (R)	FIFO_LEVEL	7'h0	The byte number of the valid data in the FIFO. If FIFO is full, the value of this register will be 0, thus the FIFO_FULL bit must be concatenated with this value in order to determine actual data counts in the FIFO.
7 (R)	FIFO_FULL	1'b0	FIFO full status, FIFO is full when it is read out as 1. This bit is concatenated with FIFO_LEVEL as the actual FIFO data count.
8 (R)	FIFO_EMPTY	1'b1	FIFO empty status. Equivalent to (FIFO_FULL, FIFO_LEVEL) == 0
31:9	-	23'h0	Reserved.

Table 504: USP TX FIFO Status Register

- USP TX FIFO Data Register (USP_TX_FIFO_DATA) –RISC: 0x118, DSP: 0x4c

Bit	Name	Default	Description
31:0 (W)	FIFO_DATA	32'h0	The FIFO data register, which is the bottom of TX_FIFO.

Table 505: USP TX FIFO Data Register

- USP RX DMA I/O MODE Register (USP_RX_DMA_IO_CTRL) –RISC: 0x120, DSP: 0x50

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b0	1: I/O mode 0: DMA mode
1 (R/W)	FRADDR_CLR_EN	1'b0	RX FIFO read address[1:0] clear enable. When this bit is set to 1, hardware will automatically clear RX FIFO read address[1:0] when IO_DMA_SEL is cleared from high level. 0: Disabled 1: Enabled
2 (R/W)	DMA_FLUSH	1'b0	Flushes the DMA receive FIFO in case the DATA_LEN set on the peripheral side does not match the DWORD size set in the DMA control.
3 (R/W)	DMAIO _SW_EN	1'b0	RX FIFO DMA to I/O mode auto switch by hardware enable when RX time out. 0: Disabled 1: Enabled
5:4 (R/W)	RX_ENDIAN_MODE	2'b0	Reserved
31:6	-	26'h0	Reserved

Table 506: USP RX DMA I/O MODE Register

- USP RX DMA I/O Length Register (USP_RX_DMA_IO_LEN) –RISC: 0x124, DSP: 0x52~0x53

Bit	Name	Default	Description
31:0 (R/W)	DATA_LEN	32'hffffffff	The total byte number of the data that can be received by USP after it is enabled. 0: No limit, USP can receive data continuously until it is diabled. 0x1~32'hffffffff: only the given number of datas can be received. If there is more data to be received, restart the USP. It is recommended that the user set this field to 0.

Table 507: USP RX DMA I/O Length Register

- USP RX FIFO Control Register (USP_RX_FIFO_CTRL) –RISC: 0x128, DSP: 0x54

Bit	Name	Default	Description
1:0 (R/W)	FIFO_WIDTH<1:0>	2'h0	Data width of FIFO 0: Byte 1: WORD 2: DWORD
8:2 (R/W)	FIFO_THD<4:0>	7'h0	A threshold in byte that triggers an interrupt. An interrupt is triggered when the data count in the FIFO reaches the threshold.
31:9	-	23'h0	Reserved

Table 508: USP RX FIFO Control Register

- USP RX FIFO Level Check Register (USP_RX_FIFO_LEVEL_CHK) –RISC: 0x12C, DSP: 0x56~0X57

Bit	Name	Default	Description
4:0 (R/W)	FIFO_SC<4:0>	5'h0	Stop check in DWORD
9:5	-	5'h0	Reserved
14:10 (R/W)	FIFO_LC<4:0>	5'h0	Low check in DWROD
19:15	-	5'h0	Reserved
24:20 (R/W)	FIFO_HC<4:0>	5'h0	High check in DWORD
31:25	-	7'h0	Reserved

Table 509: USP RX FIFO Level Check Register

- USP RX FIFO Operation Register (USP_RX_FIFO_OP) –RISC: 0x130, DSP: 0x58
  This register is different from other peripheral operation registers and the reset bit is bit 0.

Bit	Name	Default	Description
0 (R/W)	FIFO_RESET	1'b0	Set to 1 to stop the FIFO and reset the FIFO internal status, including interrupt status. Set to 0 in normal operation.
1 (R/W)	FIFO_START	1'b0	Start the read/write transfer when this bit is declared.
31:2	-	30'h0	Reserved.

Table 510: USP RX FIFO Operation Register

- USP RX FIFO Status Register (USP_RX_FIFO_STATUS) –RISC: 0x134, DSP: 0x5a

Bit	Name	Default	Description
6:0 (R)	FIFO_LEVEL	7'h0	The byte number of the valid data in the FIFO. If FIFO is full, the value of this register will be set to 0, thus the FIFO_FULL bit must be concatenated with this value in order to determine the actual data count in the FIFO.
7 (R)	FIFO_FULL	1'b0	FIFO full status. The FIFO is full when it is read as 1. This bit is concatenated with FIFO_LEVEL as the actual FIFO data count.
8 (R)	FIFO_EMPTY	1'b1	FIFO empty status. Equivalent to (FIFO_FULL, FIFO_LEVEL) == 0
31:9	-	23'h0	Reserved

Table 511: USP RX FIFO Status Register

- USP RX FIFO Data Register (USP_RX_FIFO_DATA) –RISC: 0x138, DSP: 0x5c~0x5d

Bit	Name	Default	Description
31:0 (R)	FIFO_DATA	32'h0	The FIFO data register, which is at the bottom of the RX_FIFO.

Table 512: USP RX FIFO Data Register

Audio Codec Interface

Overview

The SiRFatlasV Audio Codec interface can be used to connect to either AC97 or I^2S type Codec. The pins of the AC97 Codec are shared with that of I^2S and will be configured as the right output or input when used by the corresponding interface. The I^2S interface's DA_DATA1 and DA_DATA2 pins are shared with USP0 and USP1. When Audio controller works in I^2S floating mode, which can be set by configure register CODEC_I2S_TX_RX_EN bit 5, I^2S DOUT pin X_AC97_DOUT is input. In this mode, I^2S will not drive the X_AC97_DOUT.

The Audio Codec interface is controlled by RISC through the RISC I/O interface. Audio data transferred between memory and the Audio Codec interface is through the DMA channel. The Audio Codec interface supports both AC97 and I^2S formats. These two formats cannot work simultaneously; users need to configure the audio Codec interface register in order to select the supported format.

For I^2S interface and AC97 PCM data streams, there are two DMA channels:

- For PCM audio data in, it is channel 6.
- For audio data out, it is channel 7.

Users can configure this DMA setting for up to six-slot audio data playback. For audio input, SiRFatlasV only supports two-slot input. For AC97 I/F, there is an extra DMA channels for data in (channel 8). This DMA channel can be used for non-audio data input such as touch panel sample data.

The entire Audio Codec interface consists of four parts:

- A RISC I/O interface (register files)
- A DMA interface
- An AC97 controller
- An I^2S controller

Feature List

- For the AC97 interface
  - Intel AC97 specification 2.2 compliant
  - Normal mode (48KHz)
  - 16-bit resolution
  - Stereo, mono data input or differential non-audio data such as touch panel data is supported
  - DMA / RISC I/O mode
  - Loop-DMA mode

- For I^2S interface
  - Both the I^2S master and slave mode are supported. (The USP interface of SiRFatlasV only supports slave mode.)
  - 16-bit resolution
  - Loop-DMA mode

Pin Descriptions

External Pin Descriptions

Pin Name	I/O Type	Pin MUX	Default Function	Default Status	Description
X_AC97_BIT_CLK	Bi	AC97 I^2S	AC97	Input pull-down	BIT_CLK. 12.288M Clock signal input from external AC97 Codec. (This is the clock signal generated by SiRFatlasV or input from an external I^2S Codec. The frequency is programmable).
X_AC97_DIN	Input	AC97 I^2S	AC97	Input pull-down	AD_DATA. Serial digital input0 data from AC97 Codec.
X_AC97_SYNC	Bi	AC97 I^2S	AC97	Input pull-down	SYNC. Fixed 48KHz frame signal for synchronization (Frame signal for synchronization is generated by SiRFatlasV or input from an external I^2S Codec).
X_AC97_DOUT	Bi	AC97 I^2S	AC97	Output low	DA_DATA0. Serial digital output data from AC97 Codec (Primary serial output data to the external I^2S Codec) When Audio controller work in I^2S floating mode, this pin is used as input.

Pin Name	I/O Type	Pin MUX	Default Function	Default Status	Description
X_URFS_0 (I2S_DOUT1)	Output	I^2S USP0 GPIO	GPIO	Output	DA_DATA1, Serial digital output data from AC97 CODEC.
X_URFS_1(I2S_DOUT2)	Output	I^2S USP1 GPIO	GPIO	Output	DA_DATA2, Serial digital output data from AC97 CODEC.

Table 513: Audio External Pin Description

Functional Descriptions

Block Diagram

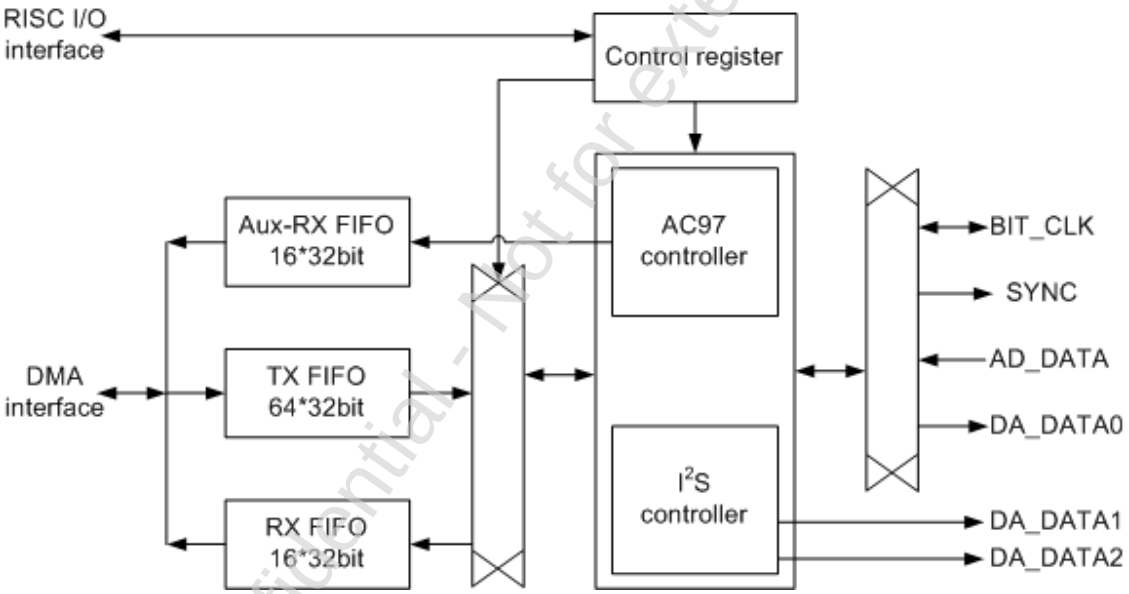


Figure 76: Block Diagram of the Audio

Module Descriptions

Top Level of the Codec Interface

This module is only used to connect the sub-module of the Codec interface. The sub-modules include a RISC interface, an AC97 interface, an I^2S interface, a RX_FIFO interface, a TX_FIFO interface and an AUX_RX_FIFO interface.

AC97 Codec Interface

In the AC97 interface, SiRFatlasV operates as an AC97 Codec controller whose interface is a 4-wire digital serial interface (with BIT_CLK, SYNC, SDATA_OUT and SDATA_IN). BIT_CLK and SDATA_IN are inputs from the external AC97 Codec chip, the SYNC and SDATA_OUT are generated by SiRFatlasV and output to an external Codec chip. This interface also allows the host (RISC) to read and write the internal registers of the Codec through AC97 frame. The AC97 interface supports normal mode (48 KHz).

I^2S Codec Interface

The I^2S interface is a serial interface that includes BIT_CLK, FRAME_SYNC, AD_DATA, DA_DATA0, DA_DATA1 and DA_DATA2. Four of the six pins are shared with the AC97 Codec interface; DA_DATA1 and DA_DATA2 are exclusively used by the I^2S interface. In this case, however, only AD_DATA is the input from external I^2S audio Codec chip. BIT_CLK, FRAME_SYNC, DA_DATA0, DA_DATA1 and DA_DATA2 will be generated by SiRFatlasV and output to an external Codec chip.

The sample frequency is programmable and the host (RISC) can determine the frequency of BIT_CLK and FRAME_SYNC. It can also separately turn on or off the left and right audio channels. I^2S-format is supported.

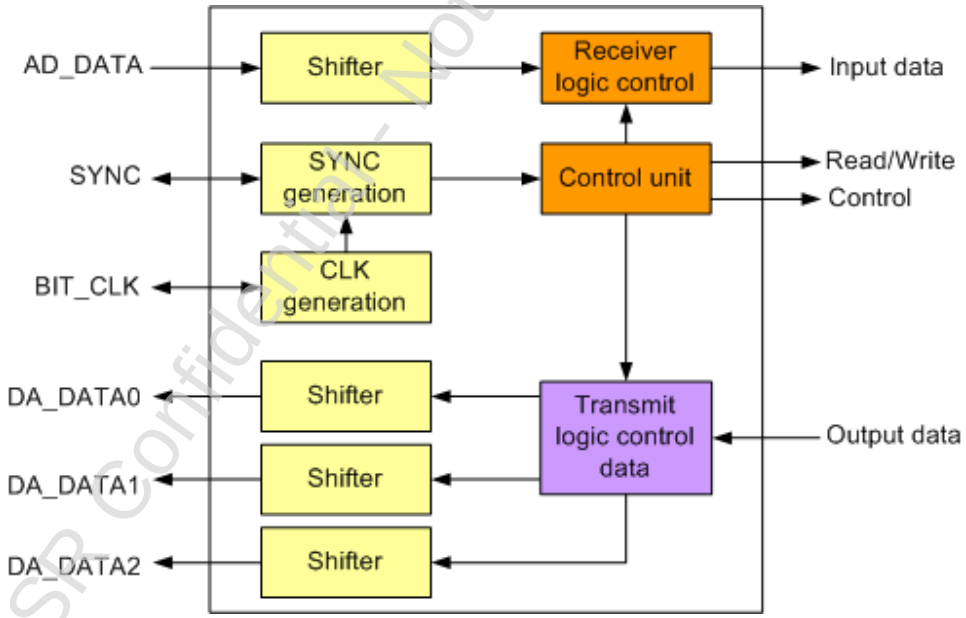


Figure 77: I^2S Controller Functional Block Diagram

RISC I/O Interface

In SiRFatlasV, the registers of audio codec controller interface occupy 64 KB of memory space ranging from 0x80060000 to 0x8006FFFF.

There are four groups of registers which are located in the register space with different offset address:

- Audio Codec controller registers (address offset 0x0)
- TX FIFO control registers (address offset 0xf80)
- RX FIFO control registers (address offset 0xfc0)
- AUX RX FIFO control registers (address offset 0xB80)

For the AC97 Interface, users can use the AC-Frame interface to program the external audio Codec registers in order to configure the external Codec working mode or query its status. RISC I/O accesses external audio Codec registers in no more than 2 frame time.

Codec FIFO to DMA Interface

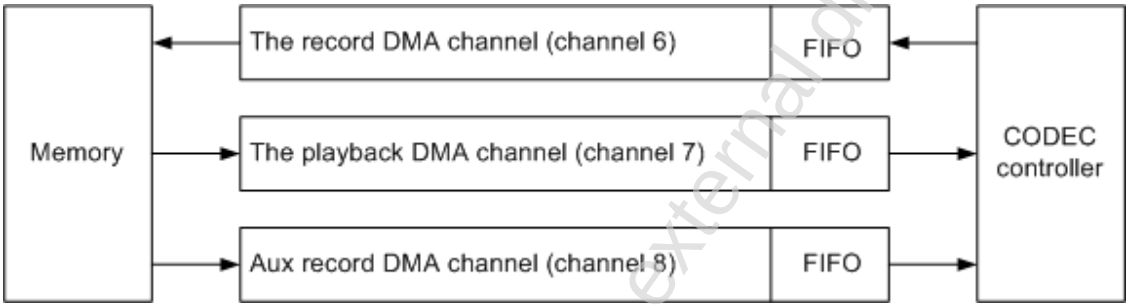


Figure 78: Codec DMA Structure

As shown in the figure above, there are 3 DMA channels in the Codec interface block. For the Playback DMA channel, users can configure the corresponding working mode register to enable to work in stereo and AC3 5.1 channel mode. For the Recode DMA channel, users may configure the corresponding working mode register to make it writing in the mono or stereo record mode. For the Aux Record DMA channel, users can only use it to sample one of the single-slot input AD data.

AC97 Controller

RISC Interface

The AC97 CODEC block can be controlled by RISC. Users can use AC-Frame interface to program the external audio CODEC in order to configure the external CODEC working mode or to query external CODEC status. The RISC I/O accesses external audio CODEC registers in no more than 2 frame time.

There are 3 register groups located in the register space with different offset addresses:

- Audio CODEC controller registers
- Primary FIFO control registers
- Auxiliary FIFO control register

Each corresponding part provides ports that can be accessed by RISC.

External AC97 CODEC Register Operations

AC-link slot 1 and slot 2 are used as AC97 command slots to control external AC97 CODEC features and monitor its status. The command port supports up to sixty-four 16-bit read/write registers, addressable on even-byte boundaries.

AC-link output frame from SiRFatlasV AC97 Controller interface to the AC97 Codec. Slot 1 indicates whether the current control transaction is a read or a write operation, as well as specifies the target register address. If it is a write operation, the output frame slot 2 will contain valid data, and the tag bit of slot 1 will be asserted. If it is a read operation, the tag bit of slot 2 will be de-asserted, and the data stream in slot 2 will be abandoned.

AC-link input frame (from AC97 Codec to SiRFatlasV AC97 Controller interface) Slot 1 echoes the control register read address (bit 18 to bit 12), and for variable sample rate, it delivers request flags for all output slots (bit 11 to bit 2). The echoed address is valid only when slot 1 is tagged valid, but the request flags are independent of the tag bit. When slot 2 is tagged valid, the input frame Slot 2 will contain 16-bit control register read data.

The AC-link output frame is illustrated in the following diagrams when the SiRFatlasV AC97 controller issues a control register read/write command.

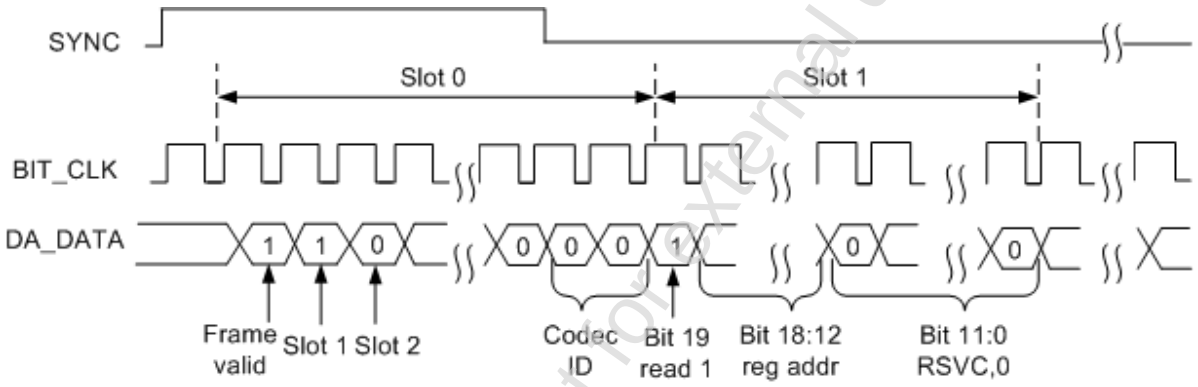


Figure 79: AC-Link Output Frame Read Command Diagram

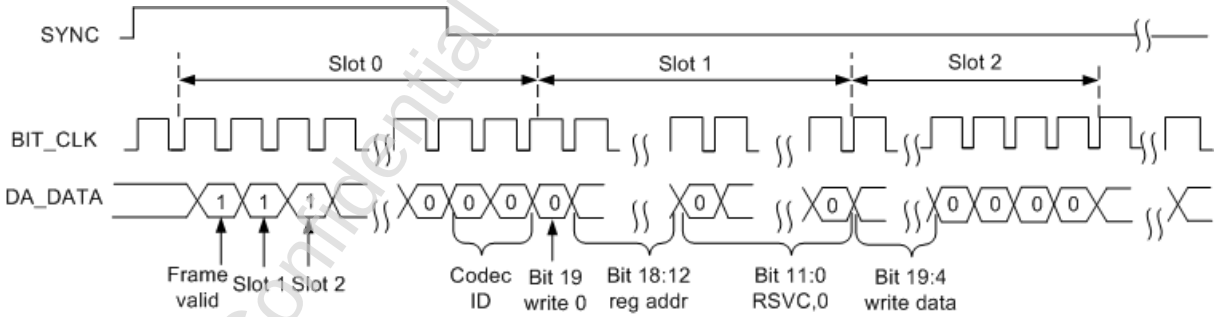


Figure 80: AC-Link Output Frame Write Command Diagram

The following figure illustrates AC97 CODEC response with the control register read data.

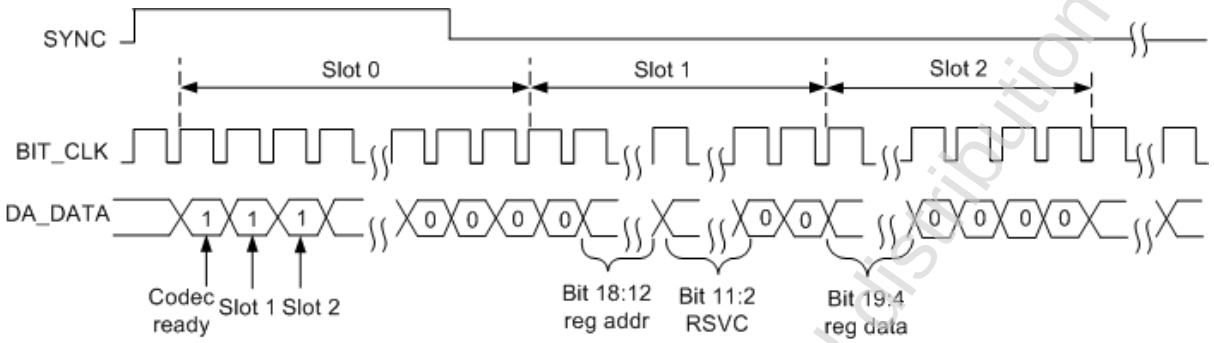


Figure 81: AC-Link Input Frame Command Diagram

PCM Record

The following figure shows the AC97 PCM record function block. There are two records mode, stereo mode and mono mode depending on the SiRFatlasV AC97 Controller configuration.

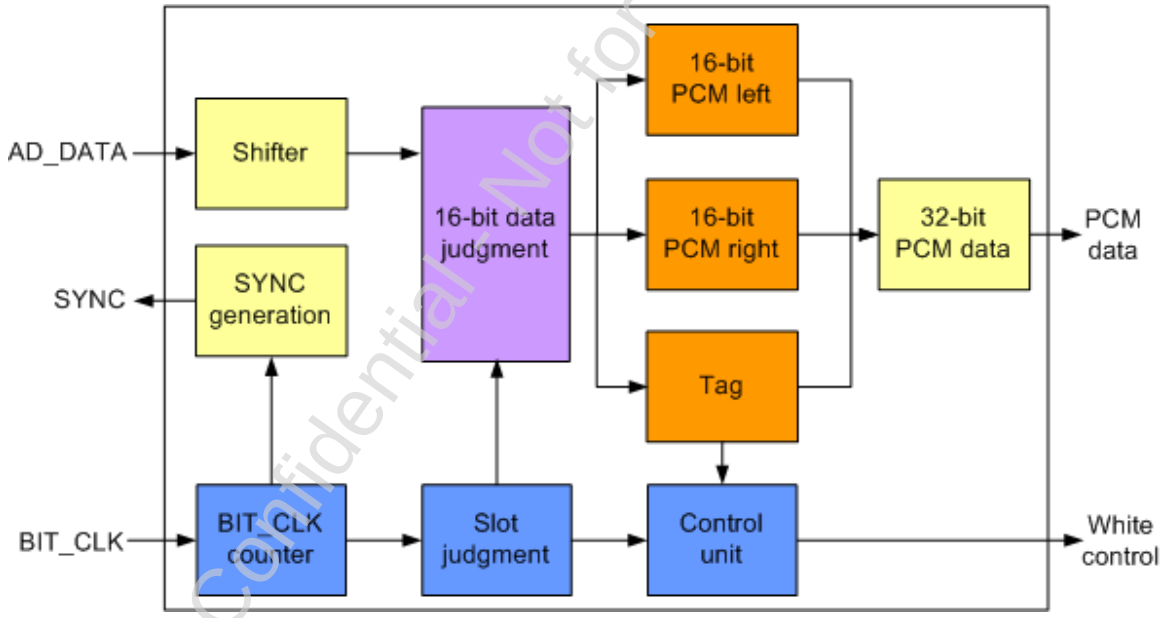


Figure 82: AC97 PCM Record Functional Block Diagram

Data transfer over AC-link data is in serial mode. First SiRFatlasV AC97 controller converts the serial input data into 16-bit, which may be an AC-link tag, status address, status data, PCM left channel or right channel data. The BIT_CLK counter will then make the slot judgment. AC-link input slot 0 is the AC-link tag. The bits in the tag indicate whether the slots contain valid data. AC-link input slot 3 and 4 are PCM record left and right channel output of AC97's input MUX, the post-ADC, and when bit 11 or bit 10 of AC-link tag are asserted, it will indicate that slot 3 or slot 4 contains valid PCM data. In 48 KHz sample rate, PCM left and right channels are always valid in each AC97 audio frame.

In stereo mode, 32-bit PCM data is made up of PCM left and right channels. Normally LSB 16-bit is left channel and MSB 16-bit right channel. In the interlace mode, the left channel is saved in MSB 16-bit while the right channel in LSB 16-bit. In mono mode, the 16-bit left or right channel data is first saved in LSB 16-bit data format, then in MSB 16-bit format.

The PCM record process is shown in the flowchart below.

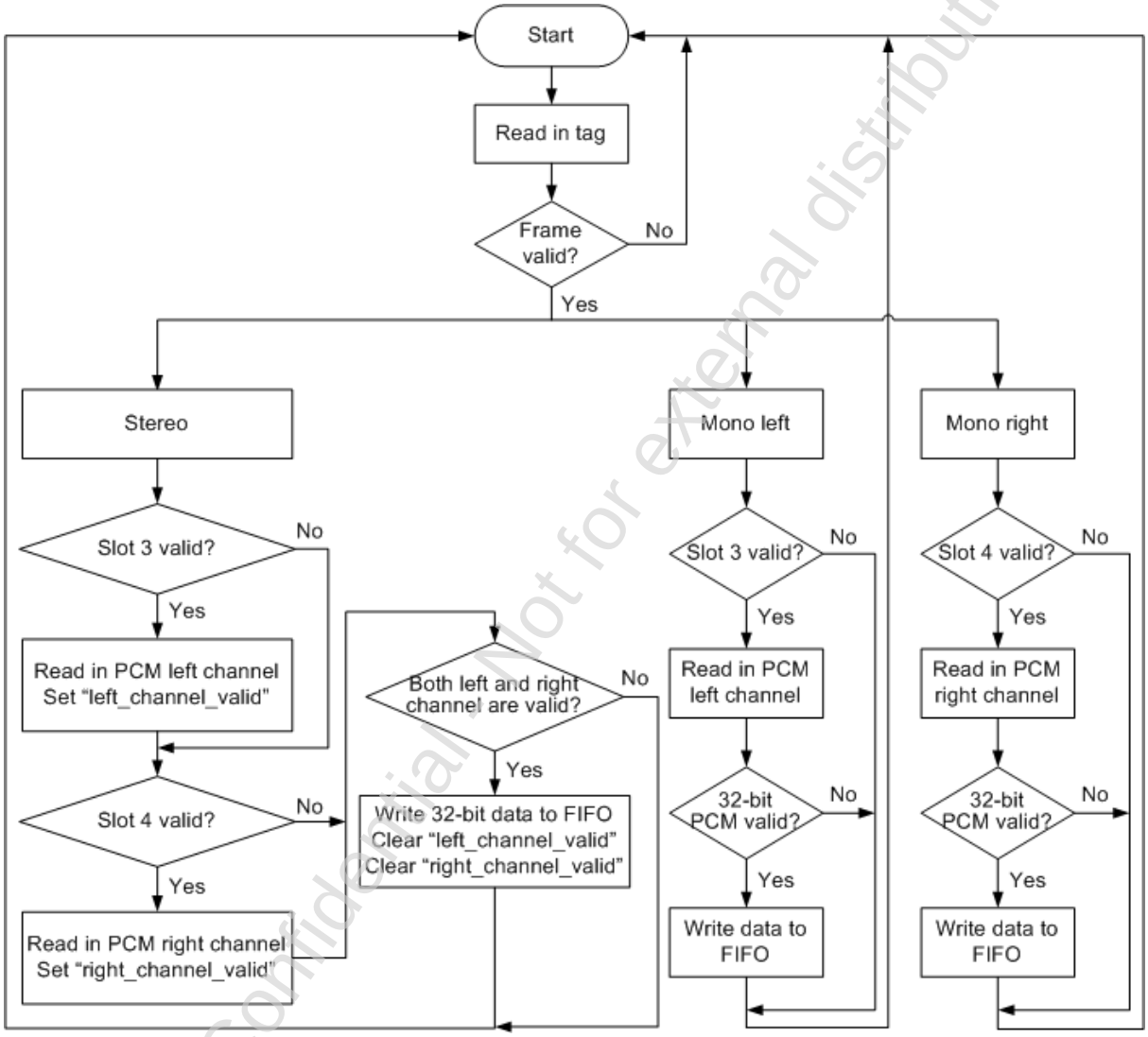


Figure 83: AC97 PCM Record Flowchart

PCM Playback

The SiRFatlasV AC97 controller supports mono and stereo playback modes. When the AC97 CODEC works in 48 KHz mono mode, the SiRFatlasV CODEC controller will fetch 16-bit audio data from the memory to the corresponding left or right channel output. When it works in 48 KHz stereo mode, each audio output frame will contain both the left and right channels valid data. For the AC97 controller, 32-bit PCM data is fetched from the FIFO at 48 KHz for each frame.

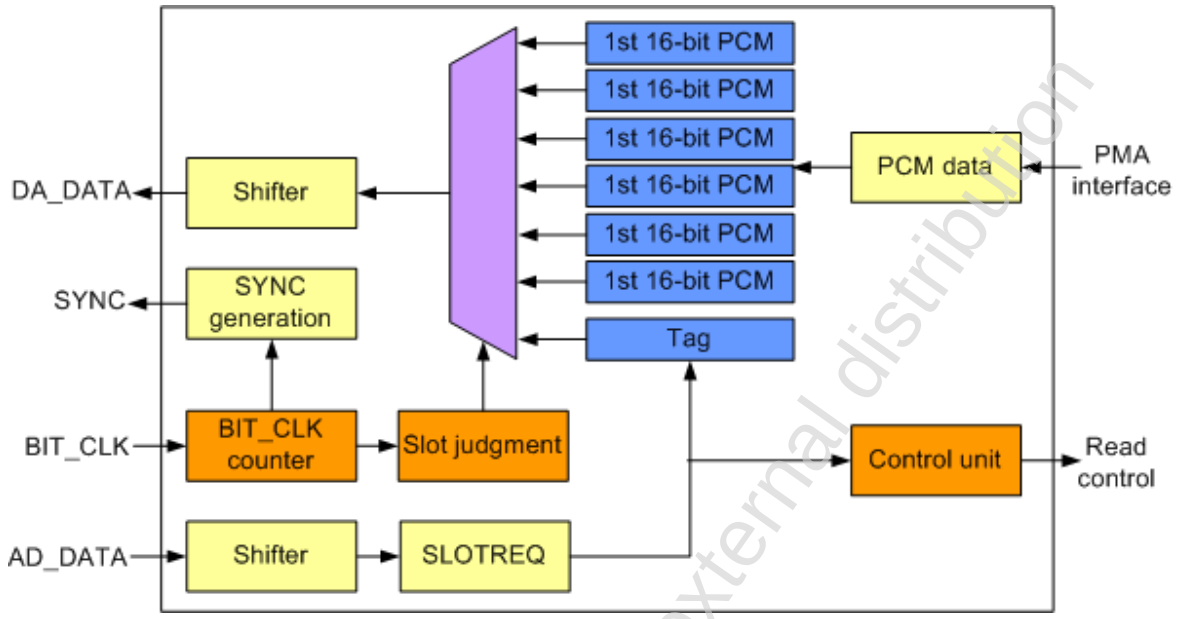


Figure 84: AC97 PCM Playback Functional Diagram

Auxiliary Data Input

The user can program the AC97 controller to select the right slot for auxiliary data input. Regardless of the slot selected for carrying data, the input data will always be transferred to the memory through DMA channel 8.

The data transfer over AC-link is in serial mode. First the SiRFatlasV AC97 controller converts the serial input data into 16-bit. The 16-bit data may be PCM data or other ADC sample data. Then the BIT_CLK counter will make the slot judgment. AC-link input slot 0 is the AC-link tag. The bits in the tag indicate whether the selected slot contains valid data.

I²S Controller

The SiRFatlasV I²S Controller supports I²S-bus format with 16-bit word length. The clock and the SYNC signal can be set to both master and slave mode.

In the master mode, the frequency of BIT_CLK is programmable and generated by the SiRFatlasV I²S controller by dividing the internal clock. In addition, SYNC is driven by the SiRFatlasV I²S controller and generated by counting BIT_CLK, which can set to 50% or not 50% duty cycle by adjusting the left channel clock number and whole frame clock number. SYNC frequency is the same as Codec sampling frequency. To get the right sampling rate, simply adjust the whole frame clock number. Refer to register CODEC_I²S_CTRL description in “Register Definitions” for more detail about how to configure I²S master mode clock.

In the slave mode, BIT_CLK and SYNC signals are provided by the external I²S codec chip. The I²S controller samples the input BIT_CLK and SYNC. In this mode, the frequency of the I/O clock should be set 10 times of BIT_CLK.

This I²S controller supports both 2-channel and 6-channel output and accepts 2-channel audio input data.

The I²S format is shown below:

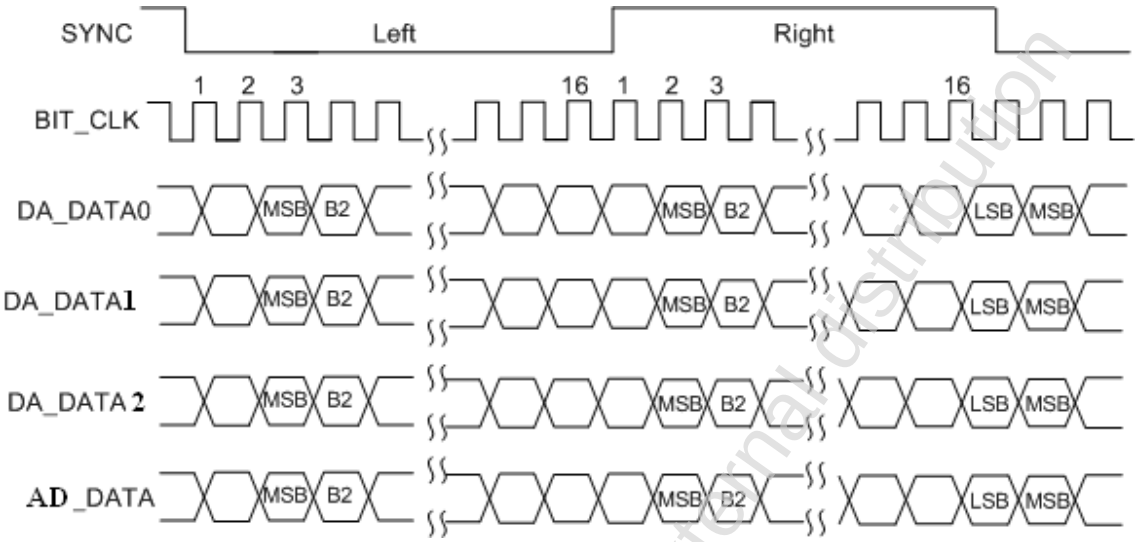


Figure 85: I²S-Bus Format

For the I²S controller interface, WS (Word Select) is synchronized with the falling edge of BIT_CLK. The input serial data AD_DATA is sampled on the rising edge of BIT_CLK. The output data is driven onto the DA_DATA bus on the falling edge of BIT_CLK. In I²S-bus format, data is driven onto the bus MSB first.

All data transaction on the I²S digital serial bus is for audio data stream.

The I²S Codec is different from AC97 Codec in that for the I²S Codec, BIT_CLK and frame rate are programmable and the sample frequency is tied to both BIT_CLK and the I²S frame clock number. For the AC97 Codec, BIT_CLK is fixed at 12.288 MHz while the frame rate is fixed at 48 KHz. For the I²S interface, all data transfer is audio data stream, but for AC97, other than the audio data stream, the register control is handled through the AC-link.

Interrupt Generation

There are 3 DMA channels in the Codec interface and all CODEC interrupt is generated by the DMA interface. There are eight FIFO statuses which can trigger CODEC interrupts if they are not masked:

- TX FIFO Full
- TX FIFO Empty
- TX FIFO Overflow
- TX FIFO Underflow
- RX FIFO Full
- RX FIFO Empty
- RX FIFO Underflow
- RX FIFO Overflow

When the TX FIFO Full status is indicated, if DMA interface continues to write audio data into TX FIFO, then a TX FIFO overflow will be triggered. In general interrupts will not occur because when the TX FIFO is full; the DMA request will not be generated, thereby guaranteeing no data written to the TX FIFO.

When the TX FIFO is in empty status and the DMA controller cannot serve the CODEC transmission channel, any further data requests from the audio CODEC will trigger a TX FIFO underflow interrupt. When the RX FIFO is in full state and the DMA controller cannot transfer data into memory, more audio input data stream will trigger RX FIFO overflow interrupts.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
RISC I/O Interface	0x80060000

Table 514: Audio Base Address

Register Mapping

RISC Address <11:0>	Register	Description
0x0000	CODEC_SHARE	Codec interface control register
0x0004	CODEC_I2S_CTRL	I^2S interface control register
0x0008	CODEC_I2S_TX_RX_EN	I^2S interface transfer enable register
0x0C~0x10	-	Reserved
0x001C	CODEC_AC97_CTRL	AC97 interface control register
0x0020	CODEC_AC97_CMD	AC97 command register
0x0024	CODEC_AC97_AD_TAG	AC97 input frame tag
0x0028	CODEC_AC97_STATUS	AC97 Codec status data
0x002C	CODEC_RD_AC97REG_STS	AC97 register read back status
0x0030	-	Reserved
0x0034	CODEC_TXSLOT_EN	AC97 TX slot selection
0x0038	CODEC_RXSLOT_EN	AC97 RX slot selection
0x003C	CODEC_AUX_RX_EN	AC97 input auxiliary slot selection
0x0040~0x0BBC	-	Reserved
0x0BC0	CODEC_RX_AUX_DMA_IO_CTRL	Codec auxiliary RX FIFO mode selection
0x0BC4~0x0BC8	-	Reserved
0x0BCC	CODEC_RX_AUX_FIFO_LEVEL_CHK	Codec auxiliary RXFIFO level check
0x0BD0	CODEC_RX_AUX_FIFO_OP	Codec auxiliary RXFIFO operation
0x0BD4	CODEC_RX_AUX_FIFO_STS	Codec auxiliary RXFIFO status register
0x0BD8	CODEC_RX_AUX_FIFO_INT_EN	Codec auxiliary RXFIFO interrupt enable

RISC Address <11:0>	Register	Description
0x0BE0~0x0F7C	-	Reserved
0x0F80	CODEC_TX_DMA_IO_CTRL	Codec transmit DMA I/O mode
0x0F84~0x0F88	-	Reserved
0x0F8C	CODEC_TX_FIFO_LEVEL_CHK	Codec TXFIFO level check
0x0F90	CODEC_TX_FIFO_OP	Codec TXFIFO operation
0x0F94	CODEC_TX_FIFO_STS	Codec TXFIFO status register
0x0F98	CODEC_TX_FIFO_INT_EN	Codec TXFIFO interrupt enable
0x0F9C~0x0FBC	-	Reserved
0x0FC0	CODEC_RX_DMA_IO_CTRL	Codec receive DMA I/O mode
0x0FC4~0x0FC8	-	Reserved
0x0FCC	CODEC_RX_FIFO_LEVEL_CHK	Codec RXFIFO level check
0x0FD0	CODEC_RX_FIFO_OP	Codec RXFIFO operation
0x0FD4	CODEC_RX_FIFO_STS	Codec RXFIFO status register
0x0FD8	CODEC_RX_FIFO_INT_EN	Codec RXFIFO interrupt enable
Others	-	Reserved

Table 515: Audio Register Address Mapping

Register Descriptions

- Codec share register (CODEC_SHARE) – 0x0

  Although the CODEC interface supports both AC97 and I^2S audio formats, they however cannot work simultaneously. CODEC_SHARE is used to select whether I^2S or AC97 will be used, it is also used to control whether the record or the playback channel in the Audio CODEC interface is enabled.

Bit	Name	Default	Description
0 (R)	-	1'b0	Reserved
1 (R/W)	INTERLACE_MODE	1'b0	Codec interlace mode selection: 0: When record, the first sample (e.g. left channel) will be stored in LSB 16-bit, the second sample (e.g. right channel) will be stored in MSB 16-bit. When playback, play LSB 16-bit first. 1: When record, the first sample (e.g. left channel) will be stored in MSB 16-bit, the second sample (e.g. right channel) will be stored in LSB 16-bit. When playback, play MSB 16-bit first.

Bit	Name	Default	Description
4:2 (R)	-	3'b0	Reserved
5 (R/W)	AC97_AD_IDLE	1'b1	AC97 AD wire status setting: 1: AC97 AD input wire is shut down in the AC97 interface. 0: AC97 AD input wire is enabled in the AC97 interface. By default, the AC97 AD wire is shut down before receiving signals through the AC97 Codec interface. User needs to clear this register bit.
6 (R/W)	AC97_DA_IDLE	1'b1	AC97 DA wire status setting: 1: AC97 DA wire is shut down in the AC97 interface; the external AC97 CODEC interface receives no valid data streams, including the CMD data. 0: AC97 DA wire is enabled in the AC97 interface, the AC97 controller is allowed to be output to the external AC97 CODEC. By default, the AC97 DA wire is shut down, before outputting data to the AC97 Codec. User needs to clear this register bit.
7 (R/W)	I2S_AC97	1'b0	Codec interface mode selection: 0: Codec interface is in AC97 format. 1: Codec interface is in I^2S format. By default, the codec interface is the AC97 controller.
8 (R/W)	ALAW_EN	1'b0	For PCM DMA playback channel: 0: ALAW expansion is enabled. 1: ALAW expansion is disabled.
9 (R/W)	ULAW_EN	1'b0	For PCM DMA playback channel: 0: ULAW expansion is enabled. 1: ULAW expansion is disabled.
31:10 (R)	-	21'h0	Reserved

Table 516: Codec Share Register (CODEC_SHARE)

- Codec I^2S Control Register (CODEC_I2S_CTRL) – 0x4
  This register is used to configure the I^2S controller:

Bit	Name	Default	Description
0 (R/W)	I2S_SLV	1'b0	I^2S work state control 0: Master mode, the BIT CLOCK and WS signals are provided by SiRFatlasV. 1: Slave mode, the BIT CLOCK and WS signals

Bit	Name	Default	Description
			are provided by the external I²S Codec.
1 (R/W)	I2S_SIX_CHN	1'b0	I²S channel select: 0: Two channels 1: Six channels
2 (R/W)	I2S_GLITCH_FREE	1'b0	Slave clock sample mode: 0: Without glitch-free circuits 1: With glitch-free circuits
3 (R/W)	I2S_LOOP_BACK	1'b0	Loop-back mode for testing and the transmitted data is looped back to the receiving module, this is designed only for two-channel outputs: 0: No loop back 1: Loop back
8:4 (R/W)	I2S_L_CHN_LEN	5'hf	The left channel clock number of the I²S frame which must be set to actual number -1
14:9 (R/W)	I2S_FRM_LEN	6'h1f	The entire I²S clock number which must be set to actual number -1
21:15 (R/W)	I2S_SYSCLK_DIV	9'h0	The divider number of the I²S system clock from I²S CLOCK I2S_SYSCLK_DIV = fi2s_clock/ (fi2ssys * 2) – 1 I²S_system clock means I²S mclk.
23:22(R/W)	AC97_FIFO_EMPTY_CTRL	2'b0	When TXFIFO underflow happens, there will be possible cause channel disorder. Setting these two bits will avoid this. 2'b00: Not mask TXFIFO read signal when TXFIFO underflow happens 2'b01: Mask 2*frame times TXFIFO read signal when TXFIFO underflow happens (Used in 3 channel playback mode) 2'b10: Reserved 2'b11: Mask 3*frame times TXFIFO read signal when TXFIFO underflow happens (Used in 6 channel playback mode)
31:24 (R/W)	I2S_BITCLK_DIV	8'h0	The divider number of I²S bit clock from I²S clock. I2S_BITCLK_DIV = fi2ssys / (fi2sbit * 2) – 1

Table 517: Codec I²S Control Register (CODEC_I2S_CTRL)

- Codec I²S Transmit/Receive Enable Register (CODEC_I2S_TX_RX_EN) –0x8
  Before enabling the I2S_TX_EN or I2S_RX_EN bit, you must first configure CODEC_I2S_CTRL.

