1. Overview

1.1 Introduction

Ky ® X1 is a high-performance and ultra-low-power SoC that integrates 8 RISC-V CPU cores with Ky® AI computing power. It comes with the following most relevant advantages:

- Integration of Ky® X60™ RISC-V core processor which adheres to the RISC-V 64GCVB architecture and RVA22 standard
- Capable of 2.0 TOPS AI computing power by leveraging customized RISC-V instructions to enable CPU AI fusion computing
- Support of the popular AI inference frameworks such as TensorFlow Lite, TensorFlow, and ONNX Runtime
- Achievement of ultra-low power consumption by incorporating multiple granular power islands and adjusting power states (this makes X1 highly competitive in terms of energy usage)
- Availability of full-feature interfaces to enrich more innovative applications and products
- Compatibility with mainstream OS to meet the needs of various application scenarios
- Compliance with the industrial-grade reliability standards

1.2 General Features

- **Application Processor (AP)**
  - Ky® X60™ RISC-V Dual-Cluster 8-Core Processor
  - Adherence to the RISC-V 64GCVB architecture and RVA22 standard
  - Cluster 0
    - Quad-Core with 2.0 TOPS AI computing power
    - 32K L1-Cache per core
    - 512K L2-Cache

        - 512KB TCM
        - 256bit vector
    - Cluster 1
        - Quad-Core
        - 32K L1-Cache per core
        - 512K L2-Cache
        - 256bit vector
    - DVFS with adaptive operating voltage from 0.6V to 1.05V
- **DDR Memory**
    - Dual-Chip selection, 32-bit LPDDR4/LPDDR4x SDRAM with 2666 Mbps transfer rate, supporting up to 16 GB of RAM
    - Dual-Chip selection, 32-bit LPDDR3 SDRAM with 1866 Mbps transfer rate, supporting up to 4 GB of RAM
- **RCPU (Real-Time CPU)**
    - R-I2C (×1)
    - R-I2S (×2)
    - R-UART (×2)
    - R-CAN_FD (×1)
    - R-IR_RX (×1)
    - R-PWM (×10)
    - R-SPI (×1)
- **Peripheral Controller**
    - GPIO (×128)
        - 128 pins
        - Pull-up/pull-down programmable
        - 104x 1.8V IO8
        - 24x 1.8V/3.3V IO
    - UART (×10)
        - AP/BT/print

- I2C (×10)
  - For camera, G-Sensor, E-COMPASS, Proximit-Sensor, Light-Sensor, Gyro, Fingerprint, NFC, PMIC, Touch, etc.
  - 8x AP_I2C (AP I2C0/1/7 dedicated for camera) + 1x HDMI I2C + 1x PWR I2C
- SPI (×4)
  - Support of both master and slave mode
  - For IMU, codec etc.
  - Platform with 4 SPI (1x QSPI, 1x SPI LCD, 2x SPI)
- USB (×3)
  - USB 2.0 OTG
  - USB 2.0 Host
  - USB 3.0 (combo PCIE PortA)
- PCIE (×3)
  - PCIE PortA Gen2x1
  - PCIE PortB Gen2x2
  - PCIE PortC Gen2x2
- GMAC (×2)
  - 10/100/1000 Mbps
  - RGMII
- SDIO (×1 for WIFI)
  - Compatible with 4-bit SDIO 3.0 UHS-I protocol, up to SDR104 (208MHz)
- SD (×1 for TF card)
  - Compatible with 4-bit SD 3.0 UHS-I protocol, up to SDR104 (208MHz)
- eMMC (×1)
  - Compatible with 8bit eMMC5.1, up to HS400 (200MHz)
- MIPI CSI (CSI-2 v1.1) 4-Lane (×2)
  - 4-Lane + 4-Lane mode
  - 4-Lane + 2-Lane mode

        - 4-Lane + 2-Lane + 2-Lane mode (triple sensor)
    - MIPI DSI (DSI v1.1) (×1)
        - 4-Lane DSI
    - PWM (×20)
    - CAN-FD (×1)
    - IR-RX (×1)
- **Security System**
    - RISC-V PMP Security
    - Secure Boot
    - Secure eFuse 4K bits
    - Cryptographic engine (TRNG, AES, RSA, ECC, SHA2, HMAC)
- **Debug System**
    - Two JTAGs for both CPU and MCU subsystem
    - UARTs
    - CPU/IO register snapshot after watchdog reboot
- **Boot System**
    - Initial AP boot from SPI-Nand/SPI-NorFlash/eMMC/SD
    - 128KB boot-ROM
- **Aided System**
    - Watchdog design for each CPU/MCU subsystem
- **Operating Temperature**
    - -40°C ~ +85°C (Industrial Standard)

1.3 Multimedia Features

- **GPU**
    - IMG BXE-2-32@819MHz, 32KB SLC
    - Support of OpenCL3.0/OpenGL ES 3.2/Vulkan1.3
- **VPU (Video Processing Unit)**

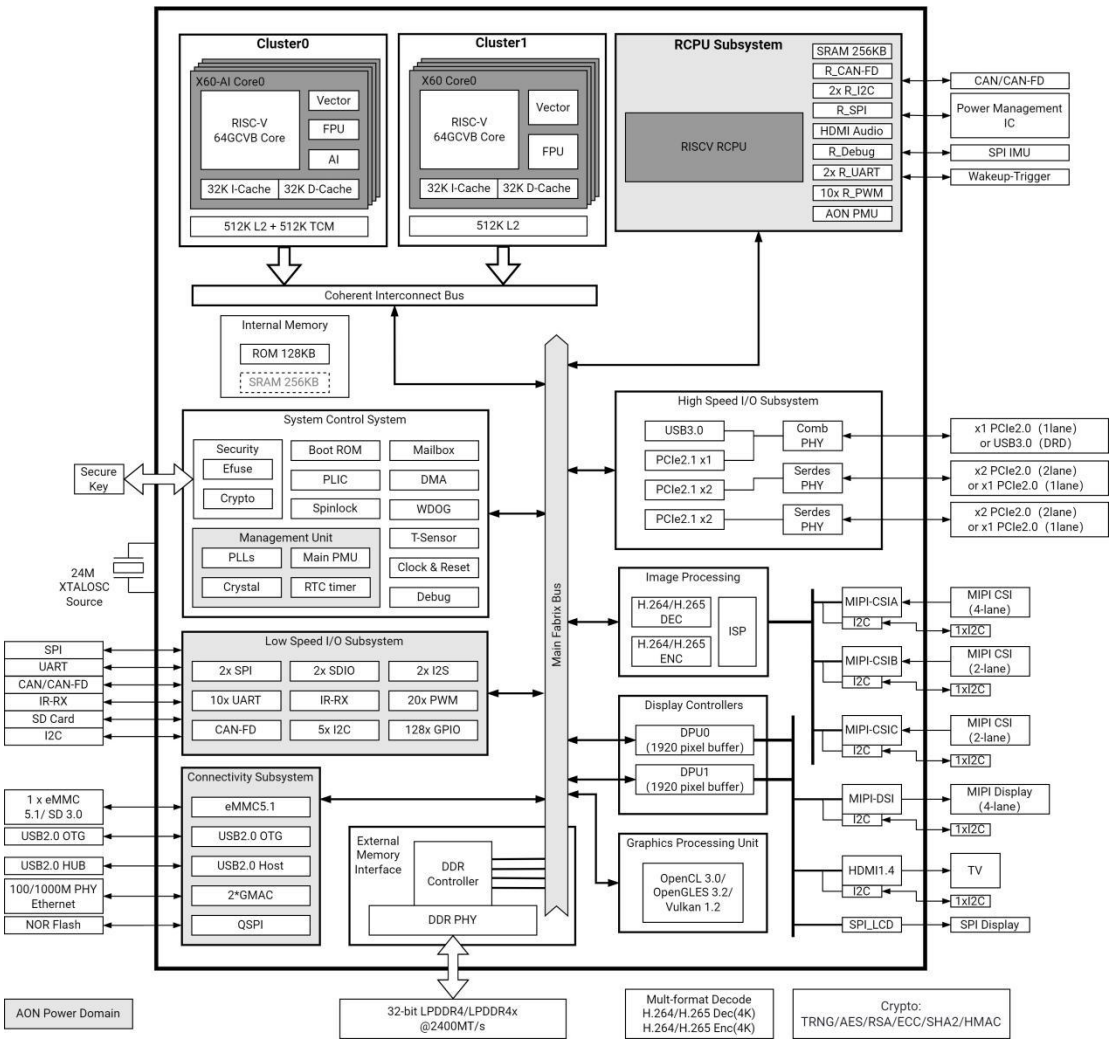
- H.265/H.264/VP8/VP9/MPEG4/MPEG2 decoder 4K@60fps
- H.265/H.264/VP8/VP9 encoder 4K@30fps
- Support of simultaneous encoding and decoding at 1080P@60fps
- Support of simultaneous H264/H265 encoding at 1080P@30fps and H264/H265 decoding at 4K@30fps

- **Display**
  - 1 MIPI DSI-4 lane or SPI interface
  - Support up to HD+ (1920x1080@60fps)
  - Support of up to 4-full-size-layer composer and maximum 8-layer composer by up-down layer reuse in RDMA channel
  - Support of *cmdlist* mechanism which can configure register parameters by hardware
  - Support of concurrent write-back with both raw and AFBC format
  - Support of dither/crop/rotation in write-back path
  - Support of an advanced MMU (virtual address) mechanism with nearly no page missing in 90/270 degree rotation
  - Support of color key and solid color
  - Support of both advanced error diffusion and pattern based dither for panel
  - Support of both raw and AFBC format image source
  - Support of color saturation/contrast enhancement
  - Support of both video mode and *cmd* mode for panel
  - Support of DDR frequency dynamic changing with embedded DFC buffer
  - HDMI 1.4
- **Camera**
  - Dual-ISP
    - 16M (max) 30fps Dual ISP
    - One 4-Lane CSI + one 4-Lane CSI, or 4-Lane + 2-Lane + 2-Lane
    - RAW sensor, output YUV data to DRAM
    - Hardware JPEG encoder, supporting up to 23M

        - Support of YUV/EXIF/JFIF format
        - AF/AE/AWB
        - Face detection
        - Digital zoom, panorama view
        - PDAF
        - PiP (Picture-in-Picture)
        - Continuous video AF
        - HW 3D denoise
- **Audio**
    - 2 × Full-Duplex I2S Interfaces
    - 1 × HDMI Audio Interface

1.4 Block Diagram

The architecture of X1 is depicted below.



2. Specifications

2.1 CPU Subsystem

2.1.1 Features

- Availability of two asymmetric CPU clusters, where
  - Cluster 0 integrates Quad RISC-V Ky® X60™ Cores with 2.0 TOPS AI-Power extension
  - Cluster 1 includes Quad RISC-V Ky® X60™ Cores without AI capability
- High-performance: low-power Ky® X60™ CPU core adheres to RISC-V 64GCVB

architecture and RVA22 standard

- Support of a processor core local interrupt controller (CLINT) and a platform level interrupt controller (PLIC)
- Compliance with RISC-V debug V0.13.2 standard
- Capture of a snapshot of the CPU critical information when a watchdog reset occurs in order to help debugging
- Power islands and two-level power strategies design for each CPU core and clusters in order to achieve ultra-low power consumption

2.1.2 Ky® X60™ RISC-V Core

2.1.2.1 Introduction

X60™ is an innovative high-efficiency processor core that adheres to RISC-V 64GCVB and RVA22 standards.

In order to meet the current and future computational demand, X60™ incorporates numerous DSA technologies and micro-architecture optimizations, and provides robust computing power for AI applications, machine learning, SLAM, etc.

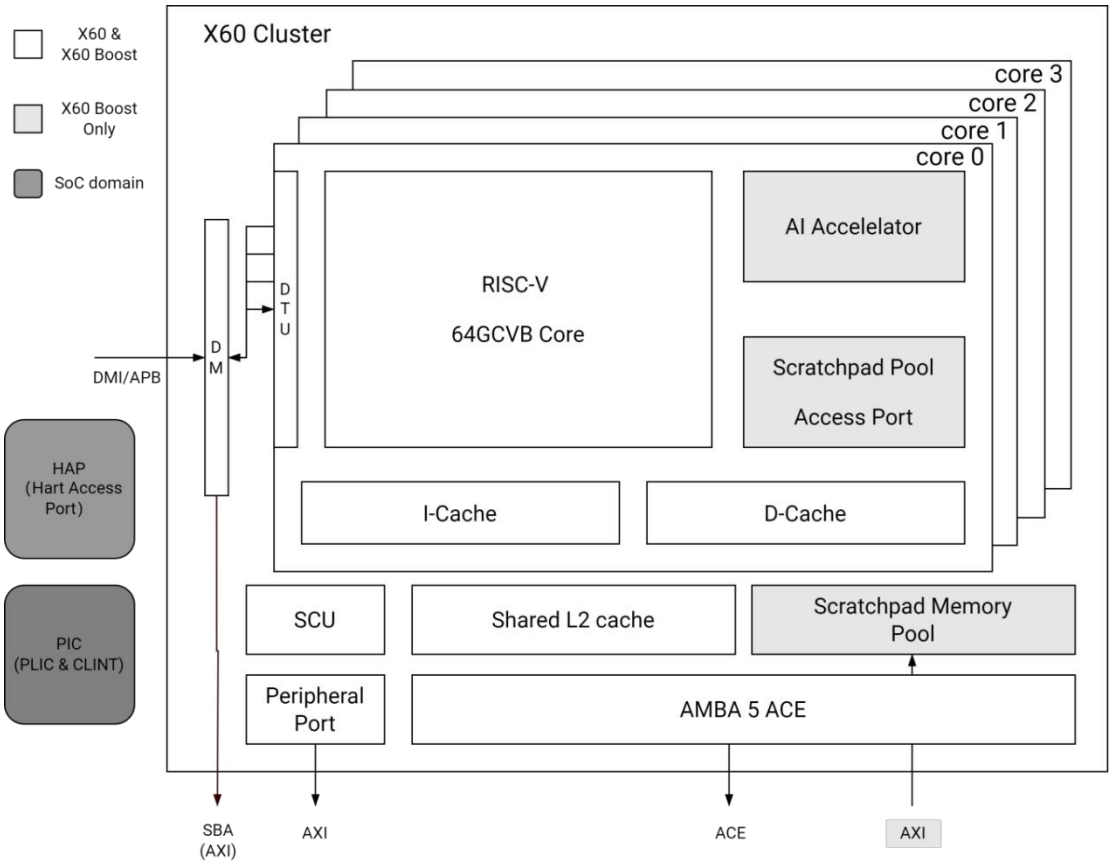
2.1.2.2 Features

- Compliance with RISC-V 64GCVB and RVA22 standards
- Each core has 32KB L1-I cache and 32KB L1-D cache
- Each cluster contains 512KB L2 cache
- Cluster 0 integrates 512KB TCM (Tight-Coupled Memory) for AI extension
- L1 cache supports MESI consistency protocol, instead L2 cache supports MOESI consistency protocol
- Vector extension: RVV1.0 with VLEN 256/128-bit and x2 execution width
- AI customized instructions explored and implemented in Cluster 0

- Support of CLINT and PLIC with a total of 256 interrupts
- Support of RISC-V performance PMU
- Support of SV39 virtual memory
- Support of 32 PMP entries adhering to RISC-V security framework
- Support of RISC-V debug framework
- Support of the following extensions:
  - RV64I
  - M
  - A
  - F
  - D
  - C
  - V
  - Sscofpmf
  - Sstc
  - Svinval
  - Svnapot
  - Svpbmt
  - Zicbom
  - Zicbop
  - Zicboz
  - Zicntr
  - Zicond
  - Zicsr
  - Zifencei
  - Zihintpause
  - Zihpm
  - Zfh

- Zfhmin
  - Zkt
- Zba
- Zbb
- Zbc
  - Zbs
- Zbkc
- Zvfh
- Zvfhmin
  - Zvkt

2.1.2.3 Block Diagram



2.1.3 Interrupt Controller

2.1.3.1 Introduction

X1 contains

- One Processor Core Local Interrupt Controller (CLINT)
- One Platform Level Interrupt Controller (PLIC)

to manage interrupts for two processor clusters.

The exception handling, which includes exceptions and external interrupts, is an important function of the processor. When specific events occur, the processor redirects to handle them. Such events can include hardware faults, instruction execution errors, user program service requests, and more.

CLINT is a memory address mapped module for handling software interrupts and timer interrupts.

Instead, PLIC samples external interrupt sources, then prioritizes and distributes them accordingly. In the PLIC model, both the machine mode and supervisor mode of each core are valid interrupt targets. PLIC supports up to 256 external interrupt sources. Each interrupt supports both level and edge formats.

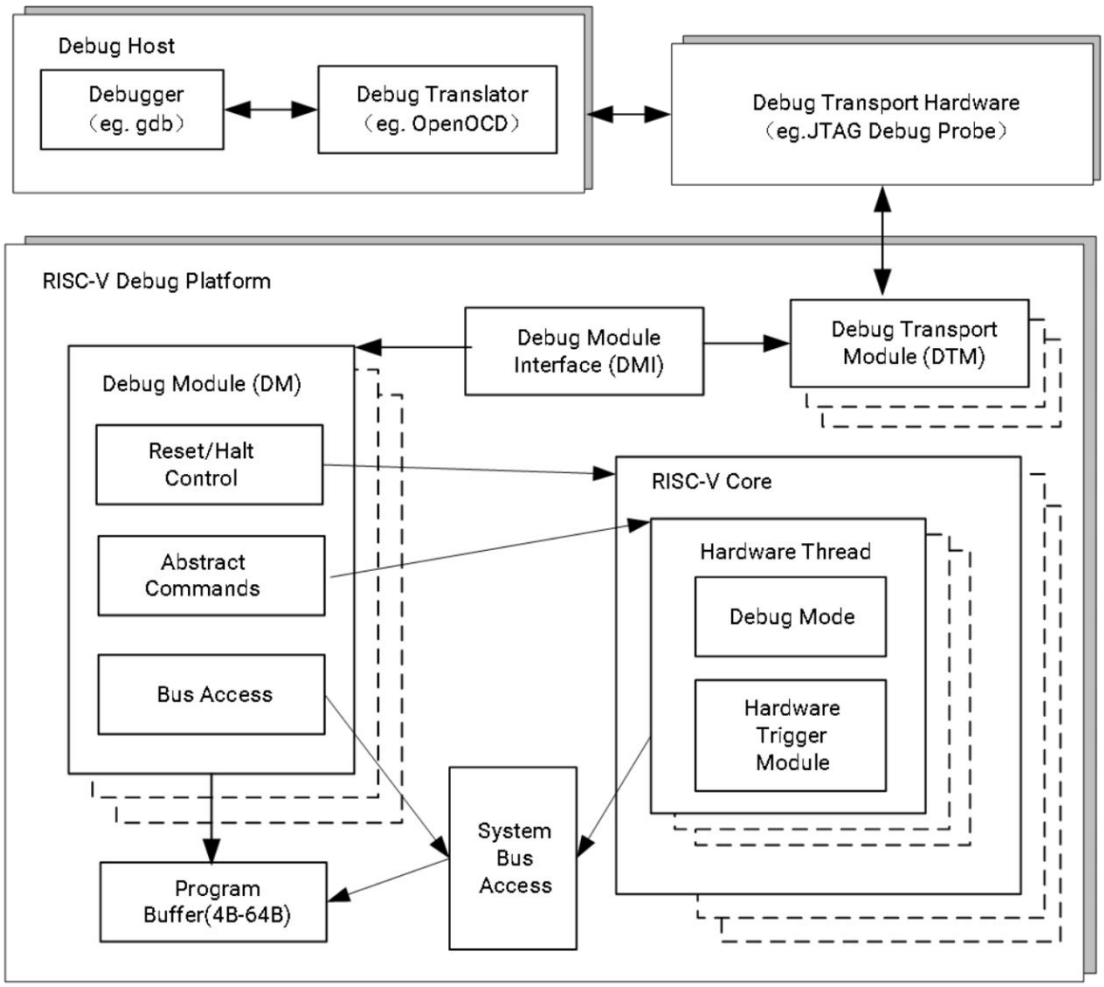
2.1.4 Debug & Trace

2.1.4.1 Introduction

The debugging interface serves as the channel for software to interact with the processor. Through this interface, users can access CPU registers and memory contents, as well as other on-chip device information. Additionally, tasks such as downloading programs can be performed via the debugging interface.

2.1.4.2 Block Diagram

The micro-architecture of the debugging interface is depicted below.



As can be seen, the debugging system consists of

- A debugging software
- A debugging agent service
- A debugger
- A debugging interface

These components are interconnected as follows:

- The debugging software communicates with the debugging agent service over a network
- The debugging agent service connects to the debugger via USB
- The debugger interacts with the CPU through the JTAG interface

The JTAG memory access method could be either *progbuf* or *sysbus* mode, where

- The *progbuf* mode is a standard JTAG method that accesses memory through the CPU

- The *sysbus* mode bypasses the CPU to access on-chip resources via the System Bus Access (SBA) port

2.2 Memory & Storage

2.2.1 On-Chip Memory

2.2.1.1 Introduction

X1 includes the following on-chip memory:

- 128KB boot-ROM
- 256KB SRAM shared between Main CPU and RCPU

2.2.2 DDR

2.2.2.1 Introduction

The DDR controller features a cutting-edge design that optimizes DRAM access by rearranging requests into an efficient order, rather than processing them in their original sequence. It uses re-ordering buffers (ROBs) to reorganize accesses to the SRAM device for improving performance, while maintaining the original transaction order for requests with the same ID on the AXI interface.

Additionally, the DDR controller includes a unified write pool to temporarily store write transactions. Such write pool minimizes write latency and reduces the performance penalty due to switching between read and write operation at the DRAM interface. With a built-in heuristic write buffer control and user-programmable write buffer control, the DDR controller dynamically balances read and write operation performance in real-time.

The DDR controller is also designed to support AMBA AXI4 bus protocols. It is fully scalable and supports up to 4 AXI ports.

