[1. Sirius Features Introduction 5](#_Toc498118273)

[1.1 Cortex-A7 5](#_Toc498118274)

[1.2 CEVA DSP 6](#_Toc498118275)

[1.2.1 Introduction 6](#_Toc498118276)

[1.2.2 Architecture 6](#_Toc498118277)

[1.2.3 Features 7](#_Toc498118278)

[1.3 Cortex-M7 9](#_Toc498118279)

[1.3.1 Description 9](#_Toc498118280)

[1.3.2 Block Diagram 9](#_Toc498118281)

[1.3.3 Embedded Characteristics 9](#_Toc498118282)

[1.4 NOC (network on-chip) 11](#_Toc498118283)

[1.4.1 General Description 11](#_Toc498118284)

[1.4.2 Architecture 11](#_Toc498118285)

[1.4.3 Functional description 11](#_Toc498118286)

[1.5 SMMU 12](#_Toc498118287)

[1.5.1 Overview 12](#_Toc498118288)

[1.5.2 Block Diagram. 13](#_Toc498118289)

[1.5.3 Features 13](#_Toc498118290)

[1.6 Image Signal Processor (ISP) 15](#_Toc498118291)

[1.6.1 Overview 15](#_Toc498118292)

[1.6.2 Features 15](#_Toc498118293)

[1.6.3 TOP Diagram 16](#_Toc498118294)

[1.7 Baseband Modem Processor 16](#_Toc498118295)

[1.7.1 Overview 16](#_Toc498118296)

[1.7.2 Architecture 17](#_Toc498118297)

[1.7.3 Features 17](#_Toc498118298)

[1.8 HEVC Codec 17](#_Toc498118299)

[1.8.1 Overview 17](#_Toc498118300)

[1.8.2 Architecture 18](#_Toc498118301)

[1.8.3 Features 18](#_Toc498118302)

[1.8.3.1 HEVC Encoder 18](#_Toc498118303)

[1.8.3.2 HEVC Decoder 19](#_Toc498118304)

[1.9 H.264/AVC Codec 20](#_Toc498118305)

[1.9.1 Overview 20](#_Toc498118306)

[1.9.2 Architecture 20](#_Toc498118307)

[1.9.3 Features 21](#_Toc498118308)

[1.9.3.1 H.264/AVC Encoder 21](#_Toc498118309)

[1.9.3.2 H.264/AVC Decoder 21](#_Toc498118310)

[1.10 JPEG Codec 22](#_Toc498118311)

[1.10.1 Overview 22](#_Toc498118312)

[1.10.2 Architecture 22](#_Toc498118313)

[1.10.3 Features 22](#_Toc498118314)

[1.11 Display Engine 23](#_Toc498118315)

[1.11.1 Overview 23](#_Toc498118316)

[1.11.2 Block Diagram 24](#_Toc498118317)

[1.11.3 Features 25](#_Toc498118318)

[1.12 DDR 26](#_Toc498118319)

[1.12.1 Overview 26](#_Toc498118320)

[1.12.2 Architecture 26](#_Toc498118321)

[1.12.3 Features 26](#_Toc498118322)

[1.13 USB 27](#_Toc498118323)

[1.13.1 Features 27](#_Toc498118324)

[1.13.2 Block Diagram 28](#_Toc498118325)

[1.13.3 USB 3.0 Device Features 28](#_Toc498118326)

[1.13.4 USB 3.0 xHCI Host Features 29](#_Toc498118327)

[1.14 MIPI CSI System 29](#_Toc498118328)

[1.14.1 Overview 29](#_Toc498118329)

[1.14.2 Architecture 29](#_Toc498118330)

[1.14.3 Features 30](#_Toc498118331)

[1.15 HDMI Receiver 31](#_Toc498118332)

[1.15.1 Overview 31](#_Toc498118333)

[1.15.2 Architecture 31](#_Toc498118334)

[1.15.3 Features 31](#_Toc498118335)

[1.16 PCI Express 32](#_Toc498118336)

[1.16.1 Overview 32](#_Toc498118337)

[1.16.2 Architecture 33](#_Toc498118338)

[1.16.3 Features 33](#_Toc498118339)

[1.17 USB/DP Type-C 34](#_Toc498118340)

[1.17.1 Overview 34](#_Toc498118341)

[1.17.2 Architecture 35](#_Toc498118342)

[1.17.3 Features 36](#_Toc498118343)

[1.18 Ethernet 36](#_Toc498118344)

[1.18.1 Overview 36](#_Toc498118345)

[1.18.2 Block Diagram 37](#_Toc498118346)

[1.18.3 Features 37](#_Toc498118347)

[1.19 Quad-SPI Flash Controller 38](#_Toc498118348)

[1.19.1 Overview 38](#_Toc498118349)

[1.19.2 Block Diagram 38](#_Toc498118350)

[1.19.3 Features 38](#_Toc498118351)

[1.20 SD Card Controller 39](#_Toc498118352)

[1.20.1 Overview 39](#_Toc498118353)

[1.20.2 Block Diagram 39](#_Toc498118354)

[1.20.3 Features 39](#_Toc498118355)

[1.21 Low-Speed Peripherals 41](#_Toc498118356)

[1.21.1 UART Controller 41](#_Toc498118357)

[1.21.1.1 Overview 41](#_Toc498118358)

[1.21.1.2 Block Diagram 41](#_Toc498118359)

[1.21.1.3 Features 41](#_Toc498118360)

[1.21.2 I2C Controller 42](#_Toc498118361)

[1.21.2.1 Overview 42](#_Toc498118362)

[1.21.2.2 Block Diagram 42](#_Toc498118363)

[1.21.2.3 Features 43](#_Toc498118364)

[1.21.3 SPI Controller 43](#_Toc498118365)

[1.21.3.1 Overview 43](#_Toc498118366)

[1.21.3.2 Block Diagram 44](#_Toc498118367)

[1.21.3.3 Features 44](#_Toc498118368)

[1.21.4 CAN controller 44](#_Toc498118369)

[1.21.4.1 Overview 44](#_Toc498118370)

[1.21.4.2 Block diagram 45](#_Toc498118371)

[1.21.4.3 Features 45](#_Toc498118372)

[1.21.5 Watchdog Timer 46](#_Toc498118373)

[1.21.5.1 Block Diagram 46](#_Toc498118374)

[1.21.5.2 Features 46](#_Toc498118375)

[1.21.6 Timer with PWM 46](#_Toc498118376)

[1.21.6.1 Overview 46](#_Toc498118377)

[1.21.6.2 Block Diagram 47](#_Toc498118378)

[1.21.6.3 Features 47](#_Toc498118379)

[1.21.7 GPIO 47](#_Toc498118380)

[1.21.7.1 Overview 47](#_Toc498118381)

[1.21.7.2 Block Diagram 48](#_Toc498118382)

[1.21.7.3 Features 48](#_Toc498118383)

[1.22 EMMC Controller 48](#_Toc498118384)

[1.22.1 Overview 48](#_Toc498118385)

[1.22.2 Block Diagram 49](#_Toc498118386)

[1.22.3 Features 49](#_Toc498118387)

[1.23 Secure Subsystem Feature 50](#_Toc498118388)

[1.23.1 Secure CPU 50](#_Toc498118389)

[1.23.2 Security Protocol Accelerator(SPACC) 50](#_Toc498118390)

[1.23.3 Public Key Accelerator (PKA) 51](#_Toc498118391)

[1.23.4 True Random Number Generator (TRNG) 52](#_Toc498118392)

[1.23.5 Secure UART 52](#_Toc498118393)

[1.23.6 Secure Timer 53](#_Toc498118394)

[1.23.7 Secure Watchdog 53](#_Toc498118395)

[1.23.8 OTP (one-time programmable) Controller 54](#_Toc498118396)

[1.23.8.1 Features 54](#_Toc498118397)

[1.23.9 Trust Zone controller 54](#_Toc498118398)

[1.23.9.1 Overview 54](#_Toc498118399)

[1.23.9.2 Block Diagram 55](#_Toc498118400)

[1.23.9.3 Features 55](#_Toc498118401)

[1.24 DVP interface 55](#_Toc498118402)

[1.24.1 Overview 55](#_Toc498118403)

[1.24.2 Architecture 56](#_Toc498118404)

[1.24.3 Functional description 56](#_Toc498118405)

[1.25 Internal Video interface block 56](#_Toc498118406)

[1.25.1 Overview 56](#_Toc498118407)

[1.25.2 Architecture 57](#_Toc498118408)

[1.25.3 Features 57](#_Toc498118409)

[1.26 I2S interface 58](#_Toc498118410)

[1.26.1 Overview 58](#_Toc498118411)

[1.26.2 Block Diagram 58](#_Toc498118412)

[1.26.3 Features 59](#_Toc498118413)

[1.27 DMA controller 60](#_Toc498118414)

[1.27.1 CEVA\_DMAC Controller 60](#_Toc498118415)

[1.27.1.1 Block Diagram 60](#_Toc498118416)

[1.27.1.2 Features 60](#_Toc498118417)

[1.27.2 TOP\_DMAC Controller 61](#_Toc498118418)

[1.27.2.1 Block Diagram 61](#_Toc498118419)

[1.27.2.2 Features 61](#_Toc498118420)

[1.27.3 AHB DMA Controller 62](#_Toc498118421)

[1.27.3.1 Block Diagram 62](#_Toc498118422)

[1.27.3.2 Features 62](#_Toc498118423)

[1.28 Analog part 63](#_Toc498118424)

[1.28.1 Overview 63](#_Toc498118425)

[1.29 MISC blocks 63](#_Toc498118426)

[1.29.1 Overview 63](#_Toc498118427)

[1.29.2 CGU 64](#_Toc498118428)

[1.29.3 RGU 64](#_Toc498118429)

[1.29.4 IO Share function 64](#_Toc498118430)

[1.29.5 TOP Global Register 64](#_Toc498118431)

[1.29.6 SPI Debug Interface 65](#_Toc498118432)

[1.29.6.1 Overview 65](#_Toc498118433)

[1.29.6.2 Block Diagram 65](#_Toc498118434)

[1.29.6.3 Features 65](#_Toc498118435)

[1.29.7 CoreSight 65](#_Toc498118436)

[1.29.7.1 Overview 65](#_Toc498118437)