Bit	Name	Default	Description
0 (R/W)	I2S_RX_EN	1'b0	Receive enable bit: 0: Disabled 1: Enabled
1 (R/W)	I2S_TX_EN	1'b0	Transmit enable bit: 0: Disabled 1: Enabled
2 (R/W)	I2S_SYSCLK_EN	1'b0	I²S mclk output enable bit: 0: Disabled 1: Enabled
3 (R/W)	I2S_CLK_SEL	1'b0	I²S system clock select: 0: From PWM 1: From ext-clk
4 (R/W)	I2S_FLOATING	1'b0	I²S DOUT pin mode selection: 0: X_AC97_DOUT is output. 1: X_AC97_DOUT is input, floating mode.
29:5 (R)	-	26'b0	Reserved
31:30 (R/W)	I2S_R2X_LP	2'h0	Loop back the received data to the transmit port directly for testing purposes: 00: Reserved 01: rx to tx0 10: rx to tx1 11: rx to tx2

Table 518: Codec I²S Transmit/Receive Enable Register (CODEC_I2S_TX_RX_EN)

Sample configuration of I²S master mode clock:

- Configure the source of PWM6 to PLL2 (take PLL2 freq is 664MHz for example)
- Set PWM6 Wait=12 Hold=13 so the output of PWM6 is 664/(12+1+13+1)=24.5926MHz
- Set I²S _SYSCLK_DIV=0 so MCLK=24.5926MHz/2*(0+1) =12.296MHz
- Set I²S_BITCLK_DIV=3 so BITCLK=12.296MHz/2*(3+1) =1.537MHz
- Set I²S _FRM_LEN =31 I2S_L_CHN_LEN=15 so SYNC=BITCLK/(I2S_FRM_LEN+1)≈48K

- Codec AC97 Control Register (CODEC_AC97_CTRL) – 0x1C
  This register is used to configure AC97 controller, including:

  - Left and right channel controls

- AC97 Codec warm wake controls
- AC-link output frame controls

Bit	Name	Default	Description
3:0 (R)	-	4'h0	Reserved.
4 (R/W)	WARM_WAKE	1'b0	AC97 warm wake control: 1: Warm wake AC97 codec. 0: Exit warm wake-up mode. WARM_WAKE should be set when BIT_CLK is absent. The asserted WARM_WAKE signal set the SYNC to be HIGH to bring AC97 Codec out of halted or low-power mode.
5 (R/W)	AC97_START	1'b0	AC97 interface start controls: 1: Start AC97 interface. 0: Stop AC97 interface. Before the AC97 interface is started, users need to enable the channel, and once AC97_START is asserted, the AC97 interface will begin to work.
12:6 (R)	-	7'h0	Reserved.
13 (R/W)	DA Slot2 Tag	1'h0	1: Indicates the AC97 interface will output valid data through Slot 2 in the next output frame. 0: Indicates that the Slot 2 in the next output frame will be disabled. After the command data is written to AC97 Codec, the tag bit will be cleared by hardware.
14 (R/W)	DA Slot1 Tag	1'h0	1: Indicates the AC97 interface will output valid data through Slot 1 in the next output frame. 0: Indicates that in the next output frame Slot 1 output is disabled. After the command is sent to AC97 Codec, this bit will be cleared by hardware.
15 (R/W)	DA Frame Valid	1'h0	1: Indicates the output frame contains the valid data. 0: Indicates the output frame is invalid. After the AC97 playback channel is initialized, the Frame_valid bit should be set. Otherwise, AC97 Codec will abandon all output frames from the AC97 interface.
31:16(R)	-	16'h0	Reserved

Table 519: Codec AC97 Control Register (CODEC_AC97_CTRL)

- Codec AC97 Command Register (CODEC_AC97_CMD) – 0x20

  For writing the command register, it will depend on the settings of the 0x1c write command register and the 0x3c target_reg_sel register. The tag bits in the output frame slot0 (Bit [14], Bit

[13]) will be different in these two operations; however the data and the address will be the same. The command data and the address will be sent out according to 0x1c.

Bit	Name	Default	Description
7:0 (R)	-	8'b0	Reserved.
14:8 (R/W)	AC97_CMD_ADDR	7'b0	The operation address of register in AC97 Codec chip
15 (R/W)	CMD_TYP	1'b0	Command type: 1: Reads AC97 Codec status commands. 0: Writes AC97 Codec control register commands.
31:16 (R/W)	AC97_CMD_DATA	16'h0	AC97 Codec control register write data.

Table 520: Codec AC97 Command Register (CODEC_AC97_CMD)

- Codec AC97 Input Frame Tag Register (CODEC_AC97_AD_TAG) – 0x24

  This is a read-only register; the tag of the AC-link input frame is saved into this register. It can be used to judge whether AC97 Codec is ready.

Bit	Name	Default	Description
14:0 (R)	-	15'h0	Reserved
15 (R)	AD FRAME VALID	1'h0	AC97 input frame valid tag: 1: Codec is ready. 0: Codec is not ready.
31:16 (R)	-	16'h0	Reserved.

Table 521: Codec AC97 Input Frame Tag Register (CODEC_AC97_AD_TAG)

- Codec AC97 CMD data register (CODEC_AC97_STATUS) – 0x28

  When reading the control register from the AC97 Codec interface, the read data is saved in this register which contains 16-bit control register read data from AC97 Codec and its register index.

Bit	Name	Default	Description
7:0 (R)	-	8'h00	Reserved.
14:8 (R)	AC97_CMD_ADR	7'h0	AC97 Codec register address.
15 (R)	-	1'h0	Reserved.
31:16 (R)	AC97_CMD_DATA	16'h0	AC97 Codec register read data.

Table 522: Codec AC97 CMD Data Register (CODEC_AC97_STATUS)

- Codec AC97 Status Read Control Register (CODEC_ RD_AC97REG_ STS) – 0x2C

  This register is used to monitor AC97 Codec command ports (slot 1 and slot 2) and the I/O status port (slot 12).

Bit	Name	Default	Description
0 (R/W)	AC97_RD_REG_ STS	1'b0	Reads AC97 external register status bit: 1: AC97 Codec has delivered the read register data, the data and echoed address is saved in CODEC_AC97_STATUS. 0: AC97 Codec delivers no status data. Before issuing a read command to AC97 Codec, users need to write 1'b1 to clear its status (This bit will be set to zero when writing 1'b1).
1 (R)	AC97_CMD_REG _STS	1'b0	Writes the AC97 external register status bit: 1: AC97 Codec is writing the external register, the next write operation cannot be carried until this bit is cleared. 0: AC97 Codec can carry out the write operation.
2	-	1'b0	Reserved
4:3 (R)	CODEC_ID	2'b0	CODEC_ID should be always set to 2'b00.
31:5	-	27'h0	Reserved

Table 523: Codec AC97 Status Read Control Register (CODEC_ RD_AC97REG_ STS)

- Codec AC97 Output Slot Selection Register (CODEC_TXSLOT_EN) – 0x34

Bit	Name	Default	Description
9:0 (R/W)	CODEC_TXSLOT_EN	10'h0	AC97 output slot selection (up to 6 slots can be valid) 0: Slot output is disabled 1: Slot output is enabled Bit 0: Output frame Slot3 selection Bit 1: Output frame Slot4 selection Bit 2: Output frame Slot5 selection Bit 3: Output frame Slot6 selection Bit 4: Output frame Slot7 selection Bit 5: Output frame Slot8 selection Bit 6: Output frame Slot9 selection Bit 7: Output frame Slot10 selection Bit 8: Output frame Slot11 selection Bit 9: Output frame Slot12 selection
31:10 (R)	-	22'h0	Reserved

Table 524: Codec AC97 Output Slot Selection Register (CODEC_TXSLOT_EN)

- Codec AC97 Input Slot Selection Register (CODEC_RXSLOT_EN) – 0x38

Bit	Name	Default	Description
9:0 (R/W)	CODEC_RXSLOT_EN	10'h0	AC97 input slot selection (Maximum 2 slots allowed to be valid). 0: Slot input is disabled 1: Slot input is enabled Bit 0: Input frame Slot3 selection Bit 1: Input frame Slot4 selection Bit 2: Input frame Slot5 selection Bit 3: Input frame Slot6 selection Bit 4: Input frame Slot7 selection Bit 5: Input frame Slot8 selection Bit 6: Input frame Slot9 selection Bit 7: Input frame Slot10 selection Bit 8: Input frame Slot11 selection Bit 9: Input frame Slot12 selection
30:10(R)	-	21'h0	Reserved
31 (W/R)	MONO_MODE	1'b0	This bit is set for AC97 to work in the Mono mode, if this bit is set to 1, then only one slot is allowed to be enabled in this register bit[9:0]. 1'b0: AC97 is not working in MONO_MODE. 1'b1: AC97 is working in MONO_MODE.

Table 525: Codec AC97 Input Slot Selection Register (CODEC_RXSLOT_EN)

- Codec AC97 Input Auxiliary Slot Selection Register (CODEC_AUX_RX_EN) – 0x3C

Bit	Name	Default	Description
9:0 (R/W)	CODEC_AUX_RX_EN	10'h0	AC97 input slot selection (only 1 slot can be valid.) 0: Slot input is disabled 1: Slot input is enabled Bit 0: Input frame Slot3 selection Bit 1: Input frame Slot4 selection Bit 2: Input frame Slot5 selection Bit 3: Input frame Slot6 selection Bit 4: Input frame Slot7 selection Bit 5: Input frame Slot8 selection Bit 6: Input frame Slot9 selection Bit 7: Input frame Slot10 selection Bit 8: Input frame Slot11 selection Bit 9: Input frame Slot12 selection
14:10 (R)	-	5'h0	Reserved
15 (R/W)	AUX_SAM_EN	1'h0	Auxiliary input data sample enable
31:16 (R)	-	16'h0	Reserved

Table 526: Codec AC97 Input Auxiliary Slot Selection Register (CODEC_AUX_RX_EN)

NOTE – This register is used for sampling auxiliary input data. The sampled data can be either audio or non-audio data.

- Codec Auxiliary RX_FIFO DMA I/O Control Register (CODEC_RX_AUX_DMA_IO_CTRL) – 0xBC0

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b0	1: I/O mode 0: DMA mode
1 (R/W)	IO_DMA_RW	1'b0	1: Read from CODEC 0: Write to CODEC
2 (R/W)	DMA_FLUSH	1'b0	Flushes the DMA received FIFO in case the DATA_LEN set at the peripheral side does not match the DWORD size set in the DMA control.
31:3	-	29'h0	Reserved

Table 527: Codec Auxiliary RX_FIFO DMA I/O Control Register (CODEC_RX_AUX_DMA_IO_CTRL)

- Codec Auxiliary RX_FIFO Level Check Register (CODEC_RX_AUX_FIFO_LEVEL_CHK) – 0XBCC

Bit	Name	Default	Description
3:0 (R/W)	FIFO_SC	4'h4	Stop check in DWORD
9:4 (R)	-	6'h0	Reserved
13:10 (R/W)	FIFO_LC	4'h8	Low check in DWORD
19:14 (R)	-	6'h0	Reserved
23:20 (R/W)	FIFO_HC	4'hC	High check in DWORD
31:24 (R)	-	8'h0	Reserved

Table 528: Codec Auxiliary RX_FIFO Level Check Register (CODEC_RX_AUX_FIFO_LEVEL_CHK)

- Codec Auxiliary RX_FIFO Operation Register (CODEC_RX_AUX_FIFO_OP) – 0XBD0

Bit	Name	Default	Description
0 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is declared.
1 (R/W)	FIFO_RESET	1'b0	Internally linked to FIFO_START_INI. Set this bit to 1 to stop the FIFO and reset the FIFO internal status, including its relevant interrupt status. Set it to 0 in normal operation.
31:2 (R)	-	30'h0	Reserved.

Table 529: Codec Auxiliary RX_FIFO Operation Register (CODEC_RX_AUX_FIFO_OP)

- Codec Auxiliary RX FIFO Status Register (CODEC_RX_AUX_FIFO_STS) – 0xBD4

Bit	Name	Default	Description
0 (R/W)	RX_AUX_FIFO_FULL	1'b0	Auxiliary RX FIFO full status: 1: AUX RX FIFO is in full state. 0: AUX RX FIFO is not in full state. It indicates the current auxiliary RX FIFO full status. Once the FIFO status is changed, the status bit will be cleared.
1 (R/W)	RX_AUX_FIFO_EMPTY	1'b0	Auxiliary RX FIFO empty status: 1: AUX RX FIFO is in empty state. 0: AUX RX FIFO is not in empty state. It indicates the current auxiliary RX FIFO empty status, once the FIFO status is changed, the status bit will be cleared.
2 (R/W)	RX_AUX_FIFO_OFLOW	1'b0	Auxiliary RX FIFO overflow status:

Bit	Name	Default	Description
			1: AUX RX FIFO overflow takes place. 0: AUX RX FIFO is not overflowed. Users can write 1'b1 to clear the register bit after the RX FIFO overflow takes place.
3 (R/W)	RX_AUX_FIFO_UFLOW	1'b0	Auxiliary RX FIFO underflow status: 1: AUX RX FIFO underflow takes place. 0: AUX RX FIFO is not underflow. Users need to write 1'b1 to clear the register bit after the auxiliary RX FIFO underflow takes place.
4 (R/W)	RX_AUX_FIFO_OFLOW_INTR	1'b0	Auxiliary RX FIFO overflow interrupt: 1: Auxiliary RX FIFO overflow interrupt takes place 0: Auxiliary RX FIFO overflow interrupt not takes place Need write 1 to clear this bit.
5 (R/W)	RX_AUX_FIFO_UFLOW_INTR	1'b0	Auxiliary RX FIFO underflow interrupt: 1: Auxiliary RX FIFO underflow interrupt takes place. 0: Auxiliary RX FIFO underflow interrupt not takes place. Need write 1 to clear this bit.
31:6(R)	-	26'h0	Reserved

Table 530: Codec Auxiliary RX FIFO Status Register (CODEC_RX_AUX_FIFO_STS)

- Codec Auxiliary RX FIFO Interrupt Enable Register (CODEC_RX_ AUX_FIFO_INT_EN) – 0xBD8

Bit	Name	Default	Description
0 (R/W)	RX_AUX_FIFO_FULL_EN	1'b0	1: AUX RX FIFO full interrupt is enabled. 0: AUX RX FIFO full interrupt is disabled.
1 (R/W)	RX_AUX_FIFO_EMPTY_EN	1'b0	1: AUX RX FIFO empty interrupt is enabled. 0: AUX RX FIFO empty interrupt is disabled.
2 (R/W)	RX_AUX_FIFO_OFLOW_EN	1'b0	1: AUX RX FIFO overflow interrupt is enabled. 0: AUX RX FIFO overflow interrupt is disabled.
3 (R/W)	RX_AUX_FIFO_UFLOW_EN	1'b0	1: AUX RX FIFO underflow interrupt is enabled. 0: AUX RX FIFO underflow interrupt is disabled.
31:4 (R)	-	28'h0	Reserved

Table 531: Codec Auxiliary RX FIFO Interrupt Enable Register (CODEC_RX_ AUX_FIFO_INT_EN)

- Codec TX_FIFO DMA I/O Control Register (CODEC_TX_DMA_IO_CTRL) – 0xF80

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b0	1: I/O mode 0: DMA mode
1 (R/W)	IO_DMA_RW	1'b0	1: Read from CODEC 0: Write to CODEC
2 (R/W)	DMA_FLUSH	1'b0	Flushes the DMA received FIFO in case the DATA_LEN set at the peripheral side does not match the DWORD size set in the DMA control.
31:3	-	29'h0	Reserved

Table 532: Codec TX_FIFO DMA I/O Control Register (CODEC_TX_DMA_IO_CTRL)

- Codec TX_FIFO Level Check Register (CODEC_TX_FIFO_LEVEL_CHK) – 0xF8C

Bit	Name	Default	Description
5:0 (R/W)	FIFO_SC	6'h30	Stop check in DWORD
9:6 (R/W)	-	4'h0	Reserved
15:10 (R/W)	FIFO_LC	6'h20	Low check in DWROD
19:16	-	4'h0	Reserved
25:20 (R/W)	FIFO_HC	6'h10	High check in DWORD
31:26	-	6'h0	Reserved

Table 533: Codec TX_FIFO Level Check Register (CODEC_TX_FIFO_LEVEL_CHK)

- Codec TX_FIFO Operation Register (CODEC_TX_FIFO_OP) – 0xF90

Bit	Name	Default	Description
0 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is declared.
1 (R/W)	FIFO_RESET	1'b0	Internally linked to FIFO_START_INI Set it to 1 to stop the FIFO and reset the FIFO internal status, including its relevant interrupt status. Set it to 0 in normal operation.
31:2 (R)	-	30'h0	Reserved

Table 534: Codec TX_FIFO Operation Register (CODEC_TX_FIFO_OP)

- Codec TXFIFO Status Register (CODEC_TX_FIFO_STS) – 0xF94
  This register indicates TX FIFO status.

Bit	Name	Default	Description
0 (R)	TX_FIFO_FULL	1'b0	TX FIFO full status: 1: TX FIFO is in full state. 0: TX FIFO is not in full state. It indicates the current TX FIFO full status, once the FIFO status is changed, the status bit will be cleared.
1 (R)	TX_FIFO_EMPTY	1'b0	TX FIFO empty status: 1: TX FIFO is in empty state. 0: TX FIFO is not in empty state. It indicates the current TX FIFO empty status, once the FIFO status is changed, the status bit will be cleared.
2 (R/W)	TX_FIFO_OFLOW	1'b0	TX FIFO overflow status: 1: TX FIFO overflow takes place. 0: TX FIFO is not overflow. Users can write 1'b1 to clear the register bit after the TX FIFO overflow takes place.
3 (R/W)	TX_FIFO_UFLOW	1'b0	TX FIFO underflow status: 1: TX FIFO underflow takes place. 0: TX FIFO is not underflow. User needs to write 1'b1 to clear the register bit after the TX FIFO underflow takes place.
4 (R/W)	TX_FIFO_OFLOW_INTR	1'b0	TX FIFO overflow interrupt: 1: TX FIFO overflow interrupt takes place 0: TX FIFO overflow interrupt not takes place Need write 1 to clear this bit.
5 (R/W)	TX_FIFO_UFLOW_INTR	1'b0	TX FIFO underflow interrupt: 1: TX FIFO underflow interrupt takes place 0: TX FIFO underflow interrupt not takes place Need write 1 to clear this bit.
31:6 (R)	-	26'h0	Reserved

Table 535: Codec TXFIFO Status Register (CODEC_TX_FIFO_STS)

- Codec TXFIFO Interrupt Enable Register (CODEC_TX_FIFO_INT_EN) – 0xF98

Bit	Name	Default	Description
0 (R/W)	TX_FIFO_FULL_EN	1'b0	1: TX FIFO full interrupt is enabled. 0: TX FIFO full interrupt is disabled.
1 (R/W)	TX_FIFO_EMPTY_EN	1'b0	1: TX FIFO empty interrupt is enabled. 0: TX FIFO empty interrupt is disabled.
2 (R/W)	TX_FIFO_OFLOW_EN	1'b0	1: TX FIFO overflow interrupt is enabled. 0: TX FIFO overflow interrupt is disabled.
3 (R/W)	TX_FIFO_UFLOW_EN	1'b0	1: TX FIFO underflow interrupt is enabled. 0: TX FIFO underflow interrupt is disabled.
31:4 (R)	-	28'h0	Reserved.

Table 536: Codec TXFIFO Interrupt Enable Register (CODEC_TX_FIFO_INT_EN)

- Codec RX_FIFO DMA I/O Control Register (CODEC_RX_DMA_IO_CTRL) – 0xFC0

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b0	1: I/O mode 0: DMA mode
1 (R/W)	IO_DMA_RW	1'b0	1: Read from CODEC 0: Write to CODEC
2 (R/W)	DMA_FLUSH	1'b0	Flushes the DMA receive FIFO in case the DATA_LEN set at the peripheral side does not match the DWORD size set in the DMA control.
31:3 (R)	-	29'h0	Reserved

Table 537: Codec RX_FIFO DMA I/O Control Register (CODEC_RX_DMA_IO_CTRL)

- Codec RX_FIFO Level Check Register (CODEC_RX_FIFO_LEVEL_CHK) – 0xFCC

Bit	Name	Default	Description
3:0 (R/W)	FIFO_SC	4'h4	Stop check in DWORD
9:4 (R/W)	-	6'h0	Reserved
13:10 (R/W)	FIFO_LC	4'h8	Low check in DWROD
19:14 (R/W)	-	6'h0	Reserved
23:20 (R/W)	FIFO_HC	4'hC	High check in DWORD

Bit	Name	Default	Description
31:24	-	8'h0	Reserved

Table 538: Codec RX_FIFO Level Check Register (CODEC_RX_FIFO_LEVEL_CHK)

- Codec RX_FIFO Operation Register (CODEC_RX_FIFO_OP) – 0xFD0

Bit	Name	Default	Description
0 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is declared.
1 (R/W)	FIFO_RESET	1'b0	Internally linked to FIFO_START_INI. Set to 1 to stop the FIFO and reset the FIFO internal status, including its relevant interrupt status. Set to 0 in normal operation.
31:2 (R)	-	30'h0	Reserved

Table 539: Codec RX_FIFO Operation Register (CODEC_RX_FIFO_OP)

- Codec RXFIFO Status Register (CODEC_RX_FIFO_STS) – 0xFD4
  This register indicates the RX FIFO status:

Bit	Name	Default	Description
0 (R/W)	RX_FIFO_FULL	1'b0	RX FIFO full status: 1: RX FIFO is in full state. 0: RX FIFO is not in full state. It indicates the current RX FIFO full status, once the FIFO status is changed, the status bit will be cleared.
1 (R/W)	RX_FIFO_EMPTY	1'b0	RX FIFO empty status: 1: RX FIFO is in empty state. 0: RX FIFO is not in empty state. It indicates the current RX FIFO empty status, once the FIFO status is changed, the status bit will be cleared.
2 (R/W)	RX_FIFO_OFLOW	1'b0	RX FIFO overflow status: 1: RX FIFO overflow takes place. 0: RX FIFO is not overflow. User can write 1'b1 to clear the register bit after the RX FIFO overflow takes place.
3 (R/W)	RX_FIFO_UFLOW	1'b0	RX FIFO underflow status: 1: RX FIFO underflow takes place. 0: RX FIFO is not underflow.

Bit	Name	Default	Description
			Users need to write 1'b1 to clear the register bit after the RX FIFO underflow takes place.
4 (R/W)	RX_FIFO_OFLOW_INTR	1'b0	RX FIFO overflow interrupt: 1: RX FIFO overflow interrupt takes place 0: RX FIFO overflow interrupt not takes place Need write 1 to clear this bit.
5 (R/W)	RX_FIFO_UFLOW_INTR	1'b0	RX FIFO underflow interrupt: 1: RX FIFO underflow interrupt takes place 0: RX FIFO underflow interrupt not takes place Need write 1 to clear this bit.
31:6 (R)	-	26'h0	Reserved

Table 540: Codec RXFIFO Status Register (CODEC_RX_FIFO_STS)

- Codec RXFIFO Interrupt Enable Register (CODEC_RX_FIFO_INT_EN) – 0xFD8

Bit	Name	Default	Description
0 (R/W)	RX_FIFO_FULL_EN	1'b0	1: RX FIFO full interrupt is enabled. 0: RX FIFO full interrupt is enabled.
1 (R/W)	RX_FIFO_EMPTY_EN	1'b0	1: RX FIFO empty interrupt is enabled. 0: RX FIFO empty interrupt is disabled.
2 (R/W)	RX_FIFO_OFLOW_EN	1'b0	1: RX FIFO overflow interrupt is enabled. 0: RX FIFO overflow interrupt is disabled.
3 (R/W)	RX_FIFO_UFLOW_EN	1'b0	1: RX FIFO underflow interrupt is enabled. 0: RX FIFO underflow interrupt is disabled.
31:4 (R)	-	28'h0	Reserved.

Table 541: Codec RXFIFO Interrupt Enable Register (CODEC_RX_FIFO_INT_EN)

DAC Interface

Overview

DAC interface is a digital Virtual Component. The DAC architecture uses entirely digital approach to convert digital source input to PWM signal output.

Feature List

- I^2S master mode (16 bits)
- Supports 8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k. 48k. 64k, 88.2k and 96k sample rate.

- Programmable volume control: +6dB to -100dB in 1dB steps.
- Support speaker and headphone mode.

Pin Descriptions

External Pin Descriptions

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_URXD_1	I/O	spk_p	USP1	Input and internal pull down	Logic output for speaker. DAC shares this pin with USP1.
X_UTFS_1	I/O	spk_n	USP1	Input and internal pull down	Complementary output for speaker. DAC shares this pin with USP1.
X_UTXD_1	I/O	headphone_l	USP1	Input	Left channel data of headphone. DAC shares this pin with USP1.
X_USCLK_1	I/O	headphone_r	USP1	Input	Right channel data of headphone. DAC shares this pin with USP1.
X_AC97_BIT_CLK	I/O	bclk	AC97	Input	AC97 clock.
X_AC97_SYNC	I/O	lro	AC97	Output low	AC97 sync signal.

Table 542: DAC External Pin Descriptions

Functional Descriptions

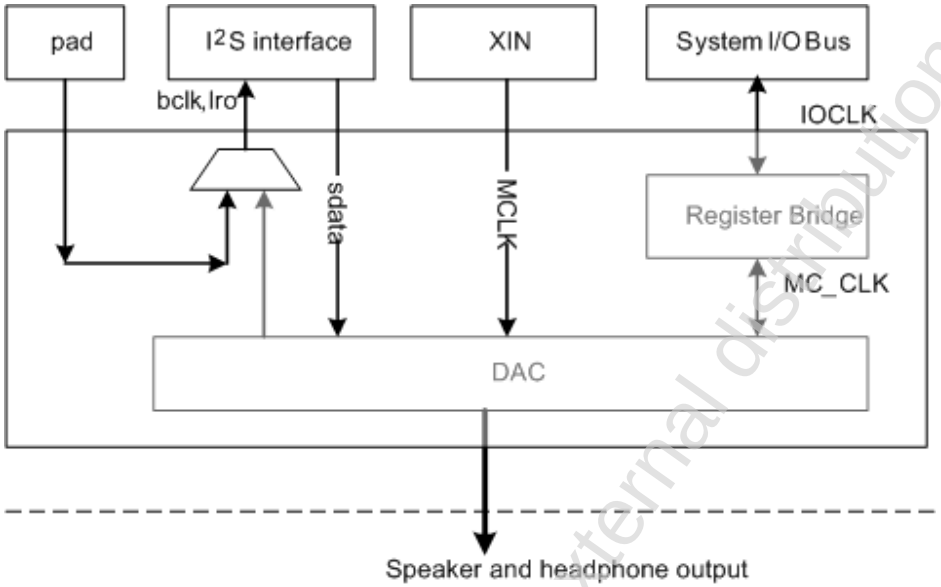


Figure 86: DAC Block Diagram

The DAC interface shares the same clock with the audio interface.

It uses the external XIN (24MHz) as the working clock (MCLK).

The register access is through an internal register bridge. It transfers the IOBUS to the internal register access interface. IOCLK will be divided according to the register setting and send the output to DAC module as MC_CLK (working clock of internal register access).

The DAC interface generates the I^2S clock and sync signal according to the selected sample clock and send to the internal I^2S interface. I^2S interface fetch the output audio data from the external memory through DMA controller and send the data to the DAC interface.

The DAC interface receives the audio data through the I^2S bus and converts it to the PWM format.

It supports speaker and headphone mode. Only one mode is available at one time.

DAC interface generates four output signals, PWMLP/PWMLN (left channel PWM data), PWMRP/PWMRN (right channel PWM data). It will be sent to the different pad according to the working mode.

In speaker mode, PWMLP and PWMLN will send to X_URXD_1 and X_UTFS_1 as speaker output.

In headphone mode, PWMLP and PWMRP will send to X_UTXD_1 and X_USCLK_1 as headphone output.

Register Definition

Register Address Mapping

Base Address

Access Type	Address Mapping
DAC base address through RISC I/O	0x80068000

Table 543: DAC Address Mapping

Register Mapping

RISC Address <15:0>	Register	Description
0x8000	DAC_BG_OP	Operation type register
0x8004	DAC_BG_ADDR	Operation address register
0x8008	DAC_BG_WDATA	Data to write
0x800C	DAC_BG_RDATA	Data to read
0x8010	DAC_DIV	Divider register
0x8014	DAC_CTRL	Control register
0x8018	DAC_FRAME_COUNT	Count THE I^2S frame for mute control
0x801C~0x807C	-	Reserved
0x8080	DAC_POP_CTRL	Control register for pop suppression.

Table 544: DAC Register Definition

Register Descriptions

- DAC Bridge Operation Type Register (DAC_BG_OP) – RISC: 0x8000

Bit	Name	Default	Description
1:0 (R/W)	OP_TYPE	2'b0	00: IDLE 01: Reserved 10: Read 11: Write Write 10 or 11 will cause the bridge to start operation. If the operation is finished, these two bits will be cleared to 00 Reference wait time (read 8*Tioclk, write 8*Tioclk). It is about 64ns when IOCLK works at 125MHz frequency.

Bit	Name	Default	Description
31:2	-	30'h0	Reserved

Table 545: DAC Bridge Operation Type Register

- Operation Address Register (DAC_BG_ADDR) – RISC : 0x8004

Bit	Name	Default	*Description*
5:0 (R/W)	OP_ADDR	6'h0	Address of the register in DAC
31:6	-	26'h0	Reserved

Table 546: Operation Address Register

- Write Data to Write Register(DAC_BG_WDATA) – RISC: 0x8008

Bit	Name	Default	*Description*
7:0 (R/W)	OP_WDATA	8'h0	Data to write to DAC
31:8	-	24'h0	Reserved

Table 547: Write Data to Write Register

- Read Data Register (DAC_BG_RDATA) – RISC: 0x800C

Bit	Name	Default	*Description*
7:0 (R)	RDATA	8'h0	Data read from DAC
31:8	-	24'h0	Reserved

Table 548: Read Data Register

- Clock Divide Register (DAC_DIV) – RISC: 0x8010

Bit	Name	Default	*Description*
3:0 (R/W)	DIV_NUM	4'h3	Divider number which used to divider the IOCLK. MC_CLK = IOCLK/(DIV_NUM+1) This value must be an odd value.
31:4	-	-	Reserved.

Table 549: Clock Divider Register

- Control Register (DAC_CTRL) – RISC: 0x8014

Bit	Name	Default	*Description*
0 (R/W)	DAC_EN	1'b0	1'b0: Disable DAC module. 1'b1: Enable DAC module
1 (R/W)	SPEAKER/HEADPHONE	1'b0	1'b0: Enable speaker mode 1'b1: Enable headphone mode
2 (R/W)	INV_PWMLP	1'b0	Invert pwmlp, 1'b0: Send the pwmlp out directly. 1'b1: Invert the pwmlp before out. This bit must be set at the initialize process and do not change it anymore.
3 (R/W)	INV_PWMLN	1'b0	Invert pwmln 1'b0: Send the pwmln out directly. 1'b1: Invert the pwmln before out. This bit must be set at the initialize process and do not change it anymore.
4 (R/W)	INV_PWMRP	1'b0	Invert pwmrp 1'b0: Send the pwmrp out directly. 1'b1: Invert the pwmrp before out. This bit must be set at the initialize process and do not change it anymore.
7:5	-	-	Reserved.
8 (R)	SB_STATUS	1'b0	This bit is for debug and read only. 0: The DAC works at normal mode. 1: The DAC enters the standby mode.
9(R)	FRAME_COUNT_STATUS	1'b0	This bit is debugging only usage. When the internal frame count is bigger than the DAC_FRAME_COUNT, it will change to 1. It can be used to wait the DAC enters the mute or quit the mute mode successfully.
31:10	-	-	Reserved.

Table 550: Control Register

- I²S Frame Count Register (DAC_FRAME_COUNT) – RISC: 0x8018

Bit	Name	Default	*Description*
15:0 (R/W)	FRAME_NUM	16'h0	This register is debugging only usage. Internal counter will count the lro number when this register is set, if it is bigger than the FRAME_NUM, the frame_count_status will be changed to 1.
31:16	-	-	Reserved.

Table 551: I²S Frame Count Register

- Control Register (DAC_POP_CTRL) – RISC: 0x8080

Bit	Name	Default	*Description*
15:0 (W)	UP_WAIT_NO	16'hf	Wait number of enable headphone process The default value is the best setting. Do not change it.
31:16 (W)	DOWN_WAIT_NO	16'hf	Wait number of disable headphone process. The default value is the best setting. Do not change it.

Table 552: Control Register

Register Bridge Address Mapping

Address <6:0>	Register	Description
0x0000	AICR	Audio interface control register
0x0001	CR1	Control register1
0x0002	CR2	Control register2
0x0003	CR3	Control register3
0x0004	VMCR	Volume and mute control register

Table 553: Internal Register Bridge Register Mapping

Register Bridge Register Description

- Audio interface control Register (AICR) – Address: 0x00

Bit	Name	Default	Description
0(R/W)	SERIAL	1'b1	This bit must be set to 1.
1(R/W)	IIS	1'b1	This bit must be set to 1.
3:2	ADWL	2'b3	This value must be set to 0.

Bit	Name	Default	Description
(R/W)			
7:4	-	4'h0	Reserved

Table 554: Audio interface control Register

- Control Register1 (CR1) – Address: 0x01

Bit	Name	Default	Description
3:0 (R/W)	FREQ	4'h7	Sample frequency configuration 4'h0: 8k, 4'h1: 11.025k, 4'h2: 12k, 4'h3: 16k, 4'h4: 22.05k, 4'h5: 24k, 4'h6: 32k, 4'h7: 44.1k. 4'h8: 48k. 4'h9: 64k, 4'ha: 88.2k 4'hb: 96k
5:4 (R/W)	QUARTZ	2'h0	This value must be set to 0.
7:6	-	2'h0	Reserved

Table 555: Control Register1

- Control Register2 (CR2) – Address: 0x02

Bit	Name	Default	Description
2:0 (R/W)	DELTAFREQ	3'h0	Sampling frequency offset, Add some positive or negative offset on Sampling frequency
7:3	-	5'h0	Reserved

Table 556: Control Register2

In order to perform operations such as audio/video synchronization, the DAC allows the users to fine tune the Sampling frequency by decreasing or increasing it by some few percent. This is achieved by register configuration as described in the followings tables.

FS @ 8kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.125%
110	-2.5%
111	-1.5625%
001	+1.25%
010	+2%
011	+3.125%
100	No change
FS @ 11.025kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]]	**Description**
000	No change
101	-2.9479%
110	-2.0408%
111	-1.1338%
001	+1.2245%
010	+2.0408%
011	+2.9479%
100	No change
FS @ 12kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3%
110	-2.0833%
111	-1.25%
001	+1.25%
010	+2.0833%

011	+3%
100	No change
FS @ 16kHz, Sampling Frequency configuration bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.125%
110	-1.75%
111	-1%
001	+1.25%
010	+2%
011	+3.125%
100	No change
FS @ 22.05kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-2.9478%
110	-2.0408%
111	-1.1337%
001	+0.9070%
010	+2.0408%
011	+3.1746%
100	No change
FS @ 24kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.125%
110	-2.0833%
111	-1.25%
001	+1.25%
010	+2.0833%

011	+3.125%
100	No change
FS @ 32kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.125%
110	-1.875%
111	-1.25%
001	+1.25%
010	+2.1875%
011	+3.125%
100	No change
FS @ 44.1kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.0612%
110	-2.0408%
111	-1.1338%
001	+0.9070%
010	+2.0408%
011	+3.1746%
100	No change
FS @ 48kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.125%
110	-2.0833%
111	-1%
001	+1%
010	+2.0833%

011	+3.125%
100	No change
FS @ 64kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.125%
110	-1.875%
111	-1.25%
001	+1.25%
010	+2.1875%
011	+3.125%
100	No change
FS @ 88.2kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.06122%
110	-2.0408%
111	-1.1338%
001	+0.9070%
010	+2.0408%
011	+3.1746%
100	No change
FS @ 96kHz, Sampling Frequency Configuration Bits DELTA FREQ[2 :0]	**Description**
000	No change
101	-3.125%
110	-2.08333%
111	-1.041167%
001	+1.04167%
010	+2.0833%
011	+3.125%
100	No change

- Control Register3 (CR3) – Address: 0x03

Bit	Name	Default	Description
2:0 (R/W)	LIMR	3'h4	Limiter Rate, Set the volume transition time corresponding to one volume step change time. 000 = 1 / Fs 001 = 2 / Fs 010 = 4 / Fs 011 = 8 / Fs 100 = 16 / Fs 101 = 32 / Fs 110 = 64 / Fs 111 = 128 / Fs
3 (R/W)	L/R	1'b0	Left/Right select bit 0 = The next read or write operations on VMCR register will concern the left channel. 1 = The next read or write operations on VMCR register will concern the right channel.
4 (R/W)	SB	1'b0	Stand-by/Normal mode 0 = Normal mode 1 = Stand-by mode
5 (R/W)	-	1'b0	Reserved
6(R/W)	DEEMPH	1'b0	De-emphasis filter enable (used only when de-emphasis option is selected) 1 = Enable the de-emphasis filter (for Fs = 44.1kHz) 0 = Disable the de-emphasis filter. The de-emphasis filter respects the standard 15µs/50µs response only at sampling rate Fs = 44.1kHz.
7(R/W)	FPWM	1'b0	1 = PWM outputs are forced to 0 when DAC is in standby mode 0 = Normal mode

Table 557: Control Register3

- Volume and mute control register (VMCR) – Address: 0x04

Bit	Name	Default	Description
6:0 (R/W)	VOL	7'h79	Volume Control 1111111 = +6Db 1111110 = +5dB 1111101 = +4dB 1111100 = +3dB … 0010110 = -99dB 0010101 = -100dB 0010100 = -∞ (mute) … 0000000 = -∞ (mute) Allows control of the left or right channel signal level in 1dB increments from +6dB to –100dB. Bit L/R must be set first to select the left or the right channel. All volume settings less than –100dB are equivalent to muting the selected channel. When the volume data is set, the volume value is changed with 1dB step to the target value, the period of transition set by LIMR[2:0] bits.
7 (R/W)	MUTE	1'h0	Soft Mute Control, acts on booth left and right channel 0 = Left and right channel mute off. 1 = Left and right channel mute on. Acts on both left and right channels whatever the L/R bit value. When entering mute mode (MUTE = 1), the volume is decreased progressively down to -∞, with a rate specified by the limiter rate bits LIMR[2:0]. When leaving the mute mode (MUTE = 0), the volume is increased progressively from -∞ up to the setting volume value, with a rate specified by the limiter rate bits LIMR[2:0].

Table 558: Volume and mute control register

UART Controller

Overview

There are two dedicated UARTs (Universal Asynchronous Receiver/Transmitter) in SiRFatlasV:

- UART0: Full UART with DMA (128byte*2 FIFO) and flow control
- UART1: Full UART without DMA (32byte*2 FIFO)

- UART1: Without DMA and DSP interfaces

Feature List

UART supports the following features:

- 5, 6, 7, or 8 bits per character
- 1, or 2 stop bit detection and generation
- Stick bit (none, even, odd, space, mark)
- Two clock dividers for more flexible clock division
- Line break detection
- Internal loop back diagnostic functionality
- Independent 128-byte transmit and receive FIFOs for DMA (UART0)
- Independent 32-byte transmit and receive FIFOs (UART1)
- Maximum baud rate of up to 10 Mbps

UART1 supports the UART0 features except for DMA-related features and flow control feature.

NOTE – If user needs more than two UARTs, please configure USP as UARTs mode. For more details, refer to the Section of USP.

Pin Descriptions

External Pin Descriptions

Pin Name	I/O Type	Pin MUX	Default Function	Default Status	Description
X_TXD_0	Output	-	UART	Output high	UART0 TXD
X_RXD_0	Input	GPIO	UART	Input with pull up	UART0 RXD
X_URFS_0	Output	USP0	UART	Input with pull down	UART0_RTS
X_URFS_1	Input	USP1	UART	Input with pull down	UART0_CTS
X_TXD_1	Output	-	UART	Output high	UART1 TXD
X_RXD_1	Input	GPIO	UART	Input with pull up	UART1 RXD

Table 559: UART External Pin Description

Functional Descriptions

Function Modes

Generally, low-speed UART serial devices come with one of the two modes, without hardware flow-control or with hardware flow-control modes.

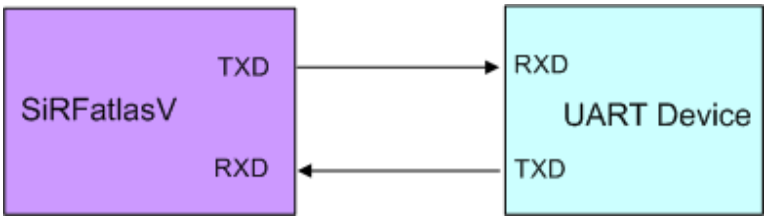


Figure 87: UART without Flow Control

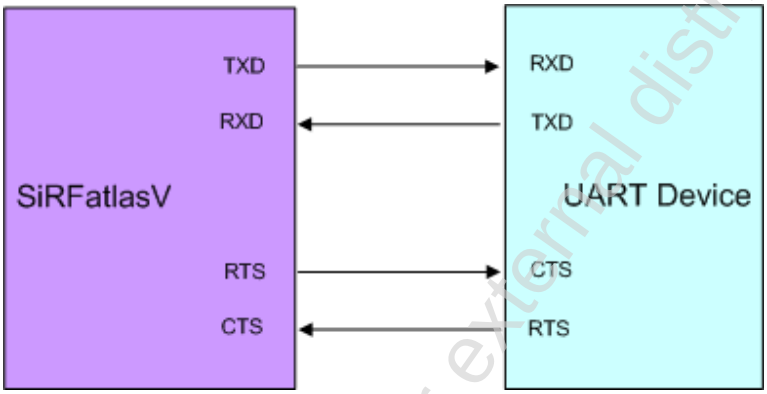


Figure 88: UART with Flow Control

In some special cases, users can connect UART with modems, and in that case, the modem control signals (DCD, RI, RTS, etc.) can connect with SiRFatlasV GPIOs.

Timing Diagram

The SiRFatlasV UART frame is composed of the following domains:

- 1bit start bit
- 5/6/7/8 bit data
- None or 1bit stick bit
- 1 or 2bit stop bit

The stick bit may be:

- Odd parity bit
- Even parity bit
- Mark (logic 1)
- Space (logic 0)

The following diagram is an example of UART frame timing:

Figure 89: UART Frame Timing

Register Definition

Register Address Mapping

Base Address

Each UART in SiRFatlasV has the same set of registers, therefore offset addresses are the same.

The actual address of the UART register is equal to the UART base address plus the offset address.