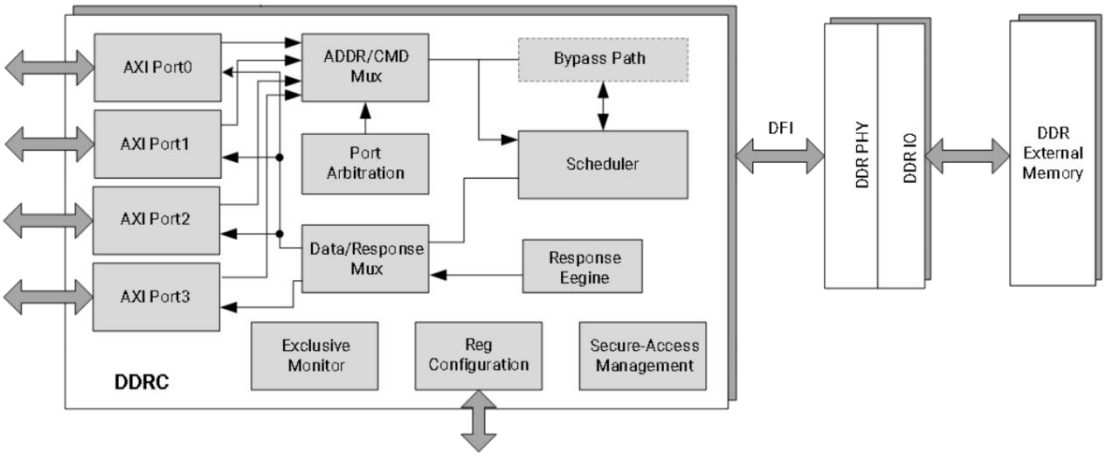
2.2.2.2 Features

- Priority-based arbitration with a starvation prevention scheme
- Merge of write operations to the same address by using a write buffer to reduce DDR write operation traffic
- Direct forward of read operations of the write buffer to the ROB without accessing DDR
- Two levels dynamic scheduling with bandwidth guarantee
- Support of power-saving features, including active/pre-charge power-off and self-refresh, with control options available automatically (via idle timer), manually (through registers) or externally (via dedicated ports)
- Support of dynamic frequency change
- Support of JEDEC compliant LPDDR3 and LPDDR4 devices
- Support of DRAM size from 64MB to 16GB
- One DRAM channel with a x32 DDR PHY, programmable by software to support x32, x16 or x8 data width
- Support of x16, x32 DRAM devices (1 DQS per 8 DQ)
- Support up to 2 Chip Select (CS) or Rank per channel
- Support up to 8 banks per CS for LPDDRx
- Each CS can be mapped to a different starting address
- Each CS can be programmed for 8MB to 16GB
- DRAM banks can be kept open after access (no auto-pre-charge)
- Support of burst length of 8 and 16 for the applicable DDR type
- Programmable address order
- Flexible bank placement between CS and data width
- Implementation of memory controller performance counters
- Global monitors for RISC-V exclusive load/store access
- Secure access management for DDR transactions
- Frequency change register update: implementation of a register table for

hardware-triggered sequence update after frequency changes

2.2.2.3 Block Diagram

The architecture of the DDR controller interface is depicted below.



2.2.3 Quad-SPI

2.2.3.1 Introduction

Quad-SPI acts as an interface to external serial flash devices with up to four bidirectional data lines.

2.2.3.2 Features

- Flexible sequence engine to support various flash vendor devices
  - Single, dual and quad mode operation
- DMA supports reading RX buffer data via AMBA AHB bus (64-bit width interface) or IP register space (32-bit access), and filling TX buffer via IP register space (32-bit access)
  - Configurable DMA inner loop size
  - Fifteen interrupt conditions

- Memory-mapped read access for connected flash devices
  - Programmable sequence engine for future command/protocol changes, and able to support all existing vendor commands and operations
  - Support of all types of addressing
  - Support of standard SPI, Fast, Dual, Dual I/O, Quad, Quad I/O mode
  - Operation up to 104MHz clock frequency

2.2.4 eMMC Interface

2.2.4.1 Introduction

The eMMC interface is a hardware block that acts as a host of the eMMC bus to transfer data between eMMC card and the internal bus master.

2.2.4.2 Features

- Compatibility with 8 bits eMMC 5.1 protocol specification
  - Use of the same SD-HCI register set for eMMC transfers, with additional vendor-specific registers
  - Support of 1-bit/8-bit MMC and CE-ATA cards
  - Support of the following data transfer types defined in the SD-HCI specification:
    - PIO
    - SDMA
    - ADMA
    - ADMA2
  - Support of the SPI mode for eMMC card
  - Support of the following speed modes defined in eMMC 5.1:
    - Legacy (up to 26MB/s, 1.8V signal)

- High-speed SDR (up to 52MB/s, 1.8V signal)
- High-speed DDR (up to 52MB/s, 1.8V signal)
- HS200 (up to 200MB/s, 1.8V signal)
- HS400 (up to 400MB/s, 1.8V signal)

- Hardware generation/checking of CRC for all command and data transactions on the card bus
- 1024-byte FIFO (2 x 512-byte data blocks) for data transmission and reception

2.2.5 SD/MMC Interface

2.2.5.1 Introduction

The SD/MMC interface is a hardware block that acts as a host of the SD/MMC bus to transfer data between SD/MMC card and the internal bus master.

2.2.5.2 Features

- Compatibility with 4-bit SD 3.0 UHS-I protocol specification
- Follow up of the SD-HCI register set with additional vendor-specific registers
- Support of 1-bit/4-bit SD memory
- Support of the following data transfer types defined in the SD-HCI specification:
  - PIO
  - SDMA
  - ADMA
  - ADMA2
- Support of the following speed modes defined in the SD 3.0 specification:
  - Default Speed (up to 12.5MB/s, 3.3V signal)
  - High Speed (up to 25MB/s, 3.3V signal)

- SDR12 (up to 25 MHz, 1.8V signal)
- SDR25 (up to 50 MHz, 1.8V signal)
- SDR50 (up to 100 MHz, 1.8V signal)
- SDR104 (up to 208 MHz, 1.8V signal)
- DDR50 (up to 50 MHz, 1.8V signal)

- Hardware generation/checking of CRC for all command and data transactions on the card bus
- Support of the read-wait control feature for SD/MMC cards
- Support of the suspend-resume feature for SD/MMC cards
- SD/MMC card insertion/removal detection feature via GPIO
- 1024 Bytes FIFO (2 x 512 Bytes data block) for data transmission and reception

2.3 Image Subsystem

2.3.1 MIPI Camera IN Interface

2.3.1.1 Introduction

The MIPI Camera IN interface features two MIPI-CSI2 v1.1 controllers both equipped with 4 lanes each of which supports a maximum transfer rate of 1.5Gbps.

2.3.1.2 Features

- Support of the following modes to allocate lanes to sensors:
  - 4-Lane + 4-Lane mode (double sensor)
  - 4-Lane + 2-Lane mode (double sensor)
  - 4-Lane + 2-Lane + 2-Lane mode (triple sensor)

> **Note.** In "4-Lane + 2-Lane + 2-Lane mode (triple sensor)", only 2 Bayer RAW

> and 1 YUV input format are supported.

- Support of the following input formats:
  - Legacy YUV420 8-bit
  - YUV420 8-bit
  - RAW8
  - RAW10
  - RAW12
  - RAW14
  - Embedded data type
- Support of the following types of data interleaving:
  - Data type interleaving
  - Virtual channel interleaving

2.3.2 ISP

2.3.2.1 Introduction

X1 includes a high-performance Image Signal Processor (ISP) which supports simultaneous processing of up to two raw video streams, with a total processing capacity of 21M@30fps.

2.3.2.2 Features

- Support of both video and picture mode
- Process of RAW sensor data and output YUV data to DRAM
- Hardware JPEG encoder/decoder (support up to 23M)
- Support of YUV, EXIF, JFIF format
- Auto-focus (AF), Auto-exposure (AE) and Auto-white balance (AWB)

- Face detection
- Digital zoom and panorama view
- Phase Detection Auto-focus (PDAF)
- Picture-in-Picture (PiP)
- Continuous video AF
- Hardware 3D denoise
- Multi-layer 2D YUV denoise
- Post-processing for lens shading correction
- Edge enhancement

> **Notes.** To be highlighted the following limitations:
>
> - The system supports dual-camera video stream processing (RAW). In the “4-Lane + 2-Lane + 2-Lane mode (triple sensor)” as per **Section 2.3.1**, one sensor must be a YUV input format source, and the write path should not use the MMU.
> - When processing dual-camera video stream (RAW), the total input width of each channel should not exceed 4750 pixels. The combination of the instantaneous speed of the output pixel from both sensors must be less than "*ISP's clock / 6*"
> - For video recording, the maximum width of the output video is 1920 pixels, regardless of the input resolution.
> - For photo capture, the output image size can match the input resolution.

2.3.3 GPU

2.3.3.1 Introduction

GPU is built around multi-threaded Unified Shading Clusters (USCs) that features an ALU architecture with high SIMD efficiency, and supports tile-based deferred rendering with concurrent processing of multiple tiles.

The GPU engine handles a number of different workloads, including:

- 3D graphics workload: vertex and pixel data processing for rendering 3D scenes
- Compute workload (GP-GPU): general purpose data processing

> **Note.** 3D graphics and compute (with barriers) workloads cannot be overlapped at the same time

The GPU core has an AXI 128bits bus for accessing SOC's DDR memory with a core frequency of up to 819MHz.

2.3.3.2 General Features

- Base architecture which is fully compliant with the following APIs:
  - OpenGL ES 1.1/3.2
  - EGL1.5
  - OpenCL 3.0
  - Vulkan 1.3
- Tile-based deferred rendering architecture (TBDR) for 3D graphics workloads, with concurrent processing of multiple tiles where data are processed in two phases as follows:
  - Geometry Processing Phase: involvement of vertex operations such as transformation and vertex lighting as well as dividing a 3D scene into tiles
  - Fragment Processing Phase: involvement of pixel operations such as rasterization, texturing and shading of pixels
- Programmable high quality image anti-aliasing
- Fine grain triangle culling
- Support of Digital Right Management (DRM) security
- Support of GPU virtualization as follows:
  - Up to 8 virtual GPUs
  - IMG hyperlane technology with 8 hyperlanes available
  - Separate IRQs per OSI
- Multi-threaded Unified Shading Cluster (USC) engine incorporating pixel shader,

vertex shader and GP-GPU (compute shader) functionality

- USC incorporates an ALU architecture with high SIMD efficiency
- Fully virtualized memory addressing (up to 64 GB address space), supporting unified memory architecture
- Fine-grained task switching, workload balancing and power management
- Advanced DMA driven operation for minimum host CPU interaction
- Cache type as follows:
  - 32KB System Level Cache (SLC)
  - Specialized Texture Cache Unit (TCU)
- Compressed Texture Decoding
- Lossless and/or visually lossless low area image compression, using imagination frame buffer compression and decompression (TFBC) algorithm
- Dedicated processor for B-Series core firmware execution
- Single-threaded firmware processor with a 2KB instruction cache and a 2KB data cache
- Separated power island for the firmware processor
- On-chip performance, power and statistics registers

2.3.3.3 3D Graphics Features

- **Rasterization**
  - Deferred pixel shading
  - On-chip tile floating point depth buffer
  - 8-bit stencil with on-chip tile stencil buffer
  - Maximum 2 tiles in flight (per ISP)
  - 16 parallel depth/stencil tests per clock
  - 1 fixed-function rasterisation pipeline(s)
- **Texture Lookups**
  - Support of loading from source instruction

- Texture write enabled through the Texture Processing Unit (TPU)

- **Filtering**
  - Point, bilinear and trilinear filtering
  - Anisotropic filtering
  - Corner filtering support for cube environment mapped textures and filtering across faces
- **Texture Formats**
  - ASTC LDR compressed texture format support
  - TFBC lossless and/or lossy compression format support for non-compressed textures and YUV textures
  - ETC
  - YUV planar support
- **Resolution Support**
  - Max frame buffer size: 8K×8K
  - Max texture max size: 8K×8K
- **Anti-Aliasing**
  - Max 4× multisampling
- **Primitive Assembly**
  - Early hidden object removal
  - Tile acceleration
- **Render to Buffers**
  - Twiddled format support
  - Multiple On-Chip Render Targets (MRT)
  - Lossless and/or lossy frame buffer compression/decompression
  - Programmable geometry shader support
  - Direct geometry stream out (transform feedback)
- **Compute**
  - 1, 2 and 3 dimensional compute primitives
  - Block DMA to/from USC Common Store (for local data)

- Per task input data DMA (to USC Unified Store)
- Conditional execution
- Execution fences
- Compute workload can be overlapped with any other workload
- Round to nearest even

2.3.3.4 Unified Shading Cluster (USC) Features

- 2 ALU pipelines
- 8 parallel instances per clock
- Local data, texture and instruction caches
- Variable length instruction set encoding
- Full support for OpenCL™ atomic operations
- Scalar and vector SIMD execution model
- USC F16 Sum-of-Products Multiply-Add (SOPMAD) Arithmetic Logic Unit (ALU)

2.3.4 V2D

2.3.4.1 Features

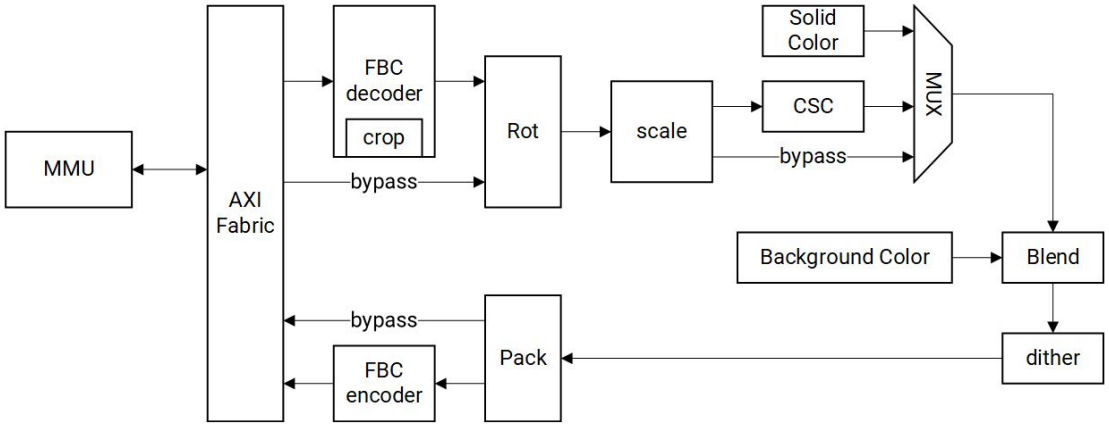
- Support of upscaling (up to 8x) and downscaling (down to 1/8x)
- Support of 0°, 90°, 180°, 270° rotation as well as mirror and flip option
- Support of simple layer and background blending
- Support of image cropping
- Support of fetch solid color
- Support of color space conversion between RGB, BT601 and BT709 (both narrow and full range)
- 4656x3596 or 4672x3504 as max NV12 resolution

- Support of dithering for smoother color transitions
- Support of MMU
- Support of APB3 and AXI3 bus interfaces
- Support of the following **input formats**:
  - RGB888 (with optional RB swap)
  - RGBX888 (with optional RB swap)
  - RGBA8888 (with optional RB swap)
  - ARGB8888 (with optional RB swap)
  - RGB565 (with optional RB swap)
  - RGBA5658 (with optional RB swap)
  - ARGB8565 (with optional RB swap)
  - A8 (8-bit alpha image)
  - Y8 (8-bit gray image)
  - YUV420 semi-planar (UV can swap)
  - AFBC 16x16 RGBA8888 (layerout0 split and non-split)
  - AFBC 16x16 NV12 (layerout1 split and non-split)
- Support of the following **output formats**:
  - RGB888 (with optional RB swap)
  - RGBX888 (with optional RB swap)
  - RGBA8888 (with optional RB swap)
  - ARGB8888 (with optional RB swap)
  - RGB565 (with optional RB swap)
  - RGBA5658 (with optional RB swap)
  - ARGB8565 (with optional RB swap)
  - A8 (8-bit alpha image)
  - Y8 (8-bit gray image)
  - YUV420 semi planar (UV can swap)
  - AFBC 16x16 RGBA8888 (layerout0 split and non-split)

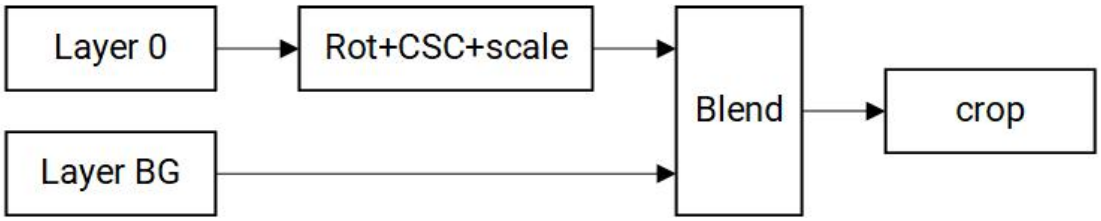
- AFBC 16x16 NV12 (layerout1 split and non-split)

2.3.4.2 Block Diagram

The micro-architecture of the V2D subsystem is depicted below.



Instead, the typical V2D work scenario is depicted below.

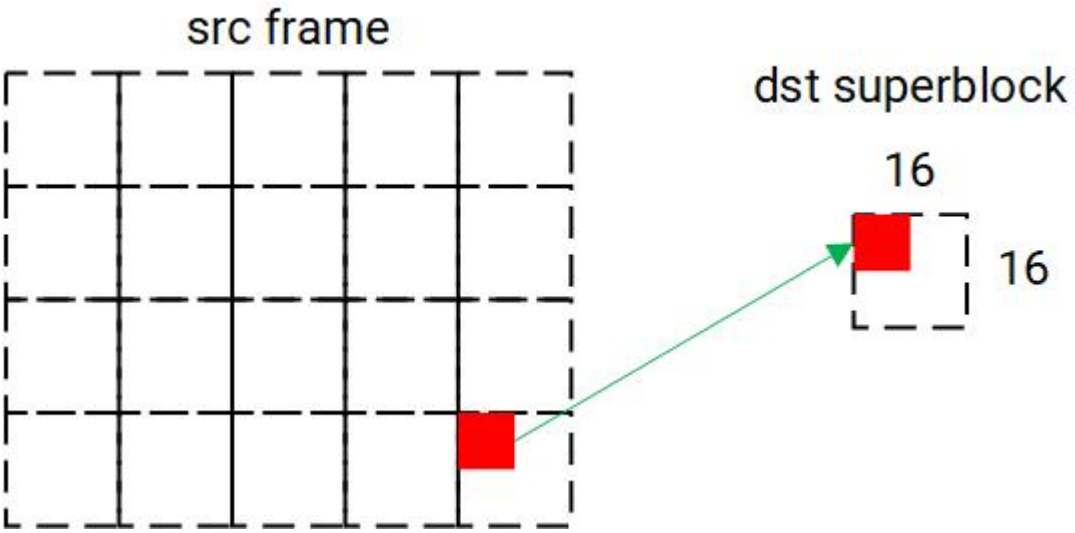


2.3.4.3 Functions

2.3.4.3.1 Fetch Data

The process of fetching a 16×16 block of data from a source frame (src frame) and related mapping to the destination superblock (dst superblock) is depicted below, where

- **AFBC**: fetch rect left, top, width, height 4 align
- **Non-AFBC**: fetch rect left, top, width, height 1 align



The code for fetching data for displaying is listed below, and the details of the specific variables and registers involved are tabled immediately after.

```
Plain Text
Input param: Rect_left, Rect_top, Rect_width, Rect_height
Rect_width = Rect_left%4 + Rect_width;
Rect_height = Rect_top%4 + Rect_height;
Rect_left = Rect_left/4 × 4;
Rect_top = Rect_top/4 × 4;
if LayerX_format == YUV420
{
    Rect_width  = ALIGN(Rect_left %2 + Rect_width, 2);
    Rect_height   = ALIGN(Rect_top%2 + Rect_height, 2);
    Rect_left = Rect_left/2 × 2;
    Rect_top = Rect_top/2 × 2;
}
Take the data in the Rect
Loop every pixel in Rect
{
    if LayerX_format == YUV420
    {
        upsample YUV420 to YUV444;
        c0 = channel 0; // Y
        c1 = channel 1; // U
        c2 = channel 2; // V
```

```
    c3 = 0xff;
}
if LayerX_format == RGB888
{
    c0 = channel 0; // R
    c1 = channel 1; // G
    c2 = channel 2; // B
    c3 = 0xff; // A
}
if LayerX_format == RGBX8888
{
    c0 = channel 0; // R
    c1 = channel 1; // G
    c2 = channel 2; // B
    c3 = 0xff; // A
}
if LayerX_format == RGBA8888
{
    c0 = channel 0; // R
    c1 = channel 1; // G
    c2 = channel 2; // B
    c3 = channel 3; // A
}
if LayerX_format == ARGB8888
{
    c0 = channel 1; // R
    c1 = channel 2; // G
    c2 = channel 3; // B
    c3 = channel 0; // A
}
if LayerX_format == RGB565
{
    c0 = byte_low &0x1f; // R5
    c1 = ((byte_high << 3) | (byte_low >> 5)) & 0x3f; // G6
    c2 = (byte_high >> 3) &0x1f; // B5
    c0 = (c0 << 3) | (c0 >> 2); // R8
    c1 = (c1 << 2) | (c1 >> 4); // G8
```

```
        c2 = (c2 << 3) | (c2 >> 2); // B8
        c3 = 0xff; // A8
    }
    if LayerX_format == YUV420 && LayerX_swap == 1
        Swap(c1, c2);
    else if LayerX_swap == 1
        Swap(c0, c2);
    Index = Rect_y%16 × 16 + Rect_x;
    data[0][index] = c0;
    data[1][index] = c1;
    data[2][index] = c2;
    data[3][index] = c3;
}
```

Variable	Bit	Comment
Rect_left Rect_top	16bit unsigned	Range [0, 65535]
Rect_width Rect_height	5bit unsigned	Range [1, 16]
Rect_x Rect_y	16bit unsigned	Range [0, 65535] Pixel global position
c0, c1, c2, c3	8bit unsigned	Range [0, 255]
byte_low byte_high	8bit unsigned	Range [0, 255] byte_low: lower byte in RGB565 byte_high: higher byte in RGB565

data[4][256]	8bit unsigned × 4 × 256	Range [0, 255]
index	8bit unsigned	Range [0, 255]

2.3.4.3.2 Solid Color

The code for applying the solid color within a specific rectangle is listed below, and the details of the specific variables and registers involved are tabled immediately after.