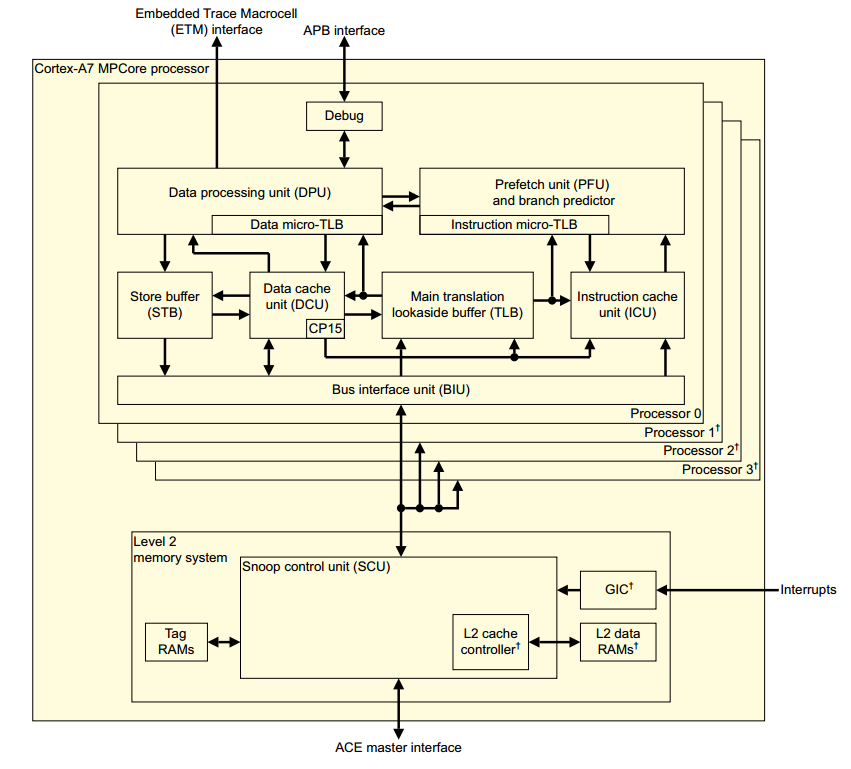
[1.29.7.2 Block Diagram 66](#_Toc498118438)

[1.29.7.3 Features 66](#_Toc498118439)

## Sirius Features Introduction

## Cortex-A7

The Cortex-A7 MPCore processor is a high-performance, low-power processor that implements the ARMv7-A architecture. The Cortex-A7 MPCore processor has four processors in a single multiprocessor device with a L1 cache subsystem, an integrated GIC, and an L2 cache controller. Picture 1-1 shows a top-level functional diagram of the Cortex-A7 MPCore processor.



Picture 1-1

It contains Data Processing Unit, System control coprocessor, Instruction side memory system, Data side memory system, L1 memory system, Media Processing Engine, Floating-Point Unit, L2 memory system, Debug and Performance monitoring.

The cortexa7 feature list:

* 4 cores in the cortexa7 MPCore. Max work frequency of every core can reach 1.2Ghz.
* The four cores can be power on/off respectively.
* Support DVFS for whole CA7 subsystem
* 192 SPI interrupts in which there are 160 external device interrupt.
* 32KB L1 I Cache.
* 32KB L1 D Cache.
* 512KB L2 Cache.
* FPU which is a VFPv4-D16 implementation of the ARMv7 floating-point architecture.
* NEON provides support for the ARMv7 Advanced SIMDv2 and Vector Floating-Pointv4 instruction sets.

## CEVA DSP

## Introduction

Cevaxm4\_Subsystem is a platform targeted for high-performance computer vision and image processing applications, which includes four CEVA-XM4 core. The CEVA-XM4 is an extremely powerful, low-power DSP processor designed and optimized for computer vision and image processing, which is based on a VLIW model combined with SIMD concept. This processor consists of 4 Scalar Processing Units (SPUs), two Load/Store Units (LSUs), a Program Control Unit (PCU), two Vector Processing Units (VPUs), a Power Scaling Unit (PSU), Memory Subsystem (MSS) and Emulation interface.

## Architecture



Figure 1 CevaXM4\_Subsystem Diagram

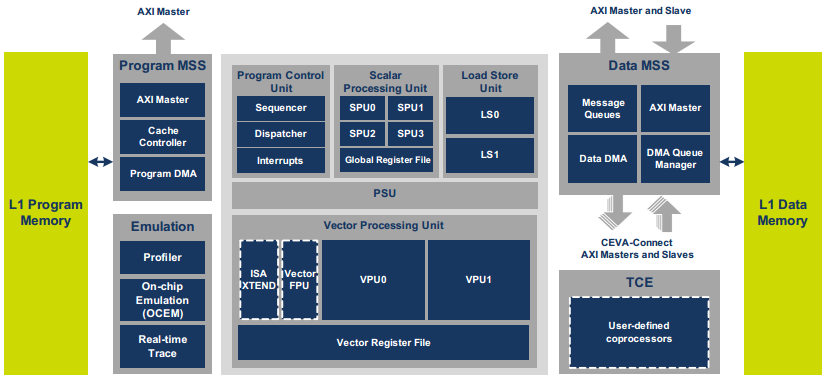


Figure 2 CEVA-XM4 DSP Block Diagram

## Features

1. The four CEVA-XM4 core can be power on/off respectively.
2. The four CEVA-XM4 core can communicate with each other in MCCI mode or interrupt mode.
3. High code compactness due to:

* Variable instruction width (16-bit, 32-bit, 48-bit and 64 bit)
* Variable-size instruction packets
* Instruction replication method

1. All instructions support predication:

* Conditional execution
* Reduces cycle count and code size on control and overhead code

1. Enhanced register file that also includes (every core)

* 32 32-bit general registers used for scalar operations and address generation
* 40 256-bit vector registers used for all vector-related operations

1. Every core includes two Vector Processing Units (VPUs):

* Parallel processing of up to 256-bits operations in each unit
* Supports 32 8-bit, 16 16-bit or 8 32-bit operations in each unit
* All operations are signed or unsigned
* Supports both inter-vector and intra-vector operations
* Up to 64 multipliers of 8x16 bits, 32 multipliers of 16x16 bits and 8 multipliers of 32x32 bits on each VPU
* Advanced filter operations for two-dimensional frames
* Up to 64 Sum of Absolute Differences (SADs) per cycle with an option to accumulate partial results
* Ability to sort vectors according to minimum, median and maximum
* Bit-manipulation operations including vector permutations
* Logical operations
* Non-linear operation support, such as , ,
* Supports up to 16 single-precision floating-point operations

1. Every core has Four Scalar Processing Units (SPUs):

* Supports 16-bit and 32-bit operations
* 16×16 bits, 32×16 bits and 32×32 bits multipliers
* 16×16 bits, 32×16 bits and 32×32 MAC operations
* Full support of bit-manipulation and logical unit
* Supports single-precision floating-point operations

1. Every core has Two Load/Store Units (LSUs) for two independent accesses to the data memories:
   * Maximum bandwidth of 512-bits when using both load units
   * Maximum bandwidth of 256-bits for store operations
   * Ability to access up to 32 different memory addresses in one memory access
   * Multiple data addressing modes, including:
   * Four gigabyte program and data address
   * Byte-addressable data address
   * Unaligned data memory access
2. Program memory subsystem that includes:
   * L1 program memory
   * L1 four-way program cache
   * Dedicated AXI master bus
   * Program Direct Memory Access (DMA) available for background transfers and cache preloading
3. Data memory subsystem that includes:

* L1 data memory
* Data DMA available for background data transfers
* DMA task queue manager
* Separate I/O space for peripherals’ connectivity

1. Fully registered memory interface
2. On-chip Emulation (OCEM) support via a JTAG port

## Cortex-M7

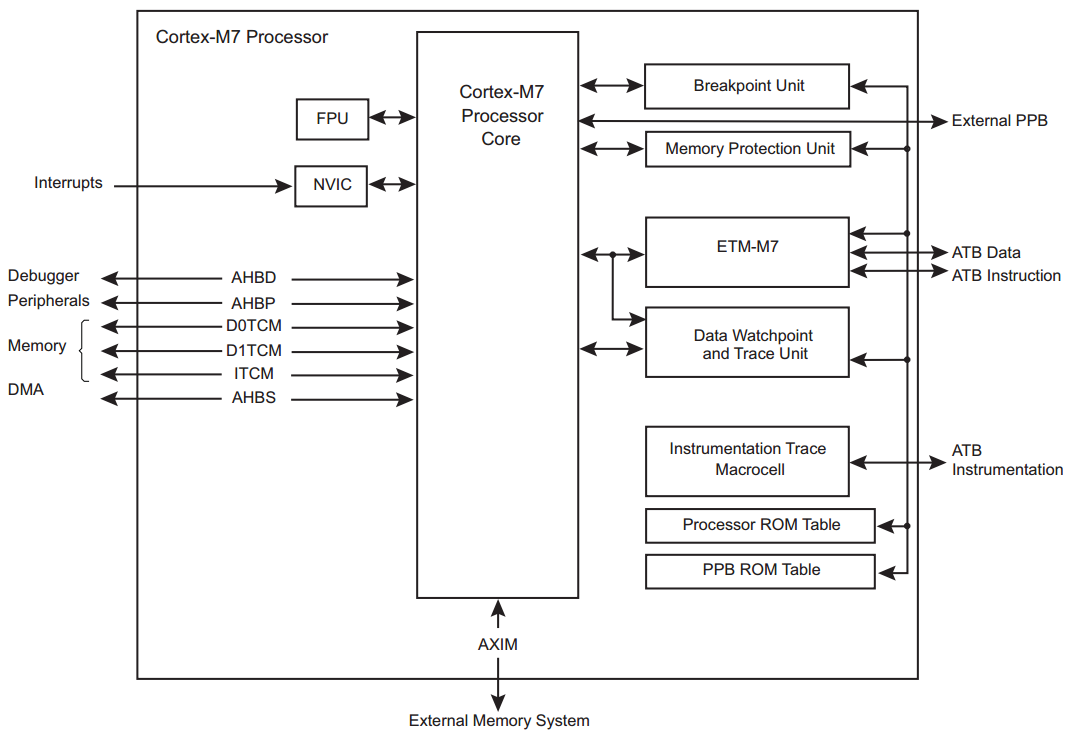
## Description

The ARM Cortex-M7 processor implements the ARMv7-M architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions.

The double-precision Floating-Point Unit (FPU) supports the ARMv7 VFPv5 architecture. It is tightly integrated to the ARM Cortex-M7 processor pipeline. It provides trapless execution and is optimized for scalar operation. It can generate an Undefined instruction exception on vector instructions that enables the programmer to emulate vector capability in software.

Note: Refer to ARM reference documents *Cortex-M7 Processor User Guide* (ARM DUI 0644) and *Cortex-M7 Technical Reference Manual* (A*RM DDI 0489)*, available on [www.arm.com](http://www.arm.com).

## Block Diagram



## Embedded Characteristics

* ARM Cortex-M7 with 16 KB of instruction cache and 16 KB of data cache
* 128KB ITCM & 64KB DTCM
* ARMv7-M Thumb instruction set combines high-code density with 32-bit performance
* Tightly Coupled Memory (TCM) interfaces:
* 64-bit ITCM interface
* 2 x 32-bit DTCM interfaces
* Memory Protection Unit (MPU): up to 16 protected memory regions for safety/critical applications
* Dedicated low-latency AHB-Lite peripheral (AHBP) interface
* Dedicated AHB slave (AHBS) interface for system access to TCMs
* Low-latency interrupt processing achieved by a Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor
* DSP extensions for efficient signal processing and complex algorithm execution
* IEEE Standard 754-2008 Floating Point Unit (FPU)
* A low-cost debug solution with the ability to:
* Implement breadpoints.
* Implement watchpoints, tracing, and system profiling
* TPIU
* DAP
* Hardware integer divide instructions
* Extensive debug and trace capabilities:
* Serial Wire Debug and Serial Wire Trace reduce the number of pins required for debugging, tracing, and code profiling

## NOC (network on-chip)

## General Description

It has the following features:

* Configurable data path widths and address width, at sockets and within the interconnect.
* Configurable, run-time programmable protected regions within the address space.
* Support AXI, AHB and APB protocol interface,etc.