- UART0 base address 0x80000000
- UART1 base address 0x80100000

Register Mapping

RISC Address <11:0>	DSP I/O Address <7:0>	Register	Description
0x40	0x20	UART_LINE_CTRL	UART line control register
0x44	-	-	Reserved
0x48	-	-	Reserved
0x4c	0x26	UART_TX_RX_EN	UART transmit and receive enable register
0x50	0x28	UART_DIVISOR	UART baud rate divider register
0x54	0x2A	UART_INT_EN	UART interrupt enable register
0x58	0x2C	UART_INT_STATUS	UART interrupt status register
0x5c	0x2E	UART_RISC_DSP_MODE	UART accessing select register
0x60~FC	-	-	Reserved
0x100	0x40	UART_TX_DMA_IO_CTRL	UART TXFIFO DMA/IO register
0x104	0x42~0x43	UART_TX_DMA_IO_LEN	UART transmit data length register

RISC Address <11:0>	DSP I/O Address <7:0>	Register	Description
0x108	0x44	UART_TXFIFO_CTRL	UART TXFIFO control register
0x10C	0x46~0x47	UART_TXFIFO_LEVEL_CHK	UART TXFIFO check level register
0x110	0x48	UART_TXFIFO_OP	UART TXFIFO operation register
0x114	0x4A	UART_TXFIFO_STATUS	UART TXFIFO status register
0x118	0x4C	UART_TXFIFO_DATA	UART TXFIFO bottom
0x11C	-	-	Reserved
0x120	0x50	UART_RX_DMA_IO_CTRL	UART RXFIFO DMA/IO register
0x124	0x52~0x53	UART_RX_DMA_IO_LEN	UART receive length register
0x128	0x54	UART_RXFIFO_CTRL	UART RXFIFO control register
0x12C	0x56~x57	UART_RXFIFO_LEVEL_CHK	UART RXFIFO check level register
0x130	0x58	UART_RXFIFO_OP	UART RXFIFO operation register
0x134	0x5A	UART_RXFIFO_STATUS	UART RXFIFO status register
0x138	0x5C	UART_RXFIFO_DATA	UART RXFIFO bottom
0x140	-	AFC_CTRL	Auto flow control register
0x144	-	-	Reserved
0x148	-	SWH_DMA_IO	Register for switching between DMA and I/O
0x14c	-	-	Reserved

Table 560: UART Register Mapping

Register Descriptions

- UART Line Control Register (UART_LINE_CTRL) – RISC: 0x40, DSP: 0x20
  It is the same for the UART0/1.

Bit	Name	Default	Description
1:0 (R/W)	DATA_BIT_LEN	2'b0	2'b00: 5 data bits in one frame 2'b01: 6 data bits in one frame 2'b10: 7 data bits in one frame 2'b11: 8 data bits in one frame
2 (R/W)	STOP_BIT_LEN	1'b0	1'b0: 1 stop bit 1'b1: 2 stop bits

Bit	Name	Default	Description
3 (R/W)	PARITY_ENA	1'b0	1'b0: Parity bit disable 1'b1: Parity bit enable
4 (R/W)	EVEN_BIT	1'b0	See STICK_BIT description
5 (R/W)	STICK_BIT	1'b0	The bit[5:3]of UART_LINE_CTRL determines the parity bit status: 3'b000: No stick bit 3'b001: Even bit parity 3'b011: Odd bit parity 3'b101: Mark bit parity 3'b111: Space bit parity
6 (R/W)	SET_BREAK	1'b0	1'b0: TXD in normal transmit state. 1'b1: TXD is forced to low level.
7 (R/W)	LOOP_BACK	1'b0	1'b0: No loop-back 1'b1: RXD is not connected to an external pin but to TXD internally and the transmitted data is looped back to the receive pin.
15:8	-	-	Reserved.
31:16 (R/W)	TIMOUT_NUM	16'h0	This register specifies the bit number for receive timeout. If there is no new data received after the bit number time has elapsed since the last data is received, then a receive timeout interrupt will take place.

Table 561: UART Line Control Register (UART_LINE_CTRL)

- UART Transmit/Receive Enable Register (UART_TX_RX_EN) – RISC: 0x4C, DSP: 0x26
  It is the same for UARTx.

Bit	Name	Default	Description
0 (R/W)	RX_EN	1'b0	Receive enable bit: 0: Disabled 1: Enabled
1 (R/W)	TX_EN	1'b0	Transmit enable bit: 0: Disabled 1: Enabled
31:2	-	30'h0	Reserved

Table 562: UART Transmit/Receive Enable Register (UART_TX_RX_EN)

- UART Clock Divisor Register (UART_CLK_DIV) – RISC: 0x50, DSP: 0x28
  It is the same for UARTx.

Bit	Name	Default	Description
15:0 (R/W)	IOCLK_DIV	16'hffff	IOCLK_DIV = fio_clock/ (baud * (SAMPLE_DIV+1)) – 1
21:16 (R/W)	SAMPLE_DIV	6'h0f	SAMPLE_DIV should be larger than or equal to 15.
31:22	-	10'h0	Reserved.

Table 563: UART Clock Divisor Register (UART_CLK_DIV)

In normal mode:

- SAMPLE_DIV is always set to 15
- IOCLK_DIV is f_{io_clock}/ (baud * 16) – 1

In order to get the accurate high baud rate when the frequency of IOCLK is low, such as 12 MHz, the value of SAMPLE_DIV must be changed to larger than 15. The exact high baud rate can be acquired if both SAMPLE_DIV and IOCLK_DIV are adjusted. For example, if IOCLK is 12 MHz, and to acquire a high baud rate of 115200, SAMPLE_DIV and IOCLK_DIV must be set to 25 and 3 respectively.

- UART Interrupt Enable Register (UART_INT_EN) – RISC: 0x54, DSP: 0x2A
  (UART1 has no DMA interfaces and related registers)

Bit	Name	Default	Description
0 (R/W)	RX_DONE_INT_EN	1'b0	Receive done interrupt enable: 0: Disabled 1: Enabled
1 (R/W)	TX_DONE_INT_EN	1'b0	Transmit done interrupt enable: 0: Disabled 1: Enabled
2 (R/W)	RX_OFLOW_INT_EN	1'b0	Receive overflow interrupt enable: 0: Disabled 1: Enabled
3 (R/W)	TX_ALLOUT_INT_EN	1'b0	All data is transmitted interrupt enable: 0: Disabled 1: Enabled
4 (R/W)	RX_IO_DMA_ INT_EN	1'b0	IO/DMA receive interrupt enable: 0: Disabled 1: Enabled
5 (R/W)	TX_ IO_DMA_ INT_EN	1'b0	IO/DMA transmit interrupt enable: 0: Disabled

Bit	Name	Default	Description
			1: Enabled
6 (R/W)	RXFIFO_FULL_INT_EN	1'b0	Receive FIFO full interrupt enable: 0: Disabled 1: Enabled
7 (R/W)	TXFIFO_EMPTY_INT_EN	1'b0	Transmit FIFO empty interrupt enable: 0: Disabled 1: Enabled
8 (R/W)	RXFIFO_THD_INT_EN	1'b0	Receive FIFO threshold interrupt enable: 0: Disabled 1: Enabled
9 (R/W)	TXFIFO_THD_INT_EN	1'b0	Transmit FIFO threshold interrupt enable: 0: Disabled 1: Enabled
10 (R/W)	FRAME_ERR_INT_EN	1'b0	UART error frame interrupt enable: 0: Disabled 1: Enabled
11 (R/W)	RXD_BREAK_INT_EN	1'b0	RXD pin break interrupt enable: 0: Disabled 1: Enabled
12 (R/W)	RX_TIMEOUT_INT_EN	1'b0	Receive timeout interrupt enable: 0: Disabled 1: Enabled
13 (R/W)	PARITY_ERR_INT_EN	1'b0	Parity error interrupt enable: 0: Disabled 1: Enabled
14 (R/W)	CTS_INT_EN	1'b0	CTS change interrupt enable 0: Disable 1: Enable
15 (R/W)	RTS_INT_EN	1'b0	RTS change interrupt enable 0: Disable 1: Enable
16 (R/W)	PLUG_IN_INT_EN	1'b0	RDS plug-in interrupt enable 0: Disable 1: Enable
31:17	-	15'h0	Reserved

Table 564: UART Interrupt Enable Register (UART_INT_EN)

- UART Interrupt Status Register (UART_INT_STATUS) – RISC: 0x58, DSP: 0x2C

  UART1 has no DMA interface and related registers. RISC/DSP writes 1'b1 to the bit will clear the interrupt.

Bit	Name	Default	Description
0 (R/W)	RX_DONE	1'b0	Valid data has been received in the RXFIFO interrupt (valid data length is define by RXFIFO_WIDTH, one transfer): 0: Invalid 1: Valid
1 (R/W)	TX_DONE	1'b0	Valid data has been transmitted from the TXFIFO interrupt (valid data length is define by the TXFIFO_WIDTH, one transfer): 0: Invalid 1: Valid
2 (R/W)	RX_OFLOW	1'b0	RXFIFO overflow Interrupt: 0: Invalid 1: Valid
3 (R/W)	TX_ALL_EMPTY	1'b0	All data in both TXFIFO and TX shifter sent out interrupt: 0: Invalid 1: Valid
4 (R/W)	DMA_IO_RX_DONE	1'b0	RXFIFO has received all data of the data package, of which the size is defined by the UART_RX_DMA_IO_LEN register: 0: Invalid 1: Valid
5 (R/W)	DMA_IO_TX_DONE	1'b0	TXFIFO has transmitted all data of a data package whose size is defined by the UART_TX_DMA_IO_LEN register: 0: Invalid 1: Valid
6 (R/W)	RXFIFO_FULL	1'b0	RXFIFO full interrupt: 0: Invalid 1: Valid
7 (R/W)	TXFIFO_EMPTY	1'b0	TXFIFO empty interrupt: 0: Invalid 1: Valid
8 (R/W)	RXFIFO_THD_REACH	1'b0	Time to read RXFIFO when the number of data in the RXFIFO reaches the threshold: 0: Invalid 1: Valid

Bit	Name	Default	Description
			This interrupt occurs only at the rising edge when the FIFO data reaches the threshold
9 (R/W)	TXFIFO_THD_REACH	1'b0	Time to write TXFIFO when the number of data in the TXFIFO reaches the threshold: 0: Invalid 1: Valid
10 (R/W)	FRM_ERR	1'b0	Received a UART frame interrupt in error: 0: Invalid 1: Valid
11 (R/W)	RXD_BREAK	1'b0	If this bit changes to 1, it means the RXD line has been in low level for at least 10 bits and the receiving action must be stopped. To correct this, disable the RX_EN bit in the USP_TX_RX_EN register, then wait some time to double check this bit. If no new break occurs, enable RX_EN and wait some time to reset and restart RXFIFO to flush invalid data and be ready to accept new data. 0: Invalid 1: Valid
12 (R/W)	UART_RX_TIMEOUT	1'b0	If this bit becomes 1, then it means there is no more new data received for the time specified by the timeout bit number defined in the TIMOUT_NUM bits of the UART_LINE_CTRL register since the last data has been received in the FIFO. 0: Invalid 1: Valid
13 (R/W)	PARITY_ERR	1'b0	0: No parity errors. 1: There is a parity error in the received frame.
14 (R/W)	CTS_CHANGE	1'b0	0: No change 1: Change, rising edge or falling edge Write it to 1'b1 for clearing CTS_INT.
15 (R/W)	RTS_CHANGE	1'b0	0: No change 1: Change, rising edge or falling edge Write it to 1'b1 for clearing RTS_INT.
16 (R/W)	PLUG_IN	1'b0	0: Unplug or normal work 1: Plug-in Write it to 1'b1 for clearing PLUG_IN_INT
31:17	-	15'h0	Reserved

Table 565: UART Interrupt Status Register (UART_INT_STATUS)

- UART RISC/DSP Mode Register (UART_RISC_DSP_MODE) – RISC: 0x5C, DSP: 0x2E

  This register can only be written by RISC. For DSP, it is read-only. Since UART1 has no DSP interface, therefore this register will not work for UART1.

Bit	Name	Default	Description
0 (R/W)	RISC_DSP_SEL	1'b0	0: UART is accessed by RISC. 1: UART is accessed by DSP.
31:1	-	31'h0	Reserved.

Table 566: UART RISC/DSP Mode Register (UART_RISC_DSP_MODE)

- UART TX DMA I/O MODE Register (UART_TX_DMA_IO_CTRL) – RISC: 0x100, DSP: 0x40

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b0	1: For I/O mode 0: For DMA mode
3:1	-	3'b0	Reserved
5:4 (R/W)	TX_ENDIAN_MODE	2'b0	Only DMA operations are affected: 00: No changes 01: Byte exchange in DWORD 10: Word exchange in DWORD 11: Byte exchange in WORD
31:6	-	26'h0	Reserved

Table 567: UART TX DMA I/O MODE Register (UART_TX_DMA_IO_CTRL)

- UART TX DMA I/O Length Register (UART_TX_DMA_IO_LEN) –RISC: 0x104, DSP: 0x42~0x43

Bit	Name	Default	Description
31:0 (R/W)	DATA_LEN	32'h0	The byte number of a DMA or I/O transfer. If it is set to zero, then the I/O or DMA transfer will work continuously until it is stopped.

Table 568: UART TX DMA I/O Length Register (UART_TX_DMA_IO_LEN)

- UARTx TX FIFO Control Register (UART_TX_FIFO_CTRL) – RISC: 0x108, DSP: 0x44
  - For UART0

Bit	Name	Default	Description
6:0 (R/W)	FIFO_THD<6:0>	7'h0	A threshold value in byte that triggers an interrupt. An interrupt is triggered when the data count in the FIFO reaches the threshold
31:7	-	25'h0	Reserved

Table 569: UARTx TX FIFO Control Register (UART_TX_FIFO_CTRL) for UART0

  - For UART1

Bit	Name	Default	Description
4:0 (R/W)	FIFO_THD<4:0>	5'h0	A threshold value in byte that triggers an interrupt. An interrupt is triggered when the data count in the FIFO reaches the threshold.
31:5	-	27'h0	Reserved

Table 570: UARTx TX FIFO Control Register (UART_TX_FIFO_CTRL) for UART1

- UARTx TX FIFO Level Check Register (UART_TX_FIFO_LEVEL_CHK) –RISC: 0x10C, DSP: 0x46~0x47
  - For UART0: Controls DMA request time based on this FIFO level check.

Bit	Name	Default	Description
4:0 (R/W)	FIFO_SC<4:0>	5'h0	Stop check in DWORD
9:5	-	5'h0	Reserved
14:10 (R/W)	FIFO_LC<4:0>	5'h0	Low check in DWROD
19:15	-	5'h0	Reserved
24:20 (R/W)	FIFO_HC<4:0>	5'h0	High check in DWORD
31:25	-	7'h0	Reserved

Table 571: UARTx TX FIFO Level Check Register (UART_TX_FIFO_LEVEL_CHK) for UART0

- UART TX FIFO Operation Register (UART_TX_FIFO_OP) – RISC: 0x110, DSP: 0x48

  This register is different from FIFOs that belong to other peripheral operation registers and its reset bit is bit 0.

Bit	Name	Default	Description
0 (R/W)	FIFO_RESET	1'b0	Set to 1 to stop the FIFO and reset the FIFO internal status, including its relevant interrupt status. Set to 0 in normal operations.
1 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is declared.
31:2	-	30'h0	Reserved.

Table 572: UART TX FIFO Operation Register (UART_TX_FIFO_OP)

- UART TX FIFO Status Register (UART_TX_FIFO_STATUS) – RISC: 0x114, DSP: 0x4A
  - For UART0

Bit	Name	Default	Description
6:0 (R)	FIFO_LEVEL	7'h0	The byte number of the valid data in the FIFO. In case FIFO is full, the value of this register should be set to 0, users must then concatenate the FIFO_FULL bit with this value to determine the actual data count in the FIFO.
7 (R)	FIFO_FULL	1'b0	FIFO full status The FIFO is full when it is read as 1. This bit is concatenated with FIFO_LEVEL as the actual FIFO data count.
8 (R)	FIFO_EMPTY	1'b1	FIFO empty status It is equivalent to (FIFO_FULL, FIFO_LEVEL) == 0.
31:9	-	23'h0	Reserved.

Table 573: UART TX FIFO Status Register (UART_TX_FIFO_STATUS) for UART0

  - For UART1

Bit	Name	Default	Description
4:0 (R)	FIFO_LEVEL	5'h0	The byte number of the valid data in the FIFO. In case FIFO is full, the value of this register should be set to 0, users must then concatenate the FIFO_FULL bit with this value to determine the actual data count in the FIFO.
5 (R)	FIFO_FULL	1'b0	FIFO full status. The FIFO is full when it's read out as 1. This bit is concatenated with FIFO_LEVEL as the actual FIFO data count.

Bit	Name	Default	Description
6 (R)	FIFO_EMPTY	1'b1	Indicates FIFO empty status. It is equivalent to (FIFO_FULL, FIFO_LEVEL) == 0.
31:7	-	25'h0	Reserved

Table 574: UART TX FIFO Status Register (UART_TX_FIFO_STATUS) for UART1

- UART TX FIFO Data Register (UART_TX_FIFO_DATA) – RISC: 0x118, DSP: 0x4C
  DSP can only access the low 16 bits of this register.

Bit	Name	Default	Description
31:0 (W)	FIFO_DATA	32'h0	The FIFO data register, which is the bottom of the TX_FIFO (only the low 8 bits will be used).

Table 575: UART TX FIFO Data Register (UART_TX_FIFO_DATA)

The following registers are UART RX_FIFO registers:

NOTE – The data flow of RX_FIFO is always from UART to RISC/DSP/DMA.

- UART RX DMA I/O MODE Register (UART_RX_DMA_IO_CTRL) – RISC: 0x120, DSP: 0x50
  Because UART1 has no DMA interfaces, so this register does not work for UART1.

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b0	1: For I/O mode. 0: For DMA mode.
1	-	1'b0	Reserved
2 (R/W)	DMA_FLUSH	1'b0	Flushes the DMA receive FIFO in case DATA_LEN is set on the peripheral side which does not match the DWORD size set in the DMA control.
3	-	1'b0	Reserved
5:4	RX_ENDIAN_MODE	2'b0	Only affects DMA operations: 00: No changes 01: Byte exchange in DWORD 10: Word exchange in DWORD 11: Byte exchange in WORD
31:6	-	26'h0	Reserved

Table 576: UART RX DMA I/O MODE Register (UART_RX_DMA_IO_CTRL)

- UART RX DMA I/O Length Register (UART_RX_DMA_IO_LEN) – RISC: 0x124, DSP: 0x52~0x53

Bit	Name	Default	Description
31:0 (R/W)	DATA_LEN	32'h0	The byte number of a DMA or I/O transfer. If it is set to 0, the I/O or DMA transfer will work continuously until it is stopped.

Table 577: UART RX DMA I/O Length Register (UART_RX_DMA_IO_LEN)

- UARTx RX FIFO Control Register (UART_RX_FIFO_CTRL) – RISC: 0x128, DSP: 0x54
  - For UART0

Bit	Name	Default	Description
6:0 (R/W)	FIFO_THD<6:0>	7'h0	A threshold value in byte that triggers an interrupt. An interrupt is triggered when the data count in the FIFO reaches the threshold.
31:7	-	25'h0	Reserved.

Table 578: UARTx RX FIFO Control Register (UART_RX_FIFO_CTRL) for UART0

  - For UART1

Bit	Name	Default	Description
4:0 (R/W)	FIFO_THD<3:0>	4'h0	A threshold value in byte that triggers an interrupt. An interrupt is triggered when the data count in the FIFO reaches the threshold.
31:5	-	27'h0	Reserved.

Table 579: UARTx RX FIFO Control Register (UART_RX_FIFO_CTRL) for UART1

- UARTx RX FIFO Level Check Register (UART_RX_FIFO_LEVEL_CHK) – RISC: 0x12C, DSP: 0x56~0x57
  - For UART0

Bit	Name	Default	Description
4:0 (R/W)	FIFO_SC<4:0>	5'h0	Stop check in DWORD
9:5	-	5'h0	Reserved
14:10 (R/W)	FIFO_LC<4:0>	5'h0	Low check in DWROD
19:15	-	5'h0	Reserved

Bit	Name	Default	Description
24:20 (R/W)	FIFO_HC<4:0>	5'h0	High check in DWORD
31:25	-	7'h0	Reserved

Table 580: UARTx RX FIFO Level Check Register (UART_RX_FIFO_LEVEL_CHK) for UART0

- UART RX FIFO Operation Register (UART_RX_FIFO_OP) – RISC: 0x130, DSP: 0x58

  This register is different from FIFOs that belong to other peripheral operation registers and its reset bit is bit 0.

Bit	Name	Default	Description
0 (R/W)	FIFO_RESET	1'b0	Set to 1 to stop the FIFO and reset the FIFO internal status, including its relevant interrupt status. Set it to 0 in normal operations.
1 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is declared.
31:2	-	30'h0	Reserved.

Table 581: UART RX FIFO Operation Register (UART_RX_FIFO_OP)

- UARTx RX FIFO Status Register (UART_RX_FIFO_STATUS) – RISC: 0x134, DSP: 0x5A
  - For UART0

Bit	Name	Default	Description
6:0 (R)	FIFO_LEVEL	7'h0	The byte number of the valid data in the FIFO. In case the FIFO is full, then the value of this register should be set to 0, users must then concatenate the FIFO_FULL bit with this value to determine the actual data count in the FIFO.
7 (R)	FIFO_FULL	1'b0	Indicates the FIFO full status; the FIFO is full when it's read out as 1. This bit is concatenated with FIFO_LEVEL as the actual FIFO data count.
8 (R)	FIFO_EMPTY	1'b1	Indicates the FIFO empty status, it is equivalent to (FIFO_FULL, FIFO_LEVEL) == 0.
31:9	-	23'h0	Reserved.

Table 582: UARTx RX FIFO Status Register (UART_RX_FIFO_STATUS) for UART0

  - For UART1

Bit	Name	Default	Description
4:0 (R)	FIFO_LEVEL	5'h0	The byte number of the valid data in the FIFO. In case the FIFO is full, then the value of this register should be

Bit	Name	Default	Description
			set to 0. Users must concatenate the FIFO_FULL bit with this value to determine the actual data count in the FIFO.
5 (R)	FIFO_FULL	1'b0	Indicates the FIFO full status. The FIFO is full when it is read as 1. This bit is concatenated with FIFO_LEVEL as the actual FIFO data count
6 (R)	FIFO_EMPTY	1'b1	The FIFO empty status. It's equivalent to (FIFO_FULL, FIFO_LEVEL) == 0.
31:7	-	25'h0	Reserved

Table 583: UARTx RX FIFO Status Register (UART_RX_FIFO_STATUS) for UART1

- UART RX FIFO Data Register (UART_RX_FIFO_DATA) – RISC: 0x138, DSP: 0x5C

Bit	Name	Default	Description
31:0 (R)	FIFO_DATA	32'h0	The FIFO data register, which is the bottom of RX_FIFO (Only the low 8 bit is used, the high 24bit is fixed at 0.)

Table 584: UART RX FIFO Data Register (UART_RX_FIFO_DATA)

- Auto Flow Control Register (AFC_CTRL) – RISC: 0x140

Bit	Name	Default	Description
7:0 (R/W)	AFC_RX_THD	8'h00	Threshold for rx_fifo to decide nRTS active or inactive.
8 (R/W)	AFC_RX_EN	1'b0	Enable bit for auto-flow-control when RX mode: 1'b1: enable 1'b0: disable (normal mode)
9 (R/W)	AFC_TX_EN	1'b0	Enable bit for auto-flow-control when TX mode: 1'b1: enable 1'b0: disable (normal mode)
10 (R/W)	CTS_CTRL	1'b1	Invert the polarity of input CTS. 1'b0: Not invert 1'b1: Invert
11 (R/W)	RTS_CTRL	1'b1	Invert the polarity of output RTS. 1'b0: Not invert 1'b1: Invert
12 (R)	CTS_IN_STATUS	1'b0	CTS status. Read only.
13 (R)	RTS_OUT_STATUS	1'b0	RTS status. Read only.
31:14	-	-	Reserved

Table 585: Auto Flow Control Register (for UART0)

- Switch between Loop-DMA and I/O Operation in RX Mode (SWH_DMA_IO) – RISC: 0x148

Bit	Name	Default	Description
0 (R/W)	TIMEOUT_IO_EN	1'b0	Active high. When get timeout interrupt, it will be changed to I/O mode automatically.
1 (R/W)	CLEAR_RX_ADDR_EN	1'b0	Active high. When found changing I/O to DMA mode, it will clear two low bits of read cint (rx_fifo_addr[1:0]).
2 (R/W)	EDGE_SC	1'b0	Active high. 1'b1: Make falling edge as start condition 1'b0: Low level as start condition.
31:3	-	-	Reserved

Table 586: Switch Register (for UART0)

SPI Controller

Overview

Serial Peripheral Interface (SPI) is one of the most popular low-speed, chip-to-chip interconnect interfaces originated from Motorola's M68xx micro-controller. The SPI in SiRFatlasV is fully compatible with the M68xx standard but yields more flexibility.

NOTE – If user needs more than one SPI, configure USP as SPI mode. For more details, refer to the section for USP.

Feature List

- Master and slave modes supported
- 8-/12-/16-/32-bit data unit
- 256 bytes receive data FIFO and 256 bytes transmit data FIFO
- Support for multi-unit frame
- Configurable SPI_EN (chip select pin) active state configurable
- Configurable SPI_CLK polarity configurable
- SPI_CLK phase configurable
- MSB/LSB first configurable

Pin Descriptions

External Pin Descriptions

Pin Name	I/O Type	Pin MUX	Default Function	Default Status	Description
X_SD_DAT_3[0] (SPI_EN)	Inout	GPIO	GPIO	Input with pull down	SPI, enable. In master mode, this pin is output. In slave mode, this pin is input.
X_ SD_DAT_3[1] (SPI_CLK)	Inout	GPIO	GPIO	Input with pull down	SPI, clock signal In master mode, this pin is output. In slave mode, this pin is input.
X_ SD_DAT_3[2] (SPI_RXD)	Inout	GPIO	GPIO	Input with pull down	SPI, data input
X_ SD_DAT_3[3] (SPI_TXD)	Inout	GPIO	GPIO	Output	SPI, data output

Table 587: SPI External Pin Description

Functional Descriptions

Function Modes

The SiRFatlasV SPI supports master and slave modes. In master mode, it supports single and multiple slave devices.

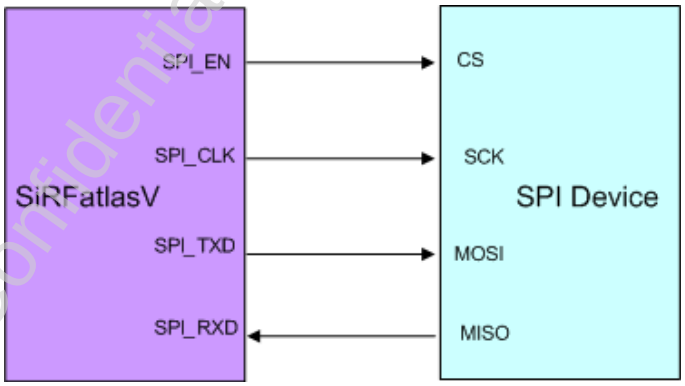


Figure 90: SPI Master Mode-Single Slave

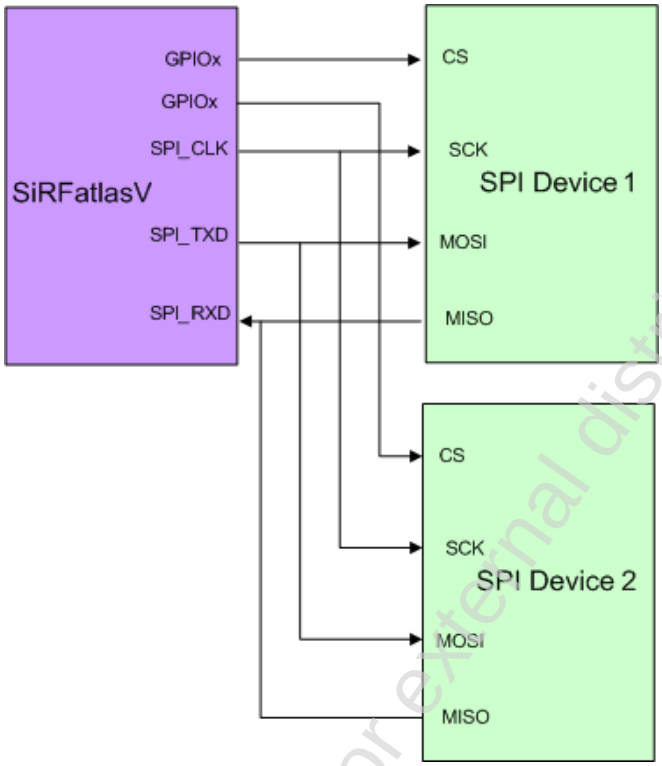


Figure 91: SPI Master Mode - Multi Slaves

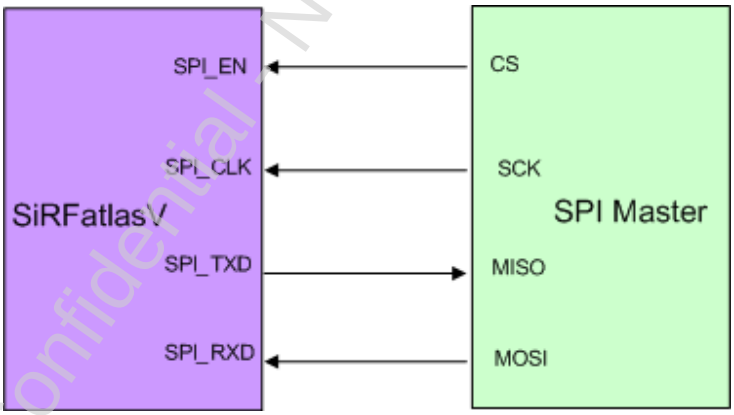


Figure 92: SPI Slave Mode

In master mode, both SPI_CLK and SPI_EN are output signals from the SPI controller. Ensure that the output SPI clock frequency dicided by the CLK_DIV of SPI_CTRL REGISTER is less than 37.5MHz.

The SPI_EN is the chip selected signal and can be set to valid of high or valid of low. As chip selected signal, SPI_EN should be held valid for some time before and after the data is transferred. The hold time can be set to one or two clock cycles of SPI_CLK according to requirements. Furthermore, SPI_EN can always be set to logic high or logic low through I/O mode when necessary.

In slave mode, both SPI_CLK and SPI_EN are inputs and driven by external devices. Ensure that the input SPI clock period is less than ($6*T_{ioclk} + 2*T_{setup}$), where T_{ioclk} is the clock period of IOCLK and T_{setup} is the external device data setup requirement.

SPI_TXD is the output data pin and the SPI_RXD is the input data pin. In both master and slave mode, these two data pins can be set to be driven at either rising edge or falling edge of SPI_CLK.

Frame Format

The SiRFatlasV SPI supports four different data units: 8-bit, 12-bit, 16-bit and 32-bit data unit. Each frame may contain:

- None or one command unit, the command unit can be 1 byte, 2 byte, 3 byte or 4 byte in length
- Either none or several data units

In the SPI bus, both single data unit and multi data unit can be transferred in one SPI frame, but only up to 64K.

SPI_TXD can transmit a command data unit before transmitting or receiving a data unit. The command data length can be set to 1, 2, 3 or 4 bytes.

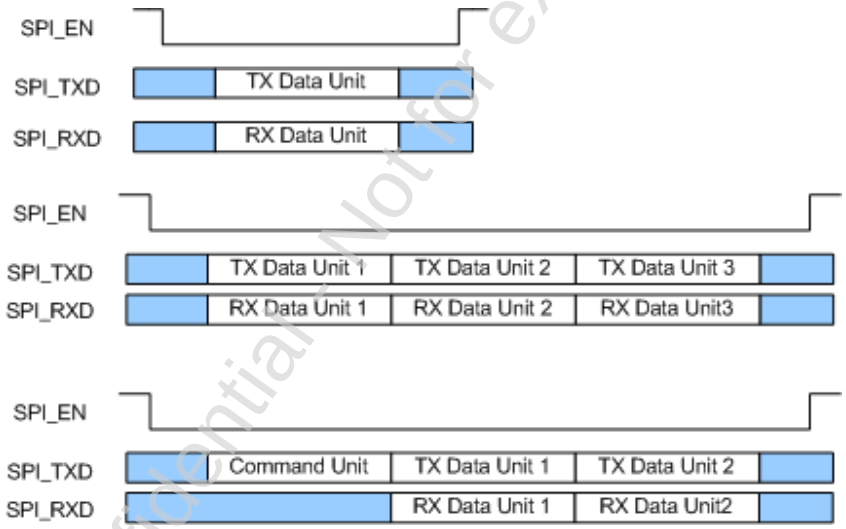


Figure 93: SPI Frame Format

Timing Diagram

The following diagram is the Motorola SPI standard. SiRFatlasV SPI supports four kinds of timing (as shown in the figures below).

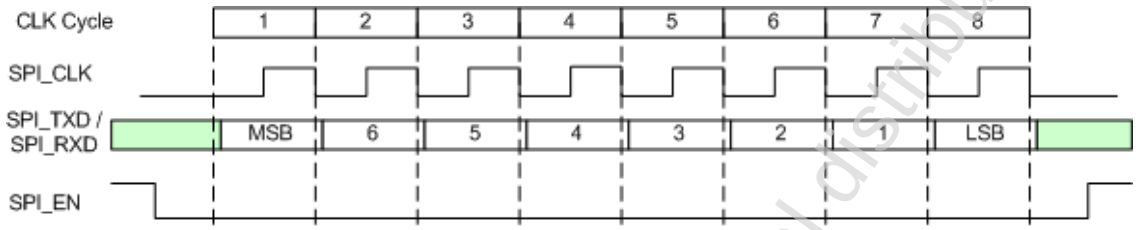


Figure 94: SPI Interface Timing DRV=0, CLK=0

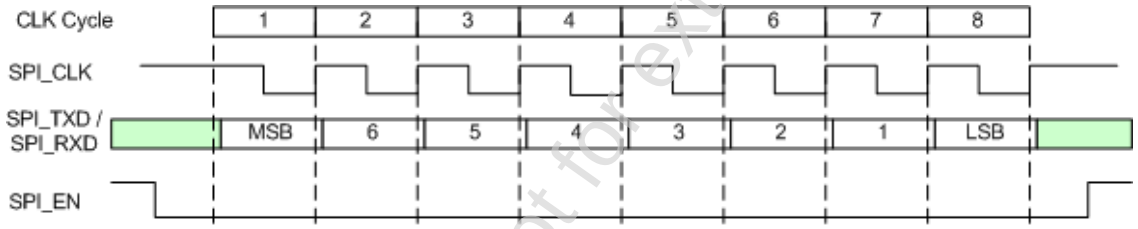


Figure 95: SPI Interface Timing DRV=1, CLK=1

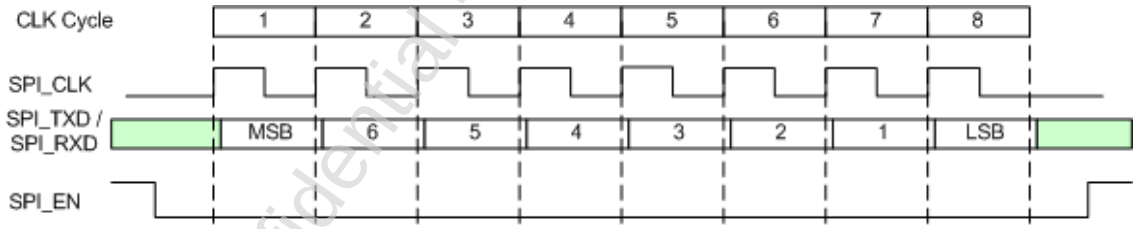


Figure 96: SPI Interface Timing DRV=1, CLK=0

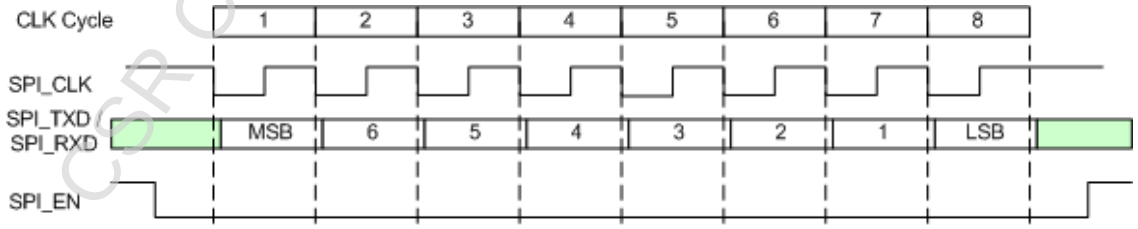


Figure 97: SPI Interface Timing DRV=0, CLK=1

Register Definition

Register Address Mapping

SPI base address: 0x80030000

RISC Address <11:0>	Register	Description
0x0000	SPI_CTRL	SPI controller configuration register
0x0004	SPI_CMD	SPI command register
0x000c	SPI_TX_RX_EN	SPI interface transfer enable register
0x0010	SPI _INT_EN	SPI interrupt enable register
0x0014	SPI _INT_STATUS	SPI interrupt register
0x0018~0xfc	-	Reserved
0x100	SPI_TX_DMA_IO_CTRL	SPI TXFIFO DMA/IO register
0x104	SPI_TX_DMA_IO_LEN	SPI transmit data length register
0x108	SPI_TXFIFO_CTRL	SPI TXFIFO control register
0x10C	SPI_TXFIFO_LEVEL_CHK	SPI TXFIFO check level register
0x110	SPI_TXFIFO_OP	SPI TXFIFO operation register
0x114	SPI_TXFIFO_STATUS	SPI TXFIFO status register
0x118	SPI_TXFIFO_DATA	SPI TXFIFO bottom
0x11C	-	Reserved
0x120	SPI_RX_DMA_IO_CTRL	SPI RXFIFO DMA/IO register
0x124	SPI_RX_DMA_IO_LEN	SPI receive length register
0x128	SPI_RXFIFO_CTRL	SPI RXFIFO control register
0x12C	SPI_RXFIFO_LEVEL_CHK	SPI RXFIFO check level register
0x130	SPI_RXFIFO_OP	SPI RXFIFO operation register
0x134	SPI_RXFIFO_STATUS	SPI RXFIFO status register
0x138	SPI_RXFIFO_DATA	SPI RXFIFO bottom
0x140	SLV_RX_SAMPLE_MODE	RX sample mode when in slave mode
0x144	DUMMY_DELAY_CTRL	Control register when inserting dummy delay

Table 588: SPI Register Mapping

Register Descriptions

- SPI Control Register (SPI_CTRL) – 0x0
  This register is used to configure the SPI work mode.

Bit	Name	Default	Description
15:0 (R/W)	CLK_DIV	16'b0	The SPI clock divides number from the I/O CLOCK. The fspi_clk = fio_clk/(2*(CLK_DIV+1))
16 (R/W)	SLV_MODE	1'b0	The SPI controller work mode: 0: Master mode 1: Slave mode
17 (R/W)	CMD_MODE	1'b1	The SPI transfer command mode: 0: Without command data in the SPI frame 1: With command data in the SPI frame
18 (R/W)	CS_IO_OUT	1'b0	If the CS pin is set to I/O mode, the output of the SPI_CS pin is controlled by this bit.
19 (R/W)	CS_IO_MODE	1'b0	The CS pin output mode. In hardware control mode, CS output is controlled by the CS hardware logic, otherwise, CS will output the value of the CS_IO_OUT bit. 0: Hardware control mode 1: RISC control I/O mode
20 (R/W)	CLK_IDLE_STAT	1b0	The SPI CLOCK idle state: 0: Clock stays at logic 0 in idle state 1: Clock stays at logic 1 in idle state
21 (R/W)	CS_IDLE_STAT	1b0	The SPI CS idle state: 0: CS stays at logic 0 in idle state 1: CS stays at logic 1 in idle state
22 (R/W)	TRAN_MSB	1b0	The SPI data transfer mode: 0: LSB 1: MSB
23 (R/W)	DRV_POS_EDGE	1b0	The SPI data is driven out: 0: Clock falling edge 1: Clock rising edge
24 (R/W)	CS_HOLD_TIME	1b0	The clock number for CS to hold valid before and after the data is transferred when CS is in hardware control mode: 0: 1 SPI_CLK cycle 1: 2 SPI_CLK cycles
25 (R/W)	CLK_SAMPLE_MODE	1b0	The slave clock sample mode

Bit	Name	Default	Description
			0: 2 sample 1: 3 sample to filter the glitch
27:26 (R/W)	TRAN_DAT_FORMAT	2'b0	The transfer data unit format: 2'b00: 8 bits 2'b01: 12 bits 2'b10: 16 bits 2'b11: 32 bits If the data format is 12-, or 16-bit, then FIFO_WIDTH in SPI_TX_FIFO_CTRL/ SPI_RX_FIFO_CTRL should be set to word width.
29:28 (R/W)	CMD_BYTE_NUM	2'b0	If the command data exists, then these two bits will decide the command byte number: 2'b00: 1 byte 2'b01: 2 bytes 2'b10: 3 bytes 2'b11: 4 bytes
30 (R/W)	ENA_AUTO_CLR	1'b0	This bit will decide whether or not to automatically clear SPI_TX_RX_EN after an SPI frame ends: 0: SPI_TX_RX_EN will still keep its value 1: SPI_TX_RX_EN will be automatically cleared by the hardware
31 (R/W)	MUL_DAT_MODE	1'b0	This bit will decide whether it is only a single data unit (the command data is not considered as a data unit). The data unit number is set in SPI_TX_DMA_IO_LEN / SPI_RX_DMA_IO_LEN: 0: Single data unit 1: Multiple data unit

Table 589: SPI Control Register (SPI_CTRL)

- SPI Command Register (SPI_CMD) – 0x4
  This register is write-only.

Bit	Name	Default	Description
31:0 (W)	SPI command register	32'b0	The SPI command register

Table 590: SPI Command Register (SPI_CMD)

- SPI Transmit/Receive Enable Register (SPI_TX_RX_EN) –0x8

  Before the SPI_TX_EN or SPI_RX_EN bit is enabled, SPI_CTRL must be configured first and write SPI_CMD if necessary.

Bit	Name	Default	Description
0 (R/W)	SPI_RX_EN	1'b0	Receive enable bit: 0: Disabled 1: Enabled
1 (R/W)	SPI_TX_EN	1'b0	Transmit enable bit: 0: Disabled 1: Enabled
2 (R/W)	SPI_CMD_TX_EN	1'b0	If only the command is transmitted, software should enable this bit, SPI_RX_EN, and SPI_TX_EN should be disabled. 0: Disabled 1: Enabled
31:3	-	29'h0	Reserved

Table 591: SPI Transmit/Receive Enable Register (SPI_TX_RX_EN)

- SPI Interrupt Enable Register (SPI_INT_EN) – RISC: 0xC

Bit	Name	Default	Description
0 (R/W)	RX_DONE_INT_EN	1'b0	Receive done interrupt enable: 0: Disabled 1: Enabled
1 (R/W)	TX_DONE_INT_EN	1'b0	Transmit done interrupt enable: 0: Disabled 1: Enabled
2 (R/W)	RX_OFLOW_INT_EN	1'b0	Receive overflow interrupt enable: 0: Disabled 1: Enabled
3 (R/W)	TX_UFLOW_INT_EN	1'b0	Transmit underflow interrupt enable: 0: Disabled 1: Enabled
4 (R/W)	RX_IO_DMA_ INT_EN	1'b0	IO/DMA receive interrupt enable: 0: Disabled 1: Enabled
5 (R/W)	TX_ IO_DMA_ INT_EN	1'b0	IO/DMA transmit interrupt enable:

Bit	Name	Default	Description
			0: Disabled 1: Enabled
6 (R/W)	RXFIFO_FULL_INT_EN	1'b0	Receive FIFO full interrupt enable: 0: Disabled 1: Enabled
7 (R/W)	TXFIFO_EMPTY_INT_EN	1'b0	Transmit FIFO empty interrupt enable: 0: Disabled 1: Enabled
8 (R/W)	RXFIFO_THD_INT_EN	1'b0	Receive FIFO threshold interrupt enable: 0: Disabled 1: Enabled
9 (R/W)	TXFIFO_THD_INT_EN	1'b0	Transmit FIFO threshold interrupt enable: 0: Disabled 1: Enabled
10 (R/W)	FRM_END_INT_EN	1'b0	The transfer frame end interrupt enable: 0: Disabled 1: Enabled
31:11	-	21'h0	Reserved

Table 592: SPI Interrupt Enable Register (SPI_INT_EN)

- SPI Interrupt Status Register (SPI_INT_STATUS) – RISC: 0x10
  RISC writes a 1'b1 to the bit will clear that interrupt.