> **Notes.**
>
> - If the register `LayerX_solid` is enabled, the fetched data is set to solid R, G, B, A
> - The coordinates of the fetch rect and solid rect are updated after rotation

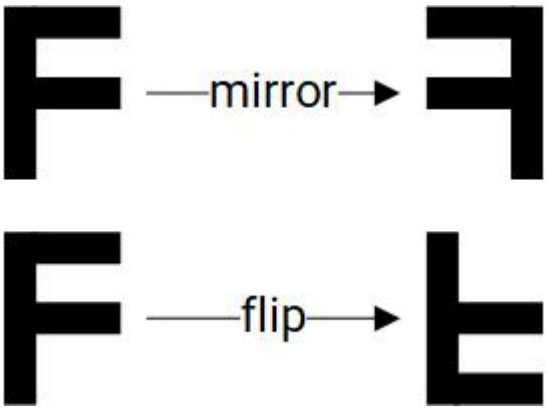
```
SQL
Input param: Rect_left, Rect_top, Rect_width, Rect_height.
if LayerX_solid_enable = 1
{
    c0 = LayerX_solid_R;
    c1 = LayerX_solid_G;
    c2 = LayerX_solid_B;
    c3 = LayerX_solid_A;
    Loop all pixels in Rect
    {
        Index = Rect_y%16 × 16 + Rect_x;
        data[0][index] = c0;
        data[1][index] = c1;
        data[2][index] = c2;
        data[3][index] = c3;
    }
    Skip fetch data from ddr
}
```

Variable	Bit	Comment
Rect_left, Rect_top	16bit unsigned	Range [0, 65535]
Rect_width, Rect_height	5bit unsigned	Range [1, 16]
Rect_x, Rect_y	16bit unsigned	Range [0, 65535] Pixel global position
c0, c1, c2, c3	8bit unsigned	Range [0, 255]
data[4][256]	8bit unsigned × 4 × 256	Range [0, 255]
index	8bit unsigned	Range [0, 255]

Register	Comment
LayerX_solid_enable	X is 0 or 1, refer to module register
LayerX_solid_R	X is 0 or 1, refer to module register
LayerX_solid_G	X is 0 or 1, refer to module register
LayerX_solid_B	X is 0 or 1, refer to module register
LayerX_solid_A	X is 0 or 1, refer to module register

2.3.4.3.3 Rotation

Support of 0°, 90°, 180°, 270° rotation (performed clockwise) as well as mirror and flip option, as depicted below (example).



The code for rotating, mirroring and flipping graphical content is listed below, and the details of the specific variables and registers involved are tabled immediately after).

```SQL
Input param: Rect_left, Rect_top, Rect_width, Rect_height,
data_in[4][256].
Output: Block_rect_left, Block_rect_top,  Block_rect_width,
Block_rect_height, data_out[4][256].
Block_rect_left = Rect_left;
Block_rect_top = Rect_top;
Block_rect_width = Rect_width;
Block_rect_height = Rect_height;
if LayerX_degree == ROT_0{
    Org_rect_left = Rect_left;
    Org_rect_top = Rect_top;
    Org_rect_width = Rect_width;
    Org_rect_height = Rect_height;
}
if LayerX_degree == ROT_90{
    Org_rect_left = Rect_top;
    Org_rect_top = ALIGN(LayerX_height,16) - Rect_left - Rect_width;
    Org_rect_width = Rect_height;
    Org_rect_height = Rect_width;
}
if LayerX_degree == ROT_180{
    Org_rect_left = ALIGN(LayerX_width,16) - Rect_left - Rect_width;
    Org_rect_top = ALIGN(LayerX_height,16) - Rect_top - Rect_height;
    Org_rect_width = Rect_width;
```

```
    Org_rect_height = Rect_height;
}
if LayerX_degree == ROT_270{
    Org_rect_left = ALIGN(LayerX_width,16)-Rect_top-Rect_height;
    Org_rect_top = Rect_left;
    Org_rect_width = Rect_height;
    Org_rect_height = Rect_width;
}
if LayerX_degree == ROT_MIRROR{
    Org_rect_left = ALIGN(LayerX_width,16) - Rect_left - Rect_width;
    Org_rect_top = Rect_top;
    Org_rect_width = Rect_width;
    Org_rect_height = Rect_height;
}
if LayerX_degree == ROT_FLIP{
    Org_rect_left = Rect_left;
    Org_rect_top = ALIGN(LayerX_height,16) - Rect_top - Rect_height;
    Org_rect_width = Rect_width;
    Org_rect_height = Rect_height;
}
//fetch data in Org_rect
Fetch_data(Org_rect, &data_in[4][256]);
Loop all pixels in data_in{
    dst_index=jx16 + i;
    if LayerX_degree == ROT_0
        src_index=jx16 + i;
    if LayerX_degree == ROT_90
        src_index=(15-i)x16 + j;
    if LayerX_degree == ROT_180
        src_index=(15-j)x16 + (15-i);
    if LayerX_degree == ROT_270
        src_index= ix16+(15-j);
    if LayerX_degree == ROT_MIRROR
        src_index = jx16 + (15-i);
    if LayerX_degree == ROT_FLIP
        src_index = (15-j)x16 + i;
    data_out[0][dst_index]= data_in[0][src_index];
```

```
    data_out[1][dst_index]= data_in[1][src_index];
    data_out[2][dst_index]= data_in[2][src_index];
    data_out[3][dst_index]= data_in[3][src_index];
}
```

Variable	Bit	Comment
Rect_left, Rect_top	16bit unsigned	Range [0, 65535]
Rect_width, Rect_height	5bit unsigned	Range [1, 16]
Block_rect_left, Block_rect_top	16bit unsigned	Range [0, 65535]
Block_rect_width, Block_rect_height	5bit unsigned	Range [1, 16]
data_in[4][256], data_out[4][256]	8bit unsigned × 4 × 256	Range [0, 255]

Register	Bit	Comment
LayerX_degree	3bit unsigned	X is 0 or 1, refer to module register
LayerX_width, LayerX_height	16bit unsigned	X is 0 or 1, refer to module register

2.3.4.3.4 CSC

Support of Color Space Conversion (CSC) as per formats below:

- BT601 and BT709: conversion between narrow and full range
- RGB to YUV
- YUV to RGB

The conversion process transforms input channels into output channels by using a transformation matrix with clamping in order to ensure valid output values, i.e. within the range [0, 255].

For that purpose, the formulas below are implemented, and the details of the specific variables and registers involved are tabled immediately after.

[Firstly for computing the intermediate channel values]

$$C0_{inter} = (Layer\_matrix[0][0] * C0_{in} + Layer\_matrix[0][1] * C1_{in} + Layer\_matrix[0][2] * C2_{in} + 512) >> 10 + Layer\_matrix[0][3]$$

$$C1_{inter} = (Layer\_matrix[1][0] * C0_{in} + Layer\_matrix[1][1] * C1_{in} + Layer\_matrix[1][2] * C2_{in} + 512) >> 10 + Layer\_matrix[1][3]$$

$$C2_{inter} = (Layer\_matrix[2][0] * C0_{in} + Layer\_matrix[2][1] * C1_{in} + Layer\_matrix[2][2] * C2_{in} + 512) >> 10 + Layer\_matrix[2][3]$$

[Then for clamping in order to ensure valid output values]

$$C0_{out} = clamp(C0_{inter}, 0, 255)$$

$$C1_{out} = clamp(C1_{inter}, 0, 255)$$

$$C2_{out} = clamp(C2_{inter}, 0, 255)$$

$$C3_{out} = clamp(C3_{in}, 0, 255)$$

Variable	Bit	Comment
C0in, C1in, C2in, C3in	8bit unsigned	Input channel
C0inter, C1inter, C2inter	10bit signed	Intermediate channel value
C0out, C1out, C2out, C3out	8bit unsigned	Output channel

Register	Index	Bit	Comment
LayerX_CSC_en	-	1bit	0: disable

able		unsigned	1: enable
Layer_matrix[#][#]	0-11	13bit signed	Range [-4096, 4095]

In the code, the conversion process is applied with the following condition:

```
Plain Text
if LayerX_CSC_enable == 0
    skip CSC function
```

2.3.4.3.5 Scaling

The scaling operation follows a systematic superblock-based approach, where

- The first four superblocks are outputted horizontally then vertically
- After the vertical output is completed, the process restarts from the first row of superblocks

2.3.4.3.6 Storing

A 16×16 image block can be stored in DDR memory, however only the portion that falls within the output crop region is stored which is converted to the specified output color format, such as YUV, RGB, etc.

The code for storing an image block is listed below, and the details of the specific variables and registers involved are tabled immediately after.

```
SQL
Input param: Rect_left, Rect_top, Rect_width, Rect_height,
data_in[4][256]
if output_format == YUV420
{
    s0=0;
    s1=1;
    s2=2;
```

```
    if(output_swap){
        Swap(s1, s2);
    }
    Loop all pixels by 2x2{
        if(pixel in output_crop_rect){
            Y00=data_in[s0][pixel_index00];
            Y01=data_in[s0][pixel_index01];
            Y10=data_in[s0][pixel_index10];
            Y11=data_in[s0][pixel_index11];
            U00=data_in[s1][pixel_index00];
            U01=data_in[s1][pixel_index01];
            U10=data_in[s1][pixel_index10];
            U11=data_in[s1][pixel_index11];
            V00=data_in[s2][pixel_index00];
            V01=data_in[s2][pixel_index01];
            V10=data_in[s2][pixel_index10];
            V11=data_in[s2][pixel_index11];
            Downsample and store to output frame
            U=(U00+U01+U10+U11+2)>>2;
            V=(V00+V01+V10+V11+2)>>2;
        }
    }
}
if output_format == RGB888
{
    s0=0;
    s1=1;
    s2=2;
    if(output_swap){
        Swap(s0, s2);
    }
    Loop all pixels{
        if(pixel in output_crop_rect){
            R=data_in[s0][pixel_index];
            G=data_in[s1][pixel_index];
            B=data_in[s2][pixel_index];
            store to output frame.
```

```
        }
    }
}
if output_format == RGBX888 || output_format == RGBA888
{
    s0=0;
    s1=1;
    s2=2;
    s3=3;
    if(output_swap){
        Swap(s0, s2);
    }
    Loop all pixels{
        if(pixel in output_crop_rect){
            R=data_in[s0][pixel_index];
            G=data_in[s1][pixel_index];
            B=data_in[s2][pixel_index];
            A=data_in[s3][pixel_index];
            store to output frame.
        }
    }
}
if output_format == ARGB8888
{
    s0=3;
    s1=0;
    s2=1;
    s3=2;
    if(output_swap){
        Swap(s1, s3);
    }
    Loop all pixels{
        if(pixel in output_crop_rect){
            R=data_in[s0][pixel_index];
            G=data_in[s1][pixel_index];
            B=data_in[s2][pixel_index];
            A=data_in[s3][pixel_index];
```

```
            store to output frame.
        }
    }
}
```

Variable	Bit	Comment
Rect_left Rect_top	16bit unsigned	Range [0, 65535]
Rect_width Rect_height	5bit unsigned	Range [1, 16]
pixel_index	8bit unsigned	Range [0, 65535]
s0, s1, s2, s3	8bit unsigned	Range [0, 255]
Y00, Y01, Y10, Y11, U00, U01, U10, U11, V00, V01, V10, V11, U, V, R, G, B, A	8bit unsigned	Range [0, 255]
data_in[4][256]	8bit unsigned × 4 × 256	Range [0, 255]

Register	Bit	Comment
Output_format	3bit unsigned	0: RGB888 (R at low address, B at high address) 1: RGBX8888 2: RGBA8888 3: ARGB8888 (A at low address, B at

		high address) 5: yuv420sp (U at low address, V at high address)
Output_swap	1bit unsigned	0: No swap 1: RGB swap RB, YUV swap UV
Output_layout	1bit unsigned	0: Linear 1: FBC compressed
Output_crop_left	16bit unsigned	Range [0, 65534] crop_left < output_left + output_width
Output_crop_top	16bit unsigned	Range [0, 65534] crop_top < output_top + output_height
Output_crop_width	16bit unsigned	Range [1, 65535] crop_left + crop_wdith ≤ output_left + output_width
Output_crop_height	16bit unsigned	Range [1, 65535] crop_top + crop_height ≤ output_top + output_height

2.4 Video Subsystem

2.4.1 Introduction

The Video Processing Unit (VPU) is a video accelerator engine with two cores designed for decoding and encoding multiple video standards. It includes a host CPU to run firmware to control the hardware engine of functions, such as bit stream parsing, control of video hardware sub-blocks and error resilience.

The VPU can work at up to 819MHz clock frequency, and supports a wide range of video standards, including H.265, H.264, VP8, VP9, MPEG4, MPEG2 and H263. It supports simultaneous

- Encoding and decoding at 1080P@60fps
- H264/H265 encoding at 1080P@30fps and H264/H265 decoding at 4K@30fps

The video codec core block executes the actual decoding and encoding for each standard by using hardwired logic. Among them, Macroblock Sequencer is the main controller that schedules process flows of the sub-blocks, and aims to reduce loads on the processor and complexity of the firmware.

As mentioned, several standard-independent blocks share common logics while they are in operation in order to ensure efficiency and streamlined performance.

2.4.2 Video Encoder

2.4.2.1 Encoding Features

- Configurable Arm Frame Buffer Compression (AFBC) 1.0 or 1.2 for input
- Support of YUV422 and YUV420 AFBC block split for 16 x 16
- Support of stride (not applicable to AFBC input formats)
- Horizontal and vertical mirroring (not applicable to AFBC input formats)
- Optional source frame rotation in 90-degree steps before encoding (not applicable to AFBC input format)

> **Note.** If YUV422 is rotated by 90 or 270 degrees and not converting to YUV420, the result will be converted to YUV440.

- Encoding support for the following source-frame input formats:
  - 1-plane YUV422, scan-line format, interleaved in YUYV or UYVY order

    > **Note.** YUV422 input scan be converted to YUV420
  - 1-plane RGB (8-bit) in byte-address order: RGBA, BGRA, ARGB or ABGR

- 2-plane YUV420, scan-line format, with chroma interleaved in UV or VU order
- 3-plane YUV420, scan-line format

> **Note.** 3-plane format is supported for testing purposes only, and should not be used for optimal performance

- AFBC YUV422
- AFBC YUV420

2.4.2.2 Supported Encoding Formats

- HEVC (H.265) Main
- H.264 Baseline Profile (BP)
- H.264 Main Profile (MP)
- H.264 High Profile (HP)
- VP8
- VP9 Profile 0

2.4.2.2.1 HEVC (H.265) Encoding Features

- Encoded bit stream is compliant with the HEVC (H.265) Main Profile
- Encoding speed of 1080p@60fps (dual cores at approximately 300 MHz)
- Bitrates up to 50MBit/s using a single core operating at 300MHz
- Max frame width and height: 4096 pixels
- 8-bit encoding with I, P, and B frames
- Progressive encoding with 64×64 CTU size
- Support of tiled mode up to four tiles with horizontal splits only
- Wave front parallel encoding
- Motion Estimation (ME) search window dimensions: ±128 pixels horizontally, ±64 pixels vertically

- ME search precision: down to Quarter Picture Element (QPEL) resolution
- Luma intra-modes: 8×8, 16×16, and 32×32
- Chroma intra-modes: 4×4, 8×8, and 16×16
- Inter-modes: 8×8, 16×16, and 32×32
- Transform size for luma: 8×8, 16×16, and 32×32
- Transform size for chromas: 4×4, 8×8, and 16×16
- Skipped CUs and Merge modes
- Deblocking
- Sample Adaptive Offset (SAO)
- Constrained intra-prediction selectable
- Fixed Quantization Parameters (QP) or rate-controlled operation.
- Rate control uses a leaky bucket model based on bitrate and buffer size settings
- Long term reference frame support
- Selectable intra-frame refresh interval
- Slice insertion on a CTU row granularity
- Selectable limits for the search window and split options
- Encoders do not prevent the output from exceeding the maximum number of bits per CTU

2.4.2.2.2 H.264 Encoding Features

- Encoded bitstream is compliant with the Baseline, Main, High Profiles
- Encoding speed of 1080p@60fps (dual cores at approximately 300 MHz)
- Bitrates up to 50MBit/s using a single core operating at 300MHz
- Max frame width and height: 4096 pixels.
- Support of I, P, and B frames
- Support of progressive encoding
- Context Adaptive Binary Arithmetic Coding (CABAC) or Context Adaptive Variable Length Coding (CAVLC) entropy coding

> **Note.** B frames are not supported with CAVLC entropy coding

- Motion Estimation (ME) search window dimensions: ±128 pixels horizontally, ±64 pixels vertically
- ME search precision: down to Quarter Picture Element (QPEL) resolution
- Luma intra-modes: 4×4, 8×8, 16×16
- Chroma intra-modes: 8×8
- Inter-modes: 8×8, and 16×16
- Transform size: 4×4 and 8×8
- Support of skipped macroblocks
- Deblocking
- Constrained intra-prediction selectable
- Fixed QP operation or rate-controlled operation
- Rate control uses a leaky bucket model based on bitrate and buffer size settings
- Support of long term reference frame
- Selectable intra-frame refresh intervals
- Slice insertion granularity of 32-pixel high rows
- Possible to limit the search window and the macroblock split options
- Always enabled the escape option to prevent the emulation of a Network Abstraction Layer (NAL) unit start code regardless of the NAL packet format setting

> **Notes.**
>
> - For further details, please refer to ITU-T H.264 Annex B: VC-1 Compressed Video Bitstream Format and Decoding Process
> - Encoders do not prevent the output from exceeding the maximum number of bits per macroblock

2.4.2.2.3 VP8 Encoding Features

- Encoding speed of 1080p@60fps (dual core at approximately 400 MHz)

- Bitrate up to 50MBit/s using a single core operating at 400MHz
- Max frame width and height: 2048 pixels
- Support of I and P frames
- Support of progressive encoding
- Motion Estimation (ME) search window dimensions: ±128 pixels horizontally, ±64 pixels vertically
- ME search precision: down to QPEL resolution
- Luma intra-modes: 4×4, 8×8, 16×16
- Chroma intra-modes: 8×8
- Inter-modes: 8x8, and 16×16
- Support of macroblocks skipping
- Deblocking
- Fixed QP operation or rate-controlled operation
- Rate control uses a leaky bucket model based on bitrate and buffer size settings
- Selectable intra-frame refresh intervals
- Possible to limit the search window and the macroblock split

2.4.2.2.4 VP9 Encoding Features

- Encoded bitstream is compliant with VP9 Profile 0 at 8-bit depth
- Encoding speed of 1080p@60fps (dual core at approximately 300 MHz)
- Bitrate up to 50MBit/s using a single core operating at 300MHz
- Max frame width and height: 4096 pixels
- Support of 8-bit sample depth
- Support of I and P frames
- Support of progressive encoding
- Tiled rows and columns
- Motion Estimation (ME) search window dimensions: ± 128 pixels horizontally, ± 64 pixels vertically

- ME search precision: down to Quarter Picture ELement (QPEL) resolution
- Luma intra-modes: 8×8, 16×16, and 32×32
- Chroma intra-modes: 4×4, 8×8, and 16×16
- Inter-modes: 8×8, 16×16, and 32×32
- Transform size for luma: 8×8, 16×16, and 32×32
- Transform size for chroma: 4×4, 8×8, and 16×16
- Support of superblocks skipping
- Deblocking
- Fixed QP operation or rate-controlled operation
- Rate control uses a leaky bucket model based on bitrate and buffer size settings
- Selectable intra-frame refresh intervals
- Support of implicit or explicit probability update using delayed contexts

2.4.2 Video Decoder

2.4.2.1 Decoding Features

- Support of the following source frame output formats:
  - 2-plane YUV420 scan line format: chroma interleaved in UV or VU order
  - 3-plane YUV420 scan line format

    > **Notes.**
    >
    > - Support of 3-plane format is included for testing purposes only, do not use such max performance for normal applications
    > - Ensure of correct alignment of YUV buffer and stride for optima performance
- YUV420 AFBC format, 8-bit color depth
- Configurable for AFBC 1.0 or AFBC 1.2 output

- Support of stride for scan-line formats only
- Decoded frame rotation is supported in 90-degree steps before output

> **Note.** Not applicable for AFBC output formats

- Support of output average luminance (brightness) and chrominance (color) values for each 32×32 pixel block in every displayed output frame

2.4.2.2 Supported Decoding Formats

- HEVC (H.265): Main Profile
- H.264: Baseline, Main, High Profile
- VP8
- VP9: Profile 0
- VC-1: SP/MP/AP
- MPEG4: SP/ASP
- MPEG2: MP
- H.263: Profile 0

2.4.2.2.1 HEVC (H.265) Decoding Features

- Fully compliance with the Main Profiles
- Support of 2160p@30fps using dual core operating at approximately 300MHz
- Handle of average bitrate up to 100MBit/s with a single core at 600MHz
- Max frame width and height: 4096 pixels
- Error concealment is performed for handling bit errors
- Output of relevant stream parameter information during decoding

2.4.2.2.2 H.264 Decoding Features

- Fully compliance with H.264 Baseline, Main, High and High 10 progressive Profiles