## Architecture



Figure : NOC Architecture

## Functional description

A NOC interconnect is built using a socket-based design methodology. It supports industry-standard interfaces to the IP cores that make up an SoC. Cores connect to agents within a NOC interconnect that decouple the functionality of each IP core from the interconnect communications required among the cores, automatically adjusting for mismatches in data width, clock frequency, and protocols. This approach allows core designers to tailor the communications for each core, while still permitting SoC integrators to balance the characteristics of intercore communication paths with respect to metrics such as latency, physical span, clock frequency usage, power domain usage, die area, and power consumption.

The masters connected to NOC\_ceva: CEVA.

The slaves connected to NOC\_ceva: CEVA, DMA\_ceva, MMU\_ceva.

The masters connected to NOC\_vision: HEVC, H264, JPEG, ISP, Video Interface, Display Control.

The slaves connected to NOC\_vision: MMU\_isp, MMU\_codec.

The masters connected to NOC\_main: CCI-400, Coresight, SPI Debug, M7 net, Baseband, DMA\_main, eMMC, SPAcc, tRoot, MMU\_ceva, USB3.0, Type-c, PCI-e, GMAC.

The slaves connected to NOC\_main: CCI-400, Coresight, M7 net, Baseband, DMA\_main, eMMC, SPAcc, tRoot, OTP, TRNG/PKA, ARM\_A7, ROM, NOC-Secure, Secure, MMU\_config, on-chip SRAM, MMU\_ddr, CEVA, USB3.0, Video Interface, MIPI, HDMI, Display Control, H264, JPEG, HEVC,ISP, Type-c, PCI-e, GMAC.

## SMMU

## Overview

The MMU-500 is a system-level **Memory Management Unit** (MMU) that translates an input address to an output address, based on address mapping and memory attribute information available in the MMU-500 internal registers and translation tables.

In Sirius there are two SMMU blocks separately for address mapping for CEVA and CodeC/ISP/DisplayEngine block.

## Block Diagram.

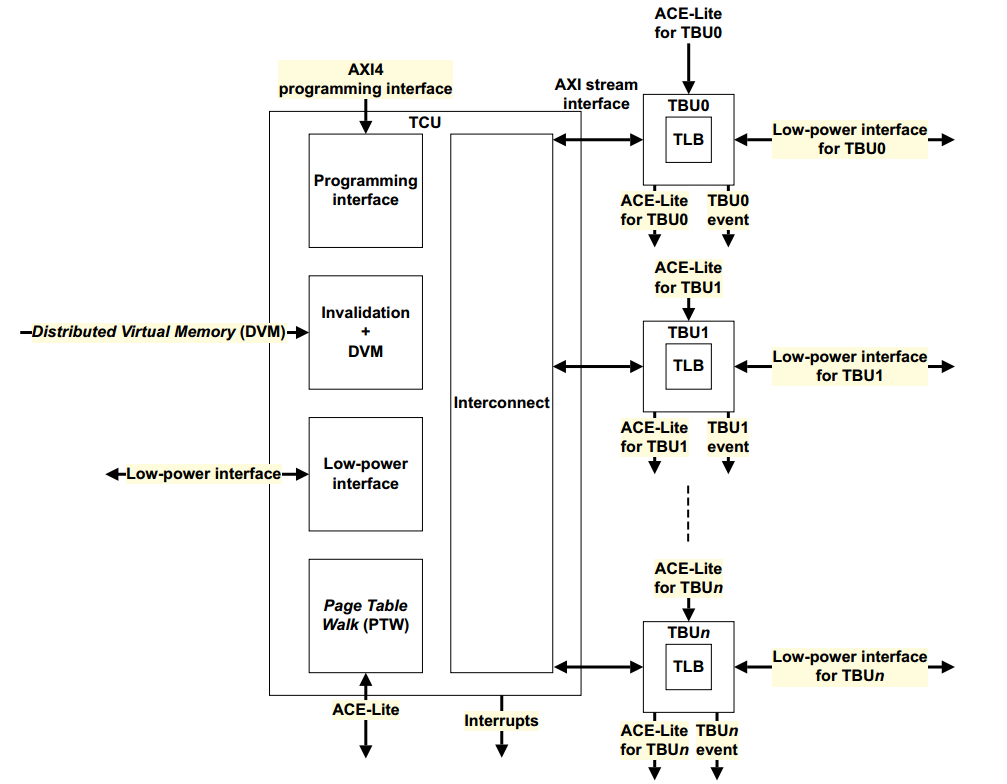


Figure 1-1

## Features

* Address virtualization to other masters in an ARM processor based system and other bus masters in the system.
* Support for the following translations:
* Stage 1.
* Stage 2.
* Stage 1 followed by stage 2.
* Programmable *Quality of Service* (QoS).
* Distributed translation support for up to 32 TBUs.
* Translation support for 32-bit to 49-bit virtual address ranges and 48-bit physical address ranges.
* Multiple transaction contexts to apply to address translations for specific streams of transactions. Supports up to 128 configurable contexts and programmable page size. The MMU-500 maps each context by using an input stream ID from the master device that requires address translation.
* Translation support for the following:
* Stage 1 ARMv7 VMSA.
* Stage 1 and Stage 2 ARMv8 AArch32.
* Stage 1 and Stage 2 ARMv8 AArch64 with 4KB and 64KB granules.
* Stage 1 followed by stage 2 translations.
* No page size restrictions. All page sizes are supported apart from the 16KB page granule defined by ARMv8 architecture.
* Arbitration of PTW requests from different TBUs by using the programmed QoS value.
* Page table walk cache for storing intermediate page table walk data.
* Page table entry cache in the TLB.
* Support for TLB Hit-Under-Miss (HUM).
* Configurable PTW depth using parallel PTWs.
* TLB invalidation through the AMBA 4 DVM signaling or register programming.
* Translation and protection check support including TrustZone® extension support.
* Fault handling, logging, and signaling that includes demand paging and support for the stall model.
* One AMBA slave interface that supports ACE-Lite per TBU for connecting the bus master device that requires address translations.
* One AMBA master interface for master device transactions or PTWs that support ACE-Lite and DVM.
* An AXI4 interface for programming.
* Page table entry cache in the TLB at two levels, namely:
* Macro TLB.
* Micro TLB.
* The TLB at two levels and the walk cache RAMs support single bit error detection and invalidation on error detection. The context disambiguation *Multi-FIFO* (MFIFO) RAM supports single bit error detection and correction.
* Debug and performance-monitoring events.
* A prefetch buffer to prefetch the next 4K or 64K leaf page entry to reduce latency.
* An IPA2PA cache to speed up stage 1 followed by stage 2 translations.
* Support for 256 outstanding transactions for each TBU master interface.
* Support for priority elevation as part of the QoS scheme.

## Image Signal Processor (ISP)

## Overview

The image signal processor (ISP) module of Sirius supports standard sensor data processing. The ISP module provide basic function such as auto white balance(AWB),automatic exposure(AE),de-mosaic, defected pixel correction(DPC) as well as advanced functions such as wide dynamic range(WDR),dynamic range compression(DRC),both RAW de-noise and 3D de-noise.

## Features

* Black level correction(BLC)
* Static and dynamic DPC and defect pixel cluster correction
* Bayer denoising
* Advanced demosaic
* Gamma Correction
* DRC
* Color management and enhancement
* Sensor Build-in WDR
* Alternate row HDR
* AWB
* AE
* AF
* 3A(AE,AF,AWB) statistics output
* Lens shading correction(LSC)
* Picture Sharpening
* Color Noise correction
* Digital Zoom
* Linear correction
* Anti-aliasing
* Contrast Enhancement
* Support YUV444/YUV420 10bit/8bit Input
* Support de-purple
* Support 3D-denoise
* Max Support 3 Resolution Output

The processing capability is as follows：

* Maximum 14bit Bayer data input
* Maximum picture resolution of 8100\*8100
* Minimum picture resolution of 480\*240
* Maximum frame of 60 fps for the 3840\*2160
* Minimum H blank region of 128 pixels
* Minimum V blank region of 30 lines
* Support real-time processing multi sensor input
* Support low-latency processing

## TOP Diagram



**Figure 1** Functional block diagram of the ISP module

## Baseband Modem Processor

## Overview

Sirius baseband modem is a flexible wireless bi-directional baseband IP that can be dynamically deployed to connect 1 AP and up to 4 users. Customers can configure this IP flexibly for broad applications such as drone, Miracast, safety protection network, and so on. The IP supports two kinds of RF transceivers, namely AD9363 and AR8003s. AR8003s can work at both 2.4G and 5.8G frequency bands.

## Architecture



## Features

* Support wireless connection between 1 AP and up to 4 users
* Support bi-directional 4K transmission
* Configurable bandwidth of 1.25M/2.5M/5M/10M/20M on broadcast/CSMA
* Configurable bandwidth of 2.5M/5M/10M/20M/40M on slot
* Configurable 1x1,1x2,1x4,2x2,2x4 MIMO with STBC
* Support 1T2R for broadcast/CSMA and 1T2R, 2T2R, 2T4R for slot
* Support BPSK/QPSK/16QAM/64QAM/256QAM modulation
* Support 1/2, 2/3, 3/4, 5/6 LDPC code-rate codec
* Support max throughput to 80Mbps bi-directionally
* Real-time variable lengths on broadcast/CSMA/slot

## HEVC Codec

## Overview

H.265/HEVC hardware codec(encoder and decoder)is targeted for high-end multimedia devices capable of recording and playing up to 4K HEVC's Main/Main10 video such as camcorders/DSC, digital TVs, set-top boxes, smartphones, tablets, security cameras, and so forth.

HEVC codec is able to encode and/or decode any resolution up to 8192x4096. It guarantees real-time performance for encoding/decoding 4K 30fps based on its sophisticated, latency tolerant architecture. HEVC Codec is highly optimized for memory bandwidth loading and excellent power management.

## Architecture



HEVC codec is easy to integrate into a SoC, since it can be connected through the industry standard interfaces : 32-bitAMBA3 APB bus for host CPU system control and 128-bit AMBA3 AXI for data transfer.