Bit	Name	Default	Description
0 (R/W)	RX_DONE	1'b0	A valid data unit has been received in the RXFIFO interrupt: 0: Invalid 1: Valid
1 (R/W)	TX_DONE	1'b0	A valid data unit has been transmitted from the TXFIFO interrupt: 0: Invalid 1: Valid
2 (R/W)	RX_OFLOW	1'b0	RXFIFO overflow Interrupt: 0: Invalid 1: Valid
3 (R/W)	TX_UFLOW	1'b0	TXFIFO underflow interrupt: 0: Invalid 1: Valid
4 (R/W)	DMA_IO_RX_DONE	1'b0	RXFIFO has received all data from the data package, the data size is defined by SPI_RX_DMA_IO_LEN: 0: Invalid 1: Valid
5 (R/W)	DMA_IO_TX_DONE	1'b0	TXFIFO has transmitted all data from the data package, the data size is defined by SPI_TX_DMA_IO_LEN: 0: Invalid 1: Valid
6 (R/W)	RXFIFO_FULL	1'b0	RXFIFO full interrupt: 0: Invalid 1: Valid
7 (R/W)	TXFIFO_EMPTY	1'b0	TXFIFO empty interrupt: 0: Invalid 1: Valid
8 (R/W)	RXFIFO_THD_REACH	1'b0	Time to read SPI_ RXFIFO when the number of data in SPI_RXFIFO reaches the threshold: 0: Invalid 1: Valid
9 (R/W)	TXFIFO_THD_REACH	1'b0	Time to write SPI_TXFIFO when the number of data in SPI_TXFIFO reaches the threshold: 0: Invalid 1: Valid

Bit	Name	Default	Description
10 (R/W)	FRM_END	1'b0	If a transfer frame is completed, then SPI_CS will change from valid to invalid, this bit will also be changed to valid. 0: Invalid 1: Valid
31:11	-	21'h0	Reserved

Table 593: SPI Interrupt Status Register (SPI_INT_STATUS)

SPI TX_FIFO Registers

NOTE – The data flow of TX_FIFO is always from RISC/DSP/DMA to SPI.

- SPI TX_FIFO DMA I/O MODE Register (SPI_TX_DMA_IO_CTRL) – RISC: 0x100

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b1	1 for I/O mode, 0 for DMA mode
31:1	-	31'h0	Reserved

Table 594: SPI TX_FIFO DMA I/O MODE Register (SPI_TX_DMA_IO_CTRL)

- SPI TX_FIFO DMA I/O Length Register (SPI_TX_DMA_IO_LEN) – RISC: 0x104

Bit	Name	Default	Description
15:0 (R/W)	DATA_LEN	16'h0	This number is the block data length to be transmitted in DMA or I/O transfer. If it is set to zero, then the I/O or DMA transfer will work continuously until it is stopped. In multi data unit frames, this number represents the data unit number in the frame, this means that in the multi-data unit frame, the DMA/IO data block will be transmitted in one SPI frame

Table 595: SPI TX_FIFO DMA I/O Length Register (SPI_TX_DMA_IO_LEN)

- SPI TX_FIFO Control Register (SPI_TX_FIFO_CTRL) – RISC: 0x108

Bit	Name	Default	Description
1:0 (R/W)	FIFO_WIDTH<1:0>	2'h0	Data width of FIFO: 0 for byte, 1 for word and 2 for DWORD. If the data unit is 12 bit, then FIFO_WIDTH should be set to word mode.
9:2 (R/W)	FIFO_THD<7:0>	8'h0	A threshold in byte that triggers an interrupt. An interrupt is triggered when the data count in the FIFO reaches the threshold.
31:10	-	22'h0	Reserved

Table 596: SPI TX_FIFO Control Register (SPI_TX_FIFO_CTRL)

- SPI TX_FIFO Level Check Register (SPI_TX_FIFO_LEVEL_CHK) – RISC: 0x10C

Bit	Name	Default	Description
5:0 (R/W)	FIFO_SC<5:0>	6'h0	Stop check in DWORD
9:6	-	4'h0	Reserved
15:10 (R/W)	FIFO_LC<5:0>	6'h0	Low check in DWROD
19:16	-	4'h0	Reserved
25:20 (R/W)	FIFO_HC<5:0>	6'h1	High check in DWORD
31:26	-	6'h0	Reserved

Table 597: SPI TX_FIFO Level Check Register (SPI_TX_FIFO_LEVEL_CHK)

- SPI TX_FIFO Operation Register (SPI_TX_FIFO_OP) – RISC: 0x110
  This register is different from FIFO of other peripheral operation registers and the reset bit is bit0.

  If SPI is set to multi-data unit mode, FIFO should be reset after each frame is finished.

Bit	Name	Default	Description
0 (R/W)	FIFO_RESET	1'b0	Set to 1 to stop the FIFO and reset the FIFO internal status, including its relevant interrupt status. Set to 0 in normal operation.
1 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is declared.
31:2	-	30'h0	Reserved

Table 598: SPI TX_FIFO Operation Register (SPI_TX_FIFO_OP)

- SPI TX_FIFO Status Register (SPI_TX_FIFO_STATUS) – RISC: 0x114

Bit	Name	Default	Description
7:0 (R)	FIFO_LEVEL	8'h0	The byte number of the valid data in the FIFO. In case the FIFO is full, the value of this register will become 0, thus users must concatenate the FIFO_FULL bit with this value to determine the actual data count in the FIFO.
8 (R)	FIFO_FULL	1'b0	This is the FIFO full status. The FIFO is full when it is read out as 1. This bit is concatenated with FIFO_LEVEL as the actual FIFO data count.
9 (R)	FIFO_EMPTY	1'b0	FIFO empty status It is equivalent to (FIFO_FULL, FIFO_LEVEL) == 0
31:10	-	22'h0	Reserved.

Table 599: SPI TX_FIFO Status Register (SPI_TX_FIFO_STATUS)

- SPI TX_FIFO Data Register (SPI_TX_FIFO_DATA) – RISC: 0x118

Bit	Name	Default	Description
31:0 (W)	FIFO_DATA	32'h0	The FIFO data register, which is at the bottom of the TX_FIFO.

Table 600: SPI TX_FIFO Data Register (SPI_TX_FIFO_DATA)

SPI RX_FIFO Registers

NOTE – The data flow of RX_FIFO is always from SPI to RISC/DSP/DMA.

- SPI RX_FIFO DMA I/O MODE Register (SPI_RX_DMA_IO_CTRL) – RISC: 0x120

Bit	Name	Default	Description
0 (R/W)	IO_DMA_SEL	1'b1	1: I/O mode 0: DMA mode
1	-	-	Reserved
2 (R/W)	DMA_FLUSH	1'b0	Flushes the DMA receive FIFO in case the DATA_LEN set at the peripheral side does not match the DWord size set in the DMA control register.
31:3	-	29'h0	Reserved

Table 601: SPI RX_FIFO DMA I/O MODE Register (SPI_RX_DMA_IO_CTRL)

- SPI RX_FIFO DMA I/O Length Register (SPI_RX_DMA_IO_LEN) – RISC: 0x124

Bit	Name	Default	Description
15:0 (R/W)	DATA_LEN	16'h0	This number is the block data length to be received in DMA or I/O transfer. If it is set to 0, the I/O or DMA transfer will work continuously until it stops. In a multi data unit frame, this number represents the data unit number in the frame, which means in the multi-data unit frame, the DMA/IO data block will be received in one SPI frame.

Table 602: SPI RX_FIFO DMA I/O Length Register (SPI_RX_DMA_IO_LEN)

- SPI RX_FIFO Control Register (SPI_RX_FIFO_CTRL) – RISC: 0x128

Bit	Name	Default	Description
1:0 (R/W)	FIFO_WIDTH<1:0>	2'h0	Data width of FIFO: 0 for byte, 1 for word and 2 for dword. If the data unit is 12 bit, then FIFO_WIDTH should be set to word mode.
9:2 (R/W)	FIFO_THD<7:0>	8'h0	A threshold in byte that triggers an interrupt. An interrupt is triggered when the data count in the FIFO reaches the threshold.
31:10	-	22'h0	Reserved.

Table 603: SPI RX_FIFO Control Register (SPI_RX_FIFO_CTRL)

- SPI RX_FIFO Level Check Register (SPI_RX_FIFO_LEVEL_CHK) – RISC: 0x12C

Bit	Name	Default	Description
5:0 (R/W)	FIFO_SC<5:0>	6'h1	Stop check in dword
9:6	-	4'h0	Reserved
15:10 (R/W)	FIFO_LC<5:0>	6'h2	Low check in dwrod
19:16	-	4'h0	Reserved
25:20 (R/W)	FIFO_HC<5:0>	6'h3	High check in dword
31:26	-	6'h0	Reserved

Table 604: SPI RX_FIFO Level Check Register (SPI_RX_FIFO_LEVEL_CHK)

- SPI RX_FIFO Operation Register (SPI_RX_FIFO_OP) – RISC: 0x130
  This register is different from FIFO of other peripheral operation registers and the reset bit is bit0.

  If the SPI is set to multi-data unit mode, reset and restart the FIFO after each frame is finished.

Bit	Name	Default	Description
0 (R/W)	FIFO_RESET	1'b0	Set it to 1 to stop the FIFO and reset the FIFO internal status as well as its interrupt status. Set it to 0 in normal operation.
1 (R/W)	FIFO_START	1'b0	Starts the read/write transfer when this bit is declared.
31:2	-	30'h0	Reserved

Table 605: SPI RX_FIFO Operation Register (SPI_RX_FIFO_OP)

- SPI RX_FIFO Status Register (SPI_RX_FIFO_STATUS) – RISC: 0x134

Bit	Name	Default	Description
7:0 (R)	FIFO_LEVEL	8'h0	The byte number of the valid data in the FIFO. If the FIFO is full, the value of this register will be 0, thus users must concatenate the FIFO_FULL bit with this value to get the actual data count in the FIFO.
8 (R)	FIFO_FULL	1'b0	The FIFO full status. The FIFO is full when it is read as 1. It is concatenated with FIFO_LEVEL as the actual FIFO data count.
9 (R)	FIFO_EMPTY	1'b0	FIFO empty status, which is equivalent to (FIFO_FULL, FIFO_LEVEL) == 0
31:10	-	22'h0	Reserved.

Table 606: SPI RX_FIFO Status Register (SPI_RX_FIFO_STATUS)

- SPI RX_FIFO Data Register (SPI_RX_FIFO_DATA) – RISC: 0x138

Bit	Name	Default	Description
31:0 (R)	FIFO_DATA	32'h0	The FIFO data register, at the bottom of RX_FIFO.

Table 607: SPI RX_FIFO Data Register (SPI_RX_FIFO_DATA)

- SPI SLV_RX_SAMPLE_MODE Control Register – RISC: 0x140

Bit	Name	Default	Description
0 (R/W)	slv2xsample	1'b0	Active high. Two times over sample mode when it is active.

Bit	Name	Default	Description
31:1	-	31'h0	Reserved.

Table 608: SPI SLV_RX_SAMPLE_MODE Control Register

- SPI DUMMY_DELAY_CTRL Control Register – RISC: 0x144

Bit	Name	Default	Description
9:0 (R/W)	delay_num	9'd0	Delay number between a command and data unit.
10	delay_mode	1'b0	1'b0: SPI_CLK idle during dummy delay. 1'b1: SPI_CLK toggle during dummy delay.
11	delay_en	1'b0	1'b0: Not insert a dummy delay. 1'b1: Insert a dummy delay.

Table 609: SPI DUMMY_DELAY_CTRL Control Register

PWM

Overview

The PWM (Pulse Wide Modulate) generator can generate 7 independent outputs. Each output duty cycle can be adjusted by setting the corresponding wait and hold registers.

Feature List

- Seven independent outputs
- Duty cycle can be adjusted by software

Pin Descriptions

External Pin Descriptions

Pin Name	I/O Type	Pin Mux	Default Function	Default Status	Description
X_GPIO[4]	Output	GPIO Group0[4]	GPIO	Input pull up	PWM0 output
X_GPIO[5]	Output	GPIO Group0[5]	GPIO	Input pull up	PWM1 output
X_GPIO[6]	Output	GPIO Group0[6]	GPIO	Input pull down	PWM2 output
X_GPIO[7]	Output	GPIO Group0[7]	GPIO	Input pull down	PWM3 output
X_CKO_0	Output	xin/xinw clock output	PWM	output	Clock 0 output
X_CKO_1	Output	GPIO Group1[20] I^2S	GPIO	Input pull down	Clock 1 output

Table 610: PWM External Pin Descriptions

Because the default function of the pads is GPIO, you need to disable the corresponding GPIOx_PAD_EN if these pads are going to work in the PWM function mode.

Register Definitions

Register Address Mapping

Base Address

Access Type	Address Mapping
PWM controller internal register base address	0x800F0000

Table 611: PWM Controller Register Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x00	PWM_PRECLKSRC	PWM pre-clock source select
0x04	PWM_OE	PWM output enable
0x08	PWM_PRECLKEN	PWM pre-clock enable
0x0c	PWM_POSTCLKEN	PWM post-clock enable
0x10	PWM_WAIT0	PWM output 0 wait state for high pulse
0x14	PWM_HOLD0	PWM output 0 hold state for low pulse
0x18	PWM_WAIT1	PWM output 1 wait state for high pulse
0x1c	PWM_HOLD1	PWM output 1 hold state for low pulse
0x20	PWM_WAIT2	PWM output 2 wait state for high pulse
0x24	PWM_HOLD2	PWM output 2 hold state for low pulse
0x28	PWM_WAIT3	PWM output 3 wait state for high pulse
0x2c	PWM_HOLD3	PWM output 3 hold state for low pulse
0x30	PWM_WAIT4	PWM output 4 wait state for high pulse
0x34	PWM_HOLD4	PWM output 4 hold state for low pulse
0x38	PWM_WAIT5	PWM output 5 wait state for high pulse
0x3c	PWM_HOLD5	PWM output 5 hold state for low pulse
0x40	PWM_WAIT6	PWM output 6 wait state for high pulse
0x44	PWM_HOLD6	PWM output 6 hold state for low pulse

Table 612: PWM Interface Register Mapping

PWM Group	Target Function
PWM 0	PWM0
PWM 1	PWM1
PWM 2	PWM2
PWM 3	PWM3
PWM 4	CKO_0
PWM 5	CKO_1
PWM 6	I2S_CLK

Table 613: PWM Channel Assign Table

Register Descriptions

- PWM Pre-Clock Source Register (PWM_PRECLKSRC) – 0x00

Bit	Name	Default	Description
1:0 (R/W)	PWM0 Clock Source	2'b0	PWM0 Pre-clock source select: 2'b00: XIN clock 2'b01: PLL1 clock 2'b10: PLL2 clock 2'b11: 32KHz clock
3:2 (R/W)	PWM1 Clock Source	2'b0	PWM1 Pre-clock source select: 2'b00: XIN clock 2'b01: PLL1 clock 2'b10: PLL2 clock 2'b11: 32KHz clock
5:4 (R/W)	PWM2 Clock Source	2'b0	PWM2 Pre-clock source select: 2'b00: XIN clock 2'b01: PLL1 clock 2'b10: PLL2 clock 2'b11: 32KHz clock
7:6 (R/W)	PWM3 Clock Source	2'b0	PWM3 Pre-clock source select: 2'b00: XIN clock 2'b01: PLL1 clock 2'b10: PLL2 clock 2'b11: 32KHz clock
9:8 (R/W)	PWM4 Clock Source	2'b0	PWM4 Pre-clock source select: 2'b00: XIN clock 2'b01: PLL1 clock

Bit	Name	Default	Description
			2'b10: PLL2 clock 2'b11: 32KHz clock
11:10(R/W)	PWM5 Clock Source	2'b0	PWM5 Pre-clock source select: 2'b00: XIN clock 2'b01: PLL1 clock 2'b10: PLL2 clock 2'b11: 32KHz clock
13:12(R/W)	PWM6 Clock Source	2'b0	PWM6 Pre-clock source select: 2'b00: XIN clock 2'b01: PLL1 clock 2'b10: PLL2 clock 2'b11: 32KHz clock
14	PWM0 bypass mode	1'b0	PWM0 bypass selection: 0: PWM0 output clock from divider. 1: PWM0 output clock from source clock selected by PWM_PRECLKSRC[1:0].
15	PWM1 bypass mode	1'b0	PWM1 bypass selection: 0: PWM1 output clock from divider. 1: PWM1 output clock from source clock selected by PWM_PRECLKSRC[3:2].
16	PWM2 bypass mode	1'b0	PWM2 bypass selection: 0: PWM2 output clock from divider. 1: PWM2 output clock from source clock selected by PWM_PRECLKSRC[5:4].
17	PWM3 bypass mode	1'b0	PWM3 bypass selection: 0: PWM3 output clock from divider. 1: PWM3 output clock from source clock selected by PWM_PRECLKSRC[7:6].
18	PWM4 bypass mode	1'b0	PWM4 bypass selection: 0: PWM4 output clock from divider. 1: PWM4 output clock from source clock selected by PWM_PRECLKSRC[9:8].
19	PWM5 bypass mode	1'b0	PWM5 bypass selection: 0: PWM5 output clock from divider. 1: PWM5 output clock from source clock selected by PWM_PRECLKSRC[11:10].
20	PWM6 bypass mode	1'b0	PWM6 bypass selection: 0: PWM6 output clock from divider. 1: PWM6 output clock from source clock selected by

Bit	Name	Default	Description
			PWM_PRECLKSRC[13:12].
31:21	-	11'h0	reserved

Table 614: PWM Pre-Clock Source Register

- PWM Output Enable Register (PWM_OE) – 0x04

Bit	Name	Default	Description
0 (R/W)	PWM0 Work Enable	1'b0	Set 1 enable Set 0 disable
1 (R/W)	PWM1 Work Enable	1'b0	Same as the above
2 (R/W)	PWM2 Work Enable	1'b0	Same as the above
3 (R/W)	PWM3 Work Enable	1'b0	Same as the above
4 (R/W)	PWM4 Work Enable	1'b0	Same as the above
5 (R/W)	PWM5 Work Enable	1'b0	Same as the above
6 (R/W)	PWM6 Work Enable	1'b0	Same as the above
31:7	-	28'h0	Reserved

Table 615: PWM Output Enable Register

For bit 4 and bit 5, the control signal for the channel is not used when the clock source is 32 kHz or 24 MHz.

- PWM Pre-Clock Enable Register (PWM_PRECLKEN) – 0x08

Bit	Name	Default	Description
0 (R/W)	PWM0 Pre-Clock Enable	1'b0	Set 1 enable PWM0 Pre-Clock Set 0 disenable PWM0 Pre-Clock
1 (R/W)	PWM1 Pre-Clock Enable	1'b0	Same as the above
2 (R/W)	PWM2 Pre-Clock Enable	1'b0	Same as the above
3 (R/W)	PWM3 Pre-Clock Enable	1'b0	Same as the above
4 (R/W)	PWM4 Pre-Clock Enable	1'b0	Same as the above
5 (R/W)	PWM5 Pre-Clock Enable	1'b0	Same as the above
6 (R/W)	PWM6 Pre-Clock Enable	1'b0	Same as the above
31:7	-	28'h0	Reserved

Table 616: PWM Pre-Clock Enable Register

- PWM Post-Clock Enable Register (PWM_PRECLKEN) – 0x0c

Bit	Name	Default	Description
0 (R/W)	PWM0 Post-Clock Enable	1'b0	Set 1 enable PWM0 Post-Clock Set 0 disable PWM0 Post-Clock
1 (R/W)	PWM1 Post-Clock Enable	1'b0	Same as the above
2 (R/W)	PWM2 Post-Clock Enable	1'b0	Same as the above
3 (R/W)	PWM3 Post-Clock Enable	1'b0	Same as the above
4 (R/W)	PWM4 Post-Clock Enable	1'b0	Same as the above
5 (R/W)	PWM5 Post-Clock Enable	1'b0	Same as the above
6 (R/W)	PWM6 Post-Clock Enable	1'b0	Same as the above
31:7	-	28'h0	Reserved

Table 617: PWM Post-Clock Enable Register

- PWM Output Wait State Registers 0~6 (PWM_WAIT<9:0>)

Bit	Name	Default	Description
9:0 (R/W)	HI_WAIT_CNT	10'h0	Number of pre-clock cycles that the PWM pin is high = HI_WAIT_CNT + 1
31:10	-	22'h0	Reserved

Table 618: PWM Output Wait State Registers 0~6

- PWM Hold State Register0~6 (PWM_HOLD<9:0>)

Bit	Name	Default	Description
9:0 (R/W)	LO_HOLD_CNT	10'h0	Number of pre-clock cycles that the PWM pin is low = LO_HOLD_CNT + 1
31:10	-	22'h0	Reserved

Table 619: PWM Hold State Registers 0~6

I²C

Overview

The I²C bus is a two-wire, bi-directional serial bus that provides a simple and efficient method for data exchange between devices. It is most suitable for applications that require occasional communications over a short distance among several devices.

Feature List

- Two I²C controller modules are on chip (they only share one INT, one clk, and have a separate reset)
- RISC I/O bus read write register
- Up to 16 bytes data buffer for issuing commands and writing data at the same time
- Up to 16 commands, and receiving read data 16 bytes at a time
- Error INT report (ACK check)
- No-ACK bus protocols (SCCB bus protocols)

Pin Description

The following table shows the I²C input/output pins:

Pin Name	I/O Type	Pin MUX	Default Function	Default Status	Description
X_SCL_0	Open-drain	GPIO3[19]	GPIO	Input no pull (GPIO mode)	I²C clock bus 0
X_SDA_0	Open-drain	GPIO3[18]	GPIO	Input no pull (GPIO mode)	I²C data bus 0
X_GPIO[2]	Open-drain	X_SCL_1	GPIO	Input no pull (GPIO mode)	I²C clock bus 1
X_GPIO[3]	Open-drain	X_SDA_1	GPIO	Input no pull (GPIO mode)	I²C data bus 1

Table 620: I²C External Pin Descriptions

Because the default function of the pads is GPIO, you need to disable the corresponding GPIOx_PAD_EN before configing I²C.

I²C Bus Descriptions

This interface supports both compliant master and slave operations, It provides the following features:

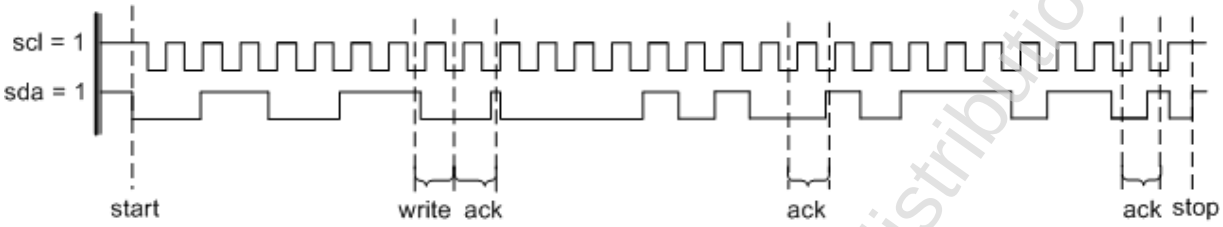


Figure 98: Timing Diagram of I²C Register Write

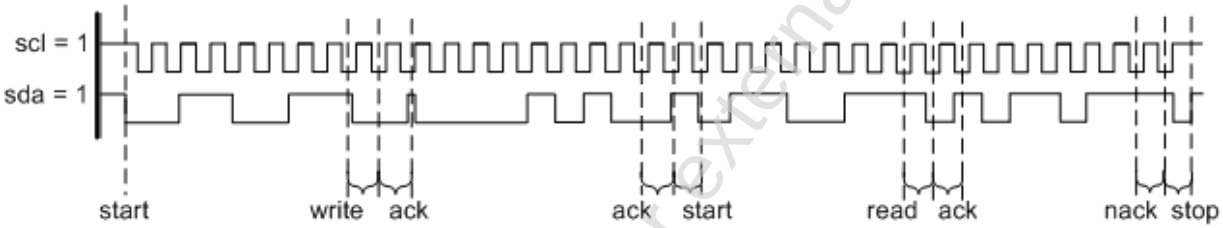


Figure 99: Timing Diagram of I²C Register Read

The I²C bus provides 2 electrical signals, SCL (clock line) and SDA (data line), and defines several transmission codes, including:

- Bus idle state

  The bus is idle when both data and clock lines are high. The control of the bus is initiated with a start bit, and the bus will be released with a stop bit. Only the master can generate the start and stop bits.

- Start bit/stop bit

  The start bit is defined as a high to low transition of the data line while the clock line is high, whereas the stop bit is defined as a low to high transition while the clock line is high.

- Read/write bit

  0 indicates write mode, 1 indicates read mode.

- Data bit transfer

  One data bit is transferred during each clock pulse. The I²C clock pulse is provided by the master. Data must be stable during the high period of the I²C clock, which changes only when the I²C clock is low. Data is transferred in 8 bits at a time followed by an acknowledge bit.

- Acknowledge (ack) / no-acknowledge (nack) bit

  The master generates the acknowledgement clock pulse, while the transmitter (which is the master when writing, or slave when reading) releases the data line. The receiver will then indicate the acknowledgement bit by pulling the data line low during the acknowledgement clock pulse. The no-acknowledgement bit is generated when the data line is not pulled down by the receiver during the acknowledgement clock pulse. A no-acknowledgement bit is used to terminate a read sequence.

A typical write sequence is executed according to the steps below:

- The master sends a start bit, and after the start bit, the master will send a byte data which is comprised of high 7-bit slave device's address and a write bit.
- The slave device acknowledges its address by sending an acknowledgement bit back to the master.
- The master then transfers the data 8 bits at a time, with the slave sending an acknowledgement bit after each 8 bits are sent.
- The master stops writing by sending a stop bit after all bytes have been transmitted.

A typical read sequence is executed as follows:

- The master sends a start bit, the slave address, a write bit and register address just as in the write sequence. The slave device acknowledges its address and data by sending an acknowledgement bit back to the master.
- The master sends a start bit and the slave address with a read bit. The slave device acknowledges its address by sending an acknowledgement bit back to the master.
- The slave device sends the register data out. The master receives the byte data and sends an acknowledgement bit after each 8 bits are transferred.
- The data transfer is completed when the master sends a stop bit after a no-acknowledgement bit.

Register Definitions

Register Address Mapping

Base Address

RISC Address<31:0>	Register	Description
0x800B_0000~0x800B_00FF	I2C_0 base address	I^2C module 0 register map
0x800B_0100~0x800B_01FF	I2C_1 base address	I^2C module 1 register map

Table 621: I^2C Base Address Mapping

Register Mapping

RISC Address <11:0>	Register	Description
0x000	I2C_CLK_CTRL	I^2C clock control register
0x004	-	Reserved
0x008	-	Reserved
0x00c	I2C_STATUS	I^2C status register
0x010	I2C_CTRL	I^2C control register
0x014	I2C_IO_CTRL	I^2C and I/O control register

RISC Address <11:0>	Register	Description
0x018	I2C_SDA_DELAY	The delay of I²C SDA driven after the SCL negative edge register.
0x01C	I2C_CMD_START	Start executing command after the command buffer is filled.
0x030	I2C_CMD_0	Command buffer
0x034	I2C_CMD_1	Command buffer
0x038	I2C_CMD_2	Command buffer
0x03C	I2C_CMD_3	Command buffer
0x040	I2C_CMD_4	Command buffer
0x044	I2C_CMD_5	Command buffer
0x048	I2C_CMD_6	Command buffer
0x04C	I2C_CMD_7	Command buffer
0x050	I2C_CMD_8	Command buffer
0x054	I2C_CMD_9	Command buffer
0x058	I2C_CMD_10	Command buffer
0x05C	I2C_CMD_11	Command buffer
0x060	I2C_CMD_12	Command buffer
0x064	I2C_CMD_13	Command buffer
0x068	I2C_CMD_14	Command buffer
0x06C	I2C_CMD_15	Command buffer
0x080	I2C_DATA_0	Read data buffer
0x084	I2C_DATA_1	Read data buffer
0x088	I2C_DATA_2	Read data buffer
0x08C	I2C_DATA_3	Read data buffer

Table 622: I²C Interface Register Mapping

Register Descriptions

- I²C Clock Control Register (I2C_CLK_CTRL) – 0x000

Bit	Name	Default	Description
15:0 (R/W)	CLK_PER	16'h00	The divider of the I²C clock from io_clk. If Fi2c is greater than 100k Value=Fio/(5.5*Fi2c) Else if Fi2c is less than 100k Value=Fio/(5*Fi2c) User can adjust the coefficients based on the waveform. Up to 300Kbps.
31:16	-	-	Reserved

Table 623: I²C Clock Control Register

- I²C (Reserved.) – 0x004
- I²C (Reserved.) – 0x008
- I²C Status Register (I2C_STATUS) – 0x00c

Bit	Name	Default	Description
0 (R)	I2C_BUSY	1'b0	Busy state flag
1 (R)	TIP	1'b0	Transfer flag. It will be cleared after each byte transfer.
2 (R)	RXACK	1'b0	Received acknowledge data. 0: ACK 1: NACK
3 (R)	-	-	Reserved
4 (R)	TR_INT	1'b0	Transfer interrupt. It will be triggered after each byte read or write operation. Write 1 will clear this interrupt. (test only)
5 (R)	-	1'b0	Reserved
6 (R)	STOP_INT	1'b0	Stop interrupt. It is triggered when a stop bit is detected in the bus. Write 1 will clear this interrupt.
7 (R)	-	-	Reserved
8 (R)	CMD_FINISH_INT	1'b0	Command buffer executes the finish interrupt
9 (R)	ERROR_INT	1'b0	Command buffer executes the error interrupt (ACK check)

Bit	Name	Default	Description
15:10 (R/W)	-	-	Reserved
20:16 (R)	CMD_INDEX	`8'h00	When cmd.exe outputs errors, the value is the command error index.
31:21	-	-	Reserved

Table 624: I²C Status Register

- I²C Control Register (I2C_CRTL) – 0x010

Bit	Name	Default	Description
0 (R/W)	I2C_CLEAR	1'b0	Set this bit to reset the module. It will clear itself. This is useful when the command buffer has errors.
1 (R/W)	CORE_EN	1'b1	1: Enables the core 0: Disables the core
2 (R/W)	MS	1'b1	1: Works in master mode. 0: Works in slave mode (not supported)
7:3	-	-	Reserved
8 (R/W)	TR_INT_EN	1'b0	Transfer interrupt enable (only for debugging)
9 (R/W)	SL_INT_EN	1'b0	Slave interrupt enable (only for debugging)
10 (R/W)	STOP_INT_EN	1'b0	Stop detected interrupt enable (only for debugging)
11 (R/W)	CMD_FINISH_INT_EN	1'b0	Command buffer execute finish interrupt enable
12 (R/W)	ERROR_INT_EN	1'b0	Command buffer execute error interrupt enable
31:13	-	-	Reserved

Table 625: I²C Control Register

- I²C I/O Control Register (I2C_IO_CTRL) – 0x014

Bit	Name	Default	Description
1:0 (R)	IO_IN	2'h00	Input data from I/O pins (test only)
3:2 (R/W)	IO_OUT	2'h00	Output data to I/O pins (test only)
5:4 (R/W)	IO_EN	2'h00	I/O pin enable (test only) 1: 2 pins used as output 0: 2 pins used as input
6 (R/W)	I2C_IO	1'b0	0: Used as I²C pins (test only) 1: Used IO_OUT (test only)

Bit	Name	Default	Description
31:7	-	-	Reserved

Table 626: I²C I/O Control Register

- I²C SDA Driven Delay after SCL Negative Edge Register (I2C_SDA_DELAY) – 0x018

Bit	Name	Default	Description
7:0 (R/W)	SDA_DELAY_REG	8'h20	SDA line delay pull-up after SCL negative edge. Users can set the bit the same as "I2C_CLK_CTRL." If it is over 0xff, set it to 0xff.
9:8 (R/W)	SCL_FILTER_REG	2'b00	SCL line input filter configuration: 2'b00: No filter 2'b01: 3 I/O CLK filter 2'b10: 6 I/O CLK filter 2'b11: 10 I/O CLK filter
31:10 (R/W)	-	-	Reserved

Table 627: I²C SDA Line Be Driven Delay after SCL Negative Edge Register

- I²C CMD Start Execute Register (I2C_CMD_START) – 0x01C

Bit	Name	Default	Description
0 (R/W)	CMD_START	1'b0	Writing this bit will start executing the command from cmd_buffer. After the command is executed, hardware will clear the bit regardless of whether the command was executed.
31:1 (R/W)	-	-	Reserved.

Table 628: I²C Command Start Execute Register

- I²C CMD Buffer Register (I2C_CMD_BUFFER) – 0x030~ 0x06C

Bit	Name	Default	Description
2:0	CMD_RP	3'b000	The repeat time of command execution. It will be cleared after the command is executed. 0: 1 time 1: 2 times 7: 8 times It will be cleared after the command is executed.
3 (R/W)	ACK	1'b0	Send the ACK bit during the read operation.

Bit	Name	Default	Description
			0: Send ACK 1: Send NACK Check ACK bit during the write operation 0: Check ACK 1: No check ACK It will be cleared after the command is executed.
4 (R/W)	WRITE	1'b0	Write a byte to the I^{2}C bus. It will be cleared after the command is successfully executed.
5 (R/W)	READ	1'b0	Read a byte from the bus. It will be cleared after the command is successfully executed.
6 (R/W)	STOP	1'b0	Send a stop after the transfer. It will be cleared after the command is successfully executed.
7 (R/W)	START	1'b0	Send the start before transfer. It will be cleared after the command is successfully executed.
31:8	-	-	Reserved

Table 629: I^{2}C Command Buffer Register

There are two command formats:

- Read Command one dword align

31:8	[7:0]
Reserved	Read command

Table 630: Read Command Format

- Write command, two dword align

31:8	[7:0]	31:8	[7:0]
Reserved	Write data	Reserved	Write command

Table 631: Write Command Format

If the previous command is a write command, the next command buffer will fill in the write data (not the next command). So if it fills in all read commands, it can fill 16 read commands, if fill the write command, it will only fill 8 write commands and 8 write data.

If errors occur during command execution, the command buffer will hold the command data rather than clear the data, and it will also set the I2C_CLEAR bit in I2C_CRTL register.

- I^{2}C Read_data_buffer Register (I2C_Read_data) – 0x080~0x8C

Bit	Name	Default	Description
7:0 (R)	DATA_BYTE_0	8'h00	The read data buffer
15:8 (R)	DATA_BYTE_1	8'h00	The read data buffer
23:16 (R)	DATA_BYTE_2	8'h00	The read data buffer
31:24 (R)	DATA_BYTE_3	8'h00	The read data buffer

Table 632: I^2C Read_data_buffer Register

The maximum data size should be 16 bytes when the data buffer receives a read request. However, if the data received exceeds 16 bytes, then data will overflow without outputting any INT error. Therefore it is recommended to take this into account when designing software.

When the current command buffer reaches 16 bytes or the data of next command buffer is 0, the controller will stop to execute and return results (the write data will not be included).

PACKAGE AND PIN SPECIFICATION

Mechanical Drawing of Package

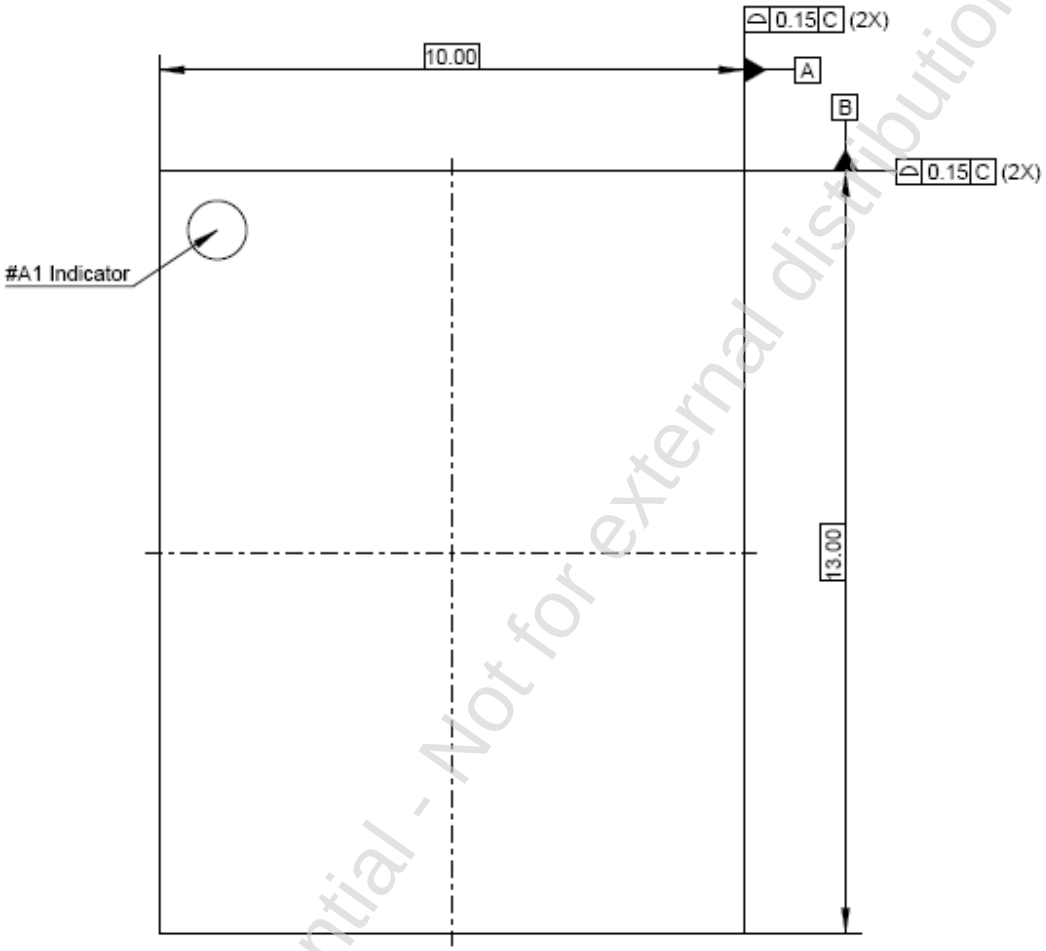


Figure 100: Top View

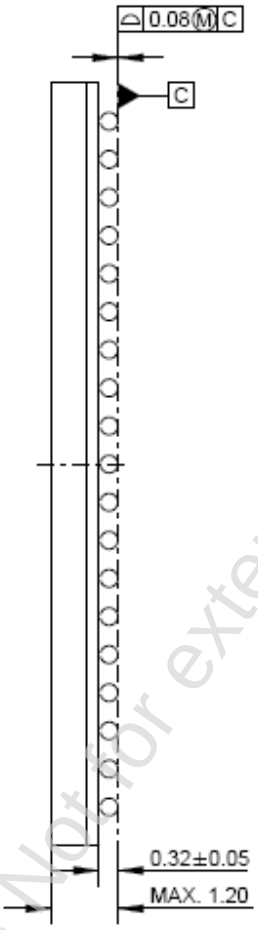


Figure 101: Side View

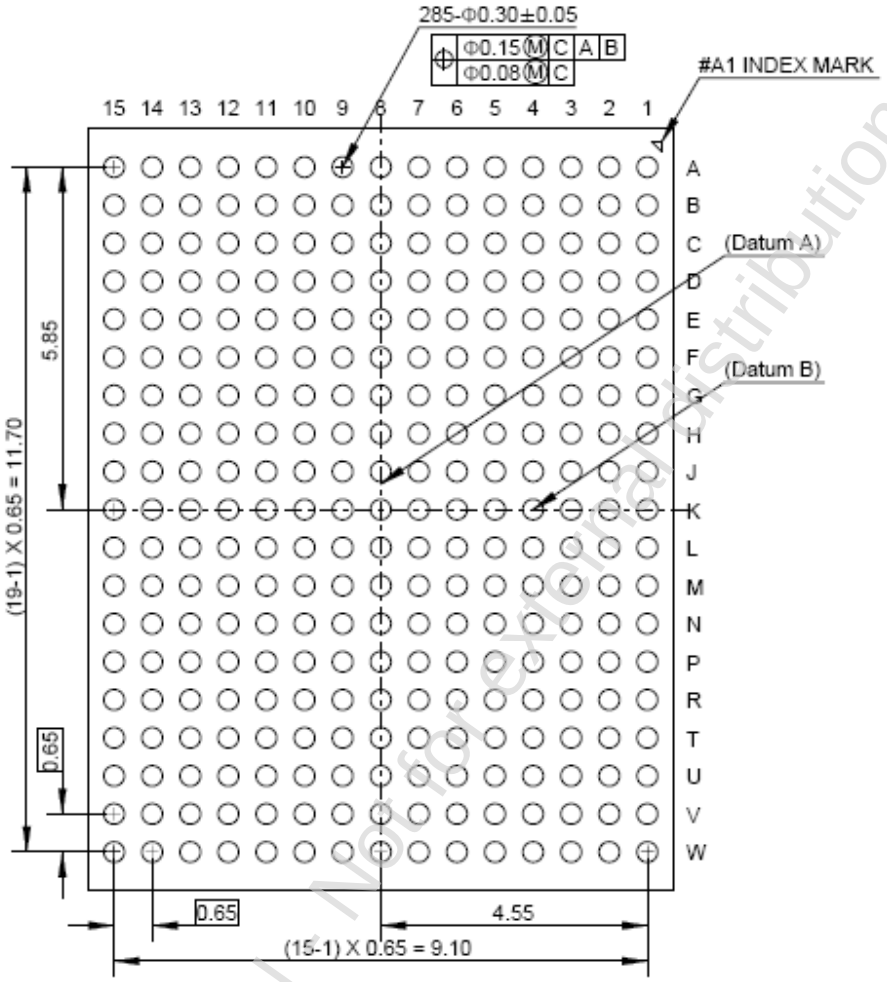


Figure 102: Bottom View

1. ALL DIMENSIONS AND TOLERANCES CONFIRM TO ASME Y14.5M-1994
2. REFERENCE SPECIFICATIONS:
   A.THIS DRAWING CONFORMS TO THE JEDEC REGISTERED OUTLINE MO-246