- For streams using Flexible Macroblock Ordering (FMO) or Arbitrary Slice Ordering (ASO) in Baseline Profile, it is used WVGA resolution with decoding speed of 30fps with a single core at 400MHz
- For streams without FMA and ASO, the decoding speeds are as follows:
    - 2160p@30fps using dual core at approximately 300MHz
    - 1080i@120fps using dual core at 400MHz
- For progressive streams:
    - Average bitrate up to 100MBit/s with a single core at 600MHz
    - Max frame width and height: 4096 pixels
- For interlaced streams:
    - Average bitrate up to 50MBit/s with a single core at 400MHz
    - Max frame width: 2048 pixels
    - Max frame height: 4096 pixels
- Error concealment is performed for managing bitstream errors
- Output of relevant stream parameter information during decoding
- Always enabled the escape option to prevent the emulation of a Network Abstraction Layer (NAL) unit start code, regardless of the NAL packet format setting

> **Note.** For further details, please refer to ITU-T H.264 Annex B: VC-1 Compressed Video Bitstream Format and Decoding Process

2.4.2.2.3 VP8 Decoding Features

- Fully compliance with the VP8 Specification
- Support of decoding speed of 1080p@60fps using dual core at approximately 400MHz
- Average bitrate up to 50MBit/s with single core at 400MHz
- Max frame width and height: 2048 pixels
- Error concealment is performed for managing bitstream errors

2.4.2.2.4 VP9 Decoding Features

- Fully compliance with Profile 0
- Support of decoding speed of 2160p@30fps using dual core at approximately 300MHz and assuming no non-visible and no Alt-Ref frames
- Support of decoding speed of 2160p@30fps using dual core at approximately 400MHz and assuming an Alt-Ref frame distance of 4
- Average bitrate up to 60MBit/s using single core at 600MHz
- Max frame width and height: 4096 pixels
- Error concealment is performed for managing bitstream errors
- Output of relevant stream parameter information during decoding

2.4.2.2.5 VC-1 Decoding Features

- Fully compliance with VC-1 Simple, Main, and Advanced Profiles
- Support of decoding speeds of 1080p@60fps and 1080i@120fps using dual core at approximately 400MHz
- Average bitrate up to 40MBit/s with single core at 400MHz
- Max frame width: 2048 pixels
- Max frame height: 4096 pixels
- Error concealment is performed for managing bitstream errors

> **Notes.**
>
> - Advanced Profile bitstream data must always include the Encapsulation Mechanism regardless of the NAL packet format setting
> - For further details, please refer to SMPTE-421M-2006 Annex E
> - The range mapping feature of the VC-1 Advanced Profile does not apply to AFBC output

2.4.2.2.6 MPEG4 Decoding Features

- Compliance with MPEG4 Simple Profile and Advanced Simple Profile

- Support of Global Motion Compensation (GMC) with a limitation of a single warp point
- Support of decoding speed of 1080p@60fps or 1080i@120fps using dual core at 400MHz
- Handle of average bitrate up to 20MBit/s with a single core operating at 400MHz
- Max frame width and height: 2048 pixels
- Error concealment is performed for managing bitstream errors

2.4.2.2.7 MPEG2 Decoding Features

- Compliance with MPEG2 Main Profile
- Support of decoding speed of 1080p@60fps or 1080i@120fps using dual core at 400MHz
- Handle of average bitrate up to 20MBit/s with single core operating at 400MHz
- Max frame width: 4906 pixels (2,048 pixels for interlaced stream)
- Max frame height: 4096 pixels
- Error concealment is performed for managing bitstream errors

2.4.2.2.8 H.263 Decoding Features

- Compliance with H.263 Profile 0
- Support of decoding speed of 1080p@60fps using dual core at approximately 400MHz
- Handle of average bitrates up to 20MBit/s with single core operating at 400MHz
- Max frame width and height: 2048 pixels
- Error concealment is performed for managing bitstream errors

2.5 Display Subsystem

2.5.1 Display Controller

2.5.1.1 Introduction

The Display Controller is a hardware block that is used to transfer display data from the display's internal memory to the DSI controller. It supports one independent display device through MIPI DSI.

2.5.1.2 Features

- Support of up to HD+ (1920x1080@60fps)
- Support of up to 4-full-size-layer composer and maximum 8 layer-composers by up-down layer reuse in the RDMA channel
- Support of *cmdlist* mechanism allowing hardware register parameters to be configured
- Support of concurrent write-back operations with both raw and AFBC format
- Support of dithering, cropping, rotation in write-back path
- Advanced MMU (virtual address) mechanism for nearly no page missing during 90° and 270° rotation
- Support of color keying and solid color generation
- Support of both advanced error diffusion and pattern-based dithering for the panel
- Support of both AFBC and raw format image sources
- Color saturation and contrast enhancement
- Support of both video mode and *cmd* mode (with frame buffer in LCM) for the panel
- Support of dynamic DDR frequency adjustment with an embedded DFC buffer
- Support of the following **input formats** (see also the map shown immediately after):
    - A2BGR101010, A2RGB101010, BGR101010A2, RGB101010A2
    - ABGR8888, ARGB8888, BGRA8888, RGBA8888
    - XBGR8888, XRGB8888, BGRX8888, RGBX8888

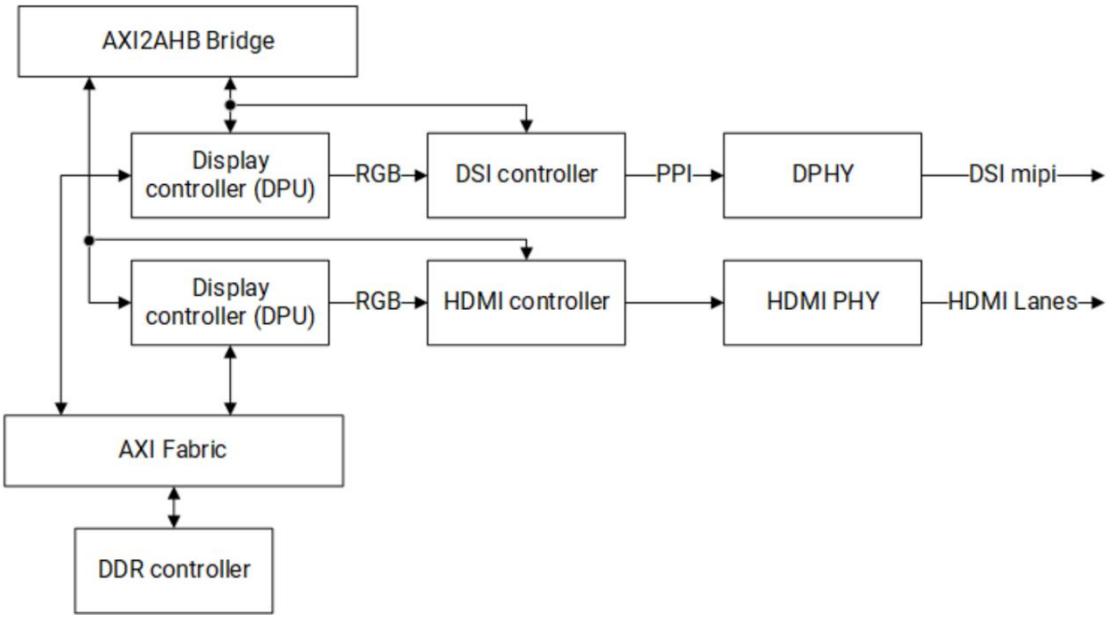
- BGR888, RGB888, ABGR1555, RGBA5551, BGR565/RGB565
- XYUV_444_P1_8, XYUV_444_P1_10, YVYU_422_P1_8, VYUY_422_P1_8
- YUV_420_P2_8, YUV_420_P3_8

Format Name	Byte Address and Order															
	127 15 120	119 14 112	111 13 104	103 12 96	95 11 88	87 10 80	79 9 72	71 8 64	63 7 56	55 6 48	47 5 40	39 4 32	31 3 24	23 2 16	15 1 8	7 0 0
ABGR_2101010	A3 [1:0]	B3[9:0]	G3[9:0]	R3[9:0]	A2 [1:0]	B2[9:0]	G2[9:0]	R2[9:0]	A1 [1:0]	B1[9:0]	G1[9:0]	R1[9:0]	A0 [1:0]	B0[9:0]	G0[9:0]	R0[9:0]
ARGB_2101010	A3 [1:0]	R3[9:0]	G3[9:0]	B3[9:0]	A2 [1:0]	R2[9:0]	G2[9:0]	B2[9:0]	A1 [1:0]	R1[9:0]	G1[9:0]	B1[9:0]	A0 [1:0]	R0[9:0]	G0[9:0]	B0[9:0]
BGRA_2101010	B3[9:0]	G3[9:0]	R3[9:0]	A3 [1:0]	B2[9:0]	G2[9:0]	R2[9:0]	A2 [1:0]	B1[9:0]	G1[9:0]	R1[9:0]	A1 [1:0]	B0[9:0]	G0[9:0]	R0[9:0]	A0 [1:0]
RGBA_2101010	R3[9:0]	G3[9:0]	B3[9:0]	A3 [1:0]	R2[9:0]	G2[9:0]	B2[9:0]	A2 [1:0]	R1[9:0]	G1[9:0]	B1[9:0]	A1 [1:0]	R0[9:0]	G0[9:0]	B0[9:0]	A0 [1:0]
ABGR_8888	A3[7:0]	B3[7:0]	G3[7:0]	R3[7:0]	A2[7:0]	B2[7:0]	G2[7:0]	R2[7:0]	A1[7:0]	B1[7:0]	G1[7:0]	R1[7:0]	A0[7:0]	B0[7:0]	G0[7:0]	R0[7:0]
ARGB_8888	A3[7:0]	R3[7:0]	G3[7:0]	B3[7:0]	A2[7:0]	R2[7:0]	G2[7:0]	B2[7:0]	A1[7:0]	R1[7:0]	G1[7:0]	B1[7:0]	A0[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
BGRA_8888	B3[7:0]	G3[7:0]	R3[7:0]	A3[7:0]	B2[7:0]	G2[7:0]	R2[7:0]	A2[7:0]	B1[7:0]	G1[7:0]	R1[7:0]	A1[7:0]	B0[7:0]	G0[7:0]	R0[7:0]	A0[7:0]
RGBA_8888	R3[7:0]	G3[7:0]	B3[7:0]	A3[7:0]	R2[7:0]	G2[7:0]	B2[7:0]	A2[7:0]	R1[7:0]	G1[7:0]	B1[7:0]	A1[7:0]	R0[7:0]	G0[7:0]	B0[7:0]	A0[7:0]
XBGR_8888	X[7:0]	B3[7:0]	G3[7:0]	R3[7:0]	X[7:0]	B2[7:0]	G2[7:0]	R2[7:0]	X[7:0]	B1[7:0]	G1[7:0]	R1[7:0]	X[7:0]	B0[7:0]	G0[7:0]	R0[7:0]
XRGB_8888	X[7:0]	R3[7:0]	G3[7:0]	B3[7:0]	X[7:0]	R2[7:0]	G2[7:0]	B2[7:0]	X[7:0]	R1[7:0]	G1[7:0]	B1[7:0]	X[7:0]	R0[7:0]	G0[7:0]	B0[7:0]
BGRX_8888	B3[7:0]	G3[7:0]	R3[7:0]	X[7:0]	B2[7:0]	G2[7:0]	R2[7:0]	X[7:0]	B1[7:0]	G1[7:0]	R1[7:0]	X[7:0]	B0[7:0]	G0[7:0]	R0[7:0]	X[7:0]
RGBX_8888	R3[7:0]	G3[7:0]	B3[7:0]	X[7:0]	R2[7:0]	G2[7:0]	B2[7:0]	X[7:0]	R1[7:0]	G1[7:0]	B1[7:0]	X[7:0]	R0[7:0]	G0[7:0]	B0[7:0]	X[7:0]
ABGR_1555	A7 [0] B7 [4:0]	G7 [4:0] R7 [4:0]	A6 [0] B6 [4:0]	G6 [4:0] R6 [4:0]	A5 [0] B5 [4:0]	G5 [4:0] R5 [4:0]	A4 [0] B4 [4:0]	G4 [4:0] R4 [4:0]	A3 [0] B3 [4:0]	G3 [4:0] R3 [4:0]	A2 [0] B2 [4:0]	G2 [4:0] R2 [4:0]	A1 [0] B1 [4:0]	G1 [4:0] R1 [4:0]	A0 [0] B0 [4:0]	G0 [4:0] R0 [4:0]
RGBA_1555	R7 [4:0]	G7 [4:0]	B7 [4:0]	A7 [0]	R6 [4:0]	G6 [4:0]	B6 [4:0]	A6 [0]	R5 [4:0]	G5 [4:0]	B5 [4:0]	A5 [0]	R4 [4:0]	G4 [4:0]	B4 [4:0]	A4 [0]
BGR_565	B7 [4:0]	G7 [5:0]	R7 [4:0]	B6 [4:0]	G6 [5:0]	R6 [4:0]	B5 [4:0]	G5 [5:0]	R5 [4:0]	B4 [4:0]	G4 [5:0]	R4 [4:0]	B3 [4:0]	G3 [5:0]	R3 [4:0]	B2 [4:0]
RGB_565	R7 [4:0]	G7 [5:0]	B7 [4:0]	R6 [4:0]	G6 [5:0]	B6 [4:0]	R5 [4:0]	G5 [5:0]	B5 [4:0]	R4 [4:0]	G4 [5:0]	B4 [4:0]	R3 [4:0]	G3 [5:0]	B3 [4:0]	R2 [4:0]
XYUV_444_P1_8	X[7:0]	Y03[7:0]	U03[7:0]	V03[7:0]	X[7:0]	Y02[7:0]	U02[7:0]	V02[7:0]	X[7:0]	Y01[7:0]	U01[7:0]	V01[7:0]	X[7:0]	Y00[7:0]	U00[7:0]	V00[7:0]
XYUV_444_P1_10	X [1:0]	Y03[9:0]	U03[9:0]	V03[9:0]	X [1:0]	Y02[9:0]	U02[9:0]	V02[9:0]	X [1:0]	Y01[9:0]	U01[9:0]	V01[9:0]	X [1:0]	Y00[9:0]	U00[9:0]	V00[9:0]
YVYU_422_P1_8	Y07[7:0]	V06[7:0]	Y06[7:0]	U06[7:0]	Y05[7:0]	V04[7:0]	Y04[7:0]	U04[7:0]	Y03[7:0]	V02[7:0]	Y02[7:0]	U02[7:0]	Y01[7:0]	V00[7:0]	Y00[7:0]	U00[7:0]
VYUY_422_P1_8	V06[7:0]	Y07[7:0]	U06[7:0]	Y06[7:0]	V04[7:0]	Y05[7:0]	U04[7:0]	Y04[7:0]	V02[7:0]	Y03[7:0]	U02[7:0]	Y02[7:0]	V00[7:0]	Y01[7:0]	U00[7:0]	Y00[7:0]
YUV_420_P2_8 Y PLANE	Y15[7:0]	Y14[7:0]	Y13[7:0]	Y12[7:0]	Y11[7:0]	Y10[7:0]	Y09[7:0]	Y08[7:0]	Y07[7:0]	Y06[7:0]	Y05[7:0]	Y04[7:0]	Y03[7:0]	Y02[7:0]	Y01[7:0]	Y00[7:0]
YUV_420_P2_8 UV PLANE	V14[7:0]	U14[7:0]	V12[7:0]	U12[7:0]	V10[7:0]	U10[7:0]	V08[7:0]	U08[7:0]	V06[7:0]	U06[7:0]	V04[7:0]	U04[7:0]	V02[7:0]	U02[7:0]	V00[7:0]	U00[7:0]
YUV_420_P3_8 Y PLANE	Y15[7:0]	Y14[7:0]	Y13[7:0]	Y12[7:0]	Y11[7:0]	Y10[7:0]	Y09[7:0]	Y08[7:0]	Y07[7:0]	Y06[7:0]	Y05[7:0]	Y04[7:0]	Y03[7:0]	Y02[7:0]	Y01[7:0]	Y00[7:0]
YUV_420_P3_8 U PLANE	U30[7:0]	U28[7:0]	U26[7:0]	U24[7:0]	U22[7:0]	U20[7:0]	U18[7:0]	U16[7:0]	U14[7:0]	U12[7:0]	U10[7:0]	U8[7:0]	U6[7:0]	U4[7:0]	U2[7:0]	U0[7:0]
YUV_420_P3_8 V PLANE	V30[7:0]	V28[7:0]	V26[7:0]	V24[7:0]	V22[7:0]	V20[7:0]	V18[7:0]	V16[7:0]	V14[7:0]	V12[7:0]	V10[7:0]	V8[7:0]	V6[7:0]	V4[7:0]	V2[7:0]	V0[7:0]

- Support of the following **output formats**:
  - RGB888, RGB565, RGB666

2.5.1.3 Block Diagram

The micro-architecture of the display subsystem is depicted below.



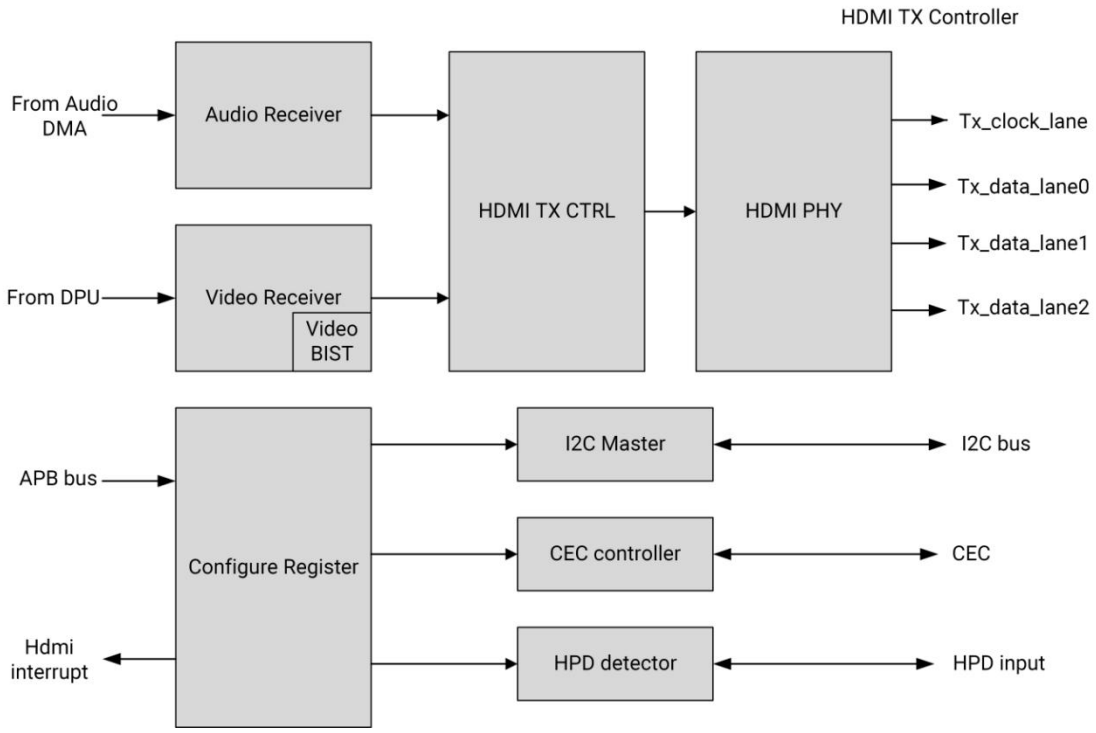
2.5.2 HDMI Interface

2.5.2.1 Features

- Compliance with HDMI Specification v1.4
- Dual-channel audio stream within the range 32~192KHz
- Physical lane speed up to 2.4Gbps/lane × 3lane
- Support of up to 1920x1440@60Hz
- Support of RGB and YcbCr 4:2:2 / 4:4:4 input video format
- Support of RGB and YcbCr 4:2:2 / 4:4:4 output video formats
- Support of 8bpc / 10bpc / 12bpc input and output color depths
- Support of EIA/CEA-861-F video timing and InfoFrame structure
- Support of L-PCM(IEC 60958), 32~192KHz dual channel audio data
- Support of Consumer Electronic Control (CEC) standard packets and user-defined packets
- Inclusion of an Internal I2C Master for remote ED access supporting 100~400Kbps speed

2.5.2.2 Block Diagram

The architecture of the HDMI interface is depicted below.



2.5.3 MIPI DSI Interface

2.5.3.1 Introduction

The MIPI Display Serial Interface (MIPI DSI) is a high-speed interface between a host processor and peripheral devices that adheres to MIPI Alliance specifications for mobile device interfaces.