## Features

## HEVC Encoder

■ Capable of encoding HEVC Main/Main 10 profile @L5.0 High-tier

■ Max resolution: 8192x4096, Min resolution: 256x128

■ Performance : 3840x2160 30fps @400MHz

■ Input video format

-YUV 4:2:0 8bit

-YUV 4:2:0 10bit

-YUV 4:2:2 8/10bit

■ I/P/B slices

■ CUT64

-support PU size : 32x32, 16x16, 8x8

-support TU size : 32x32 to 4x4

■ High performance CABAC (100Mbps)encoding

■ Low delay coding

-less than 1ms delay for starting encoder with sub-frame synchronization

■ Rate control(frame level and CU level)

-VBR , CBR and ABR

-ROI support

■1/4-pel accuracy motion estimation with search range[+/-128H,+/-64V] in IME base on pmv, and [+/-16H, +/-16V] in FME

■ Sample adaptive offset(SAO)

■ Loop filtering across slice

■ Transform skip

## HEVC Decoder

■ Capable of encoding HEVC Main/Main 10 profile @L5.0 High-tier

■ Max resolution: 8192x4096, Min resolution: 8x8

■ Performance : 3840x2160 30fps @400MHz

■ I/P/B slices

-All intra prediction modes

-All inter prediction modes

■ Variable CTU size: 64x64 to 16x16

-variable PU size : 64x64 to 4x4

-variable TU size : 32x32 to 4x4

■ High performance CABAC(100Mbps) decoding

■ Advanced Motion Vector Prediction(AMVP) and merge mode

■ 1/4 motion compensation with 8 tap filters

■ Uniform reconstruction quantization(URQ)

■ In-loop deblocking filtering

■ Sample adaptive offset(SAO)

■ Loop filtering across slice/tile boundary

■ Sequence change detection

## H.264/AVC Codec

## Overview

H.264/AVC codec is used for various kinds of multimedia products such as 4K Ultra HD TV, set-top box, and surveillance video camera. It can decode compressed video in a format of H.264 BP/MP/HP up to 4096x2304 resolution. It is also able to encode video into H.264 format of bitstream up to UHD. The H.264/AVC codec can perform simultaneous real time encoding, decoding, or both encoding and decoding of multiple video streams at multiple resolutions.

H.264/AVC is designed to optimally share most of the sub-blocks that are used in common for video processing, which contributes to the ultra-low power and low gate count.

## Architecture



H.264/AVC codec is connected with a host CPU system via 32-bit AMBA 3 APB bus for system control and 64-bit AMBA3 AXI for data.

## Features

## H.264/AVC Encoder

■ Capable of encoding BP/MP/HP profile @L4.2

■ Max resolution: 4096x2304, Min resolution: 96x16

■ Performance 1920x1080 60fps@266MHz

■ Supports MVC Stereo High profile with interview prediction only for anchor picture

■ The encoder uses only one reference frame for the motion estimation.

■ Rate Control(Frame Level and MB Level)

■ Supports CABAC/CAVLC

■ low delay coding

-less than 1ms delay for starting encoder with sub-frame synchronization

■ 2D cache for motion compensation to reduce external memory access

■ Region of Interest(ROI) picture encoding

■ 1/4-pel and 1/2-pel accuracy motion estimation with programmable search range up to [+/-64H, +/-48V]

■ Available block size can be configurable and 16x16,16x8,8x16,8x8 block size are supported

■ Intra prediction

-Luma I4x4 Mode: 9modes

-Luma I8x8 Mode: 9modes

-Luma I16x16 Mode: 4 modes(vertical ,Horizon ,DC, Plane)

-Chroma Mode: 3modes(Vertical, Horizon ,DC)

■ Filed encoding is available without PAFF, MBAFF

## H.264/AVC Decoder

■ Fully compatible with the ITU-T Recommendation H.264 specification in BP, MP and HP

■ Supports MVC Stereo High profile

■ Max resolution: 4096x2304, Min resolution: 96x16

■ Supports CABAC/CAVLC

■ Variable block size (16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4)

■ Error detection, concealment and error resilience tools with FMO/ASO support

## JPEG Codec

## Overview

JPEG Codec is a high performance hardware design that can perform the JPEG extended sequential and M-JPEG decoding and encoding. The JPEG codec supports up to 155M pixel/sec encoding for 4:2:2 color format image and 100M pixel/sec decoding for 4:4:4 color format image. Huffman coding with up to 3 tables are supported while the arithmetic coding tool is not supported.

The main features of JPEG Codec are compliant with JPEG extended sequential. The image size up to 32768x32768 is supported for decoding and encoding.

## Architecture



It is connected with the system via the 32-bit AMBA3 APB bus for system control and 64-bit AMBA3 AXI bus for data throughput, and takes advantage of on-chip memories to achieve high performance.

## Features

■ Extended sequential ISO/IEC 10918-1 JPEG compliance

■ Max Resolution: 32768x32768

■ Performance

-Encoder, 4:2:0 220M pixel/sec @200MHz

-Decoder, 4:2:0 230M pixel/sec @200MHz

■ Three component in a scan(interleaved only)

■ 8bit or 12bit samples for each component

■ Support 4:2:0 , 4:2:2, 4:4:0, 4:4:4 and 4:0:0 color format

■ Minimum encoding size is 16x16 pixels

■ Support NV12/NV16(CbCr interleaved), NV21/NV61(CrCb)

■ Support ROI(Region of Interested – decoder only)

■ Support 422/444 packed mode for all color formats

■ Encoder partial mode and rotator-mirror

■ Decoder partial mode and rotator-mirror

■ Decoder down-sampler and PPM format

## Display Engine

## Overview

Display controller defines a high-performance optimized-area display core that can be used for reading rendered images from the frame buffer to the display. In addition to providing hardware cursor patterns, the display controller performs format conversions, dithering and gamma corrections. This controller includes support for parallel pixel output and is easily adapted to external serialization logic, for example HDMI.

## Block Diagram

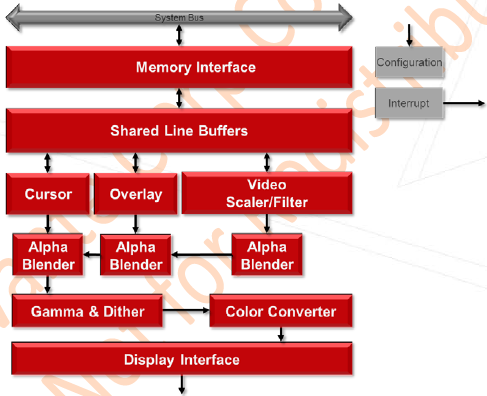


Fig. 1.1 Display controller module block diagram

Following are various modules in the Display Controller:

* **Host Interface**

Allows the core to communicate with external memory and the CPU through the AXI and AHB buses.

* **Front End and Pixel Pipeline**

Single display pipeline supports linear frame buffers for RGB and YUV inputs.

* **Cursor**

Provides hardware cursor functionality

* **Dither**

Provides and indexable look-up-table for dither function

* **Gamma Correction**

Provides a gamma correction SRAM for its function

* **Display Interface**

Supports Parallel Pixel Output with 24-bit Data, Hsync, Vsync, Data enable. Enable adapted to external serialization logic, e.g. HDMI.

## Features

* Video Timing Generation
* HSYNC, VSYNC, DE signals
* Programmable timers
* Display Interface
* Parallel Pixel Output with 24-bit Data, HSync, VSync, Data enable
* DPI 24-bit, 18-bit(2 configs) and 16-bit(3 configs) support
* Easily adaptable to external serialization logic, e.g. HDMI.
* Display
* Single display
* Maximum display size: 4Kx2K
* Sync and bland signals
* Gamma and dither tables
* Input Formats
* ARGB2101010,A/XRGB8888,A/XRGB1555,RGB565,A/XRGB4444
* Index1/2/4/8
* YUV422 packed & semi planar (YUY2, UYVY, NV16)
* YUV420 semi-planar (P10), NV12 and YUV420 semi-planar (10bit, 1 pixels 2 bytes)
* NV12 (10bit, 3 pixels 4 bytes) and NV16 (10bit, 3 pixels 4 bytes) for overlay only
* Format Conversion
* Pixel inputs accepted from multiple RGB and YUV formats
* Color Space Conversion BT.2020 and BT.709
* Pixel output is 24bit RGB in multiple formats
* Output Formats
* ARGB2101010/DPI\_D16CFG1/DPI\_D16CFG2/DPI\_D16CFG3/DPI\_D18CFG1/DPI\_D18CFG2/DPI\_D24/DPI\_D30
* Hardware Cursor
* Supports ARGB88 and Mask cursor formats
* Color
* A separate Look Up Table for Dither
* A separate Look Up Table for Gamma Correction
* Overlay with coordinate generator
* Alpha Blending
* Filter and Scaling
* Vertical and horizontal scaling
* Horizontal 3/5 tap; vertical 3 tap
* Programmable filter order
* 15.16 fixed point scaling factor

## DDR

## Overview

DDR subsystem contains two parts, DDR controller and DDR PHY. DDR controller has six axi ports which connect to axi masters through axi bus. DDR controller receives transactions from axi masters. These transactions are queued internally and scheduled for access in order to SDRAM while satisfying SDRAM protocol timing requirements, transaction priorities, and dependencies between the transactions. DDR PHY provides an interface between the controller and external SDRAM devices. DDR controller communicates with DDR PHY through FDI interface.

## Architecture

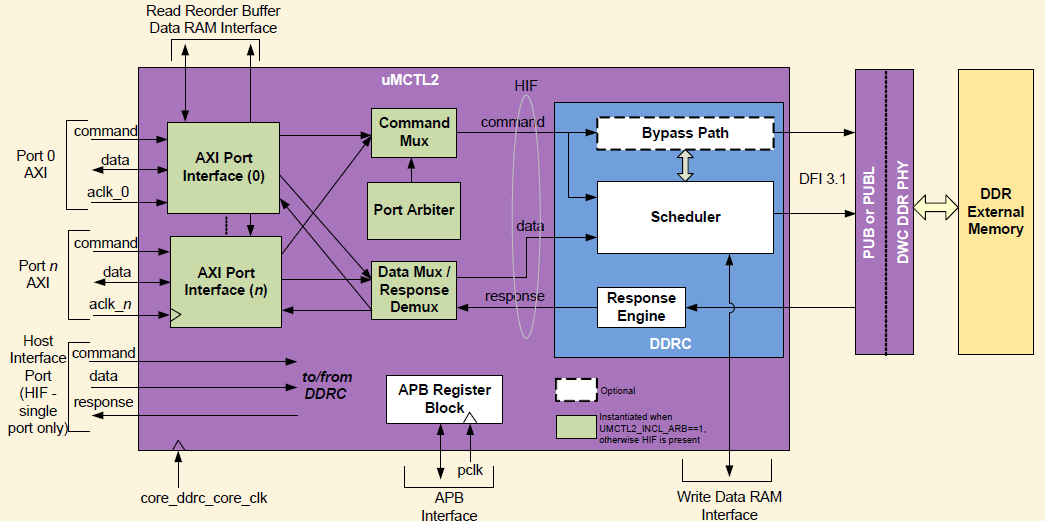


Figure : DDR subsystem Architecture

## Features

DDR subsystem has following features:

* Supported SDRAM types include:

DDR4

LPDDR3

DDR3

DDR3L

DDR3U

* Supported SDRAM speed include:

DDR4 2400Mbps

LPDDR3 2133Mbps

DDR3 2133Mbps

DDR3L 1866Mbps

DDR3U 1600Mbps

* 64/32/16bits configurable SDRAM data width
* Inline ECC supported
* Programmed supported 1T or 2T timing
* 6 axi ports
* QOS supported
* SSC(Spread Spectrum clock) supported