Figure 103: Notes for Package Information

Pin Sequence and Ball Assignment

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_sd_dat_1[3]	VDDIO	pvhbcudtbrt_s	cd0[2]:cd1[2]	0	GPIO	I(U)
x_sd_clk_3	VDDIO	pvhbcudtbrt_s	cd0[3]:cd1[3]	1	GPIO	I(U)
x_sd_cmd_3	VDDIO	pvhbcudtbrt_s	cd0[4]:cd1[4]	0	GPIO	I(U)
x_sd_dat_3[0]	VDDIO	pvhbcudtbrt_s	cd0[4]:cd1[4]	0	GPIO	I(U)
x_sd_dat_3[1]	VDDIO	pvhbcudtbrt_s	cd0[4]:cd1[4]	0	GPIO	I(U)
x_sd_dat_3[2]	VDDIO	pvhbcudtbrt_s	cd0[4]:cd1[4]	0	GPIO	I(U)
x_sd_dat_3[3]	VDDIO	pvhbcudtbrt_s	cd0[4]:cd1[4]	0	GPIO	I(U)
x_cko_1	VDDIO	pvhbcudtbrt_t	cd0[5]:cd1[5]	0	GPIO	I(D)
x_ac97_bit_clk	VDDIO	pvhbsudtart_s	cd0[6]:cd1[6]	1	AC97	I(D)
x_ac97_dout	VDDIO	pvhbcudtart_s	cd0[6]:cd1[6]	1	AC97	O(L)
x_ac97_din	VDDIO	pvhbcudtart_s	cd0[6]:cd1[6]	1	AC97	I(D)
x_ac97_sync	VDDIO	pvhbcudtart_s	cd0[6]:cd1[6]	1	AC97	O(L)
x_txd_0	VDDIO	pvhbcudtart_s	1	1	UART	O(H)
x_txd_1	VDDIO	pvhbcudtart_s	1	1	UART	O(H)
x_rxd_0	VDDIO	pvhbcudtart_s	1	1	UART	I(U)
x_rxd_1	VDDIO	pvhbcudtart_t	1	1	UART	I(U)
x_usclk_0	VDDIO	pvhbsudtart_s	cd0[7]:cd1[7]	1	GPIO	I(D)
x_utxd_0	VDDIO	pvhbcudtart_s	cd0[7]:cd1[7]	1	GPIO	I(D)
x_urxd_0	VDDIO	pvhbcudtart_s	cd0[7]:cd1[7]	1	GPIO	I(D)
x_utfs_0	VDDIO	pvhbcudtart_s	cd0[7]:cd	1	GPIO	I(D)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
			1[7]			
x_urfs_0	VDDIO	pvhbcudtart_s	cd0[7]:cd1[7]	1	GPIO	I(D)
x_usclk_1	VDDIO	pvhbsudtbrt_s	cd0[12]:cd1[12]	0	GPIO	I(Z)
x_utxd_1	VDDIO	pvhbcudtbrt_s	cd0[12]:cd1[12]	0	GPIO	I(Z)
x_urxd_1	VDDIO	pvhbcudtart_t	cd0[11]:cd1[11]	1	GPIO	I(D)
x_utfs_1	VDDIO	pvhbcudtart_s	cd0[11]:cd1[11]	1	GPIO	I(D)
x_urfs_1	VDDIO	pvhbcudtart_t	cd0[7]:cd1[7]	1	GPIO	I(D)
x_scan_en	VDDIO	pvhbcudtart_s	0	0	-	I(D)
x_usb_id	VDDA3V3_USB	pvhbr_s	-	-	-	-
x_usb_vbus	VDDA3V3_USB	pvhtbr_s	-	-	-	-
x_usb_dp	VDDA3V3_USB	pvhtbr_s	-	-	-	-
x_usb_txrtune_rkelvin	-	pvhbr_t	-	-	-	-
x_usb_dn	-	pvhtbr_s	-	-	-	-
x_xin	VDDIO	pvhsoscbrt_s	-	-	-	-
x_xout	VDDIO	pvhsoscbrt_s	-	-	-	-
x_cko_0	VDDIO	pvhbcudtbrt_s	0	0	-	O(L)
x_gps_clk	VDDIO	pvhbsudtcrt_s	1	1	GPS	I(Z)
x_gpio[0]	VDDIO	pvhbsudtart_s	1	1	GPIO	I(U)
x_gpio[1]	VDDIO	pvhbsudtart_s	1	1	GPIO	I(U)
x_gpio[2]	VDDIO	pvhbsudtart_t	1	1	GPIO	I(Z)
x_gpio[3]	VDDIO	pvhbsudtart_s	1	1	GPIO	I(Z)
x_gpio[4]	VDDIO	pvhbcudtart_t	1	1	GPIO	I(U)
x_gpio[5]	VDDIO	pvhbcudtart_s	1	1	GPIO	I(U)
x_gpio[6]	VDDIO	pvhbcudtart_t	1	1	GPIO	I(D)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_gpio[7]	VDDIO	pvhbcudtart_s	1	1	GPIO	I(D)
x_gpio[8]	VDDIO	pvhbsudtart_s	1	1	GPIO	I(U)
x_gpio[9]	VDDIO	pvhbcudtart_s	1	1	decided by boot mode	I(U):GPIO/I(U):JTAG
x_gpio[10]	VDDIO	pvhbcudtart_t	1	1	decided by boot mode	I(U):GPIO/I(U):JTAG
x_gpio[11]	VDDIO	pvhbcudtart_s	1	1	decided by boot mode	I(U):GPIO/I(Z):JTAG
x_gpio[12]	VDDIO	pvhbsudtart_t	1	1	decided by boot mode	I(U):GPIO/I(U):JTAG
x_gpio[13]	VDDIO	pvhbsudtart_s	1	1	decided by boot mode	I(U):GPIO/O(L):JTAG
x_gpio[14]	VDDIO	pvhbsudtart_t	1	1	decided by boot mode	I(U):GPIO/I(U):JTAG
x_gpio[15]	VDDIO	pvhbsudtart_s	1	1	decided by boot mode	I(D):GPIO/I(D):JTAG
x_xinw	VDDIO_RTC	pvhsoscatb_s	-	-	-	-
x_xoutw	VDDIO_RTC	pvhsoscatb_s	-	-	-	-
x_test_mode[5]	VDDIO_RTC	pvhbcudtart_s	0	0	-	I(Z)
x_test_mode[4]	VDDIO_RTC	pvhbcudtart_t	0	0	-	I(Z)
x_test_mode[3]	VDDIO_RTC	pvhbcudtart_s	0	0	-	I(Z)
x_test_mode[2]	VDDIO_RTC	pvhbcudtart_t	0	0	-	I(Z)
x_test_mode[1]	VDDIO_RTC	pvhbcudtart_s	0	0	-	I(Z)
x_test_mode[0]	VDDIO_RTC	pvhbcudtart_t	0	0	-	I(Z)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_rtc_rst_b	VDDIO_RTC	pvhbsudtart_s	0	0	-	I(Z)
x_on_key_b	VDDIO_RTC	pvhbsudtart_s	0	0	-	I(U)
x_ext_on	VDDIO_RTC	pvhbsudtart_s	0	0	-	I(Z)
x_low_batt_b	VDDIO_RTC	pvhbsudtart_t	0	0	-	I(U)
x_dram_en	VDDIO_RTC	pvhbcudtart_s	1	1	-	O(L)
x_system_en	VDDIO_RTC	pvhbcudtart_s	2	2	-	O(L)
x_reset_b	VDDIO_RTC	pvhbsudtart_t	0	0	-	I(Z)
x_sda_0	VDDIO	pvhbsudtart_s	cd0[8]:cd1[8]	1	GPIO	I(Z)
x_scl_0	VDDIO	pvhbsudtart_s	cd0[8]:cd1[8]	1	GPIO	I(Z)
x_xp	VDDA_TSC	pvhbr_s	-	-	-	-
x_xn	VDDA_TSC	pvhbr_t	-	-	-	-
x_yp	VDDA_TSC	pvhbr_s	-	-	-	-
x_yn	VDDA_TSC	pvhbr_t	-	-	-	-
x_aux2	VDDA_TSC	pvhbr_s	-	-	-	-
x_aux1	VDDA_TSC	pvhbr_t	-	-	-	-
x_aux0	VDDA_TSC	pvhbr_s	-	-	-	-
x_ref_adc	VDDA_TSC	pvhbr_t	-	-	-	-
x_mclk_o	VDDIO_MEM	pvmbsudtbrt_s	mem_clk_mode[0]:mem_clk_mode[1]	1	MEM	O(CLK)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_mclkb_o	VDDIO_MEM	pvmbsudtbrt_s	mem_clk_mode[0]: mem_clk_mode[1]	1	MEM	O(CLK)
x_ma[9]	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[5]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[4]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[8]	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[7]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[11]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[6]	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_gatei	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	-
x_gateo	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	-
x_mcke	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_ma[12]	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[13]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_mwe_b	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(H)
x_mcas_b	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(H)
x_mras_b	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(H)
x_mcs_b	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(H)
x_m_ba[0]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_m_ba[1]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[10]	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[0]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[1]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_ma[2]	VDDIO_MEM	pvmbsudtbrt_t	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_ma[3]	VDDIO_MEM	pvmbsudtbrt_s	mem_ctrl_mode[0]: mem_ctrl_mode[1]	1	MEM	O(L)
x_md[15]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[14]	VDDIO_MEM	pvmbsudtbrt_t	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[13]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[12]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_mdqm[1]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	O(L)
x_mdqs[1]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[11]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_md[10]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[9]	VDDIO_MEM	pvmbsudtbrt_t	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[8]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[7]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[6]	VDDIO_MEM	pvmbsudtbrt_t	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[5]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[4]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_mdqm[0]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	O(L)
x_mdqs[0]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_md[3]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[2]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[1]	VDDIO_MEM	pvmbsudtbrt_t	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_md[0]	VDDIO_MEM	pvmbsudtbrt_s	mem_data_mode[0]:mem_data_mode[1]	1	MEM	I(Z)
x_df_ad[7]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):SD
x_df_ad[6]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):SD
x_df_ad[5]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):SD
x_df_ad[4]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):SD
x_df_ad[3]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):SD
x_df_ad[2]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):SD
x_df_ad[1]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):SD
x_df_ad[0]	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	I(Z):Nand/I(Z):S

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
						D
x_ef_vgate	-	pvhtbr00_efuse_s	-	-	-	-
x_ef_fsource	-	pvhtbr00_efuse_s	-	-	-	-
x_df_cle	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	O(L):Nand/O(L):SD
x_df_ale	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	O(L):Nand/I(Z):SD
x_df_we_b	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	O(H):Nand/O(L):SD
x_df_re_b	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	decided by boot mode	O(H):Nand/I(Z):SD
x_df_ry_by	VDDIO_Nand	pvhbsudtbrt_s	cd0[9]:cd1[9]	1	Nand	I(Z)
x_df_wp_b	VDDIO_Nand	pvhbcudtbrt_s	cd0[9]:cd1[9]	1	Nand	O(L)
x_df_cs_b[1]	VDDIO_Nand	pvhbsudtbrt_s	cd0[9]:cd1[9]	1	Nand	O(H)
x_df_cs_b[0]	VDDIO_Nand	pvhbsudtbrt_s	cd0[9]:cd1[9]	1	Nand	O(H)
x_l_pclk	VDDIO_LCD	pvhbsudtcrt_s	cd0[1]:cd1[1]	1	LCD	I(D)
x_l_lck	VDDIO_LCD	pvhbsudtcrt_s	cd0[1]:cd1[1]	1	LCD	I(D)
x_l_fck	VDDIO_LCD	pvhbsudtcrt_s	cd0[1]:cd1[1]	1	LCD	I(D)
x_l_de	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	GPIO	I(D)
x_ldd[0]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[1]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[2]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_ldd[3]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[4]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[5]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[6]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[7]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[8]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[9]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[10]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[11]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[12]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[13]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[14]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_ldd[15]	VDDIO_LCD	pvhbcudtcrt_s	cd0[0]:cd1[0]	1	LCD	O(L)
x_gps_sgn	VDDIO	pvhbcudtcrt_s	1	1	GPIO	I(D)
x_gps_mag	VDDIO	pvhbcudtcrt_s	1	1	GPIO	I(D)
x_gps_sample_clk	VDDIO	pvhbcudtcrt_s	1	1	GPIO	I(D)
x_sd_cd_b_1	VDDIO	pvhbcudtcrt_s	1	1	GPIO	I(U)
x_sd_vcc_on_1	VDDIO	pvhbcudtcrt_s	1	1	GPIO	I(D)
x_sd_wp_b_1	VDDIO	pvhbcudtcrt_s	1	1	GPIO	I(U)
x_sd_clk_1	VDDIO	pvhbcudtbrt_s	cd0[10]:cd1[10]	1	GPIO	I(U)
x_sd_cmd_1	VDDIO	pvhbcudtbrt_s	cd0[2]:cd1[2]	0	GPIO	I(U)

A5 Ball Name	IOPST	Pad Cell	Drive	Drive Default	Default Function	Reset State
x_sd_dat_1[0]	VDDIO	pvhbcudtbrt_s	cd0[2]:cd1[2]	0	GPIO	I(U)
x_sd_dat_1[1]	VDDIO	pvhbcudtbrt_s	cd0[2]:cd1[2]	0	GPIO	I(U)
x_sd_dat_1[2]	VDDIO	pvhbcudtbrt_s	cd0[2]:cd1[2]	0	GPIO	I(U)

Table 633: Pin Status

A5 Ball Name	Function 1	Founction 1 Note	Function 2	Function 3	Function 4	Function 5	padctrl
x_sd_dat_1[3]	sd_dat_1[3]	-	-	-	gpio1[15]	-	padctrl[146]
x_sd_clk_3	sd_clk_3	-	-	-	gpio1[2]	-	padctrl[121]
x_sd_cmd_3	sd_cmd_3	-	-	-	gpio1[3]	-	padctrl[122]
x_sd_dat_3[0]	sd_dat_3[0]	-	spi_en_0	-	gpio1[28]	-	padctrl[123]
x_sd_dat_3[1]	sd_dat_3[1]	-	spi_clk_0	-	gpio1[29]	-	padctrl[124]
x_sd_dat_3[2]	sd_dat_3[2]	-	spi_din_0	-	gpio1[0]	-	padctrl[125]
x_sd_dat_3[3]	sd_dat_3[3]	-	spi_dout_0	-	gpio1[1]	-	padctrl[126]
x_cko_1	cko_1	-	i2s_mclk	-	gpio1[20]	-	padctrl[68]
x_ac97_bit_clk	ac97_bit_clk	i2s_bclk	-	-	gpio4[24]	-	padctrl[127]
x_ac97_dout	ac97_dout	i2s_dout[0]	-	-	gpio4[25]	-	padctrl[128]
x_ac97_din	ac97_din	i2s_din	-	-	gpio4[26]	-	padctrl[129]
x_ac97_sync	ac97_sync	i2s_lrclk	-	-	gpio4[27]	-	padctrl[130]
x_txd_0	txd_0	-	-	-	-	-	padctrl[159]
x_txd_1	txd_1	-	-	-	-	-	padctrl[160]
x_rxd_0	rxd_0	-	-	-	gpio3[15]	-	padctrl[118]
x_rxd_1	rxd_1	-	-	-	gpio3[16]	-	padctrl[119]
x_usclk_0	usclk_0	-	i2s_extclk	-	gpio0[18]	-	padctrl[41]

A5 Ball Name	Function 1	Founction 1 Note	Function 2	Function 3	Function 4	Function 5	padctrl
x_utxd_0	utxd_0	-	-	-	gpio0[19]	-	padctrl[42]
x_urxd_0	urxd_0	-	-	-	gpio0[20]	-	padctrl[43]
x_utfs_0	utfs_0	-	usb1_utmi_drvvbus	-	gpio0[21]	-	padctrl[44]
x_urfs_0	urfs_0	-	i2s_dout[1]	rts_0	gpio0[22]	-	padctrl[45]
x_usclk_1	usclk_1	-	headphone_r	-	gpio0[23]	-	padctrl[46]
x_utxd_1	utxd_1	-	headphone_l	-	gpio0[24]	-	padctrl[47]
x_urxd_1	urxd_1	-	spk_p	-	gpio0[25]	-	padctrl[48]
x_utfs_1	utfs_1	-	spk_n	-	gpio0[26]	-	padctrl[49]
x_urfs_1	urfs_1	-	i2s_dout[2]	cts_0	gpio0[27]	-	padctrl[50]
x_scan_en	scan_en	-	-	-	-	-	-
x_usb_id	-	-	-	-	-	-	-
x_usb_vbus	-	-	-	-	-	-	-
x_usb_dp	-	-	-	-	-	-	-
x_usb_txrtune_rkelvin	-	-	-	-	-	-	-
x_usb_dn	-	-	-	-	-	-	-
x_xin	-	-	-	-	-	-	-
x_xout	-	-	-	-	-	-	-
x_cko_0	cko_0	-	-	-	-	-	padctrl[161]
x_gps_clk	-	-	-	-	gpio4[28]	-	padctrl[5]
x_gpio[0]	-	-	-	-	gpio0[0]	-	padctrl[23]
x_gpio[1]	-	-	-	-	gpio0[1]	-	padctrl[24]
x_gpio[2]	scl_1	-	-	-	gpio0[2]	-	padctrl[25]
x_gpio[3]	sda_1	-	-	-	gpio0[3]	-	padctrl[26]
x_gpio[4]	pwm0	-	-	-	gpio0[4]	-	padctrl[27]
x_gpio[5]	pwm1	-	-	-	gpio0[5]	-	padctrl[28]
x_gpio[6]	pwm2	-	-	-	gpio0[6]	-	padctrl[29]

A5 Ball Name	Function 1	Founction 1 Note	Function 2	Function 3	Function 4	Function 5	padctrl
x_gpio[7]	pwm3	-	-	-	gpio0[7]	-	padctrl[30]
x_gpio[8]	warm_rst_b	-	-	-	gpio0[8]	-	padctrl[31]
x_gpio[9]	-	-	-	-	gpio0[9]	tms	padctrl[32]
x_gpio[10]	-	-	-	-	gpio0[10]	tdi	padctrl[33]
x_gpio[11]	-	-	-	-	gpio0[11]	tdo	padctrl[34]
x_gpio[12]	-	-	-	-	gpio0[12]	nsrst	padctrl[35]
x_gpio[13]	-	-	-	-	gpio0[13]	rtck	padctrl[36]
x_gpio[14]	-	-	-	-	gpio0[14]	ntrst	padctrl[37]
x_gpio[15]	-	-	-	-	gpio0[15]	tck	padctrl[38]
x_xinw	-	-	-	-	-	-	-
x_xoutw	-	-	-	-	-	-	-
x_test_mode[5]	-	-	-	-	-	-	-
x_test_mode[4]	-	-	-	-	-	-	-
x_test_mode[3]	-	-	-	-	-	-	-
x_test_mode[2]	-	-	-	-	-	-	-
x_test_mode[1]	-	-	-	-	-	-	-
x_test_mode[0]	-	-	-	-	-	-	-
x_rtc_rst_b	-	-	-	-	-	-	-
x_on_key_b	-	-	-	-	-	-	-
x_ext_on	-	-	-	-	-	-	-
x_low_batt_b	-	-	-	-	-	-	-
x_dram_en	-	-	-	-	-	-	-
x_system_en	-	-	-	-	-	-	-
x_reset_b	-	-	-	-	-	-	-
x_sda_0	sda_0	-	-	-	gpio3[18]	-	padctrl[0]

A5 Ball Name	Function 1	Founction 1 Note	Function 2	Function 3	Function 4	Function 5	padctrl
x_scl_0	scl_0	-	-	-	gpio3[19]	-	padctrl[1]
x_xp	-	-	-	-	-	-	-
x_xn	-	-	-	-	-	-	-
x_yp	-	-	-	-	-	-	-
x_yn	-	-	-	-	-	-	-
x_aux2	-	-	-	-	-	-	-
x_aux1	-	-	-	-	-	-	-
x_aux0	-	-	-	-	-	-	-
x_ref_adc	-	-	-	-	-	-	-
x_mclk_o	-	mclk_o	-	-	-	-	padctrl[116]
x_mclkb_o	-	mclkb_o	-	-	-	-	padctrl[215]
x_ma[9]	-	ma[9]	-	-	-	-	padctrl[180]
x_ma[5]	-	ma[5]	-	-	-	-	padctrl[176]
x_ma[4]	-	ma[4]	-	-	-	-	padctrl[175]
x_ma[8]	-	ma[8]	-	-	-	-	padctrl[179]
x_ma[7]	-	ma[7]	-	-	-	-	padctrl[178]
x_ma[11]	-	ma[11]	-	-	-	-	padctrl[181]
x_ma[6]	-	ma[6]	-	-	-	-	padctrl[177]
x_gatei	-	gatei	-	-	-	-	-
x_gateo	-	gateo	-	-	-	-	-
x_mcke	-	mcke	-	-	-	-	padctrl[183]
x_ma[12]	-	ma[12]	-	-	-	-	padctrl[182]
x_ma[13]	-	ma[13]	-	-	-	-	padctrl[216]
x_mwe_b	-	mwe_b	-	-	-	-	padctrl[184]
x_mcas_b	-	mcas_b	-	-	-	-	padctrl[185]
x_mras_b	-	mras_b	-	-	-	-	padctrl[186]
x_mcs_b	-	mcs_b	-	-	-	-	padctrl[187]
x_m_ba[0]	-	m_ba[0]	-	-	-	-	padctrl[188]
x_m_ba[1]	-	m_ba[1]	-	-	-	-	padctrl[189]

A5 Ball Name	Function 1	Founction 1 Note	Function 2	Function 3	Function 4	Function 5	padctrl
x_ma[10]	-	ma[10]	-	-	-	-	padctrl[190]
x_ma[0]	-	ma[0]	-	-	-	-	padctrl[191]
x_ma[1]	-	ma[1]	-	-	-	-	padctrl[192]
x_ma[2]	-	ma[2]	-	-	-	-	padctrl[193]
x_ma[3]	-	ma[3]	-	-	-	-	padctrl[194]
x_md[15]	-	md[15]	-	-	-	-	padctrl[214]
x_md[14]	-	md[14]	-	-	-	-	padctrl[213]
x_md[13]	-	md[13]	-	-	-	-	padctrl[212]
x_md[12]	-	md[12]	-	-	-	-	padctrl[211]
x_mdqm[1]	-	mdqm[1]	-	-	-	-	padctrl[209]
x_mdqs[1]	-	mdqs[1]	-	-	-	-	padctrl[210]
x_md[11]	-	md[11]	-	-	-	-	padctrl[208]
x_md[10]	-	md[10]	-	-	-	-	padctrl[207]
x_md[9]	-	md[9]	-	-	-	-	padctrl[206]
x_md[8]	-	md[8]	-	-	-	-	padctrl[205]
x_md[7]	-	md[7]	-	-	-	-	padctrl[204]
x_md[6]	-	md[6]	-	-	-	-	padctrl[203]
x_md[5]	-	md[5]	-	-	-	-	padctrl[202]
x_md[4]	-	md[4]	-	-	-	-	padctrl[201]
x_mdqm[0]	-	mdqm[0]	-	-	-	-	padctrl[199]
x_mdqs[0]	-	mdqs[0]	-	-	-	-	padctrl[200]
x_md[3]	-	md[3]	-	-	-	-	padctrl[198]
x_md[2]	-	md[2]	-	-	-	-	padctrl[197]
x_md[1]	-	md[1]	-	-	-	-	padctrl[196]
x_md[0]	-	md[0]	-	-	-	-	padctrl[195]
x_df_ad[7]	df_ad[7]	-	sd_dat_0[7]	sd_dat_2[7]	-	-	padctrl[110]
x_df_ad[6]	df_ad[6]	-	sd_dat_0[6]	sd_dat_2[6]	-	-	padctrl[147]
x_df_ad[5]	df_ad[5]	-	sd_dat_0[5]	sd_dat_2[5]	-	-	padctrl[148]

A5 Ball Name	Function 1	Founction 1 Note	Function 2	Function 3	Function 4	Function 5	padctrl
x_df_ad[4]	df_ad[4]	-	sd_dat_0[4]	sd_dat_2[4]	-	-	padctrl[149]
x_df_ad[3]	df_ad[3]	-	sd_dat_0[3]	sd_dat_2[3]	-	-	padctrl[150]
x_df_ad[2]	df_ad[2]	-	sd_dat_0[2]	sd_dat_2[2]	-	-	padctrl[151]
x_df_ad[1]	df_ad[1]	-	sd_dat_0[1]	sd_dat_2[1]	-	-	padctrl[152]
x_df_ad[0]	df_ad[0]	-	sd_dat_0[0]	sd_dat_2[0]	-	-	padctrl[153]
x_ef_vgate	-	-	-	-	-	-	-
x_ef_fsource	-	-	-	-	-	-	-
x_df_cle	df_cle	-	sd_clk_0	-	-	-	padctrl[132]
x_df_ale	df_ale	-	sd_cmd_0	-	-	-	padctrl[133]
x_df_we_b	df_we_b	-	sd_clk_2	-	-	-	padctrl[120]
x_df_re_b	df_re_b	-	sd_cmd_2	-	-	-	padctrl[120]
x_df_ry_by	df_ry_by	-	-	-	-	-	padctrl[162]
x_df_wp_b	df_wp_b	-	sd_vcc_on_0	-	-	-	padctrl[131]
x_df_cs_b[1]	df_cs_b[3]	-	-	-	-	-	padctrl[108]
x_df_cs_b[0]	df_cs_b[0]	-	df_cs_b[2]	-	-	-	padctrl[109]
x_l_pclk	l_pclk	-	fce_b[1]	-	-	-	padctrl[6]
x_l_lck	l_lck	-	fwe_b	-	-	-	padctrl[141]
x_l_fck	l_fck	-	foe_b	-	-	-	padctrl[142]
x_l_de	l_de	-	fa[1]	-	gpio3[23]	-	padctrl[143]
x_ldd[0]	ldd[0]	-	fd[0]	-	-	-	padctrl[163]
x_ldd[1]	ldd[1]	-	fd[1]	-	-	-	padctrl[164]
x_ldd[2]	ldd[2]	-	fd[2]	-	-	-	padctrl[165]
x_ldd[3]	ldd[3]	-	fd[3]	-	-	-	padctrl[166]

A5 Ball Name	Function 1	Founction 1 Note	Function 2	Function 3	Function 4	Function 5	padctrl
x_ldd[4]	ldd[4]	-	fd[4]	-	-	-	padctrl[167]
x_ldd[5]	ldd[5]	-	fd[5]	-	-	-	padctrl[168]
x_ldd[6]	ldd[6]	-	fd[6]	-	-	-	padctrl[169]
x_ldd[7]	ldd[7]	-	fd[7]	-	-	-	padctrl[170]
x_ldd[8]	ldd[8]	-	fd[8]	-	-	-	padctrl[171]
x_ldd[9]	ldd[9]	-	fd[9]	-	-	-	padctrl[172]
x_ldd[10]	ldd[10]	-	fd[10]	-	-	-	padctrl[173]
x_ldd[11]	ldd[11]	-	fd[11]	-	-	-	padctrl[174]
x_ldd[12]	ldd[12]	-	fd[12]	-	-	-	padctrl[154]
x_ldd[13]	ldd[13]	-	fd[13]	-	-	-	padctrl[155]
x_ldd[14]	ldd[14]	-	fd[14]	-	-	-	padctrl[156]
x_ldd[15]	ldd[15]	-	fd[15]	-	-	-	padctrl[157]
x_gps_sgn	gps_sgn	-	-	-	gpio3[20]	-	padctrl[2]
x_gps_mag	gps_mag	-	-	-	gpio3[21]	-	padctrl[3]
x_gps_sample_clk	gps_sample_clk	-	-	-	gpio3[22]	-	padctrl[4]
x_sd_cd_b_1	sd_cd_b_1	-	-	-	gpio1[7]	-	padctrl[62]
x_sd_vcc_on_1	sd_vcc_on_1	-	-	-	gpio1[8]	-	padctrl[63]
x_sd_wp_b_1	sd_wp_b_1	-	-	-	gpio1[9]	-	padctrl[64]
x_sd_clk_1	sd_clk_1	-	-	-	gpio1[10]	-	padctrl[65]
x_sd_cmd_1	sd_cmd_1	-	-	-	gpio1[11]	-	padctrl[66]
x_sd_dat_1[0]	sd_dat_1[0]	-	-	-	gpio1[12]	-	padctrl[67]
x_sd_dat_1[1]	sd_dat_1[1]	-	-	-	gpio1[13]	-	padctrl[144]
x_sd_dat_1[2]	sd_dat_1[2]	-	-	-	gpio1[14]	-	padctrl[145]

Table 634: SiRFatlasV Pin Share

Pin Name	Pin Num	Description
Power List		
VDD_Core	E10, E11, F11, H8,J8,K6,L8,M7, N11, P11	Core power.
VDDIO_N	E8	I/O power (NAND interface).
VDDIO_L	F8	I/O power (LCD interface).
VDDIO	E6,E7,N6, P6	I/O power.
VDDIO_MEM	F10,G9,H11,J9, K11,L10, M9	Memory I/O power.
VDD_PLL0/1	M1,M2	PLL power
VDD_RTC	R12	Digital power output for RTC. Bypass capacitor needed
VDDIO_RTC	U15	I/O power for RTC and power input of LDO for RTC digital power.
VDDIO_OSC	L3	Crystal oscillator power.
VDD_USB	J5	USB PHY core power.
VDDA_USB	H1	Analog power for USB.
VDD_TSC	P12	Digital power for internal touch screen controller.
VDDA3V3_TSC	N13	Analog power for touch screen controller.
VREF_ADC	P14	Reference voltage for the ADC
VDDIO_DAC	G6	MoDAC power
VINDCDC1_2	W8	Power input for switching DCDC 1 and 2.
VINLDO1_2	W12,V12	Power input for LDO1 and LDO2; Power for internal switching DC/DC and LDOs
VINLDO3_4	W3	Power input for LDO 3 and 4
L1	W9	Switch pin for DCDC1, connect to inductor
PGND1	W10	GND for DCDC1
FB_DCDC1	W11	Input to adjust output voltage of DCDC1 through external resister divider
EN_DCDC1	V11	Enable control of DCDC1, active high
L2	W7	Switch pin for DCDC2, connect to inductor
PGND2	W6	GND for DCDC2

Pin Name	Pin Num	Description
DEFDCDC2	W5	Input to adjust output voltage of DCDC2 through external resister divider
VDCDC2	V8	Feedback voltage sense input, connect directly to the output of DCDC2
EN_DCDC2	W13	Enable control of DCDC2, active high
VLDO1	V15	Power output of LDO1
EN_LDO1	U12	Enable control of LDO1, active high
VLDO2	W4	Power output of LDO2
FB2	V4	LDO2 feedback input for the external voltage divider
EN_LDO2	W14	Enable control of LDO2, active high
VLDO3	V1	Power output of LDO3
EN_LDO3	V7	Enable control of LDO3, active high
VLDO4	W1	Power output of LDO4
FB4	W2	LDO4 feedback input for the external voltage divider
EN_LDO4	V5	Enable control of LDO4, active high
Mode	V13	Select between Power Safe Mode and forced PWM Mode for DCDC1 and DCDC2. In Power Safe Mode, PFM is used at light loads, PWM for higher loads. If PIN is set to high level, forced PWM Mode is selected. If Pin has low level, then the device operates in Power Safe Mode
BP	W15	Input for bypass capacitor for internal reference, bypass capacitor needed
THRESHOLD	U5	Voltage comparator input, connect to external resister divider
HYSTERESIS	U4	Input for hysteresis on threshold, connect to external resister divider
COMPO	V3	Voltage comparator output, open-drain output, active low
Ground List		
VSS	E9, F6, F7, F9, G8, G10, G11, H6, H7, H9, H10, J6, J7, J10, J11, K7, K8, K9, K10, L6, L7, L9, L11, M6, M8, M10, M11, N7, N8, N9, N10, P8,P9, R8, R9, T7, T8, T9, T10, U7, U8,	The main digital ground domain.

Pin Name	Pin Num	Description
	U9, V9	
VSS_PLL0/1	N1, N2	Analog ground for PLL.
VSSIO_OSC	M3	I/O ground for oscillator
VSS_TSC	R13	Digital ground for internal touch screen controller.
VSSA_TSC	N12	Analog ground for the Touch Screen Controller.
VSSA_USB	J2	Analog ground for USB PHY
VSSIO_DAC	G7	MoDAC ground
FB2	V4	LDO2 feedback input for the external voltage divider
EN_LDO2	W14	Enable control of LDO2, active high
VLDO3	V1	Power output of LDO3
EN_LDO3	V7	Enable control of LDO3, active high
VLDO4	W1	Power output of LDO4
FB4	W2	LDO4 feedback input for the external voltage divider
EN_LDO4	V5	Enable control of LDO4, active high
BP	W15	Input for bypass capacitor for internal reference, bypass capacitor needed
THRESHOLD	U5	Voltage comparator input, connect to external resister divider
HYSTERESIS	U4	Input for hysteresis on threshold, connect to external resister divider
COMPO	V3	Voltage comparator output, open-drain output, active low
Ground List		
VSS	E9,F6,F7,F9,G8,G10,G11,H6,H7,H9,H10,J6,J7,J10,J11,K7,K8,K9,K10,L6,L7,L9,L11,M6,M8,M10,M11,N7,N8,N9,N10,P8,P9,R8,R9,T7,T8,T9,T10,U7,U8,U9,V9	The main digital ground domain.
VSS_PLL0/1	N1, N2	Analog ground for PLL.
VSSIO_OSC	M3	I/O ground for oscillator

Pin Name	Pin Num	Description
VSS_TSC	R13	Digital ground for internal touch screen controller.
VSSA_TSC	N12	Analog ground for the Touch Screen Controller.
VSSA_USB	J2	Analog ground for USB PHY
VSSIO_DAC	G7	MoDAC ground

Table 635: SiRFatlasV Power/Ground List

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A	x_sd_dat_1[2]	x_gps_mag	x_gps_sgn	x_ld d[15]	x_ld d[11]	x_ld d[7]	x_ld d[3]	x_l _fck	x_l_ pclk	x_df _we_b	x_df _ad[1]	x_df_ ad[4]	x_df_ ad[7]	x_m d[0]	x_m d[2]	A
B	x_sd_dat_1[1]	x_gps_sample_clk	x_gps_clk	x_ld d[14]	x_ld d[10]	x_ld d[6]	x_ld d[2]	x_l _de	x_l _lck	x_df _ale	x_df_ ad[0]	x_df_ ad[3]	x_df_ ad[6]	x_m d[1]	x_m d[3]	B
C	x_sd_dat_1[3]	x_sd_ clk_1	x_sd_dat_1[0]	x_ld d[13]	x_ld d[9]	x_ld d[5]	x_ld d[1]	x_df_cs_b[0]	x_df_ ry_by	x_df _cle	X_EF_ VGATE	x_df_ ad[2]	x_df_ ad[5]	x_md qm[0]	x_md qs[0]	C
D	x_sd_dat_3[0]	x_sd_ clk_3	x_sd_ cmd_1	x_ld d[12]	x_ld d[8]	x_ld d[4]	x_ld d[0]	x_df_cs_b[1]	x_df _wp_b	x_df _re_b	X_EF_F SOURCE	x_m a[3]	x_m d[7]	x_m d[4]	x_m d[5]	D
E	x_sd_dat_3[1]	x_c ko_1	x_sd_dat_3[3]	x_sd_ cmd_3	x_sd_ wp_b_1	VDDIO	VDDIO	VDD IO_N	VSS	VDD_ CORE	VDD_ CORE	x_m a[2]	x_m d[6]	x_m d[8]	x_md [10]	E
F	x_ac97_ bit_clk	x_us clk_0	x_ac9 7_din	x_sd_dat_3[2]	x_sd_v cc_on_1	VSS	VSS	VDD IO_L	VSS	VDDI O_MEM	VDD_ CORE	x_m a[1]	x_md [11]	x_md qs[1]	x_m d[9]	F
G	x_ac9 7_dout	x_us clk_1	x_ur xd_0	x_t xd_0	x_sd_ cd_b_1	VDDIO_D AC	VSSIO_D AC	VSS	VDDI O_MEM	VSS	VSS	x_m a[0]	x_md [13]	x_md [12]	x_md qm[1]	G
H	VDDA _USB	x_ac9 7_sync	x_ut xd_1	x_t xd_1	x_sc an_en	VSS	VSS	VDD_ CORE	VSS	VSS	VDDI O_MEM	x_ma [10]	x_m_ ba[1]	x_md [15]	x_md [14]	H
J	x_us b_dp	VSSA _USB	x_ur fs_1	x_ur fs_0	VDD _USB	VSS	VSS	VDD_ CORE	VDDI O_MEM	VSS	VSS	x_m_ ba[0]	x_m we_b	x_m a[6]	x_mr as_b	J
K	x_us b_dn	x_usb_txrt une_rkelvin	x_ur xd_1	x_ut fs_1	x_us b_id	VDD_ CORE	VSS	VSS	VSS	VSS	VDDI O_MEM	x_m cs_b	x_ma [12]	x_m a[8]	x_mc as_b	K
L	x_r xd_0	x_ut xd_0	VDDI O_OSC	x_ut fs_0	x_gpi o[11]	VSS	VSS	VDD_ CORE	VSS	VDDI O_MEM	VSS	x_ma [13]	x_m cke	x_m a[9]	x_ma [11]	L
M	VDD_ PLL0	VDD_ PLL1	VSSI O_OSC	x_r xd_1	x_gp io[9]	VSS	VDD_ CORE	VSS	VDDI O_MEM	VSS	VSS	x_m a[7]	x_m a[4]	x_m a[5]	x_mc lkb_o	M
N	VSS_ PLL0	VSS_ PLL1	x_gp io[1]	x_usb _vbus	x_test_ mode[0]	VDDIO	VSS	VSS	VSS	VSS	VDD_ CORE	VSSA _TSC	VDDA3 V3_TSC	x_g ateo	x_mc lk_o	N
P	x_xin	x_gp io[0]	x_gp io[4]	x_gpi o[10]	x_gpi o[14]	VDDIO	x_rtc _rst_b	VSS	VSS	x_sys tem_en	VDD_ CORE	VDD _TSC	x_low_ batt_b	x_re f_adc	x_g atei	P
R	x_x out	x_gp io[8]	x_gp io[5]	x_gp io[3]	x_gpi o[15]	x_test_ mode[2]	VLDO5	VSS	VSS	x_test_ mode[1]	x_re set_b	VDD _RTC	VSS _TSC	x_yn	x_yp	R
T	x_x inw	x_gp io[2]	x_gp io[7]	x_gpi o[13]	x_gpi o[12]	x_test_ mode[3]	VSS	VSS	VSS	VSS	x_s da_0	x_s cl_0	x_a ux1	x_xn	x_xp	T
U	x_x outw	x_c ko_0	x_gp io[6]	HYSTE RESIS	THRE SHOLD	x_test_ mode[4]	VSS	VSS	VSS	x_ex t_on	x_dr am_en	EN_ LDO1	x_a ux0	x_a ux2	VDDI O_RTC	U
V	VLDO3	NC	COM PO	FB2	EN_ LDO4	x_test_ mode[5]	EN_ LDO3	VDC DC2	VSS	x_on_ key_b	EN_D CDC1	VIN LDO1_2	NC	NC	VLDO1	V
W	VLDO4	FB4	VINL DO3_4	VLDO2	DEFD CDC2	PGND2	L2	VINDC DC1_2	L1	PGND1	FB_D CDC1	VIN LDO1_2	EN_D CDC2	EN_ LDO2	BP	W
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

Figure 104: Package Definition

Ball Location	Ball Name	Ball Location	Ball Name	Ball Location	Ball Name
A1	x_sd_dat_1[2]	E1	x_sd_dat_3[1]	M4	x_rxd_1
A2	x_gps_mag	E2	x_cko_1	M5	x_gpio[9]
A3	x_gps_sgn	E3	x_sd_dat_3[3]	M12	x_ma[7]
A4	x_ldd[15]	E4	x_sd_cmd_3	M13	x_ma[4]
A5	x_ldd[11]	E5	x_sd_wp_b_1	M14	x_ma[5]
A6	x_ldd[7]	E12	x_ma[2]	M15	x_mclkb_o
A7	x_ldd[3]	E13	x_md[6]	N3	x_gpio[1]
A8	x_l_fck	E14	x_md[8]	N4	x_usb_vbus
A9	x_l_pclk	E15	x_md[10]	N5	x_test_mode[0]
A10	x_df_we_b	F1	x_ac97_bit_clk	N14	x_gateo
A11	x_df_ad[1]	F2	x_usclk_0	N15	x_mclk_o
A12	x_df_ad[4]	F3	x_ac97_din	P1	x_xin
A13	x_df_ad[7]	F4	x_sd_dat_3[2]	P2	x_gpio[0]
A14	x_md[0]	F5	x_sd_vcc_on_1	P3	x_gpio[4]
A15	x_md[2]	F12	x_ma[1]	P4	x_gpio[10]
B1	x_sd_dat_1[1]	F13	x_md[11]	P5	x_gpio[14]
B2	x_gps_sample_clk	F14	x_mdqs[1]	P7	x_rtc_rst_b
B3	x_gps_clk	F15	x_md[9]	P10	x_system_en
B4	x_ldd[14]	G1	x_ac97_dout	P13	x_low_batt_b
B5	x_ldd[10]	G2	x_usclk_1	P15	x_gatei
B6	x_ldd[6]	G3	x_urxd_0	R1	x_xout
B7	x_ldd[2]	G4	x_txd_0	R2	x_gpio[8]
B8	x_l_de	G5	x_sd_cd_b_1	R3	x_gpio[5]
B9	x_l_lck	G12	x_ma[0]	R4	x_gpio[3]
B10	x_df_ale	G13	x_md[13]	R5	x_gpio[15]
B11	x_df_ad[0]	G14	x_md[12]	R6	x_test_mode[2]
B12	x_df_ad[3]	G15	x_mdqm[1]	R10	x_test_mode[1]
B13	x_df_ad[6]	H2	x_ac97_sync	R11	x_reset_b
B14	x_md[1]	H3	x_utxd_1	R14	x_yn

Ball Location	Ball Name	Ball Location	Ball Name	Ball Location	Ball Name
B15	x_md[3]	H4	x_txd_1	R15	x_yp
C1	x_sd_dat_1[3]	H5	x_scan_en	T1	x_xinw
C2	x_sd_clk_1	H12	x_ma[10]	T2	x_gpio[2]
C3	x_sd_dat_1[0]	H13	x_m_ba[1]	T3	x_gpio[7]
C4	x_ldd[13]	H14	x_md[15]	T4	x_gpio[13]
C5	x_ldd[9]	H15	x_md[14]	T5	x_gpio[12]
C6	x_ldd[5]	J1	x_usb_dp	T6	x_test_mode[3]
C7	x_ldd[1]	J3	x_urfs_1	T11	x_sda_0
C8	x_df_cs_b[0]	J4	x_urfs_0	T12	x_scl_0
C9	x_df_ry_by	J12	x_m_ba[0]	T13	x_aux1
C10	x_df_cle	J13	x_mwe_b	T14	x_xn
C11	X_EF_VGATE	J14	x_ma[6]	T15	x_xp
C12	x_df_ad[2]	J15	x_mras_b	U1	x_xoutw
C13	x_df_ad[5]	K1	x_usb_dn	U2	x_cko_0
C14	x_mdqm[0]	K2	x_usb_txrtune_rkelvin	U3	x_gpio[6]
C15	x_mdqs[0]	K3	x_urxd_1	U6	x_test_mode[4]
D1	x_sd_dat_3[0]	K4	x_utfs_1	U10	x_ext_on
D2	x_sd_clk_3	K5	x_usb_id	U11	x_dram_en
D3	x_sd_cmd_1	K12	x_mcs_b	U13	x_aux0
D4	x_ldd[12]	K13	x_ma[12]	U14	x_aux2
D5	x_ldd[8]	K14	x_ma[8]	V6	x_test_mode[5]
D6	x_ldd[4]	K15	x_mcas_b	V10	x_on_key_b
D7	x_ldd[0]	L1	x_rxd_0	-	-
D8	x_df_cs_b[1]	L2	x_utxd_0	-	-
D9	x_df_wp_b	L4	x_utfs_0	-	-
D10	x_df_re_b	L5	x_gpio[11]	-	-
D11	X_EF_FSOURCE	L12	x_ma[13]	-	-
D12	x_ma[3]	L13	x_mcke	-	-
D13	x_md[7]	L14	x_ma[9]	-	-

Ball Location	Ball Name	Ball Location	Ball Name	Ball Location	Ball Name
D14	x_md[4]	L15	x_ma[11]	-	-
D15	x_md[5]	-	-	-	-

Table 636: SiRFatlasV Ball Assignment

Pad Type Description

Bi-Directional Buffer: PV[a][b]B[c]UDT[d][e]			
[a]		[d]	
h	1.8V~3.3V Wide voltage range IO	a	A type Output Driver
m	1.8V~2.5V Wide voltage range I/O	b	B type Output Driver
		c	C type Output Driver
[b]		[e]	
None	Normal I/O	rt	Retention I/O
t	Tolerant I/O	fs	Fail-Safe I/O
[c]			
c	CMOS Input	-	-
s	Schmitt Trigger Input	-	-
Analog Bi-Directional Bypass PAD : PV[a]BR[b]			
[a]		[b]	
None	Normal Bypass PAD	00	0ohm Input Resistor
t	Tolerant Bypass PAD	10	10ohm Input Resistor
		50	50ohm Input Resistor

Type	Parameter		Description
I/O	**PVH**bcudtart	**PVH**	Normal 1.8V~3.3V Wide voltage range I/O
	PVMbcudtart	**PVM**	Normal 1.8V~2.5V Wide voltage range I/O
Tolerant	pvh_bcudtart	**None**	Normal I/O
	pvh**T**bcudtart	**T**	Tolerant I/O
Buffer Type	pvh**I**srt	**I**	Input buffer

Type	Parameter		Description
	pvh**B**cudtart	**B**	Bi-Direction
	pvhs**OSC**brt	**OSC**	Oscillator
	pvh**CKD**srt	**CKD**	Clock Driver
Input Mode	pvhb**C**udtart	**C**	Cmos Input
	pvhb**S**udtart	**S**	Schmitt trigger Input
Pull-Up/Down	pvhbc**UD**tart	**UD**	Pull-UP/Down Control
Output Type	pvhbcud**T**art	**T**	Tri-state Output
Driver Strength	pvhbcudt**A**rt	**A**	4-step Controllable Low Driver Strength Output
	pvhbcudt**B**rt	**B**	4-step Controllable High Driver Strength Output
	pvhbcudt**C**rt	**C**	4-step Controllable Slew rated Low Driver Strength Output
Power Management Function	pvhbcudta**RT**	**RT**	Retention I/O
	pvhtbcudta**FS**	**FS**	Failsafe I/O
Analog In-Out Resistor	pvhb**R**	**R**	Analog pad with 0ohm, 50ohm and 100ohm resistor
	pvhtbr**00**_efuse	**00**	Resistor Value 0ohm
PAD TYPE	pvhbcudtart	**None**	Inline CUP PAD
	pvhbcudtart**_S**	**_S**	Staggered I/O Short (inner) CUP PAD
	pvhbcudtart**_T**	**_T**	Staggered I/O Tall (outer) CUP PAD