2.5.3.2 Features

- Compliance with the MIPI DSI standard v1.0
- Compliance with the MIPI DPHY specification v1.1

- Support of MIPI DPHY up to 4 data lanes and speed up to 1200Mbps per lane
- Support of 1 active panel per DPHY link
- Compliance with the Display Command Set (DCS) standard
- Support of all pixel formats defined in DSI and DCS
- Support of video burst mode with DPHY up to 1.2Ghz per lane
- Support of virtual channels in the MIPI Link
- Support of up to 1080p resolution
- Support of command, video and burst modes
- Support of HS-TX, LP-TX, LP-RX and LP-CD signaling

2.5.4 SPI LCD Display Interface

2.5.4.1 Introduction

The SPI LCD Display Interface is used to

- Send image data commands
- Read image data
- Transmit image data

It supports the operational modes

- Single data line mode
- Dual data line mode

where each of which support the work modes

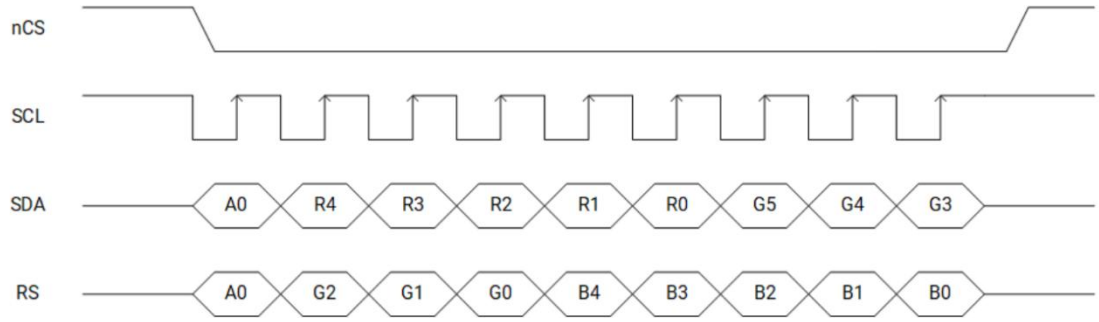
- 3-line/9bit mode
- 4-line/8bit mode

By software, it is possible to configure which line will be the first for transmitting data. Further, it is possible to configure the transfer mode choosing between

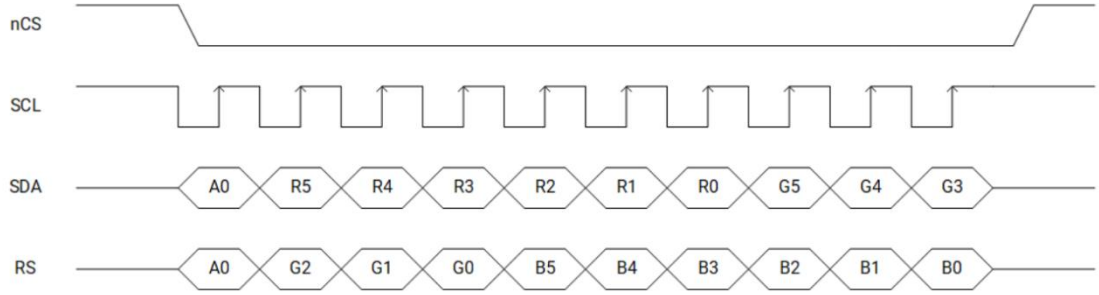
- Packet transfer mode
- Unpacked transfer mode

As example, below are depicted the transfers modes for some color formats, highlighting how data are organized and transmitted.

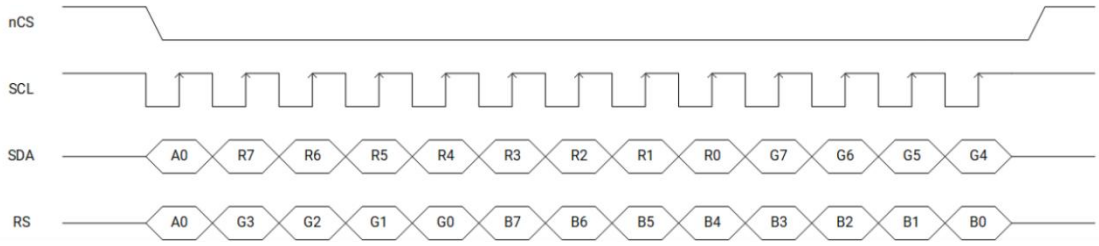
[Packet transfer mode for RGB565]



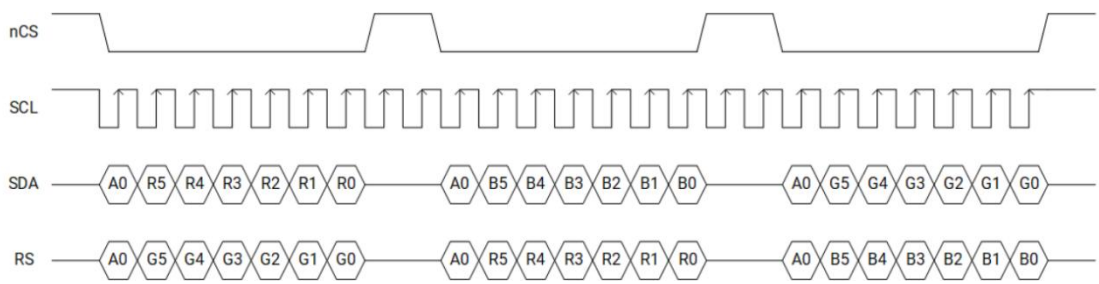
[Packet transfer mode for RGB666]



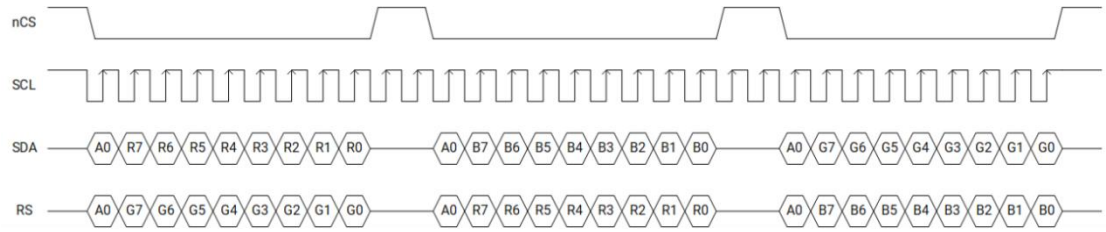
[Packet transfer mode for RGB888]



[Unpacked transfer mode for RGB666]



[Unpacked transfer mode for RGB888]



2.5.4.2 Features

- Support of SPI LCD module with resolution up to 320x240
- Support of 3-/4-line Serial Peripheral Interface (SPI) and 2-line SPI data transmission
- Support of up to 3 simultaneous overlays (2 for RGB, 1 for YUV & RGB)
- Support of dithering
- Support of gamma curve
- Alpha blending with configurable alpha values or per-pixel alpha blending
- Support of YUV to RGB color space conversion
- Support of image scaling
- Support of color keying
- Support of memory write-back
- Support of the following **input formats** for **image layer**:
  - YUV422 planar
  - YUV422 packet
  - YUV420 planar
  - RGB888
  - RGB565
  - RGB666
  - BGR888
  - BGR565
  - BGR666

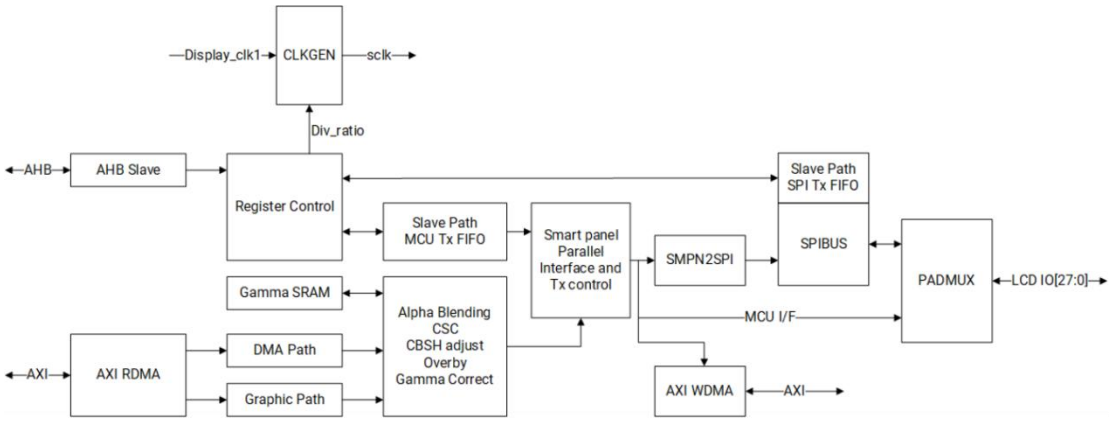
> **Note.** As can be seen, it is supported **R-B swap option** for the sake of flexibility

- Support of the following **input formats** for **OSD layer**:
  - RGB888
  - RGB565
  - RGB666
  - BGR888
  - BGR565
  - BGR666

> **Note.** As can be seen, it is supported **R-B swap option** for the sake of flexibility

2.5.4.3 Block Diagram

The architecture of the SPI LCD Display Interface is depicted below.



It is clearly understandable how the display data are efficiently processed, then converted into SPI-compatible signals, then transmitted to the connected LCD display.

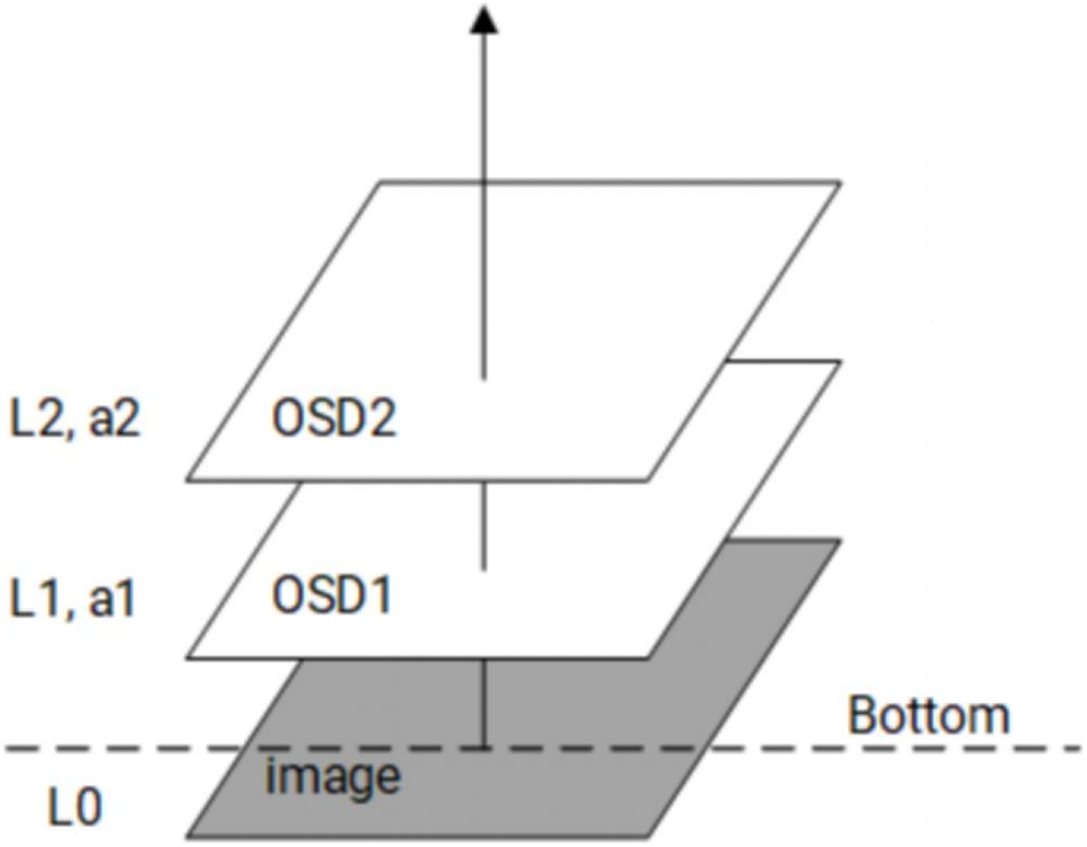
2.5.4.4 Functions

2.5.4.4.1 Blending Function

The blending function of the DSI controller is used to combine multiple layers of images or overlays with different levels of transparency (alpha values).

An example of layers and their respective alpha values is depicted below, where

- **L0**: Bottom layer, base image
- **L1**: Middle layer, alpha value **a1**
- **L2**: Top layer, alpha value **a2**



The following blending modes are supported:

- Normal Alpha Blending Mode
- Pre-Multiple Alpha Blending Mode
- Special Alpha Blending Mode

In the code, a different formula is implemented for each blending mode that uses the alpha value **a1** as per the following conditions:

```
C
```

```
if (L1 == color_key)
a1 = 8’h0;
else if (layer_alpha_sel == 1)
a1 = layer_alpha;
else
a1 = pixel_alpha;
```

Details for each blending mode are explained in the following subsections.

2.5.4.4.1.1 Normal Alpha Blending Mode

With reference to the example figure shown above,

- For **2 layers**, the formula implemented is
  - $$L^{'} = L1 \times a1 + L0 \times (1 - a1)$$
- For **3 layers** (not recommended), the formula implemented is
  - $$L^{'} = L2 \times a2 + L1 \times a1 \times (1 - a2) + L0 \times (1 - a1) \times (1 - a2)$$

> **Note.** Alpha value is not supported for write-back in this case

In the code, the pixel value **L'** depends on the alpha value **a1** as per the following conditions:

```
C
if (a1 == 8’hFF)
L' = L1;
else if (a1 == 8’h00)
L' = L0;
else
L' = (L1-L0) × a1/256 + L0
```

2.5.4.4.1.2 Pre-Multiple Alpha Blending Mode

With reference to the example figure shown above,

- For **2 layers**, the formula implemented is

- $L^{'} = L1 + L0 \times (1 - a1)$

- For **3 layers** (not recommended), the formula implemented is
  - $L^{'} = L2 + L1 \times (1 - a2) + L0 \times (1 - a1) \times (1 - a2)$

> **Note.** Alpha value is supported for write-back and its value is given by the formula $a^{'} = a1 + a2 - a1 \times a2$

In the code, the pixel value **L'** depends on the alpha value **a1** as per the following conditions:

```
C
if (a1 == 8’hFF)
L' = L1;
else if (a1 == 8’h00)
L' = L0;
else
L' = L1-L0 × (1-a1)/256;
```

2.5.4.4.1.3 Special Alpha Blending Mode

With reference to the example figure shown above,

- For **2 layers**, the formula implemented is
  - $L^{'} = L1 + L0 \times a1$

- For **3 layers** (not recommended), the formula implemented is
  - $L^{'} = L2 + L1 \times a2 + L0 \times a1 \times a2$

> **Note.** Alpha value is not supported for write-back in this case

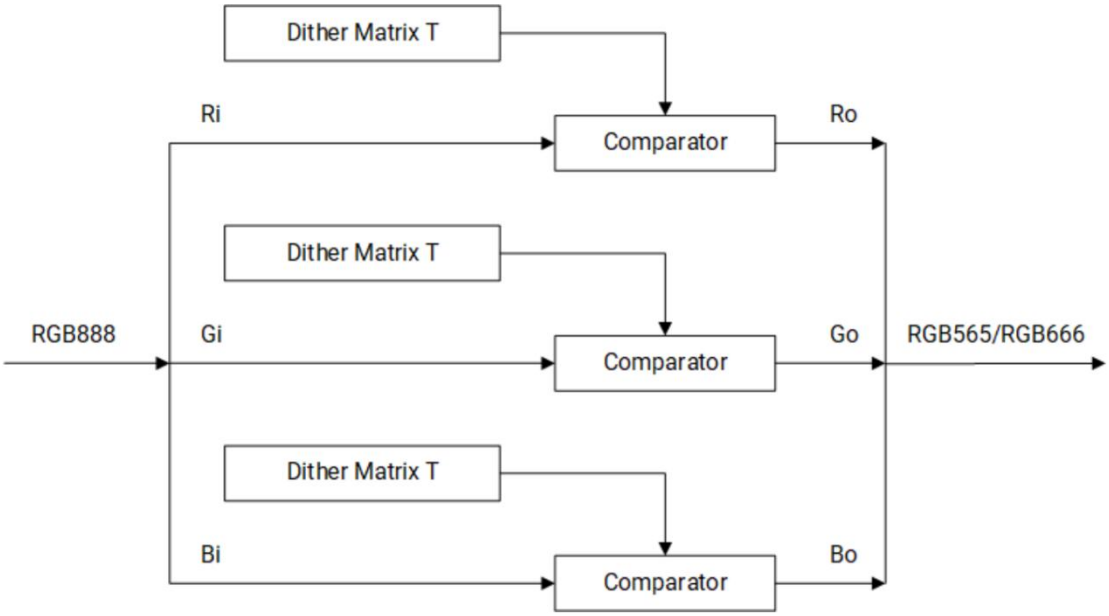
In the code, the pixel value **L'** depends on the alpha value **a1** as per the following conditions:

```
C
```

```
if (a1 == 8’hFF)
L' = L0;
else
L' = L1 + L0 × a1/256;
```

2.5.4.4.2 Dither Function

The process of the Dither function is depicted below.



The Dither function can be enabled/disabled by software.

2.5.4.4.3 Fmark Function

The Fmark function controls the start of displaying output. In particular,

- If Fmark function is **enabled**, displaying output will wait until the Fmark signal is received
- If Fmark function is **disabled**, displaying output will start immediately after initiated by software

By software is possible to enable/disable Fmark function as well as control the polarity of the Fmark signal.

It is recommended to have a register to set how long displaying output is delayed after LCDC received the Fmark signal.

2.5.4.4.4 Background Color Display Function

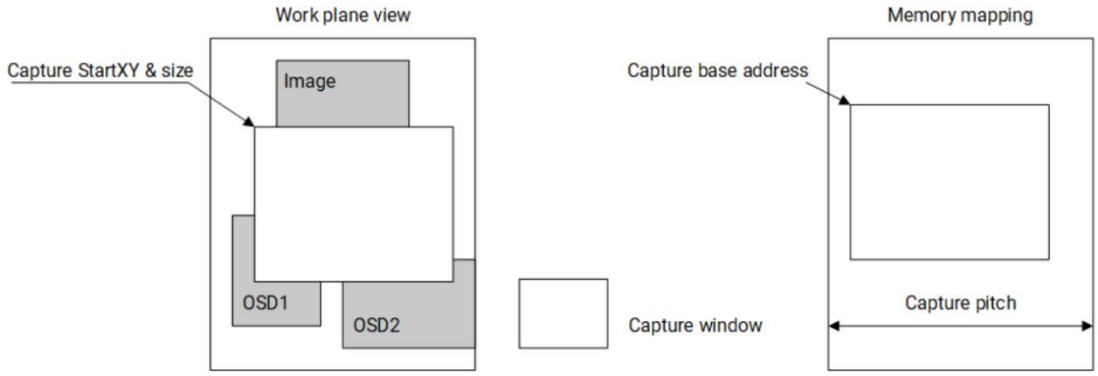
When no layer is enabled, a background color can be displayed without fetching data from DDR. The background color can be configured by software.

2.5.4.4.5 Image Capture Function

To apply the image capture function, the following parameters should be configured by software firstly:

- **startx** = X coordinate of the start point of the capture
- **starty** = Y coordinate of the start point of the capture
- **width** = Width (in pixels) of the capture from (X,Y) start point
- **height** = Height (in pixels) of the capture from (X, Y) start point
- **base_addr** = Memory start address for storing the capture
- **pitch** = Distance (in bytes) between the start of two consecutive rows of pixels stored in the memory, including any padding for alignment or hardware requirements

The process of the image capture function is depicted below.



2.6 Audio Subsystem

2.6.1 Introduction

Audio subsystem integrates two primary interfaces:

- 2 × Full-Duplex I2S Interfaces

- 1 × HDMI Audio Interface

2.6.2 Features

I2S Interfaces

- Full-duplex operation with simultaneous playback and recording support
- Compliance with standard I2S format with fixed parameters:
  - 48 kHz sample rate
  - 16-bit data depth
  - 2 channels
- Configurable system clock (sysclk) modes: 64fs, 128fs or 256fs

HDMI Audio Interface

- Playback-only functionality with fixed parameters:
  - 48 kHz sample rate
  - 16-bit data depth
  - 2 channels

2.7 Connectivity Subsystem

2.7.1 PCIe 2.0

2.7.1.1 Introduction

X1 implements three PCIe Dual-Mode ports which can be configured as either Root Complex (RC) or Endpoint (EP) device.

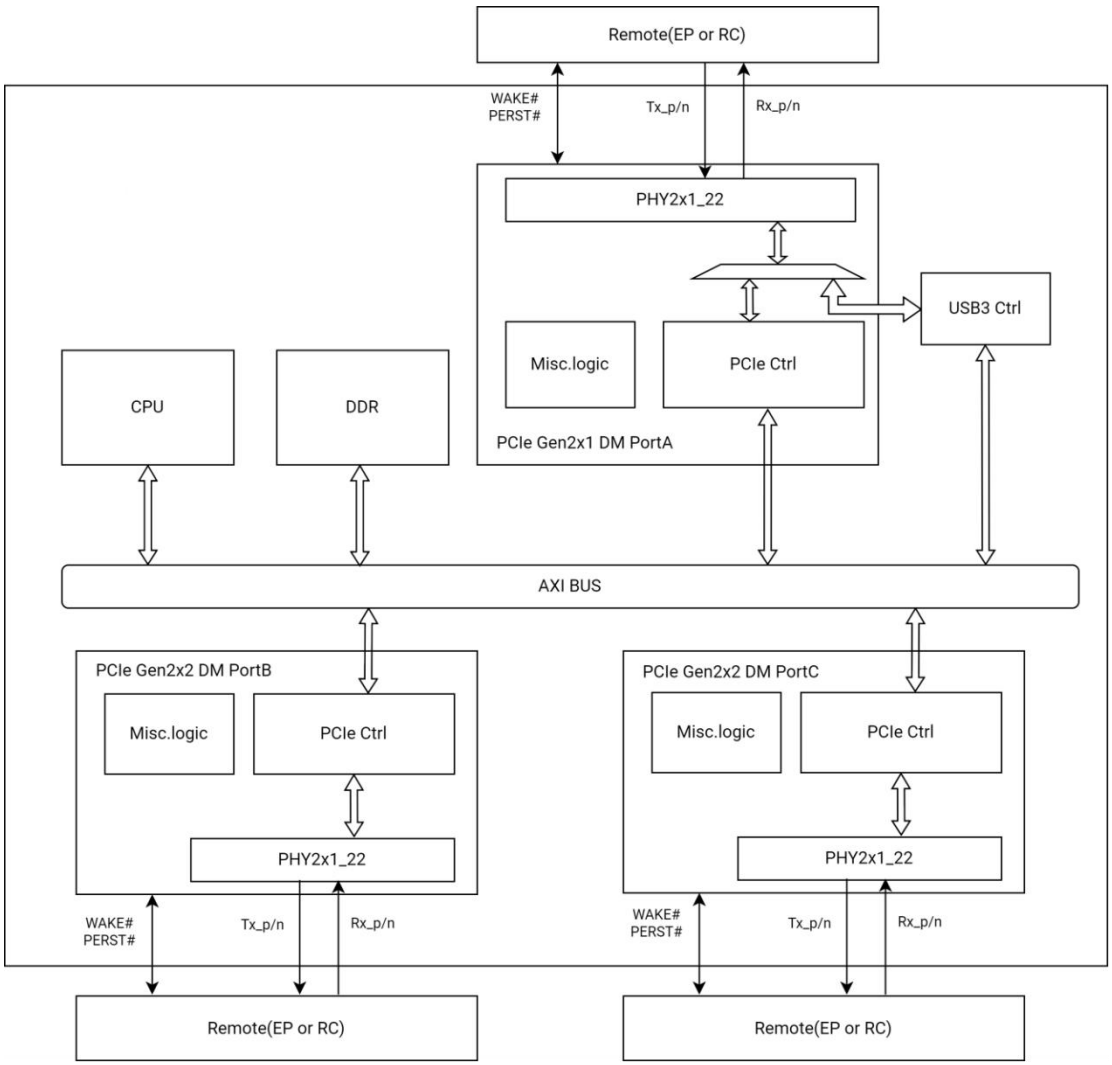
All ports support Gen2 with a data transfer speed of 5GT/s per lane. However, one port supports one lane only and two ports support two lanes each.