## USB

## Features

* Same programming model for SuperSpeed (SS), High-Speed (HS), Full-Speed (FS), and  
  Low-Speed (LS)
* 150Mhz 32-bit slave interface for CSR and RAM Debug access
* 150Mhz 64-bit master interface for internal DMA access
* Independent Little or Big Endian (byte invariant) mode selection for both Slave and Master interface
* Independent Endian selection between descriptor and data fetch
* Hardware/software race condition avoidance in systems with bridge
* Concurrent read and write operations to get the best performance of USB 3.0 duplex operation in AXI bus mode
* AHB Slave interface:
* Supports all AHB Burst types, including WRAP# supports
* Does not generate Split, Retry, or Error responses
* AXI Master interface:
* The software can select the allowable burst lengths (single, burst 4/8/16/32/64/128/ 256 as pplicable) by programming the GSBUSCFG0 register in CSR
* Handles fixed burst address alignment. Starting address of an burst is always aligned to burstsize; even though AXI specification does not require this, most memory/cache access require this. For example, if a burst of 8 is selected in a 64-bit wide data bus, the burst starting address will be 0, 64, 128, etc
* Software can select the number of outstanding read/write requests (1 to 16) made by the AXI Master Interface by programming the GSBUSCFG1 register in CSR. Read and write channels operate independently and may each have their own outstanding requests limited by the programmed value in GSBUSCFG1
* Capable of performing zero-wait state data transfers
* Handles the AXI 4k boundary, Transfers are split to prevent crossing of the 4k boundary. Optionally, handles break up transfers on the 1k boundary by utilizing a software programmable option in the GSBUSCFG1 in the CSR
* Supports cacheable accesses on the AXI Master Interface
* Does not support Atomic and Protected accesses on AXI Master

## Block Diagram



Fig.1.1 USB block diagram

## USB 3.0 Device Features

* Up to 8 bidirectional endpoints, including control endpoint 0
* Software directly handles non-timing-critical and rarely occurring tasks, such as control transactions
* Flexible endpoint configuration allows a single area optimized configuration meeting multiple applications, software can map the USB endpoint to an endpoint resource number, even if the USB endpoint numbers are not contiguous.
* Dynamic mappable TxFIFOs to support more Tx-endpoints than the physical FIFOs
* Simultaneous IN and OUT transfer support
* 4 Gbps IN and 4 Gbps OUT bandwidth (interpacket delays and protocol overhead included)
* Descriptor caching and data pre-fetching for predictable performance in high-latency systems
* Hardware handles ERDY and burst
* Hardware handles all data transfers Capability to set up multiple transfers without interrupting the host processor on every transfer
* Stream-based bulk endpoints with controller automatically initiating data movement
* Isochronous endpoints with isochronous data in data buffers or external hardware FIFOs
* Flexible Descriptor with rich set of features to support buffer interrupt moderation, multiple transfers, isochronous, control, and scattered buffering support

## USB 3.0 xHCI Host Features

* Up to 64 devices
* xHCI1.0 compatible
* Standard or open-source xHCI and class drivers
* xHCI features:
* Aggressive power management
* Clean software and hardware interface
* Memory access optimization
* Interrupt Moderation
* Descriptor caching for predictable performance in high latency systems
* Concurrent IN and OUT transfers to get the full 8 Gbps duplex throughput (interpacket delays and protocol overhead included)
* Concurrent USB 3.0/2.0/1.1 traffic:
* Designed so that USB 2.0 Devices do not degrade the overall throughput
* Net bandwidth increased to 8.48 Gbps (8 Gbps USB 3.0 bandwidth plus 480 Mbps USB 2.0 bandwidth)
* Dual power rail designs with hibernation feature

## MIPI CSI System

## Overview

CSI2 (Camera Serial Interface2) defines protocols between a peripheral device (camera) and a host processor. It provides a cost efficient solution for mobile device. MIPI CSI System integrates the function of CSI2 TX and RX. RX receives image data from MIPI CSI sensor and TX sends the processed image data out to display.

## Architecture



Figure: MIPI CSI System Architecture

The MIPI CSI System consists of four blocks:

DPHY: the physical layer of MIPI CSI and DSI.

INTERCONNECT: the control logic of data exchange between transmitter or receiver and DPHY.

TWO\_LANE\_RX: two lane controller of CSI receiver.

FOUR\_LANE\_RX: four lane controller of CSI receiver.

FOUR\_LANE\_TX: four lane controller of CSI transmitter.

## Features

* 1.5Gbps per lane in HS mode, up to 6Gbps in one four lane link.
* In RX mode, support up to 4 4lane link or 8 2\_lane link.
* In TX mode, support one 4lane (can configured as 1~4 lane) link.
* Follow data type can be supported in TX mode.
* RGB888
* YUV420 8-bit (legacy) / 8-bit / 10-bit
* YUV422 8-bit / 10-bit
* YUV444 8-bit / 10-bit
* Follow data type can be supported in RX mode.
* YUV420 8-bit (legacy) / 8-bit / 10-bit / 8-bit (CSPS) / 10-bit (CSPS)
* YUV422 8-bit / 10-bit
* RGB888 / RGB666 / RGB565 / RGB555 / RGB444
* RAW6 / RAW7 / RAW8 / RAW10 / RAW12 / RAW14 / RAW16
* User define data type

## HDMI Receiver

## Overview

HDMI 1.4 receiver enables digital video, audio, and control data reception that is transmitted via HDMI. It has a High-Bandwidth Digital Content Protection (HDCP) 1.4 decryption and control unit that enables the reception of content protected material. Apart from standards like CEC, it also supports interfaces like APB, Video Output, Audio Output, Interrupt output, and HDMI PHY interface.

## Architecture



**HDMI RX Structure**

## Features

Video input formats:

❑ RGB 4:4:4

■ 8-bit normal color mode

■ 10-, 12-, and 16-bit deep color mode

❑ YCBCR 4:2:2 with 8-, 10-, and 12-bit color depth

❑ YCBCR 4:4:4

■ 8-bit normal color mode

■ 10-, 12-, and 16-bit deep color mode

Receipt of compressed and uncompressed encoded audio data:

❑ L-PCM

❑ L-PCM multi-channel

❑ Standard bit-rate compressed audio

❑ 1-bit audio (DSD)

❑ Compressed 1-bit audio (DST) and High Bit-Rate (HBR) compressed audio

CEC Controller

CEC is a protocol that provides high-level control functions between all of the various audiovisual products in your environment. The CEC bus allows all products in the system to potentially discover and communicate with each other

## PCI Express

## Overview

The Sirius PCIe provides an AXI4 bridging capability for directly adding the PCI Express link to Network on Chip through AXI system fabric. The bridge interconnects the AXI interfaces within an AXI-embedded system with a remote PCIe link, as either a root complex port or as an endpoint port. The bridge supports three AXI interfaces:

■ one masters

■ one link slave

■ one DBI slave

The AXI master interfaces enable a remote PCIe device to read and write to an AXI slave connected to the NOC, like DDR slave. The AXI link slave interface enables an AXI master, Top DMAC for example, to read and write through the NOC to a remote PCIe device. The DBI slave enables an AXI master, A7 core for example, to access the PCIe’s registers. The Sirius PCIe supports up to x2 Gen1/Gen2 lanes, and the max throughput is 10 Gb/s.

## Architecture

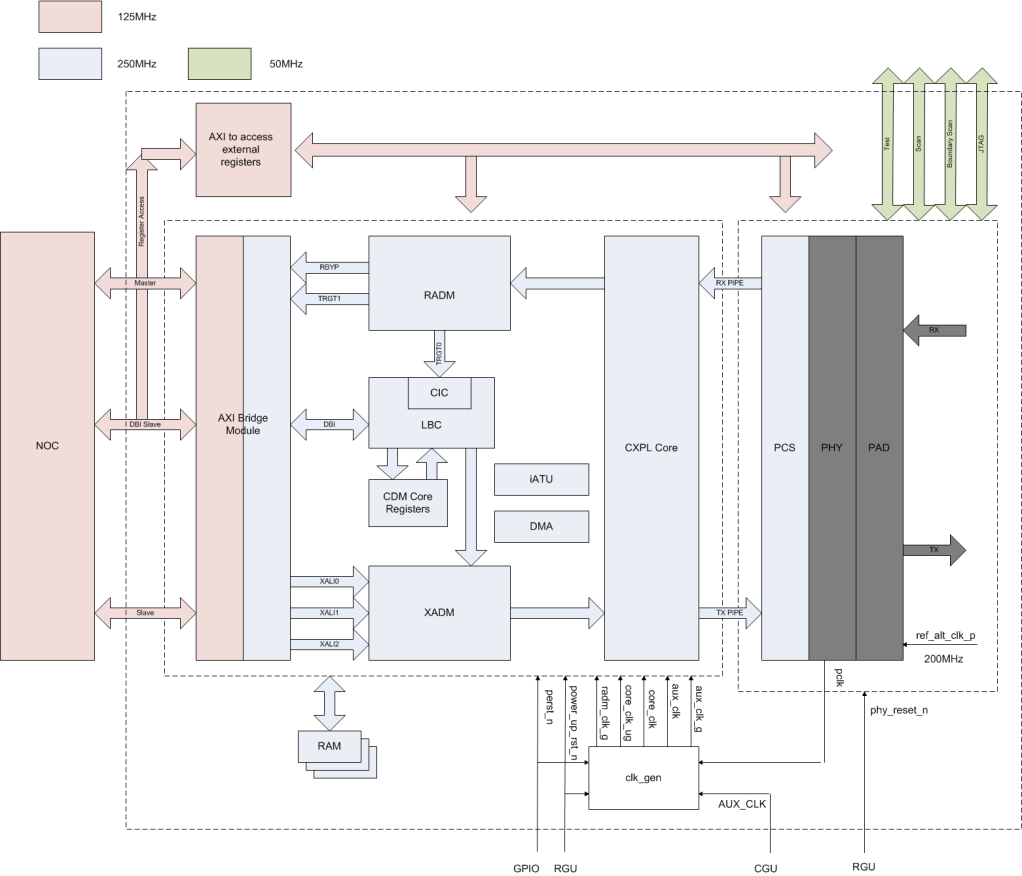


Figure 1.1 PCIe Architecture

PCIe module includes two parts: PCIe DM controller and PCIe PHY, as shown in Figure 1.1. The physical layer is split across the PIPE and controller such that the MAC functionality (LTSSM, lane-to-lane deskew) is in the controller and the PHY functionality is implemented in the PIPE-compliant PHY. The PHY is outside of the controller, interfacing through the standard PIPE interface.