Table 637: SiRFatlasV Pad Type Naming Description

Parameter (Driver Type)			Worst VDD=3.00V T=125℃ Process=Slow Isink at 0.20V Isource at 2.80V	Typical VDD=3.30V T=25℃ Process=Nominal Isink at 0.20V Isource at 3.10V	Best VDD=3.60V T=-40℃ Process=Fast Isink at 0.20V Isource at 3.40V
A, C	CD1=0,CD0=0	Isink	1.7mA	2.6mA	3.4mA
		Isource	-1.8mA	-2.6mA	-3.5mA
	CD1=0,CD0=1	Isink	3.4mA	5.2mA	6.9mA
		Isource	-3.7mA	-5.2mA	-7.1mA
	CD1=1,CD0=0	Isink	5.2mA	7.8mA	10.4mA
		Isource	-5.6mA	-7.8mA	-10.6mA
	CD1=1,CD0=1	Isink	6.9mA	10.5mA	13.9mA
		Isource	-7.5mA	-10.4mA	-14.2mA
B	CD1=0,CD0=0	Isink	3.4mA	5.2mA	6.9mA
		Isource	-3.7mA	-5.2mA	-7.1mA
	CD1=0,CD0=1	Isink	6.9mA	10.5mA	13.9mA
		Isource	-7.5mA	-10.4mA	-14.2mA
	CD1=1,CD0=0	Isink	10.4mA	15.7mA	20.9mA
		Isource	-11.3mA	-15.6mA	-21.3mA
	CD1=1,CD0=1	Isink	13.9mA	21.0mA	27.9mA
		Isource	-15.0mA	-20.9mA	-28.4mA

Table 638: SiRFatlasV pvhxxx Pad DC Current of Output Driver Type A, C and B (VDD=3.3V±0.3V)

Parameter (Driver Type)			Worst VDD=2.30V T=125°C Process=Slow Isink at 0.20V Isource at 2.10V	Typical VDD=2.50V T=25°C Process=Nominal Isink at 0.20V Isource at 2.30V	Best VDD=2.70V T=-40°C Process=Fast Isink at 0.20V Isource at 2.50V
A, C	CD1=0,CD0=0	Isink	1.4mA	2.2mA	3.1mA
		Isource	-1.5mA	-2.1mA	-2.9mA
	CD1=0,CD0=1	Isink	2.8mA	4.4mA	6.2mA
		Isource	-3.1mA	-4.3mA	-5.9mA
	CD1=1,CD0=0	Isink	4.2mA	6.6mA	9.3mA
		Isource	-4.6mA	-6.5mA	-8.9mA
	CD1=1,CD0=1	Isink	5.7mA	8.9mA	12.4mA
		Isource	-6.2mA	-8.6mA	-11.9mA
B	CD1=0,CD0=0	Isink	2.8mA	4.4mA	6.2mA
		Isource	-3.1mA	-4.3mA	-5.9mA
	CD1=0,CD0=1	Isink	5.7mA	8.9mA	12.4mA
		Isource	-6.2mA	-8.6mA	-11.9mA
	CD1=1,CD0=0	Isink	8.5mA	13.3mA	18.6mA
		Isource	-9.3 mA	-13.0mA	-17.9mA
	CD1=1,CD0=1	Isink	11.4mA	17.8mA	24.8mA
		Isource	-12.4mA	-17.3mA	-23.9mA

Table 639: SiRFatlasV pvhxxx Pad DC Current of Output Driver Type A, C and B (VDD=2.5V±0.2V)

Parameter (Driver Type)			Worst VDD=1.65V T=125℃ Process=Slow Isink at 0.20V Isource at 1.45	Typical VDD=1.80V T=25℃ Process=Nominal Isink at 0.20V Isource at 1.60	Best VDD=1.95V T=-40℃ Process=Fast Isink at 0.20V Isource at 1.75V
A, C	CD1=0,CD0=0	Isink	1.1mA	1.6mA	2.4mA
		Isource	-1.1mA	-1.6mA	-2.3mA
	CD1=0,CD0=1	Isink	2.0mA	3.3mA	4.8mA
		Isource	-2.2mA	-3.2mA	-4.6mA
	CD1=1,CD0=0	Isink	3.0mA	4.9mA	7.3mA
		Isource	-3.4mA	-4.8mA	-6.9mA
	CD1=1,CD0=1	Isink	4.0mA	6.6mA	9.7mA
		Isource	-4.5mA	-6.5mA	-9.2mA
B	CD1=0,CD0=0	Isink	2.0mA	3.3mA	4.8mA
		Isource	-2.2mA	-3.2mA	-4.6mA
	CD1=0,CD0=1	Isink	4.0mA	6.6mA	9.7mA
		Isource	-4.5mA	-6.5mA	-9.2mA
	CD1=1,CD0=0	Isink	6.0mA	9.9mA	14.6mA
		Isource	-6.8mA	-9.7mA	-13.8mA
	CD1=1,CD0=1	Isink	8.1mA	13.2mA	19.4mA
		Isource	-9.1mA	-13.0mA	-18.5mA

Table 640: SiRFatlasV pvhxxx Pad DC Current of Output Driver Type A, C and B (VDD=1.8V±0.15V)

Parameter (Driver Type)			Worst VDD=2.30V T=125°C Process=Slow Isink at 0.20V Isource at 2.10V	Typical VDD=2.50V T=25°C Process=Nominal Isink at 0.20V Isource at 2.30V	Best VDD=2.70V T=-40°C Process=Fast Isink at 0.20V Isource at 2.50V
A, C	CD1=0,CD0=0	Isink	2.1mA	3.3mA	4.6mA
		Isource	-1.6mA	-2.3mA	-3.2mA
	CD1=0,CD0=1	Isink	4.2mA	6.6mA	9.3mA
		Isource	-3.3mA	-4.7mA	-6.5mA
	CD1=1,CD0=0	Isink	6.3mA	9.9mA	14.0mA
		Isource	-5.0mA	-7.0mA	-9.7mA
	CD1=1,CD0=1	Isink	8.4mA	13.2mA	18.7mA
		Isource	-6.7mA	-9.4mA	-13.0mA
B	CD1=0,CD0=0	Isink	4.2mA	6.6mA	9.3mA
		Isource	-3.3mA	-4.7mA	-6.5mA
	CD1=0,CD0=1	Isink	8.4mA	13.2mA	18.7mA
		Isource	-6.7mA	-9.4mA	-13.0mA
	CD1=1,CD0=0	Isink	12.7mA	19.8mA	28.1mA
		Isource	-10.1 mA	-14.1mA	-19.5mA
	CD1=1,CD0=1	Isink	16.9mA	26.4mA	37.5mA
		Isource	-13.5mA	-18.9mA	-26.0mA

Table 641: SiRFatlasV pvmxxx Pad DC Current of Output Driver Type A, C and B (VDD=2.5V±0.2V)

Parameter (Driver Type)			Worst VDD=1.65V T=125℃ Process=Slow Isink at 0.20V Isource at 1.45	Typical VDD=1.80V T=25℃ Process=Nominal Isink at 0.20V Isource at 1.60	Best VDD=1.95V T=-40℃ Process=Fast Isink at 0.20V Isource at 1.75V
A, C	CD1=0,CD0=0	Isink	1.4mA	2.4mA	3.7mA
		Isource	-1.2mA	-1.7mA	-2.5mA
	CD1=0,CD0=1	Isink	2.9mA	4.9mA	7.5mA
		Isource	-2.4mA	-3.5mA	-5.0mA
	CD1=1,CD0=0	Isink	4.3mA	7.4mA	11.3mA
		Isource	-3.7mA	-5.3mA	-7.5mA
	CD1=1,CD0=1	Isink	5.8mA	9.8mA	15.1mA
		Isource	-4.9mA	-7.1mA	-10.0mA
B	CD1=0,CD0=0	Isink	2.9mA	4.9mA	7.5mA
		Isource	-2.4mA	-3.5mA	-5.0mA
	CD1=0,CD0=1	Isink	5.8mA	9.8mA	15.1mA
		Isource	-4.9mA	-7.1mA	-10.9mA
	CD1=1,CD0=0	Isink	8.7mA	14.8mA	22.7mA
		Isource	-7.4mA	-10.6mA	-15.1mA
	CD1=1,CD0=1	Isink	11.6mA	19.7mA	30.3mA
		Isource	-9.9mA	-14.2mA	-20.1mA

Table 642: SiRFatlasV pvmxxx Pad DC Current of Output Driver Type A, C and B (VDD=1.8V±0.15V)

POWER MANAGEMENT UNIT

POWER MANAGEMENT UNIT

Overview

AtlasV is integraged Power Managent Unit(PMU), which includes two step-down converters, and four LDOs.

Feature List

- Two switching DC/DC for core (1000mA) and DRAM (500mA)
- One high PSRR and low noise 300mA LDO for I/O and peripheral
- One high PSRR and low noise 150mA LDO for analog power
- One high PSRR and low noise 10mA LDO for PLL
- One high PSRR and low noise 100mA LDO for RF
- One high PSRR and low noise 10mA LDO for RTC

Functional block diagram

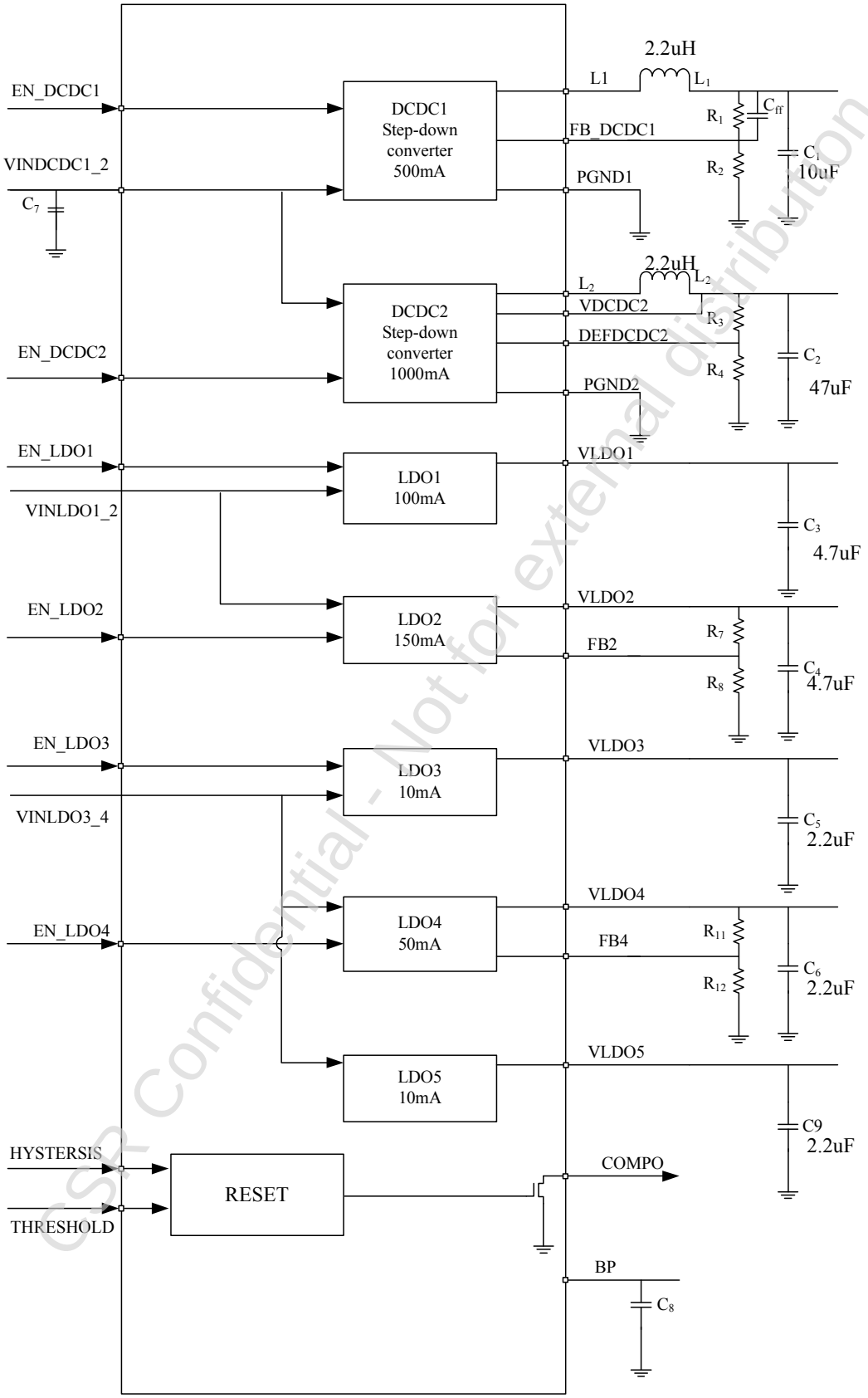


Figure 105 PMU Module Diagram

Pin Description

The fallowing table was pin description about PMU module

Pin Name	I/O	Description
VINDCDC1_2	I	Power input for switching DCDC 1 and 2.
VINLDO1_2	I	Power input for LDO1 and LDO2; Power for internal switching DC/DC and LDOs
VINLDO3_4	O	Power input for LDO 3 and 4
L1	O	Switch pin for DCDC1, connect to inductor
PGND1	I	GND for DCDC1
FB_DCDC1	I	Input to adjust output voltage of DCDC1 through external resister divider
EN_DCDC1	I	Enable control of DCDC1, active high
L2	O	Switch pin for DCDC2, connect to inductor
PGND2	I	GND for DCDC2
DEFDCDC2	I	Input to adjust output voltage of DCDC2 through external resister divider
VDCDC2	I	Feedback voltage sense input, connect directly to the output of DCDC2
EN_DCDC2	I	Enable control of DCDC2, active high
VLDO1	O	Power output of LDO1
EN_LDO1	I	Enable control of LDO1, active high
VLDO2	O	Power output of LDO2
FB2	I	LDO2 feedback input for the external voltage divider
EN_LDO2	I	Enable control of LDO2, active high
VLDO3	O	Power output of LDO3
EN_LDO3	I	Enable control of LDO3, active high
VLDO4	O	Power output of LDO4
FB4	I	LDO4 feedback input for the external voltage divider
EN_LDO4	I	Enable control of LDO4, active high
VLDO5	O	Power output of LDO5
BP	I	Input for bypass capacitor for internal reference, bypass capacitor needed
THRESHOLD	I	Voltage comparator input, connect to external resister divider
HYSTERESIS	I	Input for hysteresis on threshold, connect to external resister divider
COMPO	O	Voltage comparator output, open-drain output, active low

Table 643 PMU Pin description

Absolute Maximum Ratings

Parameter		Min.	Max.
VINDCDC1_2	Power input for switching DCDC1/2. (V)	-0.3	6.0
VINLDO1_2	Power input for LDO1 and LDO2; Power for internal switching DC/DC and LDOs (V)	-0.3	6.0
VINLDO3_4	Power input for LDO 3 and 4 (V)	-0.3	6.0

Table 644: Absolute Maximum Ratings

NOTE – NOTE - Absolute maximum ratings are stress ratings only, functional operations tested to the maximum stress capacity are not guaranteed. Stresses beyond those listed in the table above may affect device reliability and cause permanent damage.

Recommended Operating Conditions

Parameter		Min.	Typical	Max.	Unit
VINDCDC1_2	Input voltage for DCDC1_2	3.1		5.5	V
VINLDO1_2	Input voltage for LDO1_2	3.1		5.5	V
VINLDO3_4	Input voltage for LDO3_4	3.1		5.5	V

Table 645: Recommended Operating Conditions

NOTE – NOTE - The conditions in the table above are tested and recommended (unless specified as "to be qualified"). Any device operations not listed in this table may not be guaranteed.

ELECTRICAL CHARACTERISTICS

LDO1 Electrical Characteristics

Operating Conditions (unless otherwise specified) VCCL = 3.6V, C_{REFO} = 1.0µF, T_a = 25℃

Symbol	Parameter	Condition	Min	Typ	Max	Units
VOUT1	Output Voltage	Iload=50uA ~ 150mA VOUT1+0.3V≦VCCL≦5.5V	-3%	3.3	+3%	V
IOUT1	Output Current				150	mA
ILIM1	Current Limit	VOUT1=3.3V		400		mA
VDRP1	Drop-out Voltage	IOUT1=150mA		300		mV
ΔVOUT1 ΔVCCL	Line Regulation	VOUT1+0.3V≦VCCL≦5.5V IOUT1=75mA		3		mV
ΔVOUT1 ΔIOUT1	Load Regulation	50µA < IOUT1 < 150mA		25		mV
RR1	Ripple Rejection	f=10Hz-10kHz, C_{out}=1.0µF IOUT1=30mA, VOUT≦3.0V		60		dB
ISS1	Supply Current	LDO1EXON = "H" (ON)		70		uA
		LDO1EXON = "L" (OFF)			1	
EN1	Output Noise	BW=100Hz-100kHz, IOUT1=75mA		80		µVrms

Table 646 LDO1 Electrical Characteristics

LDO2 Electrical Characteristics

Operating Conditions (unless otherwise specified) VCCL = 3.6V, C_{REFO} = 2.2 µF, T_a = 25℃

Symbol	Parameter	Condition	Min	Typ	Max	Units
VOUT2	Output Voltage range	Iload=50uA ~ 300mA VOUT2+0.3V≦VCCL≦5.5V	2.5		3.3	V
VFB2	Reference Voltage		-3%	1	+3%	V
IOUT2	Output Current				300	mA
ILIM2	Current Limit	VOUT2=3.3V		500		mA
VDRP2	Drop-out	IOUT1=300mA		300		mV

	Voltage					
ΔVOUT2 ΔVCCL	Line Regulation	VOUT2+0.3V≦VCCL≦5.5V IOUT2=150mA		3		mV
ΔVOUT2 ΔIOUT2	Load Regulation	50μA < IOUT2 < 300mA		35		mV
RR2	Ripple Rejection	f=10Hz-10kHz, C_{out}=1.0μF IOUT2=75mA, VOUT≦3.0V		60		dB
ISS2	Supply Current*1	LDO2EXON = "H" (ON)		70		uA
		LDO2EXON = "L" (OFF)			1	
EN2	Output Noise	BW=100Hz-100kHz, IOUT2=150mA,VOUT2=3.0V		80		μVrms

Table 647 LDO2 Electrical Characteristics

Note*1: Supply current does not include the consumption current of external resistors

Output Voltage	R1	R2	Notes
3.30V	300kΩ	130kΩ	
3.00V	300kΩ	150kΩ	
2.85V	240kΩ	130kΩ	
2.80V	360kΩ	200kΩ	
2.50V	300kΩ	200kΩ	

Table 648 LDO2 External Resistors Table

LDO3 Electrical Characteristics

Operating Conditions (unless otherwise specified) VCCL = 3.6V, C_{REFO} = 1.0μF, T_a = 25℃

Symbol	Parameter	Condition	Min	Typ	Max	Units
VOUT3	Output Voltage	Iload=50uA ~ 10mA 3.1V≦VCCL≦5.5V	-3%	1.2	+3%	V
IOUT3	Output Current				10	mA
ILIM3	Current Limit	VOUT1=1.2V		250		mA
VDRP3	Drop-out Voltage	IOUT3=10mA		300		mV
ΔVOUT3 ΔVCCL	Line Regulation	VOUT3+0.3V≦VCCL≦5.5V IOUT3=5mA		3		mV
ΔVOUT3 ΔIOUT3	Load Regulation	50μA < IOUT3 < 10mA		10		mV
RR3	Ripple Rejection	f=10Hz-10kHz, C_{out}=1.0μF IOUT3=10mA,		60		dB
ISS3	Supply Current	LDO3EXON = "H" (ON)		50		uA
		LDO3EXON = "L" (OFF)			1	
EN3	Output Noise	BW=100Hz-100kHz, IOUT3=10mA		80		μVrms

Table 649 LDO3 Electrical Characteristics

LDO4 Electrical Characteristics

Operating Conditions (unless otherwise specified) VCCL = 3.6V, C_{REFO} = 1.0 μF, T_a = 25℃

Symbol	Parameter	Condition	Min	Typ	Max	Units
VOUT4	Output Voltage range	Iload=50uA ~ 100mA VOUT4+0.3V≦VCCL≦5.5V	2.5		3.3	V
VFB4	Reference Voltage		-3%	1	+3%	V

IOUT4	Output Current				100	mA
ILIM4	Current Limit	VOUT4=3.3V		400		mA
VDRP4	Drop-out Voltage	IOUT4=100mA		300		mV
ΔVOUT4 ΔVCCL	Line Regulation	VOUT4+0.3V≦VCCL≦5.5V IOUT4=50mA		3		mV
ΔVOUT4 ΔIOUT4	Load Regulation	50µA < IOUT4 < 100mA		25		mV
RR4	Ripple Rejection	f=10Hz-10kHz, C_{out}=1.0µF IOUT4=30mA, VOUT≦3.0V		60		dB
ISS4	Supply Current*1	LDO4EXON = "H" (ON)		70		uA
		LDO4EXON = "L" (OFF)			1	
EN4	Output Noise	BW=100Hz-100kHz, IOUT4=50mA,VOUT4=3.0V		80		µVrms

Table 650 LDO4 Electrical Characteristics

Note*1: Supply current does not include the consumption current of external resistors

Output Voltage	R1	R2	Notes
3.30V	300kΩ	130kΩ	
3.00V	300kΩ	150kΩ	
2.85V	240kΩ	130kΩ	
2.80V	360kΩ	200kΩ	
2.50V	300kΩ	200kΩ	

Table 651 LDO4 External Resistors Table

LDO5 Electrical Characteristics

Operating Conditions (unless otherwise specified) VCCL = 3.6V, C_{REFO} = 1.0µF, T_a = 25℃

Symbol	Parameter	Condition	Min	Typ	Max	Units
VOUT5	Output Voltage	Iload=50uA ~ 10mA Vout5+0.3V≦VCCL≦5.5V	-3%	3.3	+3%	V
IOUT5	Output Current				10	mA
ILIM5	Current Limit	VOUT1=3.3V		250		mA
VDRP5	Drop-out Voltage	IOUT5=10mA		300		mV
ΔVOUT5 ΔVCCL	Line Regulation	VOUT5+0.3V≦VCCL≦5.5V IOUT5=5mA		3		mV
ΔVOUT5 ΔIOUT5	Load Regulation	50µA < IOUT5 < 10mA		20		mV
ISS5	Supply Current*1	ON		1	3	uA
IRR5	Reverse Current	VOUT5=3.3V&VCCL=0V			TBD	uA

Table 652 LDO5 Electrical Characteristics

Note*1: The consumption current of the reverse protection is not included.

Step-down DC/DC Converter1 Electrical Characteristics

Operating Conditions (unless otherwise specified) VCCA, VCCP = 3.6V, T_a = 25℃ L1=2.2uH, C_{OUT}=47uF

Symbol	Parameter	Condition	Min	Typ	Max	Units
DVOUT1	Output Voltage Range	DVOUT1+0.5≦VCCA,VCCP≦5.5V	0.9	1.8	3.3	V
DIOUT1	Output Current				500	mA

FREQ1	Frequency			2.0		MHz
DISS1	Consumption Current	VCCA=VCCP=VFB=3.6V DIOUT1=0mA, no switching		70		uA
DIOFF1	Standby Current	OFF state			1	uA
DILIM1	Limit detection Current		800			mA
VFB1	FB Voltage	VCCA=VCCP=3.6V,DIOUT1=1mA	-1.5%	0.6	+1.5%	V
ΔVFB1 ΔVCC	FB Line Regulation	DVOUT1+0.5≦VCCA,VCCP≦5.5V DIOUT=DIOUTmax / 2		5		mV
ΔVFB1 ΔDIOUT1	FB Load Regulation	1mA≦DIOUT1≦500mA		20		mV
t_r	Soft-start Time			500		us

Table 653 Step-down DC/DC Converter1 Electrical Characteristics

Output Voltage	R1	R2	Cf	Notes
2.5V	510kΩ	160kΩ	47pF	
1.8V	300kΩ	150kΩ	47pF	
1.2V	330kΩ	330kΩ	47pF	

Table 654 Step-down DC/DC Converter1 External Resistors Table

Step-down DC/DC Converter2 Electrical Characteristics

Operating Conditions (unless otherwise specified) VCCA, VCCP = 3.6V, T_a = 25℃ L2=2.2uH, C_{OUT}=47uF

Symbol	Parameter	Condition	Min	Typ	Max	Units
DVOUT2	Output Voltage Range	DVOUT2+1.0≦VCCA,VCCP≦5.5V	0.9	1.2	3.3	V
DIOUT2	Output Current				1000	mA
FREQ2	Frequency			2.0		MHz
DISS2	Consumption Current	VCCA=VCCP=VFB=3.6V DIOUT2=0mA, no switching		70		uA
DIOFF2	Standby Current	OFF state			1	uA
DILIM2	Limit detection Current		1200			mA
VFB2	FB Voltage	VCCA=VCCP=3.6V,DIOUT2=1mA	-1.5%	0.6	+1.5%	V
ΔVFB2 ΔVCC	FB Line Regulation	DVOUT2+1.0≦VCCA,VCCP≦5.5V DIOUT=DIOUTmax / 2		10		mV
ΔVFB2 ΔDIOUT2	FB Load Regulation	1mA≦DIOUT2≦1000mA		25		mV
t_r	Soft-start Time			500		us

Table 655 Step-down DC/DC Converter2 Electrical Characteristics

Output Voltage	R1	R2	Notes
2.5V	510kΩ	160kΩ	
1.8V	300kΩ	150kΩ	
1.2V	330kΩ	330kΩ	

Table 656 Step-down DC/DC Converter2 External Resistors Table

Table of Graphs

Base on RECOMMENDED OPERATING CONDITIONS, TA = 25℃, L = 2.2 µH, Co = 47 µF. (unless otherwise noted).		
Efficiency converter 1	vs Output current	Figure2
Efficiency converter 2	vs Output current	Figure3
DCDC1 Output voltage ripple	Iout=1mA	Figure4
DCDC1 Output voltage ripple	Iout=500mA	Figure5
DCDC1 startup timing		Figure6
DCDC1 load transient response		Figure8
DCDC1 line transient response		Figure10
LDO2 load transient response		Figure11
LDO2 line transient response		Figure12

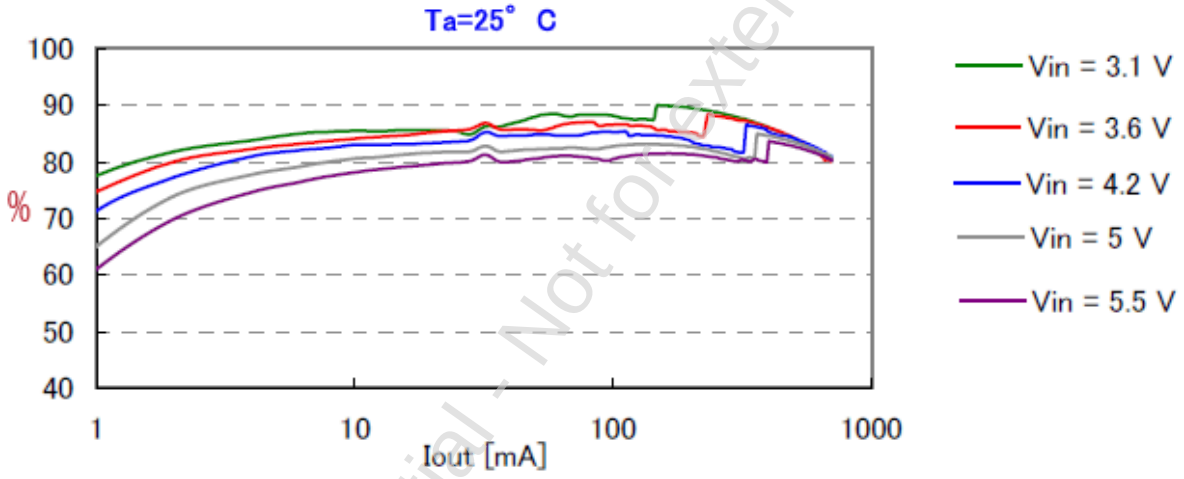


Figure 106 DCDC1 effiency

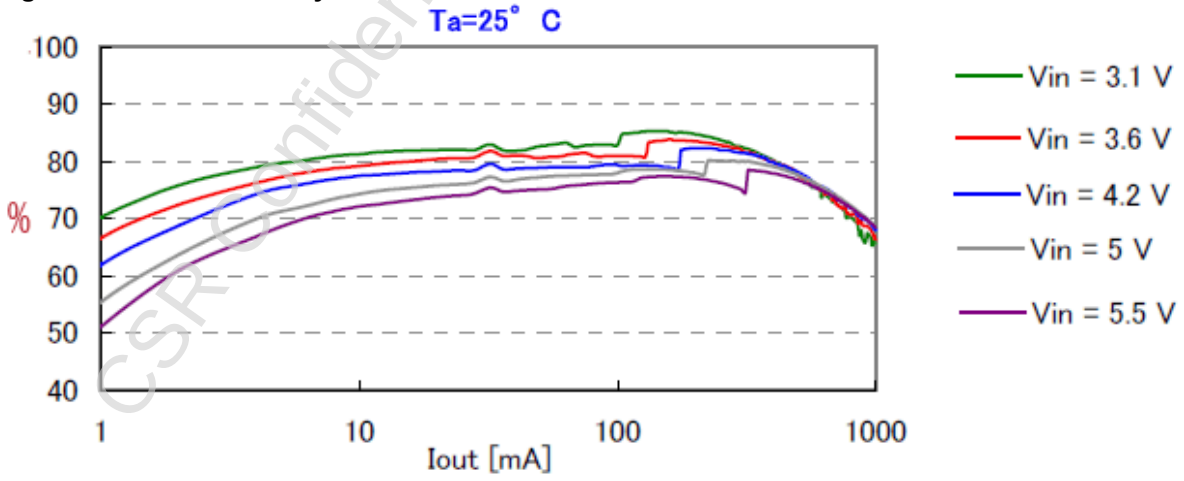


Figure 107 DCDC2 effiency

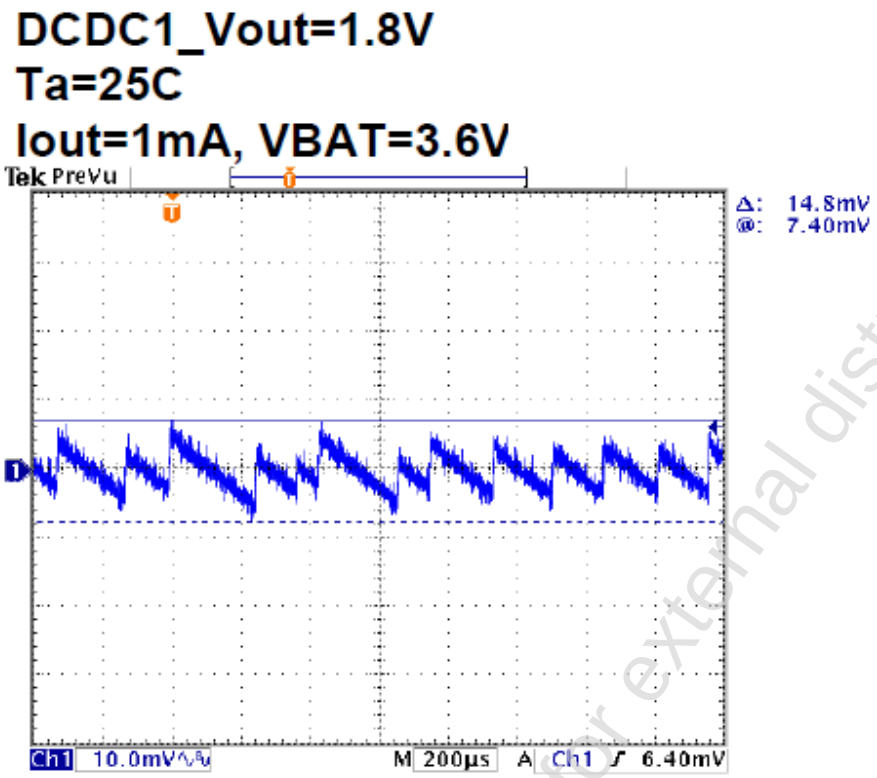


Figure 108 DCDC1 Output ripple at 1mA output current.

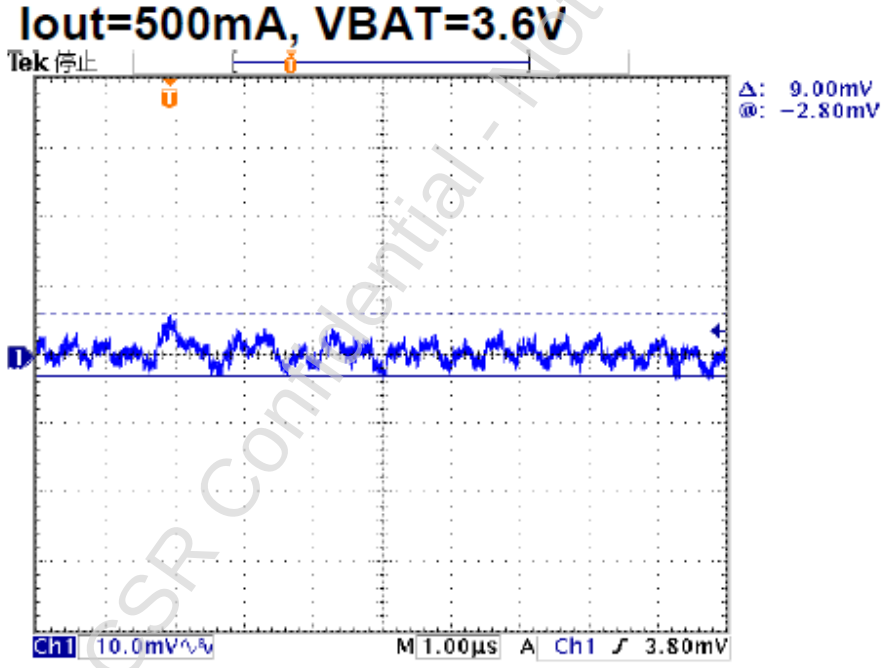


Figure 109 DCDC1 Output ripple at 500mA output current.

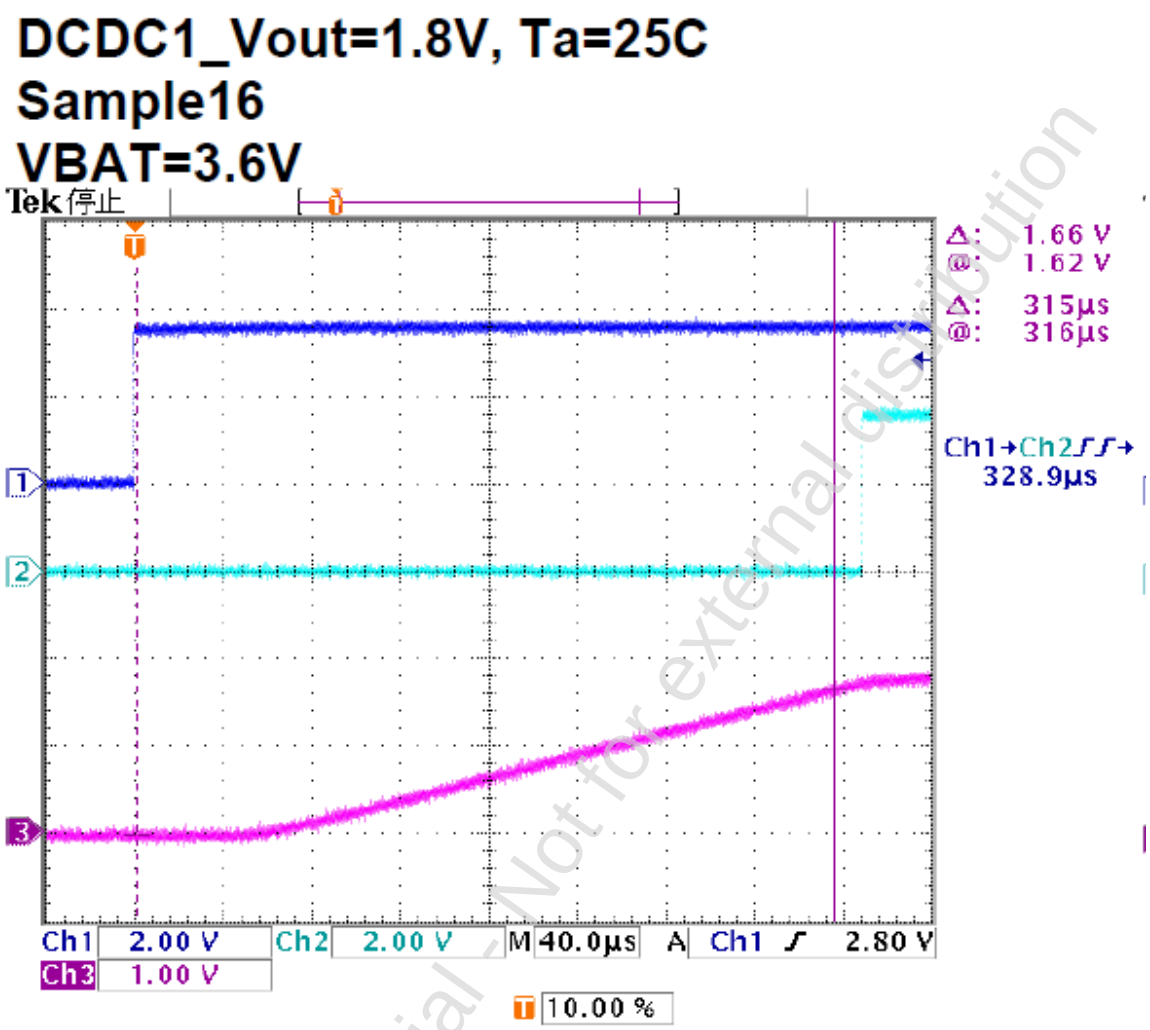


Figure 110 DCDC1 start up time

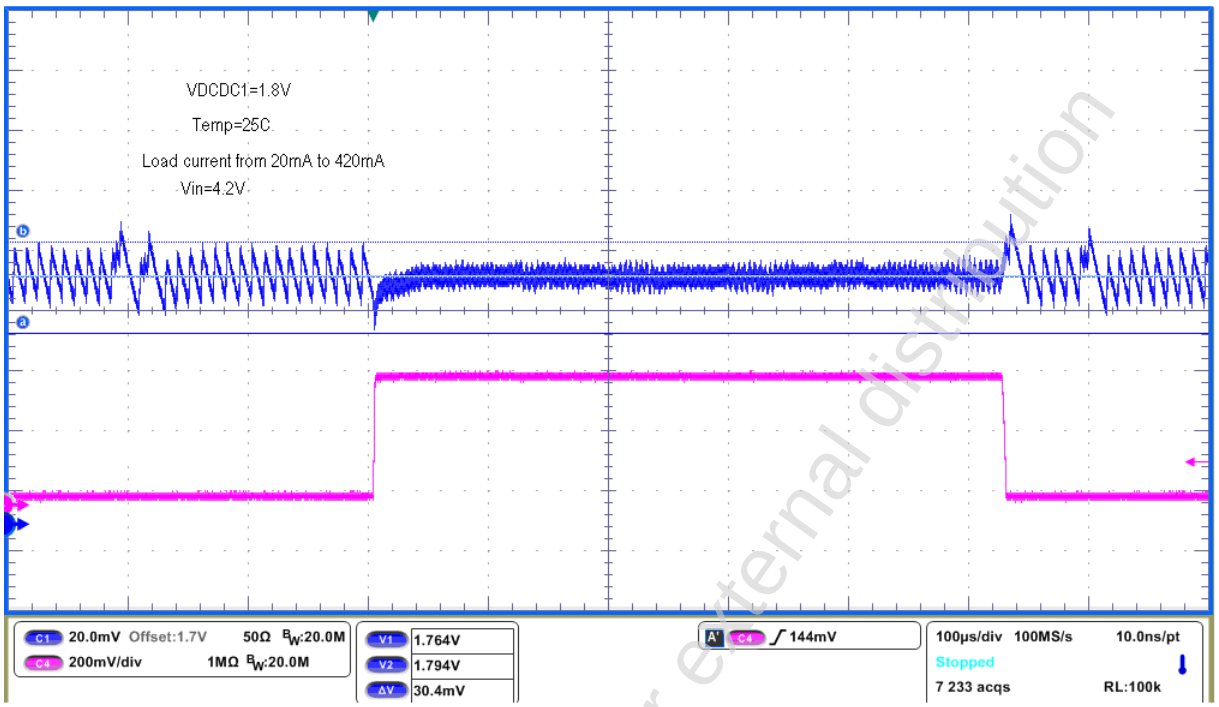


Figure 111 DCDC1 load transient response

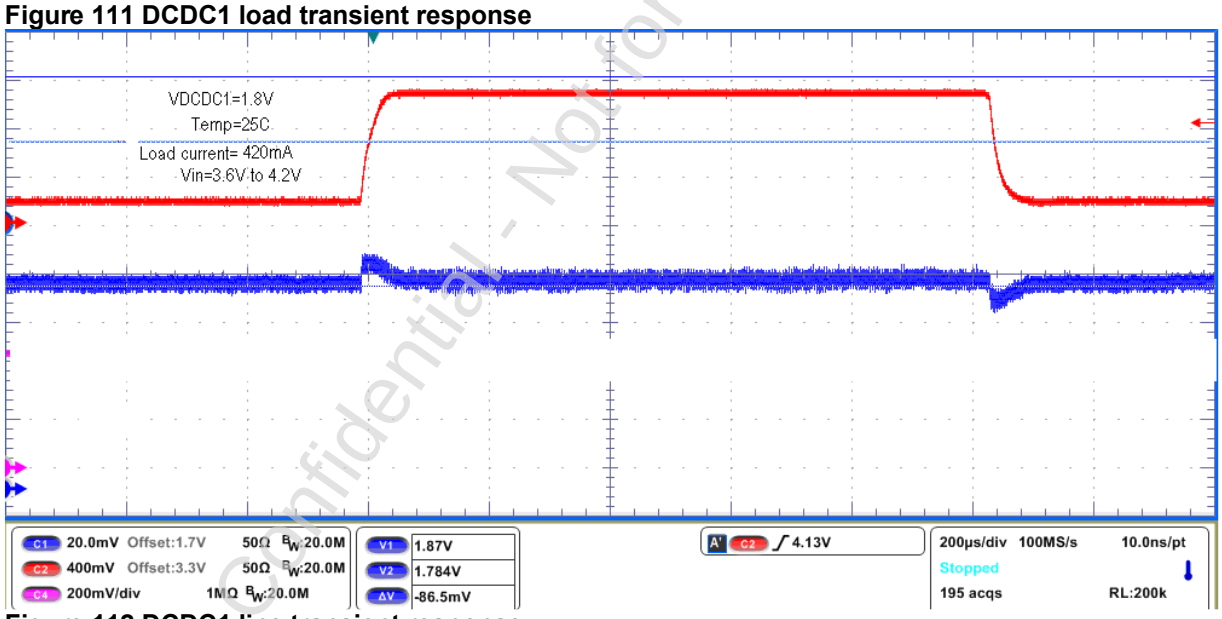


Figure 112 DCDC1 line transient response

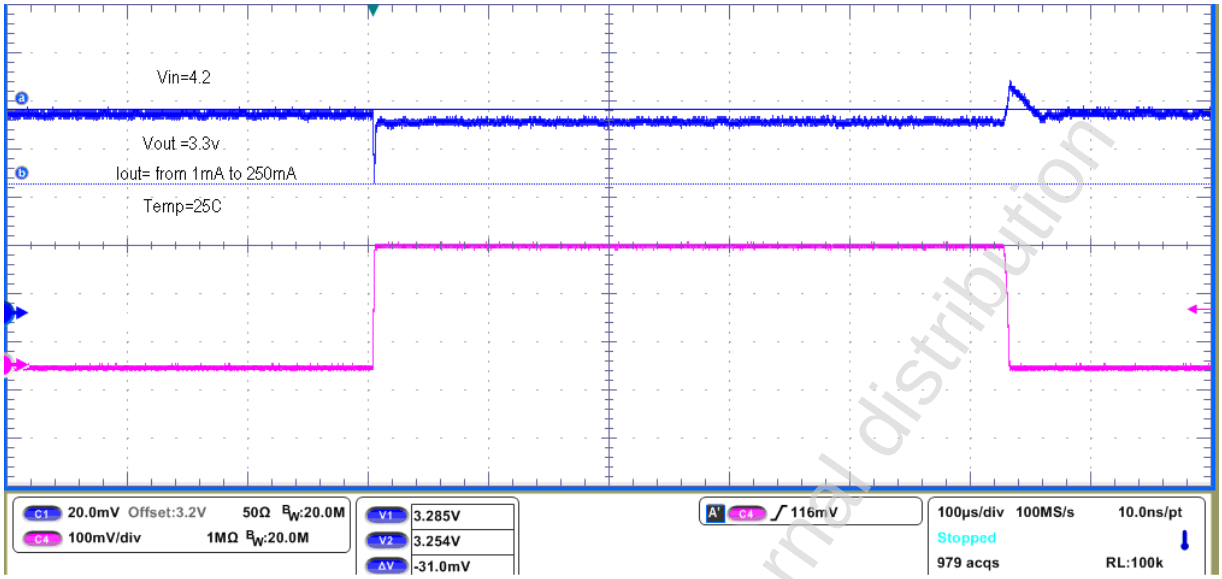


Figure 113 LDO2load transient response

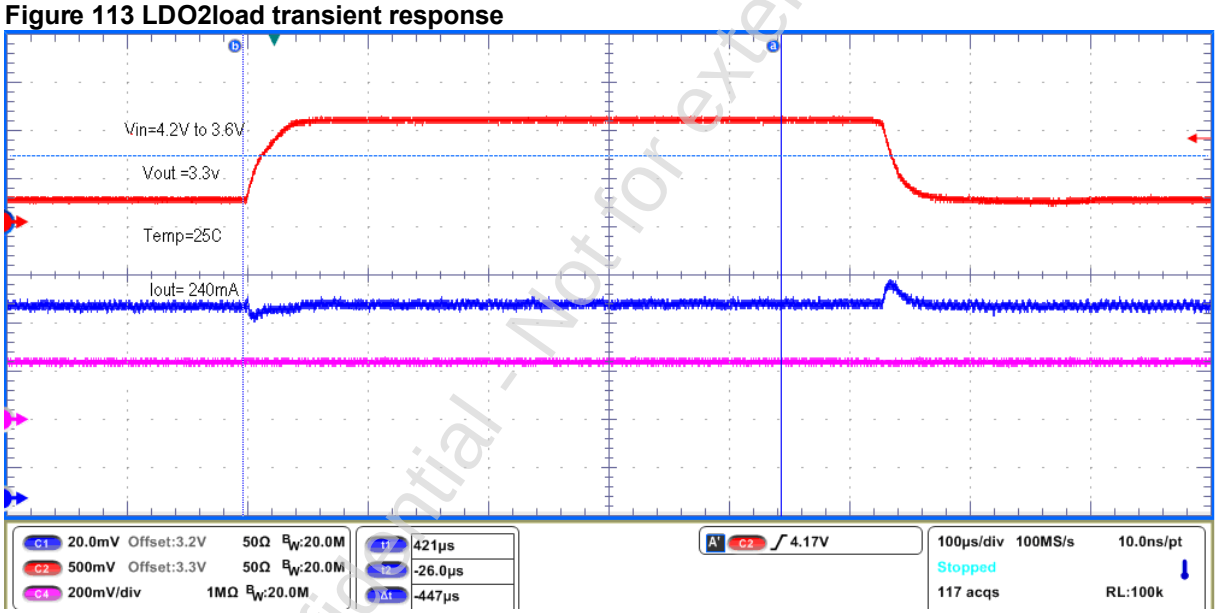


Figure 114 LDO2 Line transient response

COMPO Setting

Atlas5 PMU module contain a comparator that are used to supervise a voltage connected to an external voltage divider, and generate a pulse signal if the voltage is lower than the threshold.The rising edge is delayed by 100 ms at the open drain COMPO output. The values for the external resistors R13 to R15 are calculated as follows:

V_L = lower voltage threshold;

V_H = higher voltage threshold;

V_{REF} = reference voltage (1 V)

$$R_{13} + R_{14} = R_{15} * (\frac{V_H}{V_{ref}} - 1)$$

$$R_{14} = R_{15} * (\frac{V_H}{V_L} - 1)$$

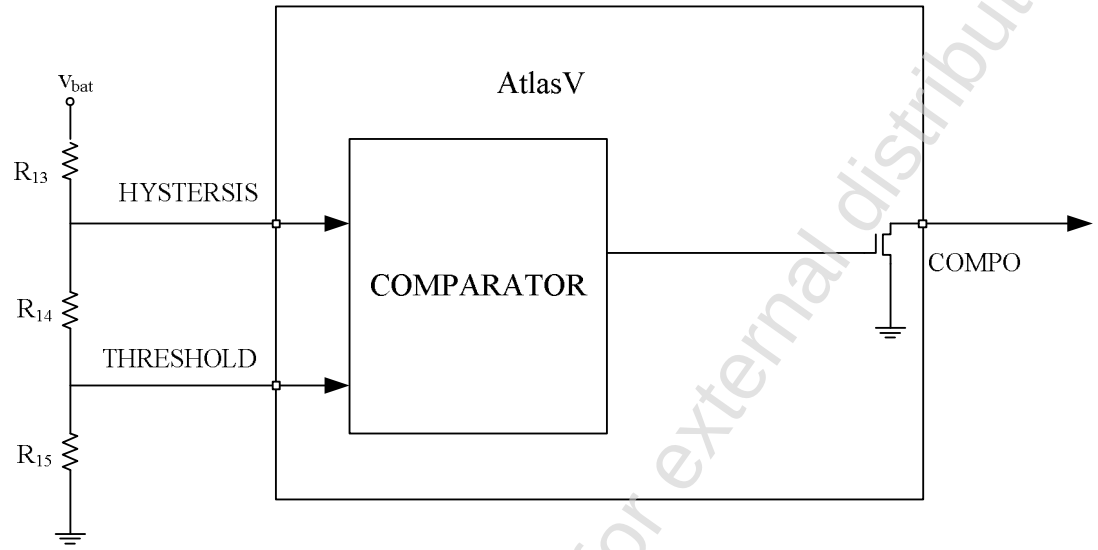


Figure 115 AtlasV COMPO setting

ELECTRICAL AND TIMING CHARACTERISTICS

NOTE – The electrical and timing characteristics in this section are only for SoC part. PMU module is not included.