2.7.1.2 Features

- Support of Dual-Mode, programmable as either Complex (RC) or Endpoint (EP) device
- Support of all non-optional features of the PCI Express Base Specification - Revision 5.0 - Version 1.0 (limited to Gen2 speed scope)
- Support of Internal Address Translation Unit (iATU) with 8 entries for outbound and 8 entries for inbound traffic
- Support of Embedded DMA with Hardware Flow Control which includes 4 write channels and 4 read channels
- Support of ECRC generation and check
- Support of max payload size up to 256 bytes
- Supports Automatic Lane Flip and Reversal
- Support of L0 and L1 Power State of Active State Link PM
- Support of Latency Tolerance Reporting (LTR)
- Support of only Virtual Channel 0
- Support of ID Based Ordering (IDO)
- Support of Completion Timeout Ranges
- Support of Separate Reference Clock With Independent Spread (SRIS)
- Support of up to 64 outbound Non-Post Requests
- Support of up to 32 outstanding AXI slave Non-Post requests
- Support of only Function 0 with 6 size-programmable BARs in EP Mode
- Support of MSI Capability in EP Mode
- Support of Integrated MSI Reception Module in RC Mode

2.7.1.3 Block Diagram

The architecture of the PCIe Dual-Mode port set is depicted below.



As can be seen, there are

- One PCIe Gen2x1 Dual-Mode port (hereafter Port A)
- Two PCIe Gen2x2 Dual-Mode ports (hereafter Port B and Port C)

as said previously, and all them consists of

- A **controller** integrated into SoC via **3 AXI ports** which are designed as
  - **AXI Master Port** to manages inbound traffic (i.e. data coming into the system) either from a remote device or through the PCIe controller's internal DMA, allowing the access to DDR memory for transferring data both to and from the remote device
  - **AXI Data Slave Port** to allows the local CPU accessing itself for outbound traffic
  - **AXI DBI Slave Port** to be used for the PCIe controller's configuration interface

- A **PHY** complied with PIPE 3 specification and distinguished in
  - **Phy2x1_22** which
    - Supports Gen2 with one lane (x1)
    - Is built using a 22nm process
    - Is shared between Port A and USB3 controller but not simultaneously, i.e. both Port A and USB3 controller can operate but not at the same time
  - **Phy2x2_22** which
    - Supports Gen2 with two lanes (x2)
    - Is built using a 22nm process
    - Comes for Port B and Port C distinctly, i.e. Port B and Port C have their own dedicated PHY
- A **miscellaneous logic**, in particular **chip I/O with remote links partner** as follows:
  - **Differential Data Signals**: Rx_p/n, Tx_p/n (x2 lanes for Port B/C, x1 lane for Port A)
  - **Reference Clock Signals**: refclk_p/n (support of both input and output mode)
  - **Warm Reset Signal**: PERST# (input in EP mode, output in RC mode)
  - **Wake-Up signal**: WAKE# (output in EP mode, input in RC mode)

2.7.2 USB

2.7.2.1 Introduction

X1 includes three USB ports as follows:

- A USB2.0 OTG Port
- A USB2.0 Host Only Port
- A USB3.0 Port with a USB2.0 DRD interface

2.7.2.2 Features

2.7.2.2.1 USB2.0 OTG Port Features

- **Controller:**
  - Support of both USB2.0 Host and Device mode
  - Compliance with the USB2.0 standard
  - Support of USB2.0 High Speed (480Mb/s) and Full Speed (12Mb/s) for both Host and Device modes
  - Support of USB2.0 Low Speed (1.5Mb/s) for Host Only Mode
  - Host controller registers and data structures are compliant with the Intel EHCI specification
  - Device controller registers and data structures are implemented as extensions to the EHCI programming interface
  - Bus interface is compliant with AMBA-AHB specification
- **Communication Interface:**
  - Implementation of UTMI+ interface to communicate with USB2.0 PHY
- **Protocols:**
  - Support of the Session Request Protocol (SRP)
  - Support of the Host Negotiation Protocol (HNP)
- **Channel & Endpoint:**
  - Support of up to 16 host channels
  - In Device mode, support of 16 IN and 16 OUT endpoints, where
    - 16KB buffer is for transmitting data
    - 2KB buffer is for receiving data

2.7.2.2.2 USB2.0 Host Only Port Features

- **Controller:**

- Support of USB2.0 HS, USB2.0 FS, USB2.0 LS Host modes
- Compliance with the USB2.0 standard
- Support of High Speed (480Mb/s), Full Speed (12Mb/s), Low Speed (1.5Mb/s) for Host mode
- Host controller registers and data structures are compliant with the Intel EHCI specification
- Bus interface is compliant with AMBA-AHB specification

- **Communication Interface:**
  - Implementation of UTMI+ interface to communicate with USB2.0 PHY
- **Channel Support:**
  - Support of up to 16 host channels

2.7.2.2.3 USB3.0 Port with a USB2.0 DRD Interface Features

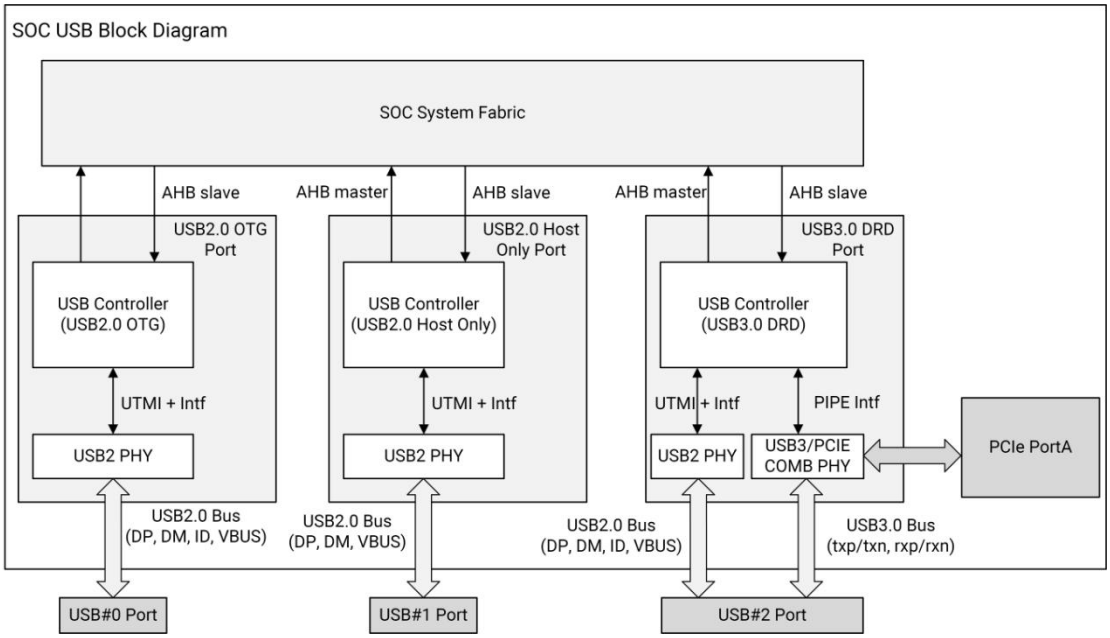
- **Controller**
  - Support of both USB3.0 Host and Device modes
  - Support of both USB2.0 Host and Device modes
  - Compliance with both the USB3.0 and USB2.0 standards
  - Support of USB3.0 (Super Speed) and USB2.0 Host and Device mode
  - USB3.0 Host Controller registers and data structures are compliant with the Intel xHCI specification
  - USB3.0 Device controller registers and data structures are self-defined requiring software configuration
  - Support of one USB3.0 port and one USB2.0 port
  - Support of High Speed (480Mb/s) and Full Speed (12Mb/s) for Host and Device mode
  - Support of Low Speed (1.5Mb/s) for Host-Only mode
- **Communication Interface:**
  - Use of PIPE3 (125MHz) interface for USB3.0 PHY
  - Use of UTMI+ (30/60MHz) interface for USB2.0 PHY

- **Clock Domains:**
  - PIPE3 PHY (125MHz)
  - UTMI+ PHY (30/60MHz)
  - MAC (nominal 125MHz)
  - BUS clock domain
  - RAM clock domain
- **System & Power Management:**
  - Internal DMA controller
  - Support of USB2.0 suspend mode
  - Support of U1/U2/U3 low-power modes for USB3.0
- **Endpoint & Memory:**
  - Support of up to 32 endpoints in Device mode
  - Flexible endpoint FIFO sizes (not limited to powers of 2) allowing the use of contiguous memory locations
  - Descriptor caching and data pre-fetching for improving performance in high-latency systems
- **Additional Features:**
  - Software-controlled standard USB commands (USB SETUP commands forwarded to application for decoding)
  - Hardware-level error handling for USB bus and packet-level issues
  - Support of interrupts

2.7.2.3 Block Diagram

The architecture of the USB port set is depicted below, where

- **USB#0 Port =** USB2.0 OTG Port
- **USB#1 Port =** USB2.0 Host-Only Port
- **USB#2 Port =** USB3.0 Port with a USB2.0 DRD interface



2.7.3 Ethernet GMAC

2.7.3.1 Introduction

X1 features a GMAC IP core which includes the essential protocol requirements for the operation of 10/100/1000 Mbps Ethernet/IEEE 802.3-2012 compliant node.

The GMAC IP core can operate at 10 Mbps, 100 Mbps (Fast Ethernet) or 1000 Mbps (Gigabit Ethernet). Additionally, it includes a powerful 64-bit Scatter-Gather DMA to transfer packets between HOST Memory and Internal FIFOs to achieve high performance.

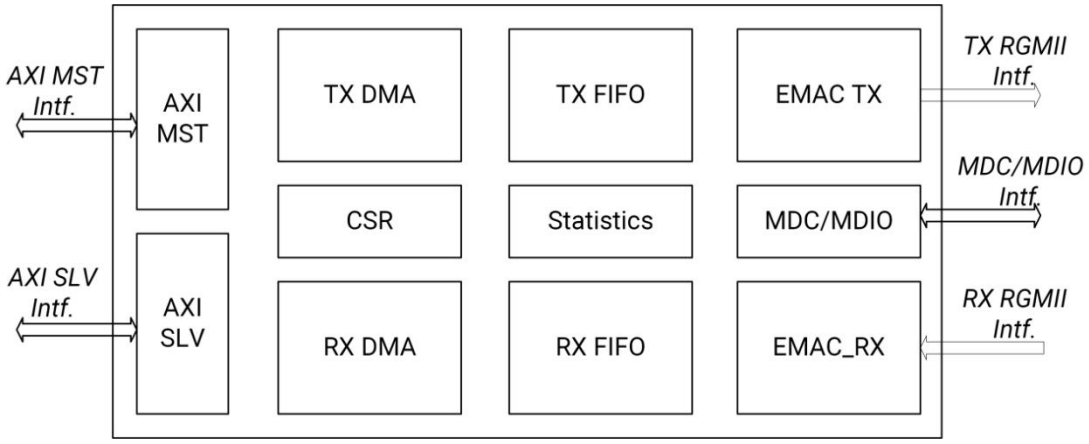
2.7.3.2 Features

- Handle of transmit/receive data encapsulation functions, including Framing (frame boundary delimitation, frame synchronization) and Error Detection (physical medium transmission errors)
- Media access management with medium allocation (collision avoidance) and contention resolution (collision handling) in Half-Duplex Mode of operation at speeds of 10/100 Mbps

- Retransmission of frames that result in Collision in Half-Duplex mode
- Support of Flow Control functions in Full Duplex mode by decoding PAUSE control frames, disabling the transmitter and generating PAUSE control Frames
- Support of a 4-bit data path based RGMII Interface to connect with RGMII-based PHY
- Support of Management Interface by generating management frames on the MDC/MDIO pins to communicate with external PHY devices
- Bus mastering on the AXI interface to transfer packets between the HOST memory and the internal FIFOs using 64-bit transfer mode
- Automatic transfer of packets between the HOST memory and internal FIFOs (based on descriptors) to minimize CPU overhead

2.7.3.3 Block Diagram

The micro-architecture of Ethernet GMAC unit is depicted below.



2.7.4 SDIO Interface

2.7.4.1 Introduction

The SDIO interface is a hardware block that serves as the host of the SDIO bus to transfer data between the SDIO Wi-Fi module and the internal bus master.

2.7.4.2 Features

- Compliance with with 4-bit SDIO 4.10 protocol specification
- Consistent with the register set defined in SD-HCI specification with additional vendor-specific registers
- Support of 1-bit and 4-bit SDIO bus
- Support of the following data transfer type defined in the SD-HCI specification:
  - PIO
  - SDMA
  - ADMA
  - ADMA2
- Support of the following speed modes defined in SD 3.0 specification:
  - Default Speed mode, up to 12.5MB/s, 3.3V signal level
  - High Speed mode, up to 25MB/s, 3.3V signal
  - SDR12, SDR up to 25 MHz, 1.8V signal
  - SDR25, SDR up to 50 MHz, 1.8V signal
  - SDR50, SDR up to 100 MHz, 1.8V signal
  - SDR104, SDR up to 208 MHz, 1.8V signal
  - DDR50, DDR up to 100MHz, 1.8V signal
- Hardware-based CRC generation and check for all command and data transactions on the card bus
- Support of read-wait control in SDIO cards
- Support of suspend/resume functionality in SDIO cards
- 1024 Bytes (2 x 512 Bytes data block) FIFO for sending and receiving data

2.7.5 CAN-FD Interface

2.7.5.1 Introduction

The CAN-FD controller is a full implementation of the CAN protocol specification which is compliant with both the CAN with Flexible Data-Rate (CAN-FD) protocol and CAN 2.0 Part B protocol.

2.7.5.2 Features

- Full implementation of the CAN-FD protocol and CAN specification 2.0 Part B with
  - Standard data frames
  - Extended data frames
  - Data lengths from 0 to 64 bytes
  - Programmable bit rate
  - Content-related addressing
- Compliant with the ISO 11898-1 standard
- Silicon-proven implementation passing ISO 16845-1:2016 CAN conformance tests
- Flexible mailboxes configurable to store 0, 8, 16, 32 or 64 bytes of data
- Each mailbox configurable to either receive or transmit supporting both standard and extended messages
- Distinct receive mask registers per mailbox
- Full-featured receive FIFO with a storage capacity of up to 6 frames with automatic internal pointer handling and DMA support
- Transmission abort capability
- Support of flexible message buffers with a total of 128 message buffer slots (8 bytes each) which can be configurable as transmitter or receiver
- Programmable clock source for the CAN Protocol Engine, either peripheral clock or oscillator clock
- RAM is not used for reception or transmission but can be used as general purpose RAM space

- Support of Listen-Only Mode (LOM)
- Programmable Loop-Back mode for self-test operation
- Programmable transmission priority scheme: based on lowest ID, lowest buffer number or highest priority
- 16-bit free-running timer for time stamps with an optional external time tick
- Global network time synchronized by a specific message
- Maskable interrupts
- Independence from the transmission medium (required an external transceiver)
- Short latency for high-priority messages due to an arbitration scheme
- Low-power modes with programmable wakeup on bus activity or frame matching (pretended networking)
- Transceiver Delay Compensation (TDC) when transmitting CAN-FD messages at faster data rates
- Remote request frames can be managed automatically by software
- CAN bit time settings and configuration can only be written in Freeze mode
- Configurable transmission mailbox status: either lowest priority buffer or empty buffer
- Support of Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- SYNCH bit in Error Status 1 register indicates synchronization with the CAN bus
- Support of CRC status for transmitted message
- Support for reception FIFO Global Mask register
- Selectable priority between mailboxes and reception FIFO during matching process
- Advanced receive FIFO ID filtering, capable of matching incoming IDs against either 128 extended IDs, 256 standard IDs, or 512 partial (8 bit) IDs, with up to 32 elements in the ID Filter Table
- Fully backward compatibility with previous CAN-FD version
- Support of detection and correction of errors in memory read accesses. Each byte of CAN-FD memory is paired with 5 parity bits, forming a 13-bit word. The error correction mechanism can

    - Detect and correct single-bit errors (correctable errors)
    - Detect, but not correct, two-bit errors (non-correctable errors)
- Support of pretended networking functionality in low-power modes: Doze mode and Stop mode

2.7.6 SPI Interface

2.7.6.1 Introduction

The SPI interface is a synchronous serial interface that allows the communication with external devices using Motorola Serial Peripheral Interface (SPI) protocol for data transfer. It can be configured to operate in either Master mode (where the attached peripheral functions as a slave) or Slave mode (where the attached peripheral functions as a master).

2.7.6.2 Feature

- Support of four combinations of CPOL and CPHA for Serial Peripheral Interface (SPI)
- Configurable to operate in either Master mode (where the attached peripheral functions as a slave) or Slave mode (where the attached peripheral functions as a master)
- Support of Receive-without-Transmit operation
- Support of serial bit rate from 6.3Kps (min recommended) to 52Mbps (max)
- Data size configurable to 8, 16, 18 or 32 bits in length
- Availability of a transmit FIFO (TXFIFO) and another independent receive FIFO (RXFIFO), where
    - In Non-Packed Data mode, both FIFOs are 32 rows deep x 32 bits wide supporting a total of 32 samples
    - In Packed Data mode, double-depth FIFOs are used when the data samples

are 8 bits or 16 bits wide, and both FIFOs are 64 locations deep x 16 bits wide supporting a total of 64 samples

- Both FIFOs can be loaded or emptied by using either programmed I/O (PIO) or DMA burst transfers

2.7.7 UART Interface

2.7.7.1 Introduction

The Universal Asynchronous Receiver/Transmitter (UART) interface is controlled via Direct-Memory Access (DMA) or programmed I/O.

2.7.7.2 Features

- Support of up to 10 UART interfaces
- Compatible with the 16550A and 16750 UART standards
- Support of adding and deleting standard asynchronous communication bits (start, stop and parity) in the serial data stream
- Independent control of transmission, reception, line status, data-set interrupts
- Modem control functions (CTSn and RTSn for both UART2 and UART3)
- Auto-flow capability for data I/O management without generating interrupts, where
  - RTSn (output) is controlled by the UART receive FIFO
  - CTSn (input) is from UART modem transmission controls
- Programmable serial interface with configurable options as follow:
  - 7-bit or 8-bit character length
  - Even, odd or no parity detection
  - 1 stop-bit generation
  - Baud rate generation up to 3.6Mbps for the 4 Fast UARTs

    - False start-bit detection
- 64-byte transmit FIFO
- 64-byte receive FIFO
- Support of complete status reporting
- Support of generating and detecting line breaks
- Support of internal diagnostics including:
    - Loopback control for fault isolation in communications link
    - Break, parity and framing error simulation
- Fully prioritized interrupt system
- Support of separated DMA requests for both transmit and receive data services
- Serial infrared asynchronous interface compliant with the Infrared Data Association (IrDA) specification

2.7.8 I2C Bus Interface

2.7.8.1 Introduction

The Inter-Integrated Circuit (I2C) bus is a true multi-master bus including collision detection and arbitration.

A dedicated I2C module, referred to as the power I2C module, is used to interface to the power management IC.

The I2C bus interface can function as both a master and a slave device on the I2C bus. This serial bus, developed by Philips Corporation, uses a 2-pin interface as follows:

- **SDA**: Data pin for input and output functions
- **SCL**: Clock pin for timing reference and control of the I2C bus

The I2C bus allows the I2C unit to interface with other I2C peripherals and microcontrollers. It requires minimal hardware, providing an economical solution for communicating status and control information between chips and external devices.