## Features

* The Sirius PCI Express supports：  
  ■ All non-optional features of the *PCI Express 2.0*  
  ■ x2 Gen1, Gen2 lanes  
  ■ 32-bit Internal Datapath Operating at 125 or 250 MHz  
  ■ Advanced Power and Clock Management

■ Internal Address Translation Unit  
 ■ Internal MSI-X Generation Module

■ AXI4 bridge

■ Two Embedded DMA

■ Automatic Lane Reversal

■ Upconfigure Support

■ Bypass, Cut-through, and Store-and-forward Queue Modes for Rx TLPs

■ Three Application Transmit Clients

■ Type 0 / 1 Configuration space

■ PHY Control Registers access

■ 5-Gbps data transmission rate

■ Integrated PHY includes transmitter, receiver, PLL, digital core, and electrostatic discharge (ESD) protection circuits

■ Excellent performance margin and receiver sensitivity

■ Robust PHY architecture that tolerates wide process, voltage, and temperature variations

■ Low-jitter PLL technology with excellent supply isolation

■ Built-in Self-Test (BIST) features for production, at-speed testing on any digital tester

■ Visibility and controllability of hard macro functions through programmable registers in the design

## USB/DP Type-C

## Overview

USB Type-C is one kind of high speed interface in Sirius. The USB Type-C receptacle, plug

and cable provide a smaller, thinner and more robust alternative to existing USB interconnects. The Sirius USB Type-C solution targets its use in a variety of platforms ranging from notebooks, PCs, Monitors, down to tablets and smart phones. The mainly external feature of this block is enhances ease of use by being plug-able in either direction between host and device. Besides, DP alternate mode is supported, which means Display port can be combined with USB operation over the same USB Type-C cable. It supports USB3.0 DRD mode, which is GEN1 SS 5.0Gbps; it supports VESA DP v1.2 version, where the maximum link rate is HBR2 5.4Gbps, with lane up to 4, supports HDCPv1.3 and HDCP v2.2 authentication and encryption.

## Architecture

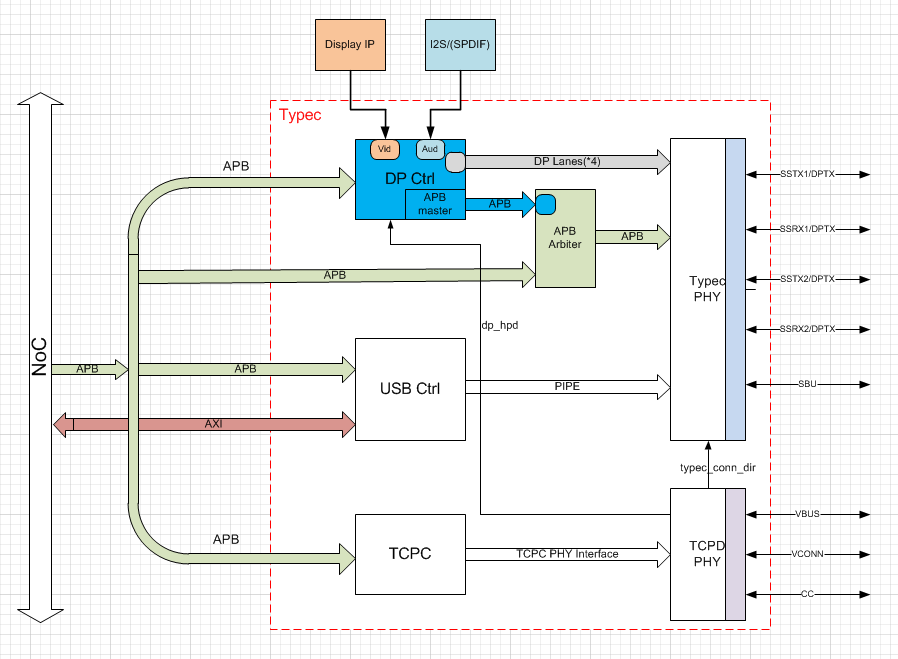


Figure USB Type-C architecture

It mainly includes five parts: DP controller, USB3.0 controller, Type-C Port controller, Multi-protocol PHY and TCPD PHY.

* DP controller: DP transmitter controller; enabled by Type-C Alternate mode.
* USB3.0 controller: USB3.0 protocol supported; dual role capability that enables connection to external USB devices as either a Host or Peripheral Device.
* Type-C Port controller: received Power Delivery Message from TCPM, then encapsulates and encodes PD message; hardware acceleration for CRC generation/checking; BMC encoding/decoding; Configuration channel(CC) logic.

## Features

* Support DP v1.2 only
  + Maximum support 4 lanes，maximum link rate is 5.4Gpbs
  + Video timing of VIF compiles with CEA-861-F, supported video format are RGB, YCbCr444, YCbCr422 and YCbCr420, supported 6bpc, 8bpc and 10bpc
  + Supported Color Space Conversion(CSC): RGB to YCbCr, YCbCr to RGB, YCbCr422 to YCbCr444 and YCbCr444 to YCbCr422
  + Support I2S interface audio, maximum physical channel is 4
  + Support HDCP v1.3 and HDCP v2.2 authentication and content encrypted
* Support USB3.0 SuperSpeed only
  + USB3.0 PIPE interface compliant
  + USB3.0 U1/U2/U3 support
  + Host Mode is based on xHCI 1.0 protocol, maximum of 64 configurable total slots support, 32 endpoints per slot
  + Up to 15 IN and 15 OUT configurable/programmable endpoints
* Support concurrent mode, where both DP and USB trans at the same time.
* BMC used for USB PD communication over the CC1/CC2 wire.
* Dynamic overcurrent protection on Vbus and Vconn.

## Ethernet

## Overview

Implements an Ethernet Media Access Controller compatible with the 10/100/1000 Mbps IEEE 802.3 and 1Gbps IEEE 802.3-2002 specifications. The controller provides half- or full-duplex operation, supports jumbo frames, and optionally provides a reach set of sta-tistics counters enabling station management. A host processor can control the operation of the core via a slave interface that provides access to its control and status registers. The controller features two master ports for data transfers, one for transmit and one for receive. The two DMA engines use buffer descriptors to automatically transfer data from local FIFOs to an external shared memory. The core supports 32-bit AMBA/AHB SoC buses;

## Block Diagram



Fig. gmac controller module block diagram

## Features

■ Programmable 10/100 or 1000 Mbps operation   
■ IEEE 802.3-2002 specification with preamble, start-of-frame delimiter (SFD), and CRC gener-ation and checking   
■ Full- or half-duplex operation   
■ CSMA/CD procedures for half duplex   
■ Flow control for full duplex   
■ Jumbo frames   
■ Flexible address filtering

■ Extensive statistics counters   
■ Detection of too long or too short packets, with programma-ble length limits   
■ Media Independent Interface (MII) for 10/100Mbps   
■ Gigabit Media Independent In-terface (GMII) for 1Gbps   
■ MDIO interface for PHY configu-ration and management

## Quad-SPI Flash Controller

## Overview

The SPI flash controller in Sirius is one instruction based controller. The command instruction code is 64 bit pattern. The controller decodes the command code and issue SPI instruction to flash chip. The controller can support fast read/dual read/quad read mode. The SPI clock frequency can be set to 25Mhz/50Mhz/100Mhz.

## Block Diagram



## Features

The SPI Flash Controller has the following features:

* Support single/quad/qpi operation
* SPI clock can be programmed to 25Mhz/50Mhz/100Mhz
* Support 3-byte/4-byte address mode
* Support DMA read operation
* Total 32 instruction space
* Instruction can be programmed
* Default instruction is used for winbond flash

## SD Card Controller

## Overview

The SD card module in Sirius has one slave AHB interface, one AHB master interface and an internal DMA engine. The SD card controller is configured by the AHB slave interface. There are two ways to transfer data from the host memory to the card device and vice versa. One is that the data is transferred through the AHB slave interface by the AHB master such as CPU and son on in the system, and the other one is that the internal DMA engine of the SD card controller transfers data between the host and the card through the AHB master interface. The latter data transfer approach is much more efficient in most situations.

## Block Diagram



Fig. SD card module block diagram

The SD card module consists of two main functional blocks, which are illustrated in Fig. 4.1.

* Bus Interface Unit (BIU) – Provides AMBA AHB and DMA interfaces for register and data read/writes.
* Card Interface Unit (CIU) – Takes care of the SD card protocols and provides the clock management.

## Features

* The following are features of the SD card module in Sirius:  
  ■ Supports Secure Digital memory protocol commands  
  ■ Supports Secure Digital I/O protocol commands   
  ■ 32-bit addressing supported for Master Interface and Slave Interface  
  ■ Support for 1.8/3.3V of operation control
* Internal DMA Block Features   
  ■ Single-channel; single engine used for Transmit and Receive, which are mutually exclusive   
  ■ Dual-buffer and chained descriptor linked list  
  ■Descriptor architecture allows large blocks of data transfer with minimum CPU intervention; each descriptor can transfer up to 4KB of data in chained mode and 8KB of data in dual-buffer mode  
  ■ Comprehensive status reporting for normal operation and transfers with errors   
  ■ Programmable interrupt options for different operational conditions
* AHB Master Interface Features  
  ■ Supports 32-bit data and 32-bit addressing  
  ■ Supports split, retry, and error AHB responses; does not support wrap   
  ■ Allows selection of AHB burst type through software
* Bus Interface Unit (BIU) Features   
  ■ Supports data widths of 32 bits   
  ■ Does not generate split, retry, or error responses on the AMBA Slave AHB bus  
  ■ Supports FIFO over-run and under-run prevention by stopping card clock
* Card Interface Unit (CIU) Features  
  ■ Supports Command Completion Signal and interrupts to host  
  ■ Supports Command Completion Signal disable  
  ■ Supports CRC generation and error detection  
  ■ Supports programmable baud rate. Supports up to 4 clock dividers to support simultaneous  
  operation of multiple cards with different clock speed requirements   
  ■ Supports card detection and initialization  
  ■ Supports write protection  
  ■ Supports SDIO interrupts in 1-bit and 4-bit modes  
  ■ Supports SDIO suspend and resume operation  
  ■ Supports SDIO read wait  
  ■ Supports block size of 1 to 65,535 bytes  
  ■ Supports Busy Clear Interrupt for the write data transfers to the card

UHS-1 and Voltage-Switching Features  
■ Support for UHS 50 and UHS 104 cards with the following speeds, frequencies, and voltages, as appropriate for each card:  
❑ Default Speed Mode – 25 MHz, 3.3V  
❑ High Speed mode – 50 MHz, 3.3V  
❑ SDR12 – 25 MHz, 1.8V  
❑ SDR25 – 50 MHz, 1.8V  
❑ SDR50 – 100 MHz, 1.8V  
❑ SDR104 – 208 MHz, 1.8V  
❑ DDR50 – 50 MHz, 1.8V  
■ Voltage switching

## Low-Speed Peripherals

## UART Controller

## Overview

The UART is modeled after the widely used 16550 UART. The register definition is quite the same except that the register offset address in aligned to 32-bit word boundaries due to the 32bit APB bus interface of this UART module.