Absolute Maximum Ratings

Parameter		Min.	Max.
VDD_Core	Core power (V)	-0.3	1.8
VDDIO_N	I/O power (NAND interface) (V)	-0.3	4.6
VDDIO_L	I/O power (LCD interface) (V)	-0.3	4.6
VDDIO	I/O power. (V)	-0.3	4.6
VDDIO_MEM	Memory I/O power. (V)	-0.3	3.6
VDD_PLL0/1	PLL power (V)	-0.3	1.8
VDDIO_RTC	I/O power for RTC and power input of LDO for RTC digital power (V)	-0.3	4.6
VDDIO_OSC	Crystal oscillator power. (V)	-0.3	4.6
VDD_USB	USB PHY core power. (V)	-0.3	1.8
VDDA_USB	Analog power for USB. (V)	-0.3	4.6
VDD_TSC	Digital power for internal touch screen controller. (V)	-0.3	1.8
VDDA3V3_TSC	Analog power for touch screen controller.(V)	-0.3	4.6
VREF_ADC	Reference voltage for the ADC (V)	-0.3	4.6
VDDIO_DAC	MoDAC power (V)	-0.3	4.6
Ts	Storage ambient temperature (°C)	-65	125

Table 657: Absolute Maximum Ratings

NOTE – Absolute maximum ratings are stress ratings only, functional operations tested to the maximum stress capacity are not guaranteed. Stresses beyond those listed in the table above may affect device reliability and cause permanent damage.

Recommended Operating Conditions

Parameter		Min.	Typical	Max.	Comments
VDD_Core	Core power (V)	1.14	1.20	1.26	
VDDIO_N	I/O power (NAND interface) (V)	3.0	3.3	3.6	-
		1.7	1.8	1.9	
VDDIO_L	I/O power (LCD interface) (V)	3.0	3.3	3.6	-
		1.7	1.8	1.9	
VDDIO	I/O power. (V)	3.0	3.3	3.6	-
VDDIO_MEM	Memory I/O power. (V)	1.7	1.8	1.9	
VDD_PLL0/1	PLL power (V)	1.14	1.2	1.26	-
VDDIO_RTC	I/O power for RTC and power input of LDO for RTC digital power (V)	2.6	3.3	3.6	-
VDDIO_OSC	Crystal oscillator power. (V)	3.0	3.3	3.6	-
VDD_USB	USB PHY core power. (V)	1.14	1.2	1.26	-
VDDA_USB	Analog power for USB. (V)	3.135	3.3	3.465	-
VDD_TSC	Digital power for internal touch screen controller. (V)	1.14	1.20	1.26	-
VDDA3V3_TSC	Analog power for touch screen controller.(V)	3.0	3.3	3.6	-
VREF_ADC	Reference voltage for the ADC (V)	3.0	3.3	3.6	-
VDDIO_DAC	MoDAC power (V)	3.0	3.3	3.6	-
Vinos	Input voltage overshoot (V)	-	0	0.3	-
Vinus	Input voltage undershoot (V)	-	0	0.3	-
To	Operation ambient temperature (˚C)	-20	25	70	Industry grade is TBD
Tj	Internal transistor junction temperature (˚C)	-20	25	125	-

Table 658: Recommended Operating Conditions

NOTE – The conditions in the table above are tested and recommended (unless specified as “to be qualified”). Any device operations not listed in this table may not be guaranteed.

Power-On and Power-Off Sequences

There are no requirements for the power-off sequence. Customers can decide which voltage is to be turned off according to the application requirements.

While the stable time of RTC clock can be affected by external element such as crystal and capacitor. So the limit of stable time can be guaranteed by choice of proper external element.

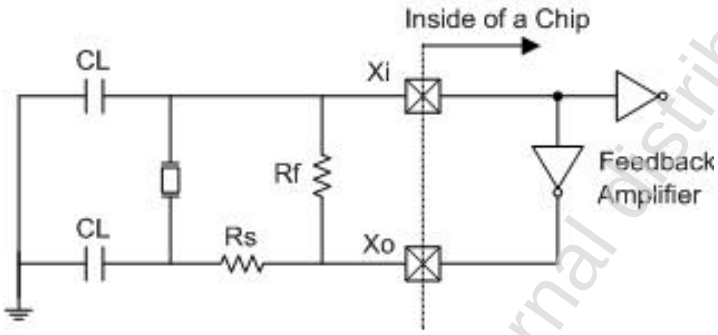


Figure 116: External Component of RTC Clock

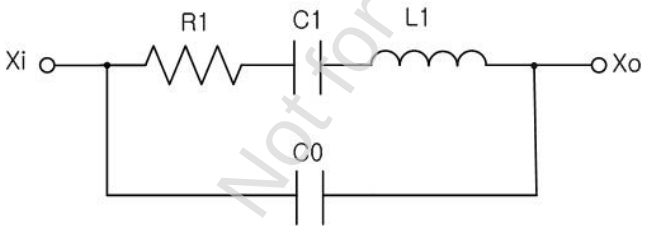


Figure 117: Quartz Crystal Equivalent Circuit

Target Frequency		32 kHz
Crystal parasitic	R1	< 80 kΩ (max.)
	C0	< 2 pF
External components	Rs	0
	Rf	5 MΩ~15 MΩ
	CL	10 pF ~ 25 pF

Table 659: RTC Clock External Component Selection Guide

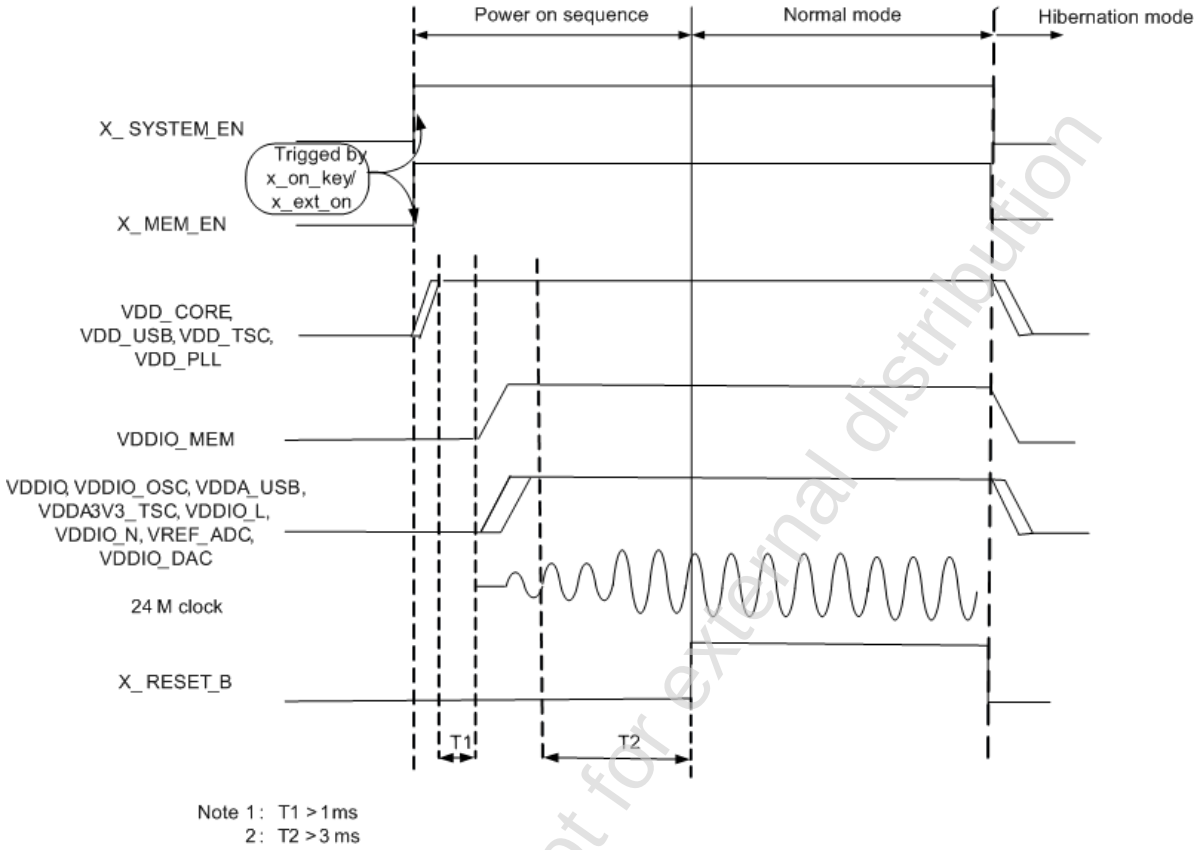


Figure 118: Power-On Sequence of Hibernation Mode

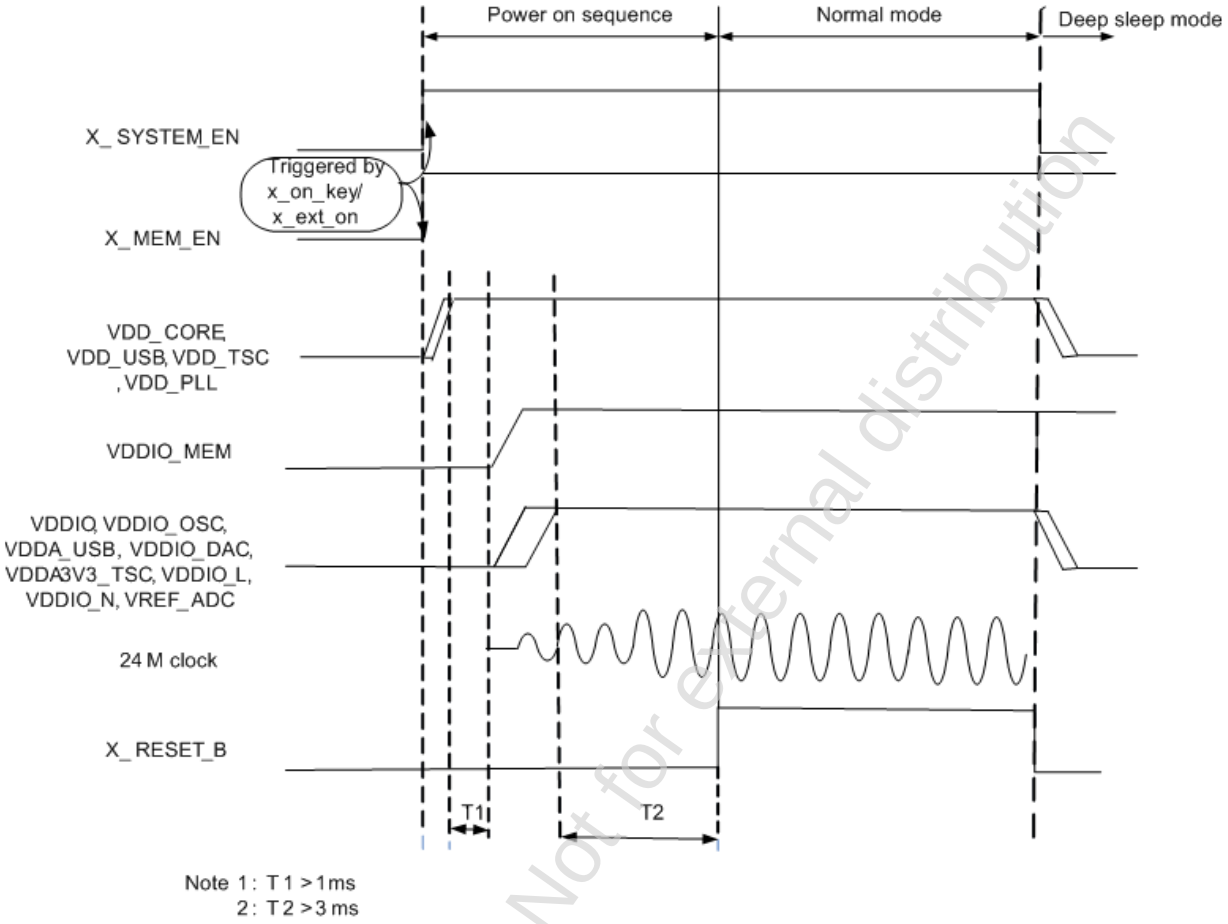


Figure 119: Power-On Sequence of Deep-Sleep Mode

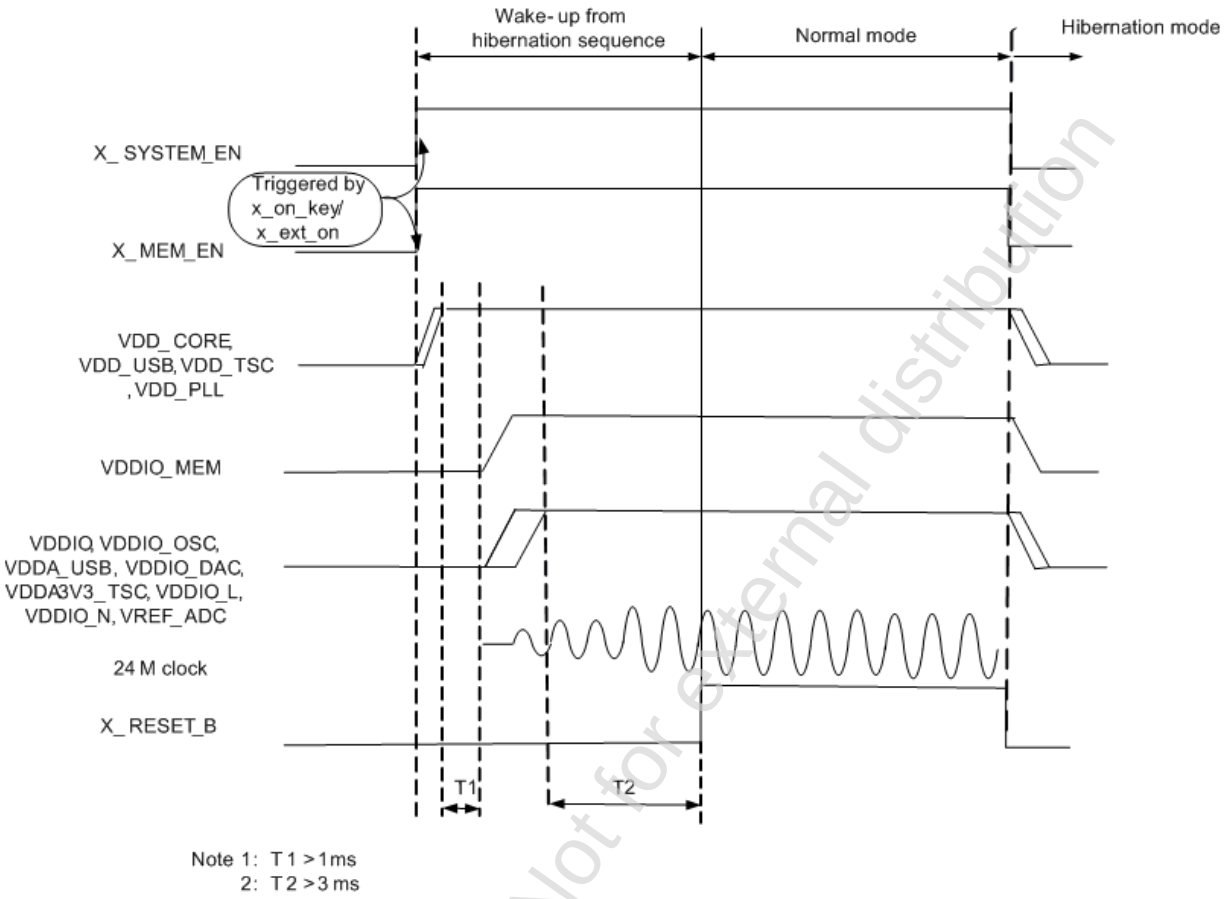


Figure 120: Power-On Sequence of Hibernation Wake-Up Process

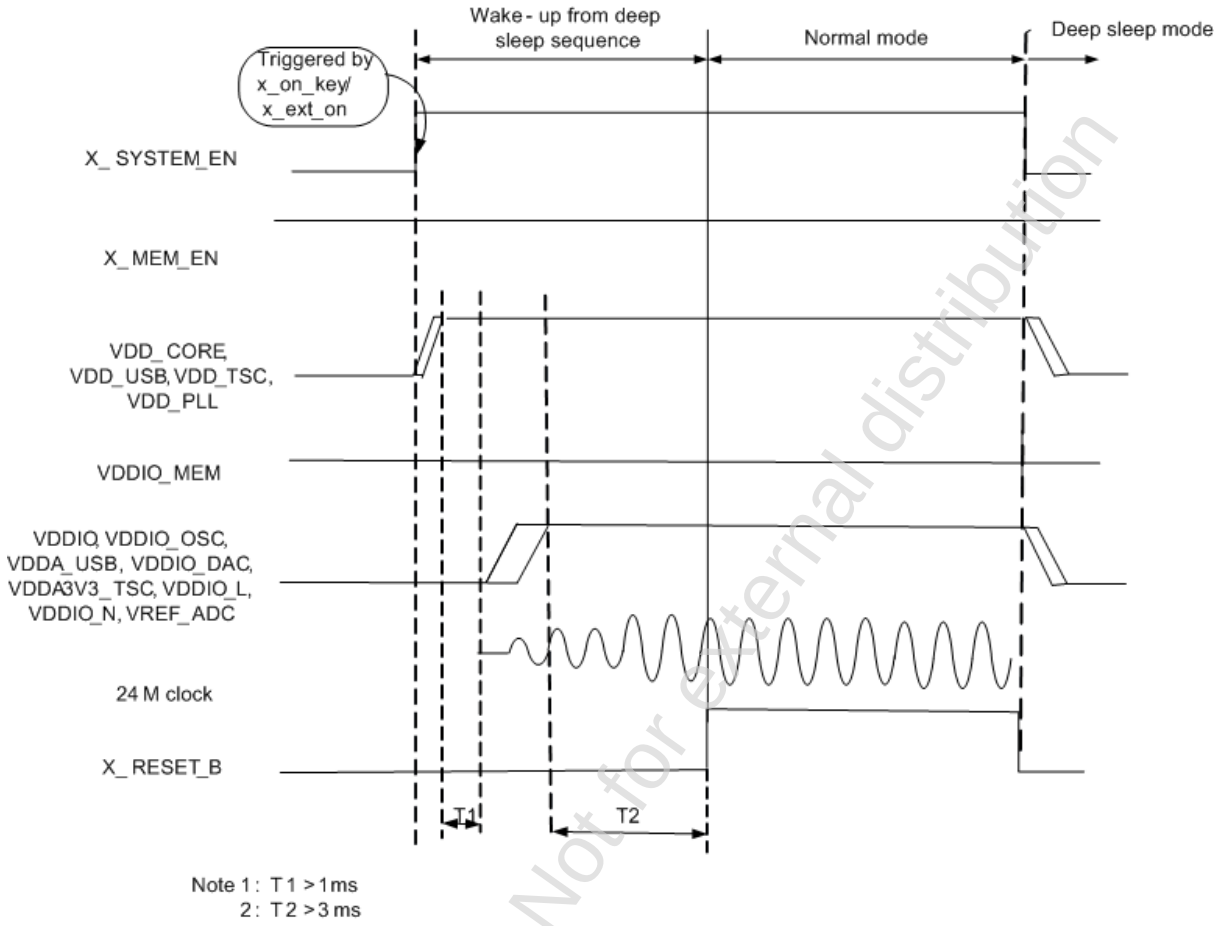


Figure 121: Power-On Sequence of Deep-Sleep Wake-Up Process

Thermal Characteristics

Major thermal dissipation paths can be illustrated as follows:

- T_J: the maximum junction temperature
- T_A: the ambient or environment temperature
- T_C: the maximum compound surface temperature
- T_B: the maximum surface temperature of PCB bottom
- P: total input power

The thermal parameter can be defined as the following:
1. Junction to ambient thermal resistance:

$$\Theta_{JA} = \frac{T_J - T_A}{P} \qquad (1)$$

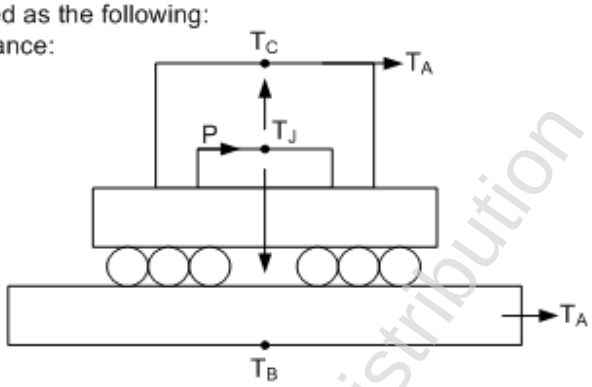


Figure 122: Junction to Ambient Thermal Resistance

Thermal Performance

Thermal measurement environment and thermal test board are based on the JEDEC standard JESD51-2 and JESD51-9.

θ_{JA} = 33.9°C/W

Reliability Test Specifications

The reliability test specifications of SiRFatlasV follow the JEDEC testing standard as part of product qualification plan. The following table lists the detailed testing items and conditions.

Qualification Stress Test Name	Test Method	Fail/ Sample Size	Results
ESD-HBM	EIA/JESD22-A114-A, 2.0kV	0/3	Passed
ESD-CDM	JESD22-C101, 500V	0/3	Passed
ESD-CDM	JESD22-C101, 750V corner pins	0/3	Passed
ESD-MM	JESD-22-A115, 200V	0/3	Passed
Latch-Up	EIA/JES78; 25C, 85C, Inom+100mA / 1.5x Vopmax	0/6	Passed
High Temp Operating Life	JESD22-A108 (125C/1000hrs, Vopmax) dynamic bias	0/77	Passed
Preconditioning	JESD22-A113, JEDEC J-STD-020 (30C/60%RH 192Hrs; L3, 260 Deg C IR Reflow)	0/135 0/90 0/90	Passed
Temp Cycle	JESD22-A104 (Condition C, -65 to 150C; 1000 Cycles)	0/45 0/45 0/45	Passed
Biased HAST	JESD22-A110 (130 C, 85% RH, 33.3 PSI, Vopmax)	0/45	Passed

Qualification Stress Test Name	Test Method	Fail/ Sample Size	Results
Pressure Cooker Test	JESD22A-102b (121C,100%RH, 15PSI)	0/45	Passed

Table 660: Reliability Qualification Summary

IR Reflow Profile

- Compliant with IPC/JEDEC J-STD-020 D
- Conditions:
  - Average ramp-up rate (217°C to peak): 1~2°C/sec max.
  - Preheat: 150°C - 200°C, 60 - 180 seconds
  - Temperature maintained above 217°C: 60~150 seconds
  - Time within 5°C of actual peak temperature: 20 ~ 40 seconds
  - Peak temperature: 260+0/-5°C
  - Ramp-down rate: 3°C/sec. max.
  - Time 25°C to peak temperature: 8 minutes max.
  - Cycle interval: 5 minutes

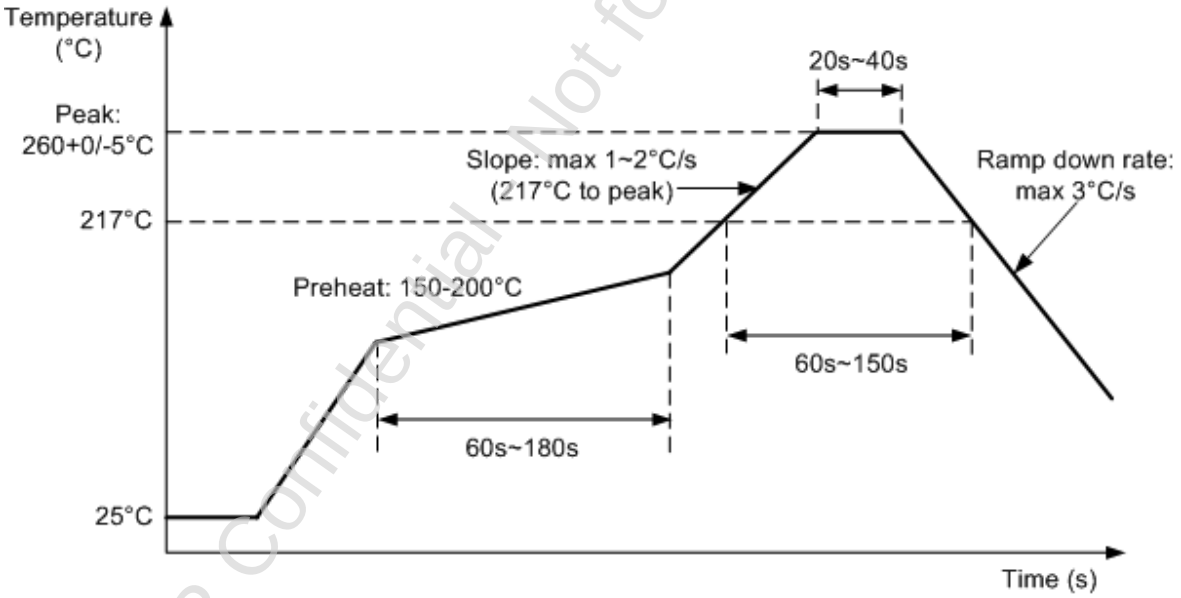


Figure 123: IR-Reflow Profile

DC Electrical Specifications

Normal Pad I/O DC Characteristics

- VDD = 1.65V~3.60V
- Vext = 3.0V~5.5V
- TA = -20°C to 70°C

Parameter		Condition		Min	Typ	Max	Unit
Vtol	Tolerant external voltage**	VDD Power Off		-	-	3.6	V
		VDD Power On	VDD=3.3V	-	-	5.5	V
			VDD=1.8V	-	-	3.6	
Vih	**High Level Input Voltage**						
	LVCMOS Interface	-		0.7VDD	-	VDD+0.3	V
Vil	**Low Level Input Voltage**						
	LVCMOS Interface	-		-0.3	-	0.3VDD	V
ΔV	Hysteresis Voltage	-		0.1VDD	-	-	V
Iih	**High Level Input Current**						
	Input Buffer	Vin=VDD		-10	-	10	uA
	Tolerant Input Buffer**	Vin=Vext		-10	-	10	uA
	Input Buffer with pull-down	Vin=VDD	VDD=3.3V	20	70	130	uA
			VDD=1.8V	5	20	40	
	Tolerant Input Buffer with pull-up**	Vin=5V	VDD=3.3V	10	30	60	uA
		Vin=3.3V	VDD=1.8V	2	8	18	
Iil	**Low Level Input Current**						
	Input Buffer	Vin=VSS		-10	-	10	uA
	Input Buffer with pull-up	Vin=VSS	VDD=3.3V	-130	-70	-20	uA
			VDD=1.8V	-40	-20	-5	
Voh	Type A,B,C	Ioh=-100uA		VDD-0.2	-	-	V
Vol	Type A,B,C	Iol=100uA		-	-	0.2	V
Ioz	Tri-State Output Leakage Current	Vout=VSS or VDD		-10	-	10	uA
R_{PU}	Pull-Up resistor	VDD=3.3V±0.3V		50	60	75	kΩ
R_{PD}	Pull-Down resistor			60	68	78	kΩ
R_{PU}	Pull-Up resistor	VDD=1.8V±0.1V		111	134	152	kΩ

Parameter		Condition	Min	Typ	Max	Unit
R_{PD}	Pull-Down resistor		122	148	183	kΩ
CIN	Input capacitance	Any input and Bidirectional buffers	-	-	5	pF
COUT	Output capacitance	Any output buffer	-	-	5	pF

Table 661: Pad I/O DC Characteristics

In the above table:

- **specification is only available on tolerant cells.
- Driver type A, B, and C: refer to DC currents table of output driver.
- The specification can be changed depending on interface voltage.

TSC and ADC Characteristics

Characteristics	Symbol	Min.	Typ.	Max.	Unit	Test Condition
Maximum conversion rate	fs	-	-	600	KSPS	-
Normal operation mode current consumption	IOP	-	3.3	4.5	mA	-
Power-down mode current consumption	IPD	-	0.001	0.015	mA	-
Total harmonic distortion	THD	-	-60	-56	dB	-
Signal-to-noise & distortion ratio	SNDR	54	58	-	dB	-
Differential nonlinearity	DNL	-	±1	±2	LSB	VREF = 3.3V AGND = 0.0V
Integral nonlinearity	INL	-	±2	±4	LSB	VREF = 3.3V AGND = 0.0V
Offset voltage	TOPOFF BOTOFF	-	10	-	LSB	VREF = 3.3V AGND = 0.0V
Gain error	-	-1	-	+1	(%FS)	-

Table 662: TSC and ADC Characteristics

USB PHY DC Electrical Characteristics

Characteristics	Symbol	Min.	Typ.	Max.	Unit
DP / DM Electrical Characteristics					
Suspended Current					

Suspended Current	Room Temp (25℃)	-	-	500	uA
	Hot Temp (70℃)	-	-	3	mA
Input Levels for Full speed					
Differential Input Sensitivity	V_{DI}	0.2	-	-	V
Differential Common Mode Range	V_{CM}	0.8	-	2.5	V
Input Levels for High speed					
Differential Common Mode Range	V_{HSCM}	-50	-	500	mV
Output Levels for FS					
Low	V_{OL}	0	-	0.3	V
High	V_{OH}	2.8	-	3.6	V
Output Levels for HS					
HS data signaling high	V_{HSOH}	360	-	460	mV
HS data signaling low	V_{HSOL}	-15.0	-	15.0	mV
VBUS Comparator					
A-Device Vbus Valid	$V_{A\_VBUS\_VLD}$	4.0	-	-	V
A-Device Session Valid	$V_{A\_SESS\_VLD}$	0.8	-	2.0	V
B-Device Session Valid	$V_{B\_SESS\_VLD}$	0.8	-	4.0	V
B-Device Session End	$V_{B\_SESS\_END}$	0.2	-	0.8	V

Table 663: USB PHY DC Electrical Characteristics

Power Consumption

Power Consumption in Typical Cases

The following table lists typical power consumption details under the following testing conditions:

- ARM 500MHz, DSP 250MHz, System-bus 250MHz, I/O 125MHz, DDR2 (clock rate is 200MHz) / mDDR (clock rate is 166MHz)
- Ambient temperature: 25°C
- GPS acquisition + Mp3 decoding + MediaQ running + Mosquito running + Memory pattern test+ RTC timer + Temperature acquisition.
- All powers are in typical voltage

Power Domain	Included Power Pins	Typical Power Consumption (mW)	Comments
PLL	VDD_PLL0/1 VDDIO_OSC	8	-
RTC	VDDIO_RTC	1	-
Core	VDD_Core	400	-
TSC	VDD_TSC VDDA_TSC VREF_ADC	<1	This value depends on application. In this scenario, it is less 1mW.
USB	VDD_USB VDDA_USB	<5	This value depends on application. It is less than 5mW in this scenario.
I/O pads	VDDIO VDDIO_L VDDIO_N VDDIO_DAC	100	This value depends on application.
Memory pads	VDDIO_MEM	55 (mDDR mode)	Memory chips are not included.
		65 (DDR2 mode)	
Total	mDDR mode	570	Memory chips are not included.
	DDR2 mode	580	

Table 664 : SiRFatlasV Power Consumption in Typical Case (500MHz)

Power Consumption in Deep-Sleep Mode

In deep-sleep mode, RTC is working and memory interface is in retention mode, other powers are shut down. The power consumption data in this mode is shown in the table below. The testing conditions are:

- Ambient temperature: 25°C
- Application scenario: in deep-sleep mode

Power	Voltage (V)	Current (µA)
VDDIO_RTC	3.3	15 - 30
VDDIO_MEM	1.8	<5

Table 665: SiRFatlasV Power Consumption in Deep-Sleep Mode

Power Consumption in Hibernation Mode

In hibernation mode, only RTC is working while other powers are shut down. The power consumption data in this mode is shown in the table below. The testing conditions are:

- Ambient temperature: 25°C
- Application scenario: in hibernation mode

Power	Voltage (V)	Current (µA)
VDDIO_RTC	3.3	15 - 30

Table 666: SiRFatlasV Power Consumption in Hibernation Mode

AC Timing Characteristics

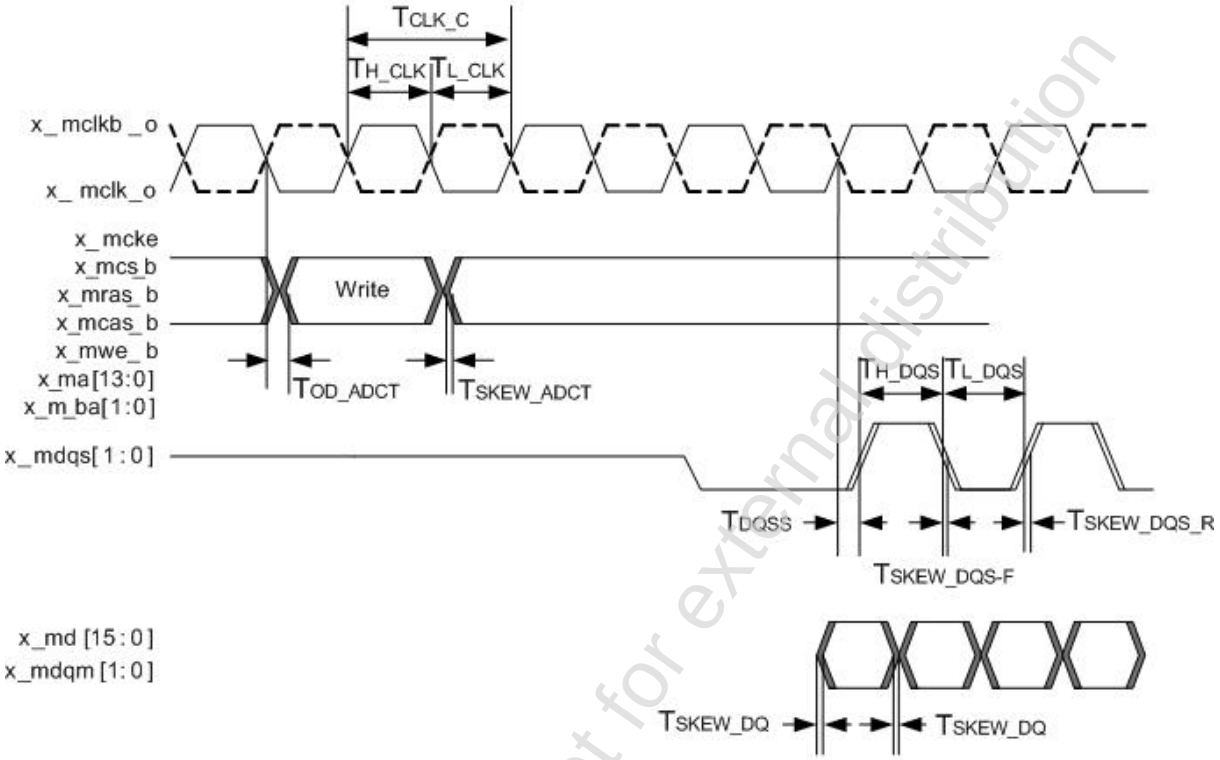
AC Operation Frequency

The following table lists the operating frequencies of SiRFatlasV.