The I2C bus interface is a peripheral device residing on the peripheral bus that performs

- **Data transfer**, handled through a buffered interface for reliable communication
- **Control and status management**, accessed via memory-mapped registers

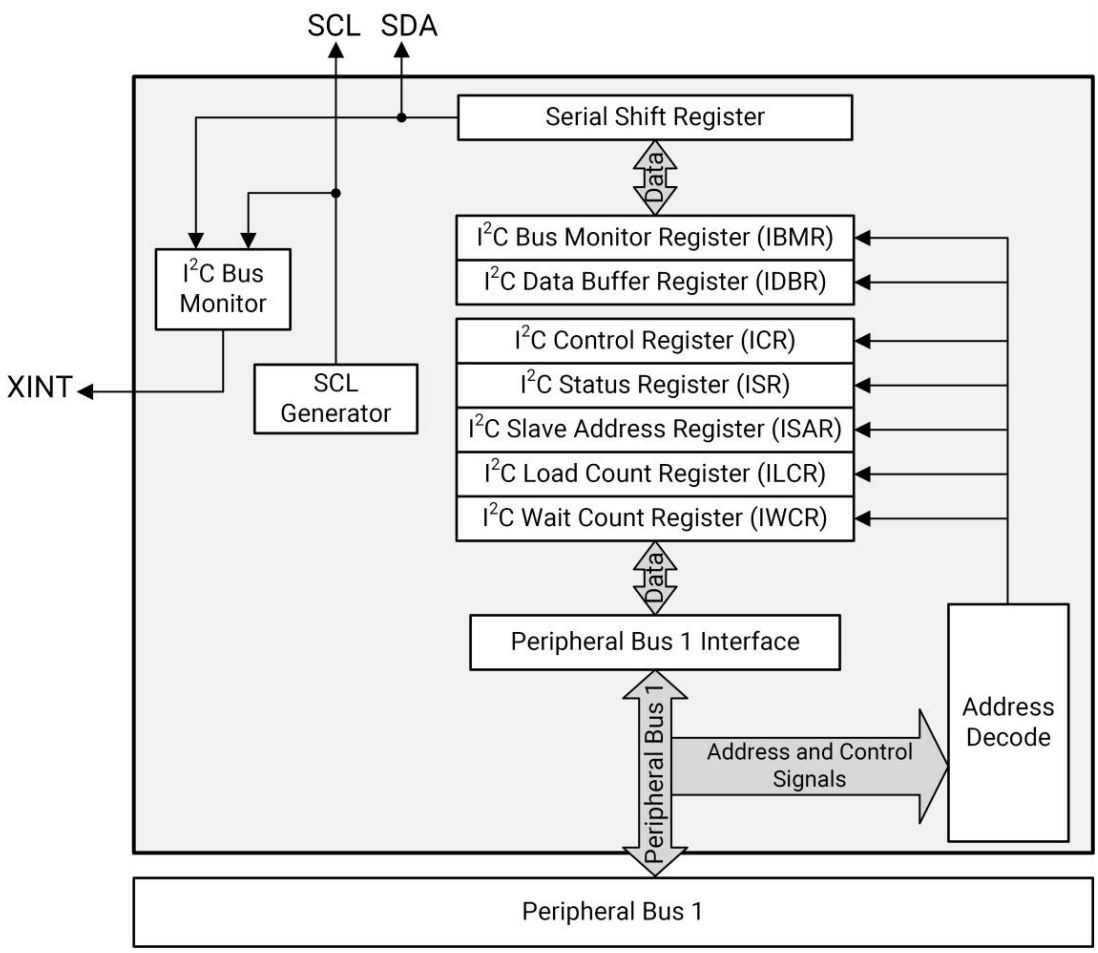
2.7.8.2 Features

- Compliance with I2C bus specification with the exception of the support for the hardware general call, 10-bit slave addressing and CBUS compatibility
- Support of Multi-Master and Arbitration
- **Operation modes and speeds** as follows:
  - Standard Operation Mode: up to 100 Kbps
  - Fast Operation Mode: up to 400 Kbps
  - High-Speed Slave Operation Mode: up to 3.4 Mbps (High-Speed I2C only)
  - High-Speed Master Operation Mode: up to 3.3 Mbps (High-Speed I2C only)

> **Note.** In High-Speed Master Operation Mode, I2C operational frequencies decrease due to the pull-up resistors on the bus. The SCL frequency is inversely proportional to the pull-up resistor value (1/R).

2.7.8.3 Block Diagram

The architecture of the I2C bus interface is depicted below.



2.7.9 IR-RX Interface

2.7.9.1 Features

- Infrared input signals are transformed into the Run-Length-Code (RLC) format
- Configurable signal width threshold for noise detection
- 32 Bytes FIFO for received data storage

2.7.10 One-Wire Bus Master Interface

2.7.10.1 Introduction

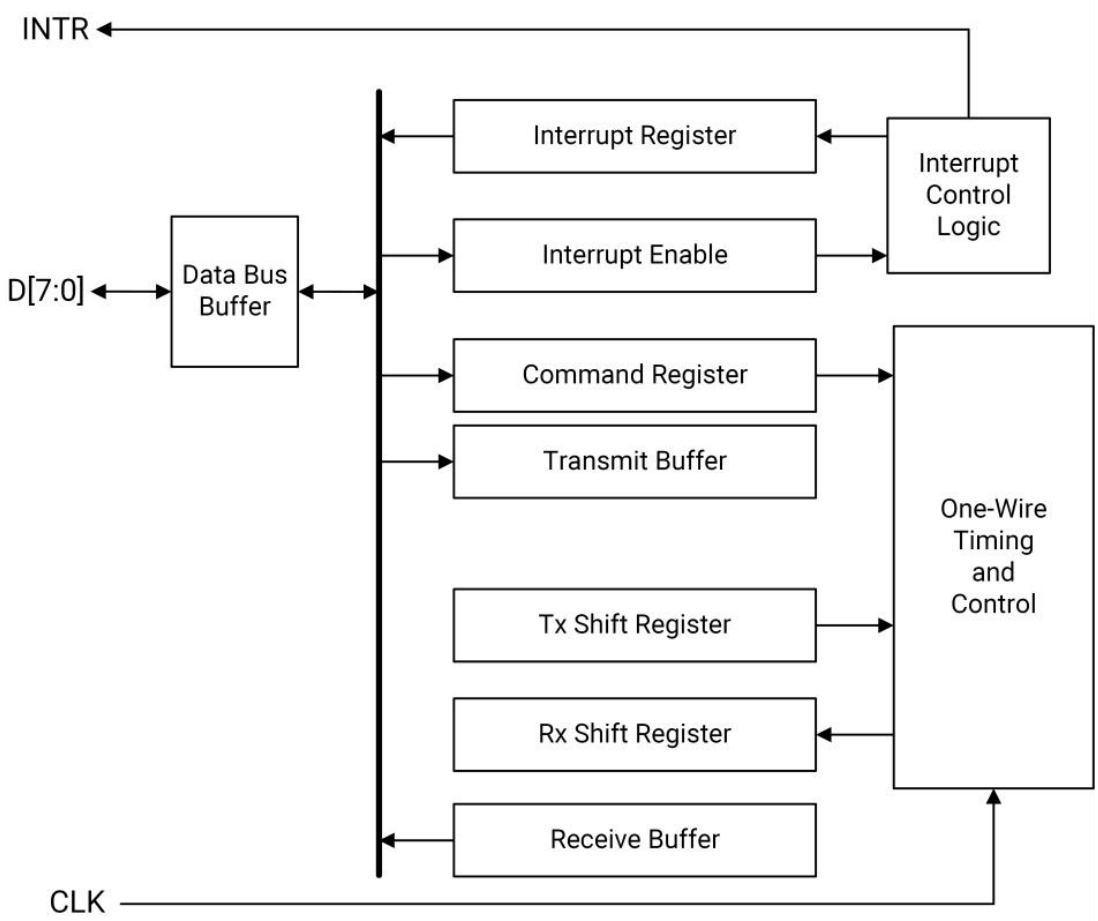
The One-Wire Bus Master Interface Controller is responsible for receiving and transmitting data on the One-Wire bus. It fully controls the One-Wire bus using 8-bit commands. The processor interacts with the controller by loading commands, reading and writing data, and configuring interrupt controls through 5 specific registers.

All One-Wire bus timing and control are generated within the One-Wire Bus Master Interface Controller once a command or data is loaded by the host. When there is activity on the bus that requires the CPU to respond, the One-Wire Bus Master Interface Controller sets a status bit and, if enabled, sends an interrupt to the CPU.

For detailed information about specific slave implementations, please refer to **Book of iButton® Standards** which describes the operation of the One-Wire bus master interface.

2.7.10.2 Block Diagram

The architecture of the One-Wire Bus Master Interface is depicted below.



2.7.11 I2S Interface

2.7.11.1 Introduction

The I2S interface is a synchronous serial interface designed to connect to various external devices, including Analog-to-Digital converters (ADC), audio and telecommunication codec. It directly supports the Inter-IC Sound (I2S) Protocol for data transfer.

2.7.11.2 Features

- Configurable to operate in either Master mode (where the attached peripheral functions as a slave) or Slave mode (where the attached peripheral functions as a master)
- Support of Receive-without-Transmit operation
- Support of serial bit rate from 6.3Kbps (min recommended) up to 52Mbps (max)
- Data sizes configurable to 8, 16, 18 or 32 bits in lenght
- Availability of a transmit FIFO (TXFIFO) and another independent receive FIFO (RXFIFO), where
    - In Non-Packed Data mode, both FIFOs are 32 rows deep x 32 bits wide supporting a total of 32 samples
    - In Packed Data mode, double-depth FIFOs are used when the data samples are 8 bits or 16 bits wide, and both FIFOs are 64 locations deep x 16 bits wide supporting a total of 64 samples
    - Both FIFOs can be loaded or emptied by using either programmed I/O (PIO) or DMA burst transfers
- Support of up to eight time slots with independent transmit/receive operation in any/all/none of the time slots
- Audio clock control provides a 4x or 8x output clock to support most standard audio frequencies

2.8 Security Subsystem

2.8.1 Encryption Engine

2.8.1.1 Features

- Support of symmetric encryption algorithms including AES
- Support of public key algorithms including RSA/ECC
- Support of HASH algorithms including SHA2

2.8.2 TRNG

2.8.2.1 Features

- Support of True Random Number Generator (TRNG) for security applications

2.8.3 eFuse

2.8.3.1 Features

- Support of total 4K eFuse bits organized into 16 banks
- User keys storage
- Anti-Rollback bits for secure firmware update
- Life Cycle Stage (LCS) bits for secure life cycle management
- Hardware lock for each eFuse bank

2.8.4 AES Engine

2.8.4.1 Features

- Dedicated high-performance AES Engine for massive data encryption/decryption

2.9 System Peripherals

2.9.1 DMA

2.9.1.1 Introduction

The Direct-Memory Access (DMA) controller is designed to transfer data between memory and peripheral devices without CPU intervention.

Peripheral devices do not directly supply addresses or commands to the memory controller. Each DMA request from a peripheral triggers a memory-bus transaction. The processor can directly access the peripheral bus by using the DMA controller which acts as a DMA bridge to bypass the DMA of the system

The DMA controller can manage different data transfer types in DMA Flow-Through Mode through 16 configurable DMA channels as tabled below.

	Internal Memory	**External Memory**	**Internal Peripheral**	**External Peripheral**
Internal Memory	Flow-Through Mode	___	___	___
External Memory	Flow-Through Mode	Flow-Through Mode	___	___

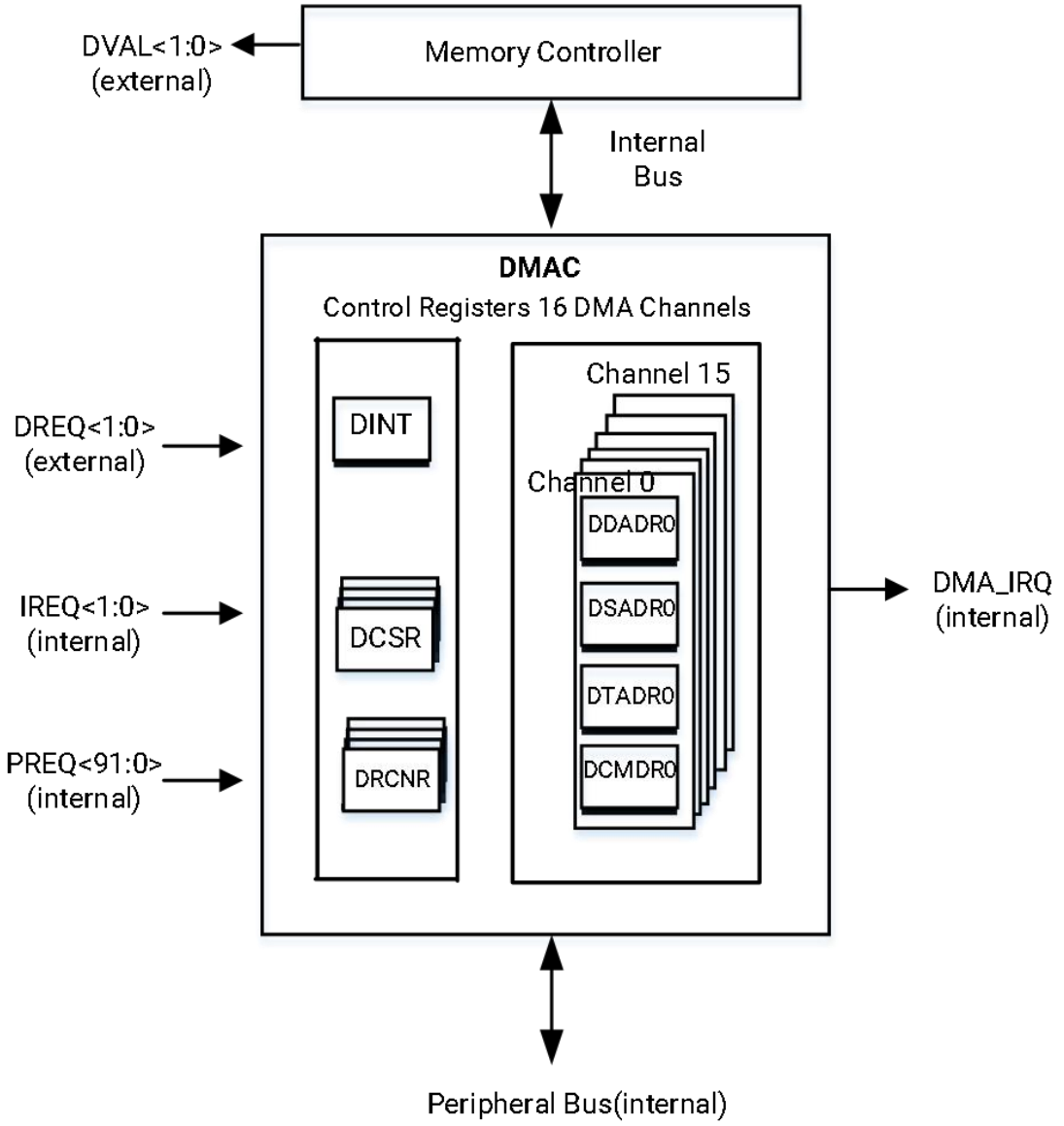
Internal Peripheral	Flow-Through Mode	Flow-Through Mode	___	___
External Peripheral	Flow-Through Mode	Flow-Through Mode	___	___

2.9.1.2 Features

- Handle of data transfers by two instances of the DMA controller, in particular
  - One for secure domains
  - One for non-secure domains
- Support of the following data transfer types in DMA Flow-Through Mode:
  - Memory-to-memory
  - Peripheral-to-memory
  - Memory-to-peripheral
- Support of DMA Flow-Through Mode for data transfers between Flash and DDR
- Implementation of a priority mechanism to process active channels at any time (up to 4 channels with outstanding DMA requests)
- Each of the 16 DMA channels is allow to operate for descriptor-fetch or non-descriptor-fetch transfers
- Support of the following special descriptor modes:
  - Descriptor Comparison
  - Descriptor branching
- Retrieve of the trailing bytes from the receive peripheral-device buffers
- Support of programmable data-burst sizes (8, 16, 32 or 64 bytes) and configurable peripheral device data widths (byte, half-word or word)
- Support of up to 8191 bytes of data transfer per descriptor (larger data transfers can be performed by chaining multiple descriptors)
- Support of a flow control bit to manage requests from peripheral device (requests are not processed unless a flow control bit is set)

2.9.1.3 Block Diagram

The architecture of the DMA controller is depicted below.



2.9.2 Timer

2.9.2.1 Introduction

X1 includes three general-purpose 32bit timers for system applications, and each one has its own 32bit Timer Counter Control Register (TCCRn) functioning as an up counter.

2.9.2.2 Features

- Programmable count mode as follows:
  - Fast count mode by input clock frequency of 12.8 MHz, 6.4 MHz, 3 MHz or 1 MHz
  - Slow count mode by input clock frequency of 32.768 KHz

2.9.3 WatchDog

2.9.3.1 Introduction

X1 includes one 16bit WatchDog Timer (WDT).

2.9.3.2 Features

- Programmable count mode as follows:
  - Fast count mode by input clock frequency of 12.8 MHz, 6.4 MHz, 3 MHz or 1 MHz)
  - Slow count mode by input clock frequency of 32.768 KHz

2.9.4 Temperature Sensor

2.9.4.1 Introduction

The Temperature Sensor Module (TSEN) provides temperature sensing and conversion functions, using a temperature-dependent voltage to time conversion method.

TSEN has an alarm function that triggers an interrupt when the temperature exceeds a specified warning threshold. It also includes a programmable self-repeating mode which performs temperature sensing operations automatically at intervals by a programmed delay.

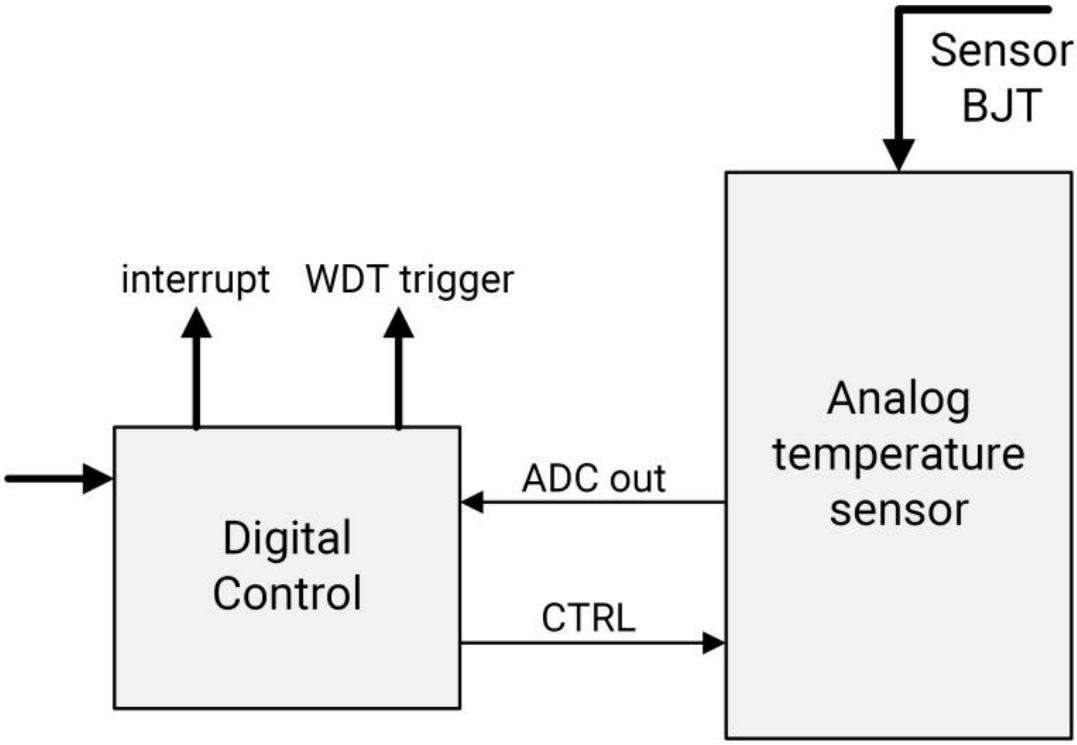
TSEN can be used by software to monitor the on-die temperature to let take all necessary actions, such as reducing the core frequency when a temperature interrupt is triggered.

2.9.4.2 Features

- Possibility to turn on/off TSEN (by software)
- Possibility to configure (by software) a high and low warning threshold of a BJT temperature for triggering related interrupts
- Record of the highest detected temperature of a BJT and its corresponding ID, and keeping track of the two most recent detected temperatures
- Possibility to enable (by software) the emergency system reset/reboot when a temperature violation occurs (the temperature sensor will trigger a system reset/reboot similar to the one performed by the Watchdog if the detected temperature exceeds the configured threshold)

2.9.4.3 Block Diagram

The architecture of the Temperature Sensor Module is depicted below.



2.9.5 PWM

2.9.5.1 Introduction

X1 contains 20 Pulse-Width Modulation (PWM) channels labeled as PWMx where x=[0,19].

Each PWM channel operates independently with its own configuration registers and generates an output PWM signal on a multi-function pin.

Each PWM channel allows controlling over both the leading-edge timing and the trailing-edge timing of its output signal.

The timing of each PWM channel can be set to run continuously or be adjusted dynamically to meet the change of requirements.

The power-saving mode allows stopping the internal clock of a PWM channel (PSCLK_PWM), resulting to a constant high or low state of the output signal of that PWM channel (PWM_OUT), thus saving power when the output signal of that PWM channel is not needed.

2.9.5.2 Features

- Support of 50% duty-cycle ranging from 198.4Hz to 6.5MHz (additional duty-cycle options depend on the choice of the preferred frequency)
- Enhanced period time controlled through 6-bit clock divider and 10-bit period time counter
- 15-bit pulse counter control

2.9.6 Mailbox

2.9.6.1 Introduction

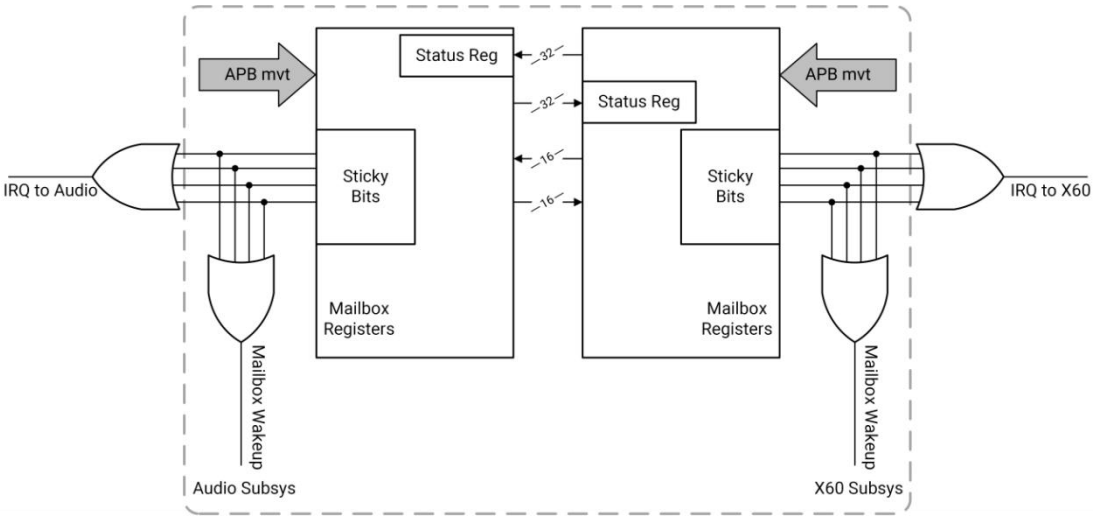
The Mailbox is designed to deliver messages or signals between SoC and MCU subsystem.