## Block Diagram



Fig. 2.1 UART functional block diagram

## Features

* 9 on-chip general UART controller inside Sirius
  + 5 simple UART without DMA feature
  + 1 full UART with DMA feature
  + 3 simple UART with DMA feature
* DMA-based or interrupt-based operation
* For all UART, 16-byte depth transmit and receive FIFO
* 16550 compatible function
  + 5-8 data bits per charater
  + Optional parity bit
  + 1/1.5/2-bit stop bit
* Programmable serial data baud rate as calculated by the following:

Buadrate = sclk/(16xdivisor)

## I2C Controller

## Overview

The I2C controller provides support for a communication link between integrated circuits on a board. It is a sample two-wire bus which consists of a serial data line (SDA) and a serial clock (SCL). The Sirius provides five I2C Controller to enable system software to communicate serially with I2C buses. Each I2C controller operates as a master.

## Block Diagram



Fig. 2.1 I2C Controller functional block diagram

The I2C controller consists of an APB interface, an I2C interface, FIFO logic to buffer data and shift logic for parallel-to-serial and serial-to-parallel conversion. Control logic is responsible for implementing the I2C protocol. The I2C controller is instantiated in Sirius through APB interface and APB clock input (PCLK) is for both APB bridge and I2C internal control logic.

## Features

* 5 on-chip I2C controller in Sirius
* All I2C support DMA feature
* Support multi-slave operation
* 16 depth transmit and receive buffers
* Three speeds:
  + Standard mode (0 to 100Kb/s)
  + Fast mode (≤400Kb/s)
  + High-speed mode (≤3.4Mb/s)
* Clock synchronization
* 7-or-10-bit addressing
* 7-or-10-bit combined format transfers
* Bulk transmit mode
* Handles bit and byte waiting at all bus speeds

## SPI Controller

## Overview

There are 4 SPI master controllers and 2 SPI slaves instantiated in Sirius through APB interface. The APB input clock (PCLK) is for both APB interface and SPI controller internal control logic. The PCLK is 300MHz for AHB/APB bridge.

## Block Diagram



Fig. 2.3 SPI Controller functional block diagram

## Features

* 6 on-chip SPI masters inside Sirius
  + 4 SPI masters, 2 support one chip-select and 2 support 5 chip-selects output
  + 2 SPI slaves
  + All SPI support DMA feature
* DMA-based or interrupt-based operation
* All SPI with 16 depth transmit and receive buffers
* Independent masking of interrupts
* Programmable features:
  + Clock bit-rate – dynamic control of the serial bit rate of the data transfer
  + Data Item size (4 to 16 bits)

## CAN controller

## Overview

There are four controller area network (CAN) controllers instantiated in Sirius through APB interface. The CAN controller bus consists of two wire, CAN-H and CAN-L, and the bus level is determined by their potential difference.

## Block diagram



Fig. 2.4 CAN controller functional block diagram

## Features

The CAN controller in Sirius offers the following features:

* Support CAN2.0B
* Data rate up to 1Mbit/s
* Programmable baud rate prescaler (1 to 1/256)
* 11-bit standard and 29-bit extended identifiers
* Two transmit buffers
  + One Primary Transmit Buffer (PTB)
  + Optional configurable Secondary Transmit Buffer (STB)
* Independent and programmable internal 29 bit acceptance filters
* Configurable interrupt sources

## Watchdog Timer

## Block Diagram



Fig. 2.5 Watchdog timer functional block diagram

## Features

* 32-bit APB bus interface, compliance with AMBA 2.0 specification
* Configurable APB data bus widths of 8, 16 and 32 bits
* Configurable watchdog counter width of 16 to 32 bits
* Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
* If a timeout occurs watchdog timer can perform one of the following operations:
  + Generate a system reset
  + Generate an interrupt, restarts the timer, and if the timer is not cleared before a second timeout occurs, generate a system reset
* Test mode signal to decrease the time required during functional test

## Timer with PWM

## Overview

There are ten 32-bit timers connected to AHB/APB bus bridge in Sirius. The timer is used for both general-purpose time counting and PWM generating. The timer optionally generates an interrupt when the 32-bit binary count-down timer reaches zero.

## Block Diagram



Fig. 2.6 Timer functional block diagram

## Features

* 32-bit APB bus interface, compliance with AMBA 2.0 specification
* Up to 8 programmable timers
* Supports for two operation modes: free-running and user-defined count
* Supports interrupt generation
* Supports PWM generation

## GPIO

## Overview

There are 5 GPIO modules instantiated in Sirius. Each GPIO includes 32 I/O pins. So Sirius supports up to 160 I/Os.

## Block Diagram



Fig. 2.7 GPIO functional block diagram

## Features

* 32-bit APB bus interface, compliance with AMBA 2.0 specification
* Up to four ports, A to D, which are separately configurable
* Separate data registers and data direction registers for each signal
* Configurable interrupt mode for Port A
* Supports digital debounce
* Supports up to 160 I/O pins
* All GPIOs are always in input direction in default after power-on-reset
* The driver strength for all of GPIOs is software-programmable

## EMMC Controller

## Overview

DWC\_mshc provides a flexible bus interface that enables you to integrate DWC\_mshc into embedded applications for system-on-a-chip (SoC) designs. DWC\_mshc has an AXI and AHB master interface that supports 32-bit and 64-bit address and data bus. Besides supporting non-DMA mode, DWC\_mshc supports various DMA options such as SDMA, ADMA2, and ADMA3 as specified in the SD Host Controller Standard.

## Block Diagram



Fig. 1.1 emmc controller module block diagram

## Features

■ Supports eMMC protocols including eMMC 5.1  
■ Supports the following data transfer types for eMMC modes

❑ CPU

❑ SDMA

❑ ADMA2

❑ ADMA3  
■ Supports independent Core, Slave Interface and Master Interface clocks  
■ Supports gating of core base clock if Host Controller is inactive  
■ Support context aware functional clock gates  
■ Applications can gate the slave interface clock if Host Controller is inactive  
■ Data Buffering  
 ❑ Configurable buffer depth. Configurable in coreConsultant/coreAssembler

❑ Automatic packing/unpacking of data to fit buffer width

■ Interrupt Outputs

❑ Combined and separate interrupt outputs

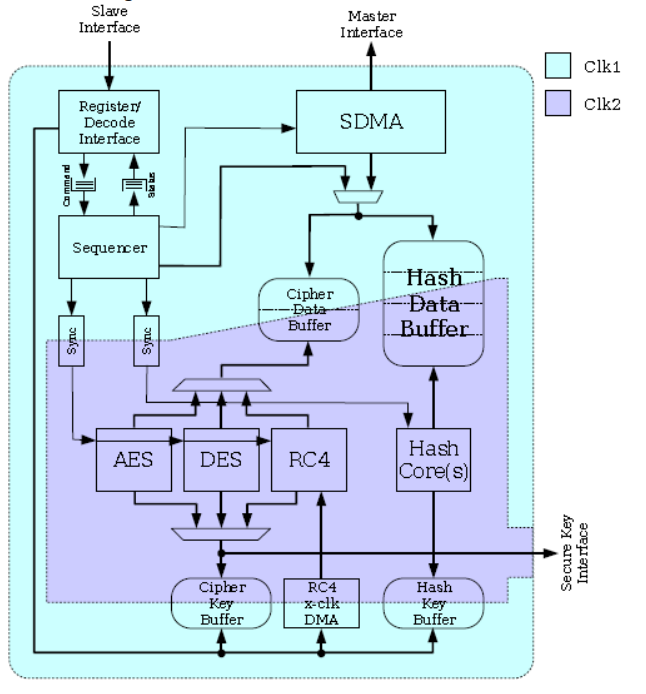
❑ Supports interrupt enabling and masking  
■ Supports Slave-Only mode as a configuration option for better area and power efficiency  
■ Supports tuning  
■ eMMC Tuning using CMD21 (eMMC)  
■ Mode 1 Re-Tuning – Host driver maintains the re-tune timer  
■ Fully Software driven Tuning/Re-tuning operations

## Secure Subsystem Feature

## Secure CPU

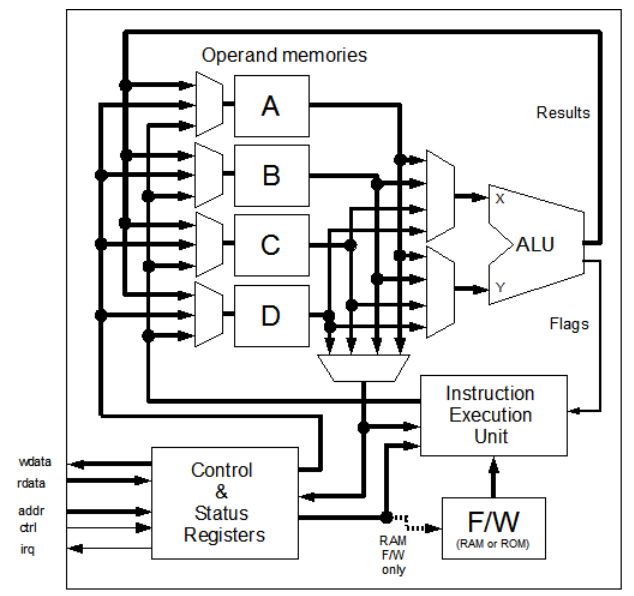
* Secure CPU contains an embedded microprocessor. As such, it requires access on a bus to fetch instructions from memory which is then stored in a local secure cache and occurs over the master interface
* Support secure boot and secure storage

## Security Protocol Accelerator(SPACC)



* Cipher algorithms: AES, DES
* Cipher modes: ECB, CBC, CTR, OFB, CFB, f8, XTS, CCM, GCM
* Hash (MAC) algorithms: MD5, SHA-1, SHA-256, SHA-512, SHA-512/256, AES-XCBC-MAC, AES-CMAC, CRC-32-IEEE 802.3
* Hash modes: Raw, SSLMAC, HMAC
* DDT packet descriptors
* Multiple cryptographic context support. The number of contexts supported is a configurable parameter
* AXI bus interfaces
* Full DMA master capabilities to bring data into the core and write back. Packet data traverses the bus only twice per packet (once for the read operation, once for the write operation)
* Endian support is configurable to either big-endian or little-endian via an external pin
* Support for secure key area for block cipher algorithms (AES)
* Secure bus available to securely operate in systems which differentiate between secure and normal processing modes

## Public Key Accelerator (PKA)



* RSA (with or without CRT): 256, 512, 768, 1024, 1536, 2048, 3072, and 4096-bit
* ECC-GF(p): 160, 192, 224, 256, 320, 384, 512bit

## True Random Number Generator (TRNG)

* Internal random (re)seed operation
* 256-bit random number generation
* Two separate reseed reminder schedules provide autonomous background reseeding (build-time configuration option)
* Shift register compatible output stream for auxiliary uses such as, DPA, TA, IPsec, and so on (for TRNG product only)
* Status/control I/O for allowing external monitoring of internal actions and states (for TRNG product only)
* Glue-less interface to ESM family entropy port (TRNG for ESM product only)

## Secure UART



* Work at 150MHz
* Simple UART without DMA feature
* For all UART, 16-byte depth transmit and receive FIFO
* 16550 compatible function
  + 5-8 data bits per charater
  + Optional parity bit
  + 1/1.5/2-bit stop bit
* Programmable serial data baud rate as calculated by the following:

Buadrate = sclk/(16xdivisor)

## Secure Timer



* 32-bit APB bus interface, compliance with AMBA 2.0 specification
* Up to 8 programmable timers
* Supports for two operation modes: free-running and user-defined count
* Supports interrupt generation

## Secure Watchdog



* 32-bit APB bus interface, compliance with AMBA 2.0 specification
* Configurable APB data bus widths of 32 bits
* Configurable watchdog counter width of 16 to 32 bits
* Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
* If a timeout occurs watchdog timer can perform one of the following operations:
  + Generate a system reset
  + Generate an interrupt, restarts the timer, and if the timer is not cleared before a second timeout occurs, generate a system reset
* Test mode signal to decrease the time required during functional test

## OTP (one-time programmable) Controller

## Features

* Read and program 32bits from/to XPM by AXI interface.
* The XPM is divided into 35banks, each bank’s right is decided by access right table. the 0~31 bank has 1Kbs, the 32~34 banks has 32Kbs
* The OTP controller can by accessed in secure mode only.
* The first 3Kbs are protected by checksum.
* The password register can only be written once after power-on reset.