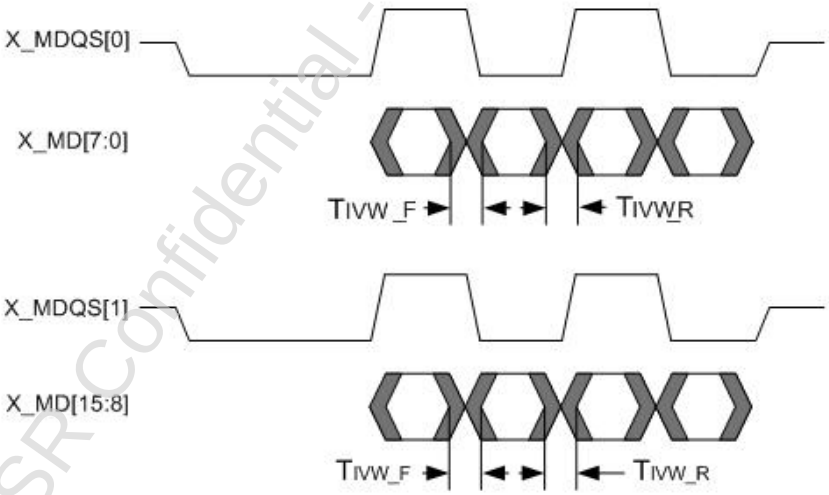
Part	Max. for 500MHz Parts
ARM clock	500
DSP clock	250
SYS clock	250
DDR2 SDRAM clock	200
Mobile DDR SDRAM clock	166
I/O clock	125

Table 667: Clock Frequencies (Unit: MHz)

Memory Interface AC Timing



DDR2 Write Timing



DDR2 Read Timing

Figure 124: Memory Interface DDR2 Timing

Symbol	Description	Min.	Typical	Max.
T_{CLK_C}	Differential clock cycle	5	-	8
T_{H_CLK}	Differential clock high pulse width	0.45* T_{CLK_C}	-	0.55* T_{CLK_C}
T_{L_CLK}	Differential clock low pulse width	0.45* T_{CLK_C}	-	0.55* T_{CLK_C}
T_{OD_ADCT}	Address/control signals output delay relative to falling edge of differential clock	-	-	0.9
T_{SKEW_ADCT}	Address/control signals skew	-	-	0.3
T_{SKEW_DQ}	Write data and data mask signal output skew.	-	-	0.3
$T_{SKEW\_DQS\_R}$	Write DQS skew at rising edge	-	-	0.1
$T_{SKEW\_DQS\_F}$	Write DQS skew at falling edge	-	-	0.1
T_{DQSS}	Write DQS latching rising transitions to associated clock edge	-0.8	-	+0.8
T_{H_DQS}	Write DQS high pulse width	0.35* T_{CLK_C}	-	-
T_{L_DQS}	Write DQS low pulse width	0.35* T_{CLK_C}	-	-
T_{IVW_R}	Invalid read data window relative to rising edge of read DQS	0.8	-	-
T_{IVW_F}	Invalid read data window relative to falling edge of read DQS	0.8	-	-

Table 668: Memory Interface AC Timing – DDR2 Mode with 200MHz Clock (Unit: ns)

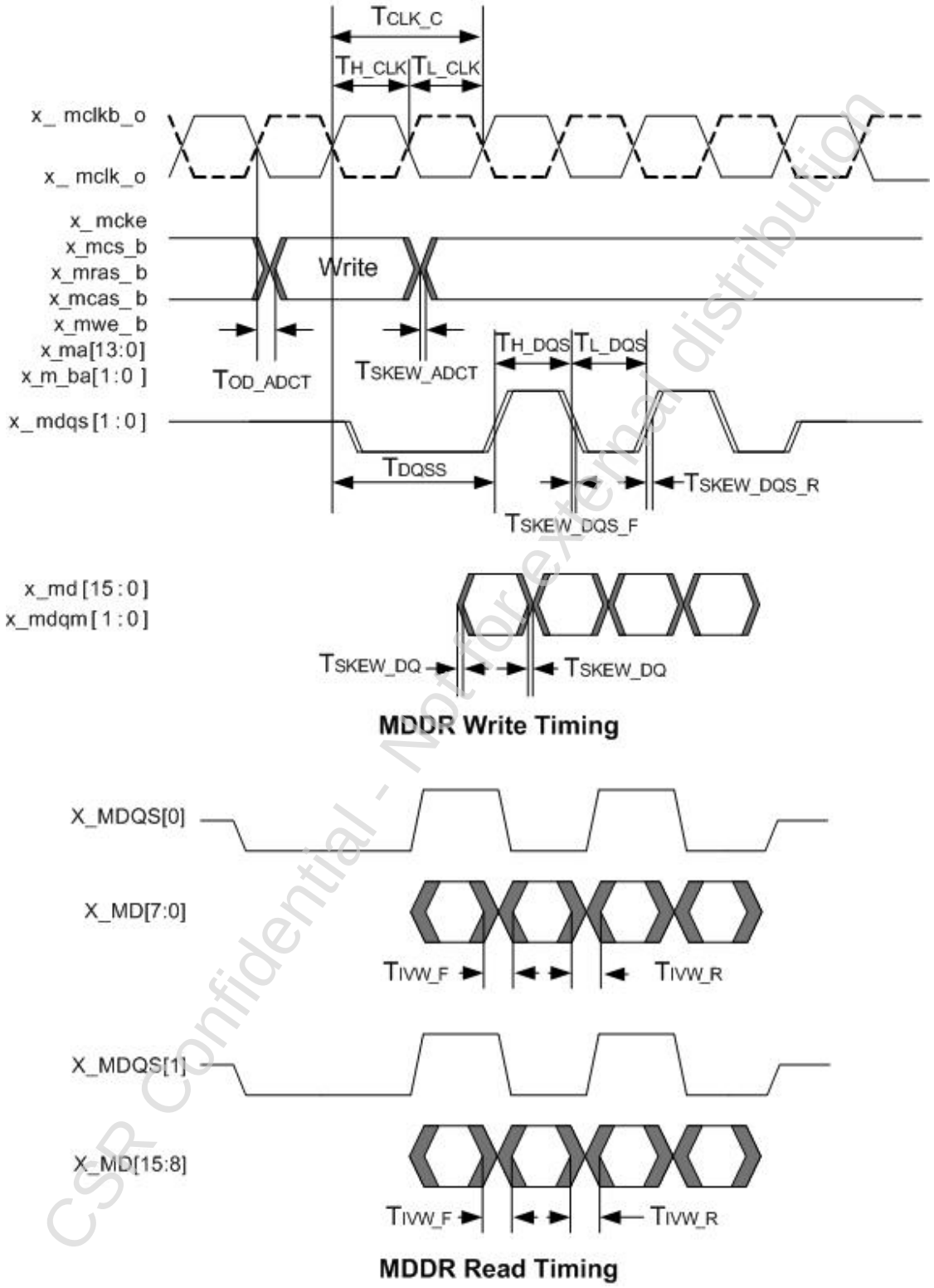


Figure 125: Memory Interface MDDR Timing

Symbol	Description	Min.	Typical	Max.
T_{CLK_C}	Differential clock cycle	6	-	-
T_{H_CLK}	Differential clock high pulse width	0.45* T_{CLK_C}	-	0.55* T_{CLK_C}
T_{L_CLK}	Differential clock low pulse width	0.45* T_{CLK_C}	-	0.55* T_{CLK_C}
T_{OD_ADCT}	Address/control signals output delay relative to falling edge of differential clock	-	-	0.9
T_{SKEW_ADCT}	Address/control signals skew	-	-	0.3
T_{SKEW_DQ}	Write data and data mask signal output skew.	-	-	0.3
$T_{SKEW\_DQS\_R}$	Write DQS skew at rising edge	-	-	0.1
$T_{SKEW\_DQS\_F}$	Write DQS skew at falling edge	-	-	0.1
T_{DQSS}	WRITE command to first DQS latching transition	-0.8+ T_{CLK_C}	-	0.8+ T_{CLK_C}
T_{H_DQS}	Write DQS high pulse width	0.35* T_{CLK_C}	-	-
T_{L_DQS}	Write DQS low pulse width	0.35* T_{CLK_C}	-	-
T_{IVW_R}	Invalid read data window relative to rising edge of read DQS	0.8	-	-
T_{IVW_F}	Invalid read data window relative to falling edge of read DQS	0.8	-	-

Table 669: Memory Interface AC Timing – Mobile DDR Mode (Unit: ns)

SD0/1/2/3 Interface AC Timing

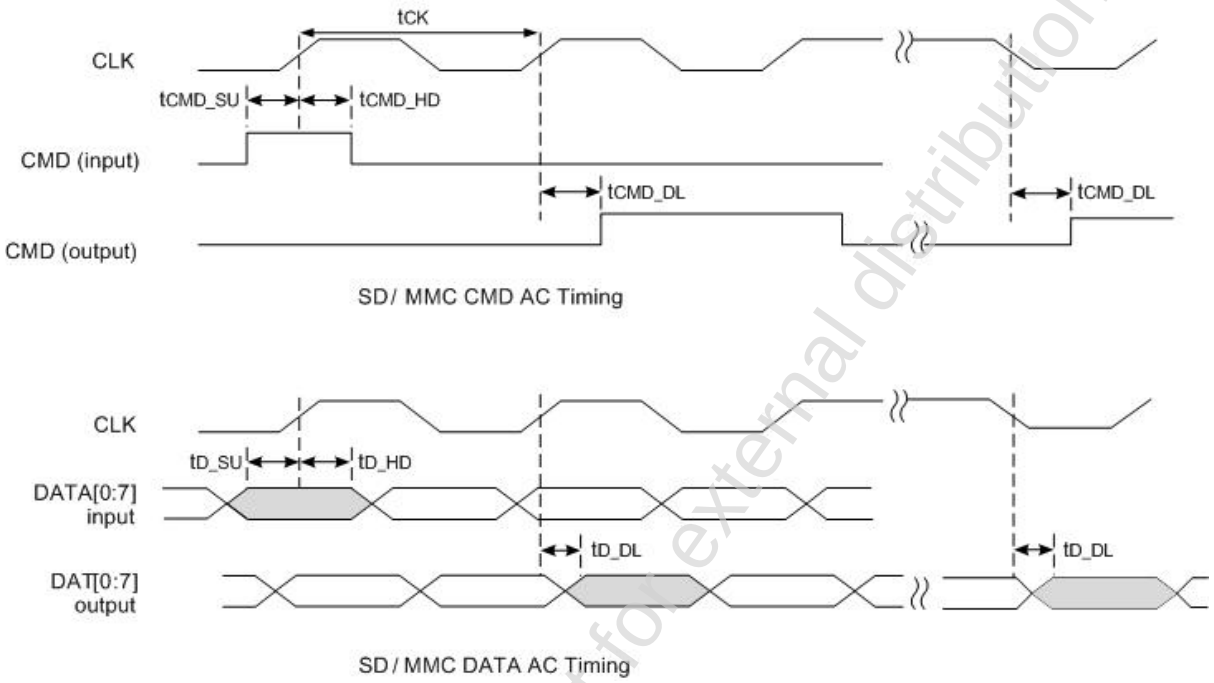


Figure 126: SD0/2 and MMC AC Timing

Symbol	Parameter	Min.	Typical	Max.
T_{CK}	Clock period at normal mode	-	40	-
	Clock period at high speed mode	-	20	-
$T_{CMD\_SU\_0}$	X_DF_ALE,SD0 CMD input setup time request	5	-	-
$T_{CMD\_HD\_0}$	X_DF_ALE,SD0 CMD input hold time request	5	-	-
$T_{CMD\_DL\_0}$	X_DF_ALE,SD0 CMD output delay	-	-	13
$T_{D\_SU\_0}$	X_DF_AD[0:7],SD0 data input setup time request	5	-	-
$T_{D\_HD\_0}$	X_DF_AD[0:7],SD0 data input hold time request	5	-	-
$T_{D\_DL\_0}$	X_DF_AD[0:7],SD0 data output delay	-	-	14

Table 670: SD Slot 0 AC Timing Data (Unit: ns)

Symbol	Parameter	Min.	Typical	Max.
T_{CK}	Clock period at normal mode	-	40	-

Symbol	Parameter	Min.	Typical	Max.
	Clock period at high speed mode	-	20	-
$T_{CMD\_SU\_1}$	X_DF_RE,SD2 CMD input setup time request	5	-	-
$T_{CMD\_HD\_1}$	X_DF_RE,SD2 CMD input hold time request	5	-	-
$T_{CMD\_DL\_1}$	X_DF_RE,SD2 CMD output delay	--	-	13
$T_{D\_SU\_1}$	X_DF_AD[0:7] SD2 data input setup time request	5	-	-
$T_{D\_HD\_1}$	X_DF_AD[0:7] SD2 data input hold time request	5	-	-
$T_{D\_DL\_1}$	X_DF_AD[0:7] SD2 data output delay	-	-	14

Table 671: SD Slot 2 AC Timing Data (Unit: ns)

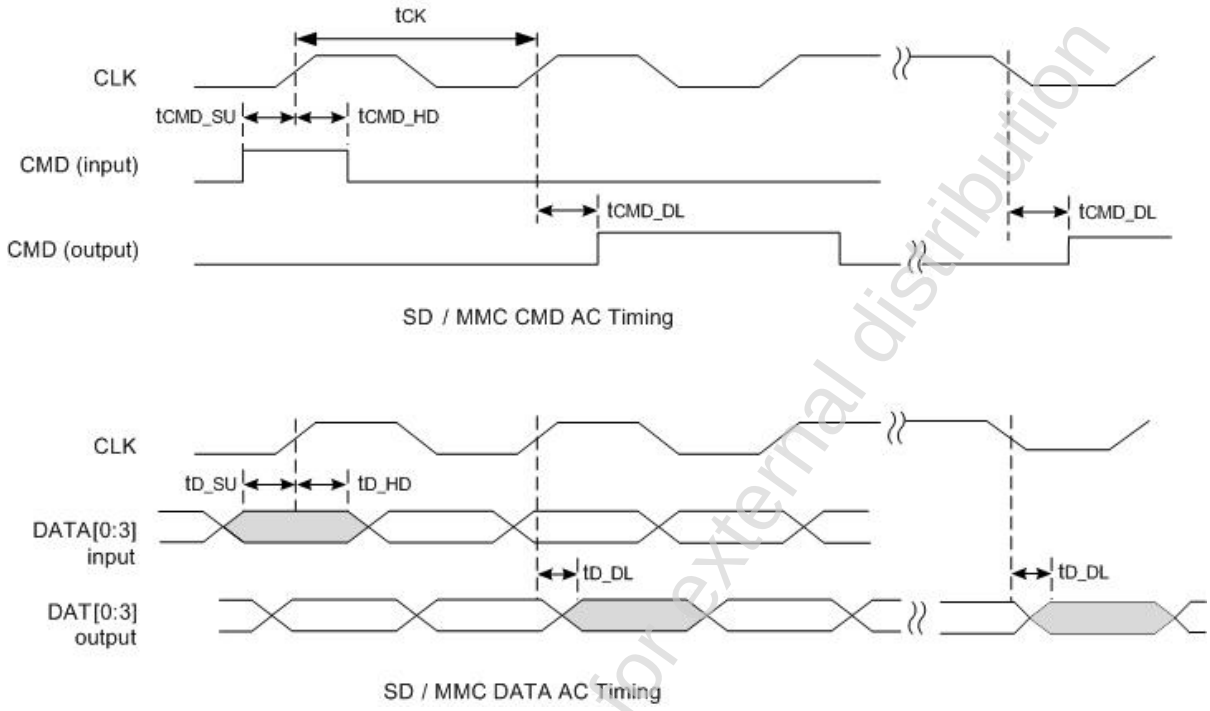


Figure 127: SD1/3 and MMC AC Timing

Symbol	Parameter	Min.	Typical	Max.
T_{CK}	Clock period at normal mode	-	40	-
	Clock period at high speed mode	-	20	-
$T_{CMD\_SU\_0}$	X_SD_CMD_1 input setup time request	5	-	-
$T_{CMD\_HD\_0}$	X_SD_CMD_1 input hold time request	5	-	-
$T_{CMD\_DL\_0}$	X_SD_CMD_1 output delay	-	-	13
$T_{D\_SU\_0}$	X_SD_DAT_1[0:3] data input setup time request	5	-	-
$T_{D\_HD\_0}$	X_SD_DAT_1[0:3] data input hold time request	5	-	-
$T_{D\_DL\_0}$	X_SD_DAT_1[0:3] data output delay	-	-	14

Table 672: SD Slot 1 AC Timing Data (Unit: ns)

Symbol	Parameter	Min.	Typical	Max.
T_{CK}	Clock period at normal mode	-	40	-

Symbol	Parameter	Min.	Typical	Max.
	Clock period at high speed mode	-	20	-
$T_{CMD\_SU\_0}$	X_SD_CMD_3 input setup time request	5	-	-
$T_{CMD\_HD\_0}$	X_SD_CMD_3 input hold time request	5	-	-
$T_{CMD\_DL\_0}$	X_SD_CMD_3 output delay	-	-	13
$T_{D\_SU\_0}$	X_SD_DAT_3[0:3] data input setup time request	5	-	-
$T_{D\_HD\_0}$	X_SD_DAT_3[0:3] data input hold time request	5	-	-
$T_{D\_DL\_0}$	X_SD_DAT_3[0:3] data output delay	-	-	14

Table 673: SD Slot 3 AC Timing Data (Unit: ns)

NAND Flash Interface AC Timing

Figure 128: NAND Flash Interface AC Timing

Symbol	Parameter	Min.	Typical	Max.
T_{RE_PULSE}	X_DF_RE_B low pulse	-	(RD_PULSE+1) x T_{IOCLK}	-
T_{RE_HI}	X_DF_RE_B hi width between low pulses	-	(RD_WT_HI+1) x T_{IOCLK}	-
T_{AD_SU}	X_DF_AD setup time request when input	8	-	-
T_{AD_HD}	X_DF_AD hold time request when input	0	-	-
T_{WE_PULSE}	X_DF_WE_B low pulse	-	Command/address cycle: (WT_PULSE) xT_{IOCLK} Write data output cycle: (1+WT_PULSE) xT_{IOCLK}	-
T_{WE_HI}	X_DF_WE_B hi width between low pulses	-	Command/address cycle: (2+RD_WT_HI) xT_{IOCLK} Write data output cycle: (1+RD_WT_HI) xT_{IOCLK}	-
T_{AD_O}	X_DF_AD output time	-	-	(WT_PULSE +1) x ioclk-20
T_{CLE_O}	X_DF_CLE output time	-	1xT_{IOCLK}	-
T_{ALE_O}	X_DF_ALE output time	-	1xT_{IOCLK}	-
T_{CLE_H}	X_DF_CLE output hold time	-	(RD_WT_HI+1)xT_{IOCLK}	-
T_{ALE_H}	X_DF_ALE output hold time	-	(RD_WT_HI+1)xT_{IOCLK}	-
T_{AD_H}	X_DF_ALE output hold time	-	(RD_WT_HI)xT_{IOCLK}	-

Table 674: NAND Flash Interface AC Timing Data (VDDIO_N=3.3V) (Unit: ns)

LCD Interface AC Timing

- Master mode
  When the LCD controller works in master mode, all pins are output.

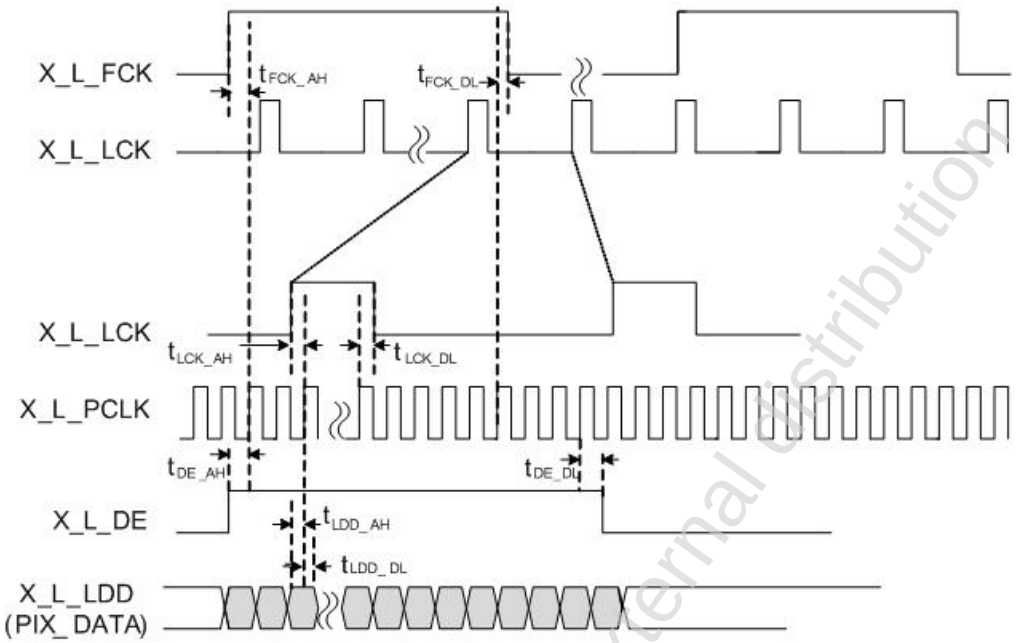


Figure 129: LCD Interface Timing Diagram in Master Mode

Symbol	Description	Value
T_{pclk}	Pix clock period	(Ratio+1) x Tsys
T_{FCK_AH}	X_L_FCK output ahead before X_L_PCLK	(Ratio-SYNC_DLY) x Tsys
T_{FCK_DL}	X_L_FCK output delay than X_L_PCLK	(SYNC_DLY+1) x Tsys
T_{LCK_AH}	X_L_LCK output ahead before X_L_PCLK	(Ratio-SYNC_DLY) x Tsys
T_{LCK_DL}	X_L_LCK output delay than X_L_PCLK	(SYNC_DLY+1) x Tsys
T_{DE_AH}	X_L_DE output ahead before X_L_PCLK	(Ratio-SYNC_DLY) x Tsys
T_{DE_DL}	X_L_DE output delay than X_L_PCLK	(SYNC_DLY+1) x Tsys
T_{LDD_AH}	X_LDD output ahead before X_L_PCLK	Tpclk/2
T_{LDD_DL}	X_LDD output delay than X_L_PCLK	Tpclk/2

Table 675: LCD Interface AC Timing in Master Mode (VDDIO_L=3.3V) (Unit: ns)

- Slave mode
  When the LCD controller works in slave mode:
  - PIXCLK is input.
  - X_VSYNC and X_HSYNC can be input or output.
  - X_L_DE and X_LDD are output.

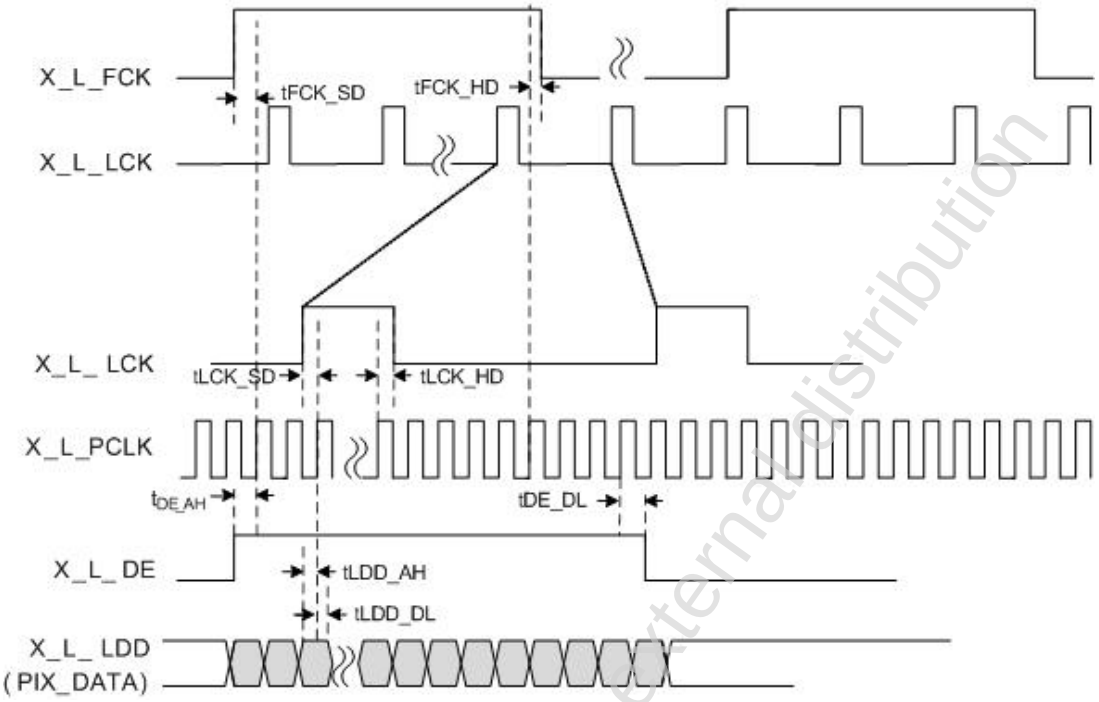


Figure 130: LCD Interface Timing in Slave Mode

Symbol	Description	Min.	Typical	Max.
T_{pclk}	Pix clock period.	-	39	-
T_{FCK_SU}	X_L_FCK setup time for X_L_PCLK when X_L_FCK is input.	5	-	-
T_{FCK_HD}	X_L_FCK hold time for X_L_PCLK when X_L_FCK is input.	5	-	-
T_{FCK_AH}	X_L_FCK output ahead before X_L_PCLK, when X_L_FCK is output.	-	Tpclk – (2+SYNC_DLY) x Tsys	-
T_{FCK_DL}	X_L_FCK output delay than X_L_PCLK when X_L_FCK is output.	-	(2+SYNC_DLY) x Tsys	-
T_{LCK_SU}	X_L_LCK setup time for X_L_PCLK when X_L_LCK is input.	5	-	-
T_{LCK_HD}	X_L_LCK hold time for X_L_PCLK when X_L_LCK is input.	5	-	-
T_{LCK_AH}	X_L_LCK output ahead before X_L_PCLK when X_L_LCK is output.	-	Tpclk – (2+SYNC_DLY) x Tsys	-
T_{LCK_DL}	X_L_LCK output delay than X_L_PCLK when X_L_LCK is output.	-	(2+SYNC_DLY) x Tsys	-
T_{DE_AH}	X_L_DE output ahead before X_L_PCLK.	-	Tpclk – (2+SYNC_DLY) x Tsys	-

Symbol	Description	Min.	Typical	Max.
T_{DE_DL}	X_L_DE output delay than X_L_PCLK.	-	(2+SYNC_DLY) x Tsys	-
T_{LDD_AH}	X_LDD output ahead before X_L_PCLK.	-	Tpclk/2	-
T_{LDD_DL}	X_LDD output delay than X_L_PCLK.	-	Tpclk/2	-

Table 676: LCD Interface AC Timing in Slave Mode (VDDIO_L=3.3V) (Unit: ns)

Audio Codec AC Timing

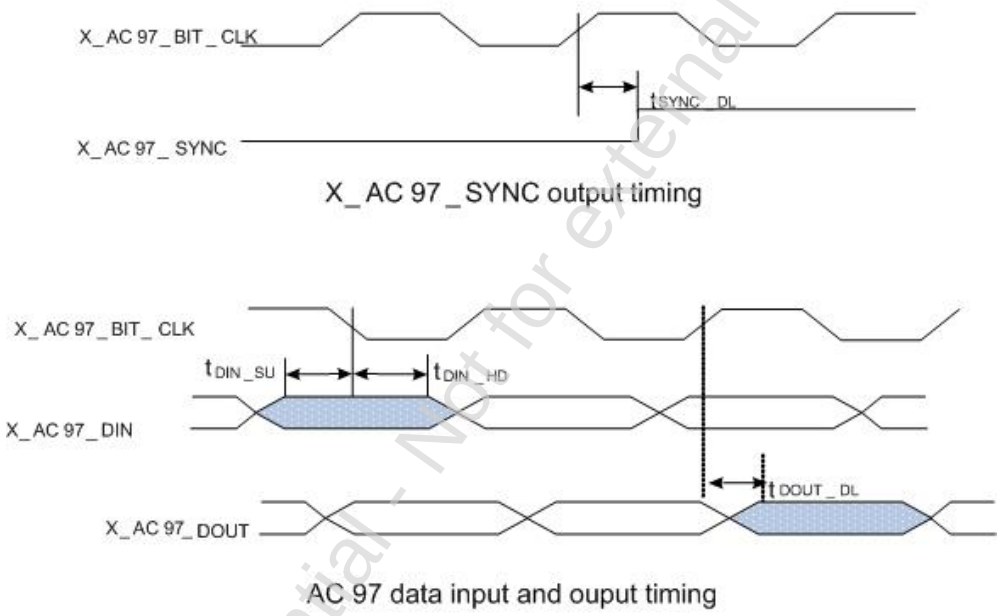


Figure 131: AC97 Interface AC Timing (AC97 Mode)

Symbol	Parameter	Min.	Typical	Max.
T_{SYNC_DL}	X_AC97_SYNC output delay from X_AC97_BIT_CLK rising edge	-	-	15
T_{DIN_SU}	X_AC97_DIN setup time to X_AC97_BIT_CLK falling edge	10	-	-
T_{DIN_HD}	X_AC97_DIN hold time from X_AC97_BIT_CLK falling edge	10	-	-
T_{DOUT_DL}	X_AC97_DOUT output delay from X_AC97_BIT_CLK rising edge	-	-	15

Table 677: AC97 Interface AC Timing Data (AC97 Mode) (Unit: ns)

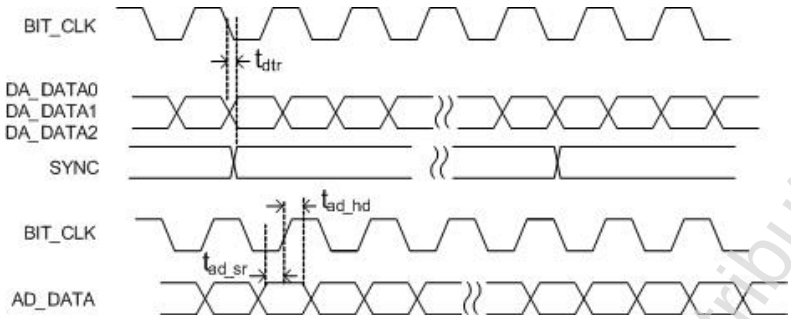


Figure 132: I²S Interface Timing (Master Mode)

Symbol	Parameter	Min.	Typical	Max.
T_{dtr}	I²S data and word select output from BIT_CLK falling edge.	-	-	60
T_{ad_sr}	I²S data input setup time from BIT_CLK rising edge	40	-	-
T_{ad_hd}	I²S data input hold time from BIT_CLK rising edge	100	-	-

Table 678: I²S Interface Master Mode Timing (Unit: ns)

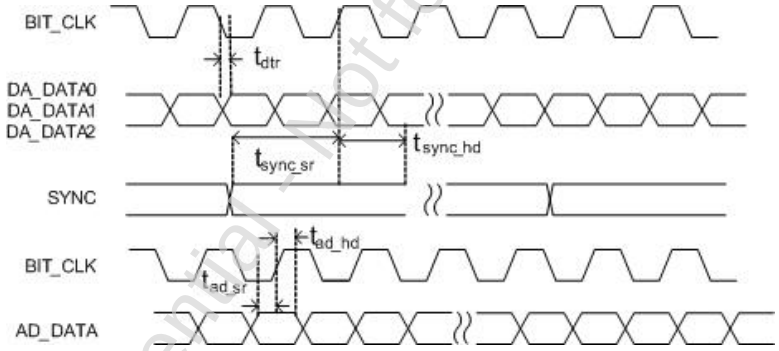


Figure 133: I²S Interface Timing (Slave Mode)

Symbol	Parameter	Min.	Typical	Max.
T_{dtr}	I²S data output from BIT_CLK falling edge.	-	-	120
T_{sync_sr}	I²S word select setup time from BIT_CLK rising edge	80	-	-
T_{sync_hd}	I²S word select hold time from BIT_CLK rising edge	100	-	-
T_{ad_sr}	I²S data input setup time from BIT_CLK rising edge	80	-	-
T_{ad_hd}	I²S data input hold time from BIT_CLK rising edge	100	-	-

Table 679: I²S Interface Slave Mode Timing (Unit: ns)

SPI Interface AC Timing

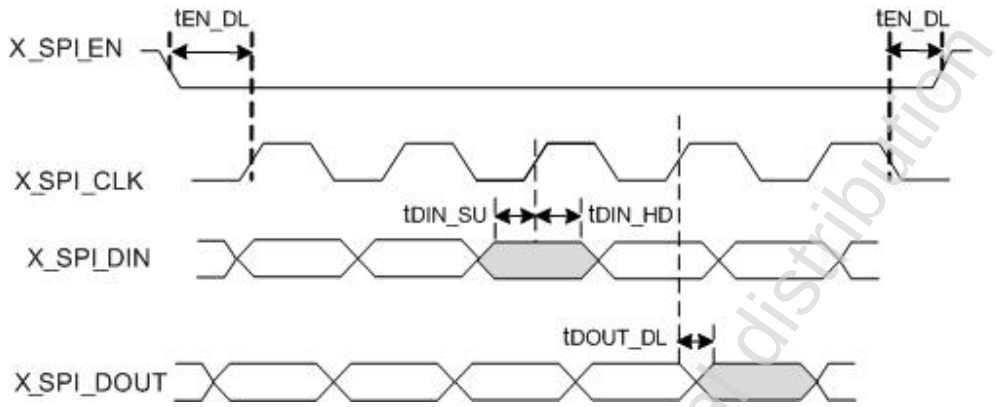


Figure 134: SPI Interface AC Timing (Master Mode)

Symbol	Parameter	Min.	Typical	Max.
T_{DIN_SU}	X_SPI_DIN setup time to X_SPI_CLK receive edge	8	-	-
T_{DIN_HD}	X_SPI_DIN hold time from X_SPI_CLK receive edge	5	-	-
T_{DOUT_DL}	X_SPI_DOUT output delay time from X_SPI_CLK transfer edge	-	-	5
T_{EN_DL}	X_SPI_EN output delay time to X_SPI_CLK transfer edge	1 x T_{SPI_CLK}	-	2 x T_{SPI_CLK}

Table 680: SPI AC Timing Data (Master Mode) (Unit: ns)

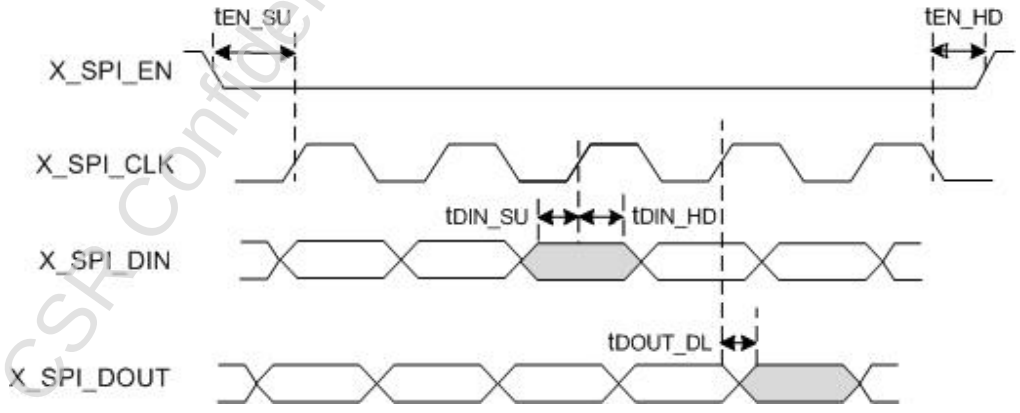


Figure 135: SPI Interface AC Timing (Slave Mode)

Symbol	Parameter	Min.	Typical	Max.
T_{DIN_SU}	X_SPI_DIN setup time to X_SPI_CLK receive edge	10	-	-
T_{DIN_HD}	X_SPI_DIN hold time from X_SPI_CLK receive edge	4 x T_{IOCLK}	-	-
T_{DOUT_DL}	X_SPI_DOUT output delay time from X_SPI_CLK transfer edge	4 x T_{IOCLK}	-	-
T_{EN_SU}	X_SPI_EN setup time to X_SPI_CLK first edge	If DRV==CLK 4*Tioclk+0.5Tspi_clk If DRV!=CLK 4*Tioclk	-	-
T_{EN_HD}	X_SPI_EN hold time from X_SPI_CLK last edge	If DRV==CLK 4*Tioclk-0.5Tspi_clk If DRV!=CLK 4*Tioclk	-	-

Table 681: SPI AC Timing Data (Slave Mode) (Unit: ns)

I²C0 and I²C1 Interface AC Timing

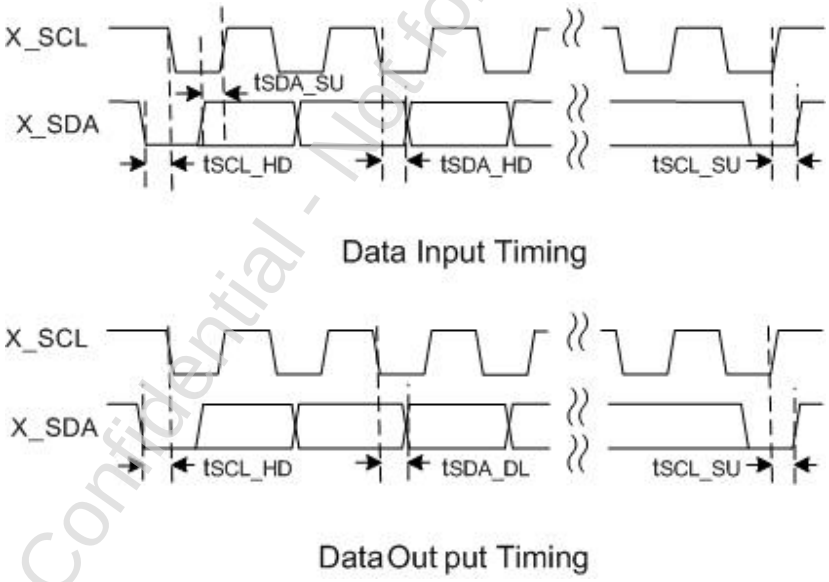


Figure 136: I²C Interface AC Timing Diagram

Symbol	Parameter	Min.	Typical	Max.
$F_{C(SCL\_0)}$	X_SCL_0 operating frequency (KHz)	-	F_{IOCLK}/(CLK_PERx5)	400
$T_{SDA\_0\_DL}$	X_SDA_0 output delay time after X_SCL_0 falling edge (µs)	-	T_{IOCLK} x (SDA_DELAY_REG+4)	-
$T_{SDA\_0\_SU}$	X_SDA_0 setup time to X_SCL_0 rising edge (ns)	250	-	-
$T_{SDA\_0\_HD}$	X_SDA_0 hold time from X_SCL_0 rising edge (ns)	0	-	-
$T_{SCL\_0\_HD}$	X_SCL_0 hold time after X_SDA_0 falling edge (µs)	-	T_{IOCLK} x CLK_PER	-
$T_{SCL\_0\_SU}$	X_SCL_0 setup time before X_SDA_0 rising edge (µs)	-	T_{IOCLK} x (CLK_PER + 2)	-

Table 682: I²C0 AC Timing Data (Master Mode)

Symbol	Parameter	Min.	Typical	Max.
$F_{C(SCL\_1)}$	X_SCL_1 operating frequency (KHz)	-	F_{IOCLK} / (CLK_PER x 5)	400
$T_{SDA\_1\_DL}$	X_SDA_1 output delay time after X_SCL_0 falling edge (us)	-	T_{IOCLK} x (SDA_DELAY_REG+4)	-
$T_{SDA\_1\_SU}$	X_SDA_1 setup time to X_SCL_1 rising edge (ns)	250	-	-
$T_{SDA\_1\_HD}$	X_SDA_1 hold time from X_SCL_1 falling edge (ns)	0	-	-
$T_{SCL\_1\_HD}$	X_SCL_1 hold time after X_SDA_1 falling edge (µs)	-	T_{IOCLK} x CLK_PER	-
$T_{SCL\_1\_SU\_S}$	X_SCL_1 setup time before X_SDA_1 rising edge (µs)	-	T_{IOCLK} x (CLK_PER + 2)	-

Table 683: I²C1 AC Timing Data (Master Mode)

USP0 and USP1 Interface AC Timing

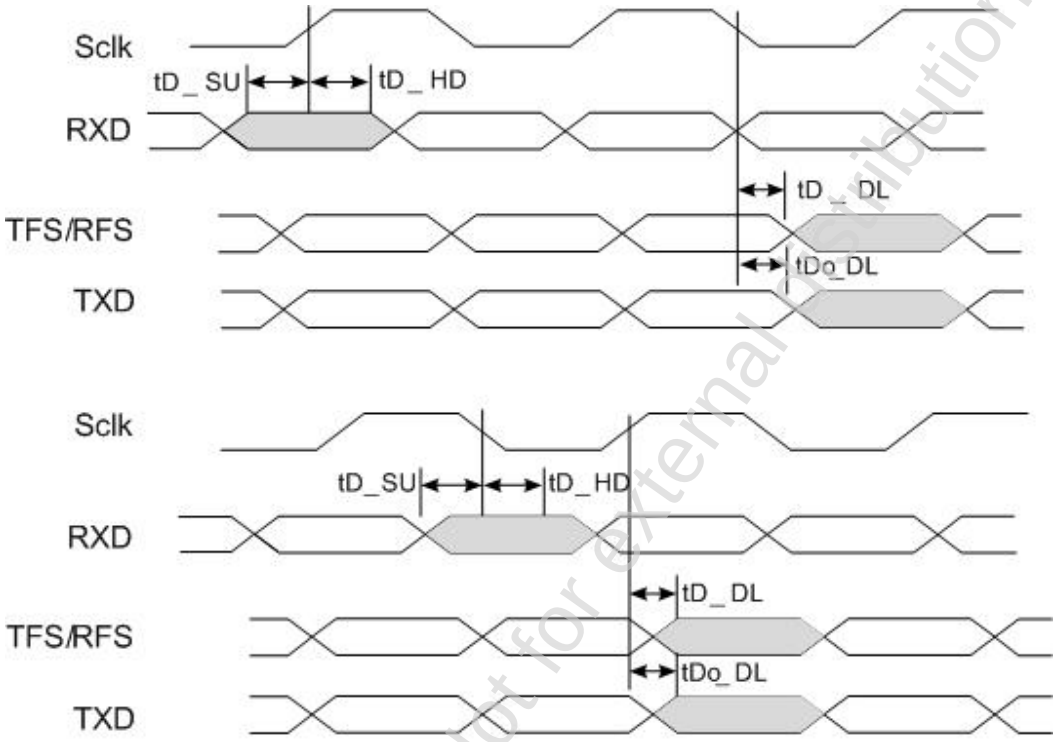


Figure 137: USP Interface AC Timing (Master Mode)

Symbol	Parameter	Min.	Typical	Max.
T_{D_SU}	X_URXD_0/1, data input setup time request	24	-	-
T_{D_HD}	X_URXD_0/1, data input hold time request	8	-	-
T_{D_DL}	X_UTFS_0/1, X_URFS_0/1, data output delay	-	-	8
T_{Do_DL}	X_UTXD_0/1, data output delay	-	-	TXD_DELAY_LEN *Tsclk+8

Table 684: USP Interface AC Timing (Master Mode) (Unit: ns)

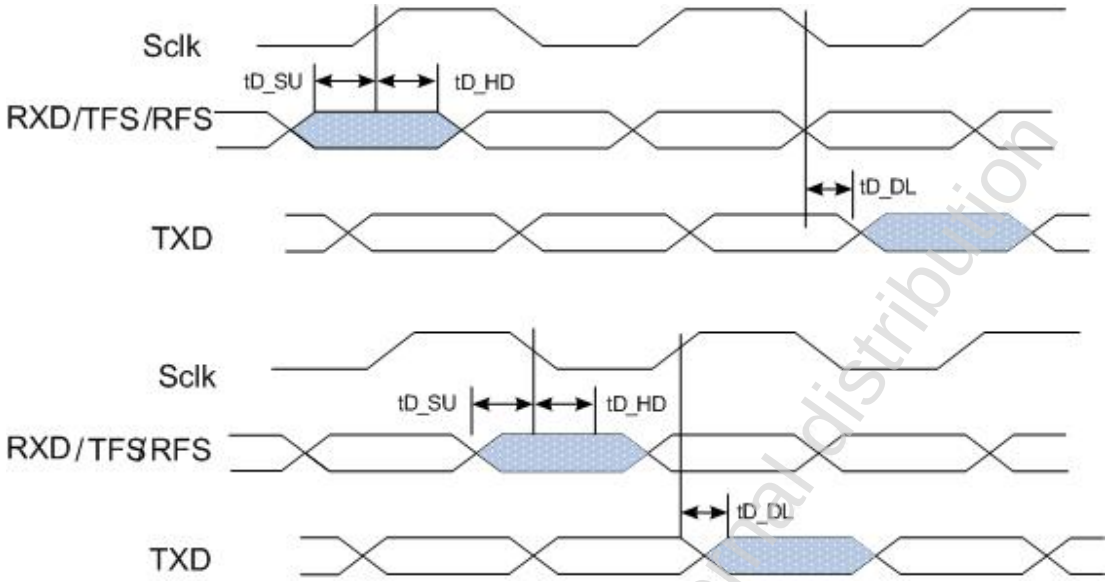


Figure 138: USP Interface AC Timing (Slave Mode)

Symbol	Parameter	Min.	Typical	Max.
T_{D_SU}	X_URXD_0/1, X_UTFS_0/1, X_URFS_0/1, data input setup time request	8	-	-
T_{D_HD}	X_URXD_0/1, X_UTFS_0/1, X_URFS_0/1, data input hold time request	32	-	-
T_{D_DL}	X_UTXD_0/1, data output delay	-	-	TXD_DELAY_LEN *Tsclk+56

Table 685: USP Interface AC Timing (Slave Mode) (Unit: ns)

REVISION HISTORY

Version	Date	Author	Comments
Issue 1	Mar., 2011	Zhanqiang Su	Initial
Issue 2	May., 2011	Zhanqiang Su	Draft.

ADDITIONAL INFORMATION

Additional technical information including Application Notes are available through the CSR web site at: http://www.CSR.com

ORDERING INFORMATION

Part Number	Marking	Package	Frequency	Dimension (mm)	Packing	Operating Environment
AT5511	AT5511	TFBGA-285	500MHz	10 x 13	Tray	Extended Commercial grade

WORLDWIDE SALES OFFICES

North America
United States
(1) (408) 467-0410
Sales@csr.com

Europe
United Kingdom
(44) (1344) 668390
SalesUK@csr.com

Germany
(49) (81) 529932-90
SalesGermany@csr.com

Belgium
(32) (496) 152969
SalesBelgium@csr.com

Asia Pacific
China
(86) (21) 5854-7153
SalesChina@csr.com

Taiwan
(886) (2) 2650-2000
SalesTaiwan@csr.com

Japan
(81) (44) 829-2186
SalesJanpan@csr.com

India
(91) (80) 41966000
SalesIndia@csr.com

South Korea
(82) (2) 3424-3150
SalesKorea@csr.com

DRAFT