## Trust Zone controller

## Overview

Trust zone Controller(TZC) is an AMBA compliant SoC peripheral. It performs security checks on transactions to memory or peripherals. You can use the TZC-400 to create up to eight separate regions in the address space, each with an individual security level setting. Any transactions must meet the security requirements to gain access to the memory or peripheral. You can program the base address, top address, enable, and security parameters for each region.

## Block Diagram



Fig. 1.1 TZC functional block diagram

## Features

* 6 TZC filters included inside Sirius
* The ability to define up to 8 address regions on the address map
* A default base region to cover all remaining portions of the address map
* Software programmable security access permissions for each address region through an APB4 interface
* Identity-based filtering of Non-secure accesses
* Support for 16 outstanding transactions on the normal paths

## DVP interface

## Overview

* DVP interface contains up to 2 DVP input (BT656/BT1120/BT601), 1 DVP output for BT656/BT1120/BT601, 1 RGB (24bit RGB data) LCD output.

## Architecture

* 
* **DVP Interface Structure**

## Functional description

* DVP interface support up to 2 DVP input (BT656/BT1120/BT601), up to 1080p@60fps
* Support 1 14bit camera interface up to 1080p@60fps
* Support 1 DVP output for BT656/BT1120/BT601, up to 1080p@60fps
* Support 24bit RGB parallel interface LCD panel, up to 1080p@60fps

## Internal Video interface block

## Overview

The VIF (Video Interface) is used to sync and control the multi video source data from peripheral camera. Up to 11 video sources are supported in the system, contains 1 HDMI interface, 2 DVP interface and 8 MIPI CSI interface. The video data can be written to DDR or sent to ISP directly. What’s more, audio data input is also supported and can be written to DDR.

## Architecture



Figure: VIF Architecture

The VIF mainly consists of three blocks:

VIDEO2SYNC: Sync the source data from different source clock domain to one clock domain.

VIDEO\_ISPIF: Process the source data and send it to ISP.

VIDEO\_BPIF: Process the source data and write it to DDR.

## Features

* Build in test with pattern generate and CRC check.
* Configure and reconstruct input video sequence timing.
* Support 4K video in both ISPIF and BPIF mode.
* Support up to 8 path of video input in BPIF mode and 2 path of video input in ISPIF mode.
* Support multi datatype.
* YUV420 8-bit (legacy) / 8-bit / 10-bit / 8-bit (CSPS) / 10-bit (CSPS)
* YUV422 8-bit / 10-bit
* RGB888 / RGB666 / RGB565 / RGB555 / RGB444
* RAW6 / RAW7 / RAW8 / RAW10 / RAW12 / RAW14 / RAW16

## I2S interface

## Overview

The DW\_apb\_i2s is a configurable, synthesizable, and programmable component designed to be used in

systems that process digital audio signals, such as:

■ A/D and D/A converters

■ digital signal processors

■ error correction for compact disc and digital recording

■ digital filters

■ digital input/output interfaces

The Inter-IC Sound (I2S) Bus is a simple three-wire serial bus protocol developed by Philips to transfer

stereo audio data. The bus only handles the transfer of audio data; hence control and subcoding signals need to be transferred separately using a different bus protocol (such as I2C).

## Block Diagram

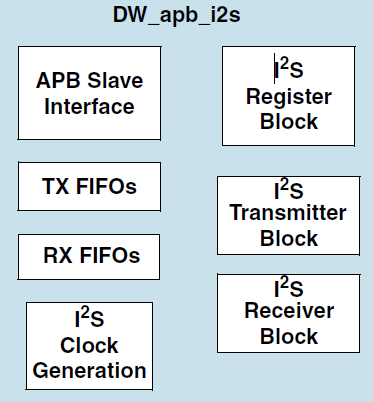


Fig. I2S module block diagram

## Features

DW\_apb\_i2s has the following features:

■ APB data bus widths of 8, 16, and 32 bits  
■ I2S transmitter and/or receiver based on the Philips I2S serial protocol

■ Configurable number of stereo channels (up to 4) for both transmitter and receiver   
■ Full duplex communication due to the independence of transmitter and receiver   
■ Asynchronous clocking of APB bus and I2S sclk

■ Master or slave mode of operation

■ Audio data resolutions of 12, 16, 20, 24, and 32 bits   
■ External sclk gating and enable signals

■ Configurable FIFO depth of 2, 4, 8, and 16 words, where the wordsize is determined by

*I2S\_RX\_WORDSIZE\_x* or *I2S\_TX\_WORDSIZE\_*r

■ Configurable support for programmable DMA registers   
■ Programmable FIFO thresholds

■ Component parameters for configurable software driver support

## DMA controller

## CEVA\_DMAC Controller

## Block Diagram



Fig. 1.1 ceva dma controller module block diagram

## Features

■ Independent core, slave interface and master interface clocks

■ Up to four channels, one per source and destination pair   
■ Data transfers in one direction only (each channel is unidirectional)   
■ Only one AXI master interfaces

■ Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral DMA transfers   
■ AMBA 4 AXI-compliant master interface  
■ AHB slave interface for programming the DMA controller

■ AXI master data bus width up to 128 bits (for both AXI master interfaces)

■ Independent control for endian scheme of linked list access on master interfaces

■ Channel locking support

❑ Supports locking of the internal channel arbitration for the master bus interface at different transfer hierarchy

■ DMA hold function

■ Multiple levels of DMA transfer hierarchy

❑ DMA transfer split into transaction, block, and complete DMA transfer levels

## TOP\_DMAC Controller

## Block Diagram



Fig. 1.1 top dma controller module block diagram

## Features

■ Independent core, slave interface and master interface clocks

■ Up to two channels, one per source and destination pair   
■ Data transfers in one direction only (each channel is unidirectional)   
■ Up to two AXI master interfaces

■ Memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral DMA transfers   
■ AMBA 4 AXI-compliant master interface  
■ AHB slave interface for programming the DMA controller

■ AXI master data bus width up to 128 bits (for both AXI master interfaces)

■ Independent control for endian scheme of linked list access on master interfaces

■ Channel locking support

❑ Supports locking of the internal channel arbitration for the master bus interface at different transfer hierarchy

■ DMA hold function

■ Multiple levels of DMA transfer hierarchy

❑ DMA transfer split into transaction, block, and complete DMA transfer levels

## AHB DMA Controller

## Block Diagram



## Features

* AHB slave interface – used to program the ahb\_dmac
* Channels
* Up to eight channels, one per source and destination pair
* Unidirectional channels – data transfers in one direction only
* Programmable channel priority
* AHB master interface(s)
* Up to four independent AHB master interfaces that allows:
* Up to four simultaneous DMA transfers
* Masters that can be on different AHB layers (multi-layer support)
* Source and destination that can be on different AHB layers (pseudo fly-by

performance)

* Configurable data bus width (up to 256 bits) for each AHB master interface
* Configurable endianness for master interfaces
* Transfers
* Support for memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral DMA transfers
* DW\_ahb\_dmac to or from APB peripherals through the APB bridge
* Configurable identification register
* Component ID parameters for configurable software driver support

## Analog part

## Overview

Sirius abb\_top contains

1. 4 channels of pipelined I/Q ADCs, which is 12 bit 100MS/s, fully differential input.
2. 2 channels of current I/Q DACs, which is 12 bit 200MS/s, fully differential output
3. 2 10bit 9MS/s SAR ADCs with 3 channel single end input
4. 1 10bit 9MS/s SAR ADCs with 8 channel single end input
5. Embedded bias voltage/current bias circuit
6. Embedded crystal driver
7. Temperature sensor
8. PVT sensor
9. ADDAPLL
10. DSPPLL0/1/2
11. SD card PLL
12. Audio PLL
13. Pixel PLL
14. DDRPLL
15. A7 PLL
16. CEVA PLL

## MISC blocks

## Overview

There are clock generation unit (CGU), reset generation unit (RGU), IO share unit and top global registers unit in non-secure block. All units are work at 20MHz and access through APB interface connected to NOC.

There is SPI debug interface and DS5 JTAGE interface in this chip, which is easy to use for debugging system.

## CGU

* The CGU generates clocks for all IP.
* Select different frequency for specified IP.
* Gate clock for specified IP.
* Generate divided clock for specified IP.
* Access through APB interface.

## RGU

* The RGU generates resets for all IP.
* Generate debounce and delay logic for power up reset.
* Generate soft reset for specified IP.
* Access through APB interface.

## IO Share function

* All IOs can be configurable via registers
* Function switch
* Pull-up/down
* Driver strength
* Up to 5 IOs shared on one.
* Access through APB interface.

## TOP Global Register

* This unit includes global registers for specified IP, for example SRAM, Boot ROM, H265, H264, JPEG, DISP, SMMU.
* Includes PMU registers.
* Access through APB interface.

## SPI Debug Interface

## Overview

SPI debug interface is developed to access whole system registers from PC side when CPU/DSP is not ready. It can be easily and simply used to tune DDR/PLL setting bypassing CPU/DSP. This IP has one external SPI interface and provide APB bridge to connect NOC system. The APB clock is 250MHz, and SPI interface clock can be up to 35MHz.

## Block Diagram



Figure 1.1 SPI debug interface diagram

## Features

* The Sirius SPI debug interface offers the following features：  
  ■ SPI command format with big endian  
  ■ APB access with little endian  
  ■ Access whole system registers

## CoreSight

## Overview

CoreSight SoC is a solution for debug and trace of complex SoCs. We can access system AXI, AHB and APB via JTAG interface. In Sirius, CoreSight SoC includes 5 trace soruces, one ETM in cortex-M7 and quad ETMs in cortex-A7 MPcore.

## Block Diagram



Fig. 1.1 CoreSight SoC block diagram

## Features

* Access to debug features and on-chip AXI, AHB, APB buses through a JTAG or Serial Wire Debug(SWD) interface
* 16KB on-chip sram ETB embedded in CoreSight SoC
* Trace source includes one M7 ETM, quad A7 ETMs
* Support up to 8-bit trace data output
* Support up to 150MHz trace clock output
* Support one M7 and quad A7 trace simultaneously
* 64-bit timestamp embedded in CoreSight SoC