2.9.6.2 Features

- A processor is allow to generate an interrupt for another processor
- Support of a polling word to enable signaling an event from one party to another without the need of interrupts
- Reception of an ACK interrupt indicates that the other party is active
- A processor can wake up another processor (supported)

2.9.6.3 Block Diagram

The architecture of the Mailbox is depicted below.



2.9.7 GPIO

2.9.7.1 Introduction

X1 provides General-Purpose Input/Output (GPIO) ports for generating and capturing application-specific input and output. These ports are accessed through the alternate function muxing, and the GPIO unit manages their control and status.

2.9.7.2 Features

- A GPIO port configured as an input can also serve as an interrupt source
- At system reset, by default all GPIO ports are configured as an input until changed by the boot process or user software
- Each GPIO port has a dedicated control signal
- Support of separated interrupts over either leading-edge timing or trailing-edge timing or both
- The GPIO port output can be individually set or cleared
- The GPIO port input can be individually read

2.9.8 RTC

2.9.8.1 Features

- Count of the number of seconds basing on the internal 1-Hz clock
- Possibility to calibrate the frequency of the internal oscillator
- Support of an alarm interrupt and 1-Hz interrupt

2.9.9 Time-Out Monitor

2.9.9.1 Features

- Configurable time-out monitor threshold
- Configurable auto response function for time-out monitor events
- Save of the address and ID of the first time-out monitor transaction for debugging purposes
- Configurable check for AW/ARREADY signals

2.10 Sensor-Hub Subsystem

2.10.1 Features

- Support of 1 I2C interface
- Support of 1 SPP interface
- Support of 2 UART interfaces
- Support of 1 CAN interface

2.11 Clock & Reset

2.11.1 Introduction

X1 comes with the following clocks:

- One 32K RTC clock
- One 24M OSC clock

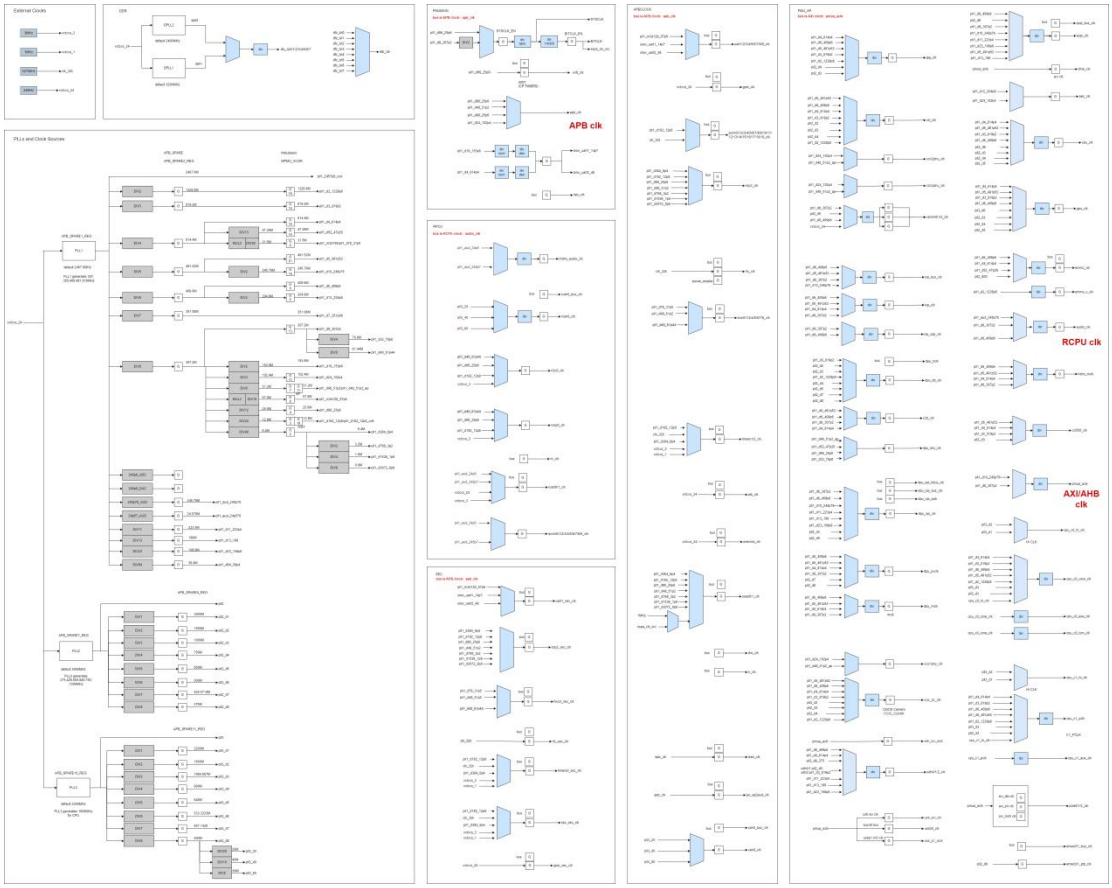
2.11.2 Features

- Three PLLs implemented inside to provide various frequencies to meet different scenario requirements
- DVFS feature supported to balance the tradeoff between power and performance
- Glitch-free clock switches and clock dividers implemented to provide all required frequencies with limited PLLs cost
- Clock gating and software reset schemes applied to modules in fine granularity to achieve power saving and flexible management

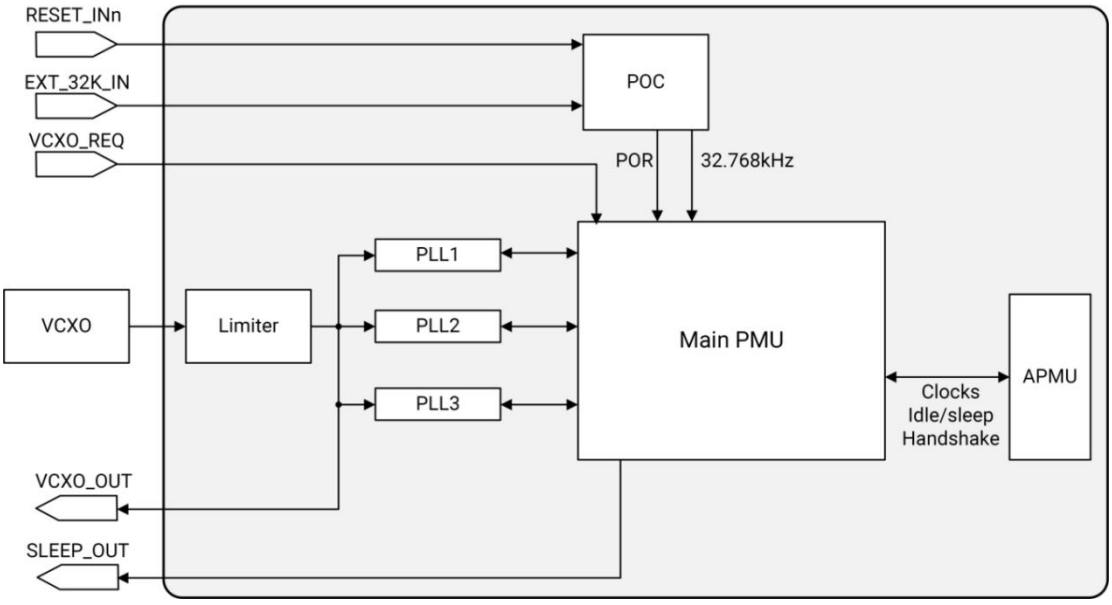
2.11.3 Block Diagram

2.11.3.1 Clock System

The detailed clock tree structure is depicted below, where is highlighted how the clock signals are generated, managed and distributed across the system to support various modules and functions.



Instead, the high-level architecture of the clock system is depicted below.



VCXO_OUT is driven with the OSC frequency if either of the following occurs:

- VCXO_REQ is asserted, and the relevant REQ_EN bit field is set in the VCXO software request control register
- Software request bit field is enabled in the VCXO software request control

register

There are three Phase-Locked Loop (PLL) designed to accept a wide range of input frequencies, and generate a broad range of output frequencies to all modules for functioning properly in different application scenario. Details for each PLL are provided in the following subsections.

2.11.3.1.1 PLL1

PLL1 is designed to generate fixed frequency points for the CPU cores and other peripherals, where

- Changes of the run-time frequency in the PLL1 output are only available for debugging purposes and should not be used in production systems
- PLL1 is enabled by default at system reset and shutdown only when the entire chip entered sleep mode with VCXO shutdown enabled
- The settings configured in the PLL1 and oscillator control registers of the Main PMU control the delay required for the PLL1 output clocks to stabilize after system reset or shutdown
- Updating the PLL1 configuration registers to change frequency during normal operations is not recommended

2.11.3.1.2 PLL2

PLL2 is designed to generate various fixed frequencies, working alongside PLL1 to provide a full range of frequencies required for different modules, where

- Changes of run-time frequency in the PLL2 output are only available for debugging purposes and should not be used in production systems
- PLL2 is disabled at system reset and must be enabled through software when required
- The settings configured in the PLL2 and oscillator control registers of the Main PMU control the delay required for the PLL2 output clocks to stabilize after system reset or shutdown
- Updating the PLL2 configuration registers to change frequency during normal operations is not recommended

2.11.3.1.3 PLL3

PLL3 is designed to provide frequencies for CPU frequency scaling and switching, where

- PLL3 is disabled at system reset and must be enabled through software when required
- The settings configured in the PLL3 and oscillator control register of the Main PMU control the delay required for the PLL3 output clocks to stabilize after system reset or shutdown
- Updating the PLL3 configuration registers to change frequency during normal operations is not recommended

2.11.3.2 Resource Reset Schemes

X1 allows applying different schemes of resource reset as tabled below.

No.	**Resource Reset Scheme**	**Description**
1	Power-On-Reset	Reset the whole chip during power-on sequence
2	WatchDog Reset	Reset the whole chip excluding pinmux registers and debug registers
3	Module Software Reset	Reset each module individually through software
4	Power Island POR Reset	Reset the whole power island during its power-on sequence

2.12 Boot Modes

2.12.1 Introduction

X1 supports booting from

- SPI NAND Flash
- SPI NOR Flash
- eMMC
- SD/TF Card

The details of the boot mode selection are tabled below.

No.	QSPI_DATA[1] / STRAP[1]	QSPI_DATA[0] / STRAP[0]	Boot Mode
1	Down	Down	SD/TF Card -> EMMC (default)
2	Up	Down	SD/TF Card -> SPI NAND Flash
3	Down	Up	SD/TF Card -> SPI NOR Flash
4	Up	Up	SD/TF Card

2.13 Power Management Unit

2.13.1 Introduction

A two-level power management strategy is implemented to control various granularities of power consumption. Different power domains and power states are also defined to achieve ultra-low power consumption.

A total of 9 power domains are implemented, and they are for

- CPU cores

> **Note.** Each CPU core has its own power domain independently controlled

- CPU clusters

> **Note.** Each CPU cluster has its own power domain independently controlled

- Video Encoder/Decoder
- GPU
- HDMI Display Subsystem
- MIPI DSI Subsystem
- Video Input Subsystem
- RCPU (including N308, Audio Codec, RCPU Peripherals)
- Always-On-Domain (AON)

All those power domains, except AON, can be powered off depending on specific application scenarios.

In order to achieve the minimal power consumption, different power states are designed as tabled below:

No.	Power State Name	Description
1	ACTIVE	The system is alive and active, with all power domains on, except those power domains with power switches that can be turned off selectively and independently.
2	CORE-IDLE	Each core stops executing instructions and enters an idle state, with clock gating automatically after a Wait-for-Interrupt (WFI) execution. The core exits this state when receiving an interrupt routed to it and continues execution.
3	Core-Power-Off	Each core, when voted, enters a power-off state after Core-Idle sleep mode. The core exits this state when receiving an interrupt, with power turned on and reset released.
4	CPU-Cluster-Power-Off	Each CPU cluster, when voted, enters this low-power state after all cores within this cluster have entered the Core-Power-Off state, with L2/TCM memory also shut

		down. Any active interrupt routing to CPU cores in this cluster would bring CPU cluster out of this state, then power on, clock resume and reset release.
5	Home-Screen	The main bus fabric AXI clock is gated off (if voted) after both CPU clusters enter CPU-Cluster-Power-Off mode. Any interrupt will wake up the chip from this state by resuming the main bus AXI clock, and powering up the corresponding CPU cluster and CPU core to which the interrupt is routed, resuming the CPU clock, and releasing the reset to service the interrupt routine.
6	Chip-Sleep	This is the most ultra-low power state, with all PLLs/Power islands off. Only 32K RTC clock remains alive, and the 24M VCXO can be configured to be on or off. In this state only the logic/IO in AON domain alives, and a pin named SLEEP_OUT connected to PMIC would be deasserted to signal PMIC to lower the VCC power supply voltage to reduce lower power comsumption.
7	RCPU with SOC LP	RCPU power domain is an independent power island and can function in any of above PMU states. RCPU can vote for different SoC low-power states according to its specific scenario requirements. The RCPU itself has four low-power states as follows: • Active Mode: Clock running • ClkGate Mode: Clock gating • PLL Off Mode: PLL powered off • Power Off Mode: RCPU power is shut down, but the RCPU AON domain remains alive

> **Note.** VPU, GPU, ISP, DPU power islands can be turned on or off by software, and are independent of the power states **No. 1~5** in the table above

In the **Chip-Sleep low power state** (see **No. 6** in the table above), the following interrupts or events can wake up the chip:

- Pad edge detection
- Keypad press
- RTC/Timer/WDT
- USB/RCPU/AP2AUDIO_IPC
- SD/EMMC/PCIE
- PMIC

In the **RCPU power off state** (see **No. 7** in the table above), the following interrupts or events can wake up RCPU PMU to resume its power supply:

- Audio plug interrupt / Hook key interrupt / Class-G short power interrupt / Audio OCP interrupt
- AP IPC power-on request
- RCPU AON Timer wakeup request
- Sensor-Hub GPIO wakeup request

3. Package

3.1 Introduction

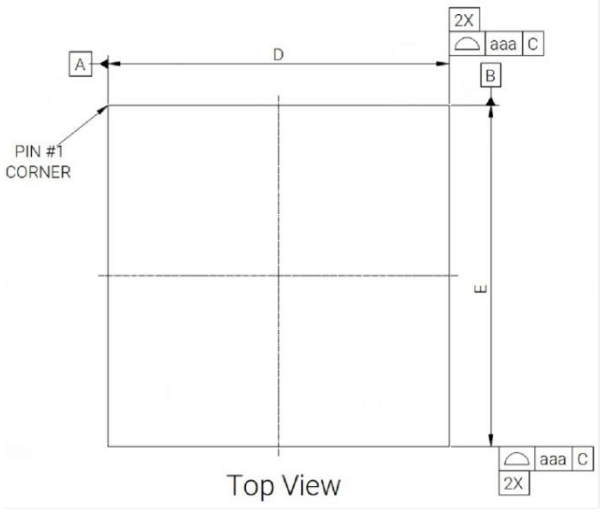
X1 is available in two packages as tabled below.

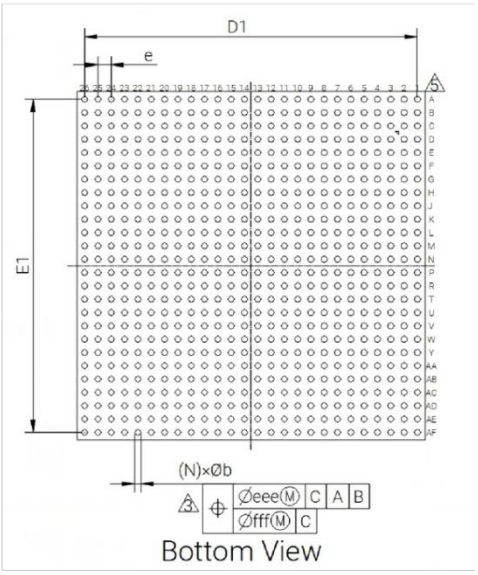
Type	Size	Pin Pitch	Pin Count
FCCSP	17×17 mm	0.65 mm	676 (26x26)

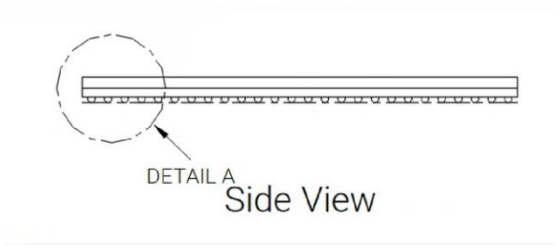
FCBGA	19×19 mm	0.65 mm	676 (26x26)

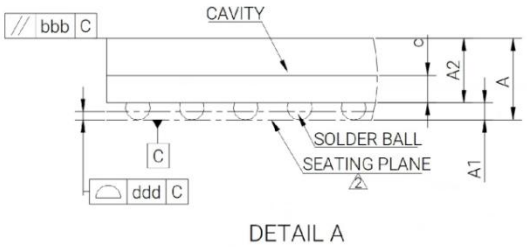
The related package outline drawing (POD) are depicted in the following sections.

3.2 FCCSP Type





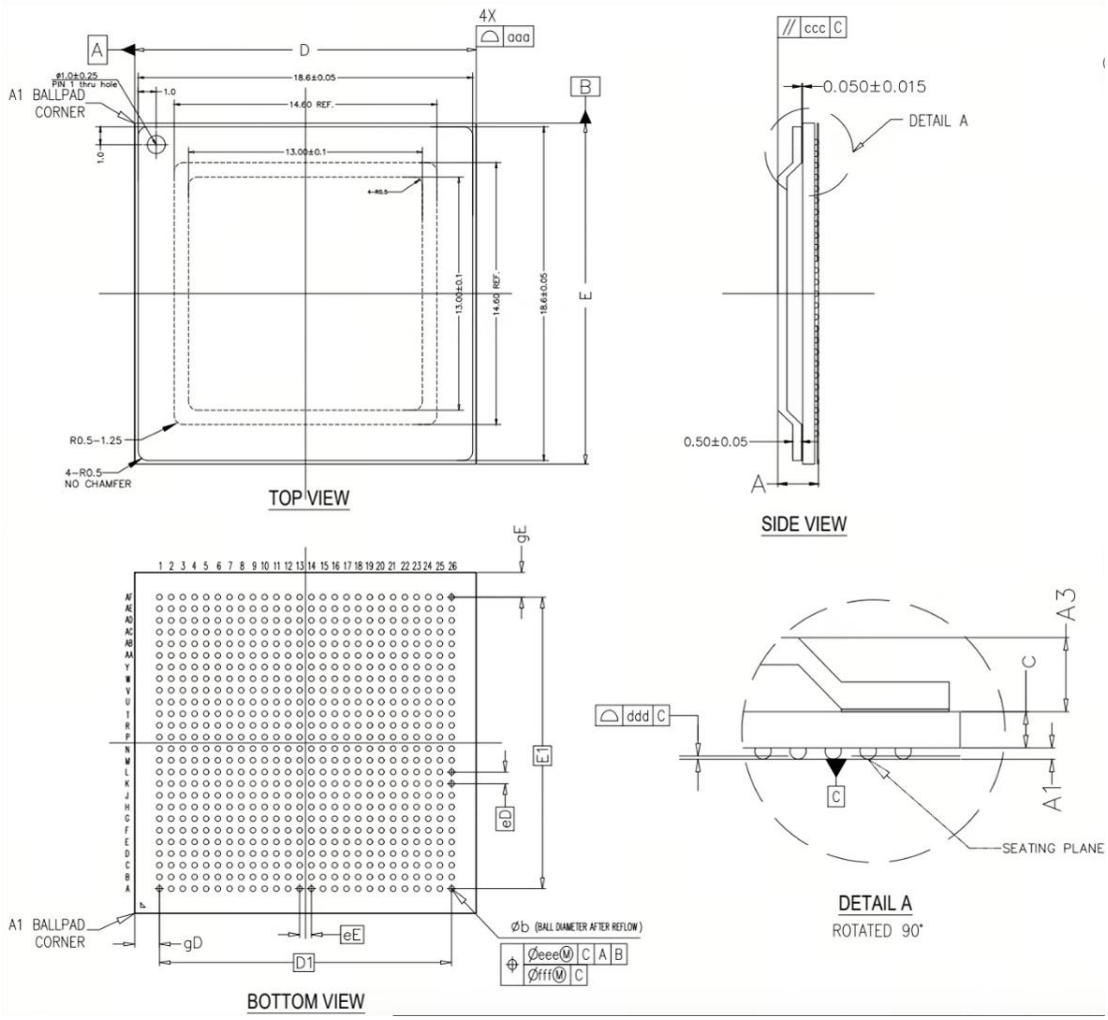




Item	Symbol	Dimension (in mm)		
		Min	Typ	Max
Total thickness	A	0.890	0.990	1.090
Pin stand off	A1	0.160	0.210	0.260
Substrate + Die + Mold	A2	0.710	0.780	0.850

Substrate + Die		c	0.290	0.330	0.370
Body size	X direction	D	16.900	17.000	17.100
	Y direction	E	16.900	17.000	17.100
Edge pin center to center	X direction	D1	—	16.250	—
	Y direction	E1	—	16.250	—
Pin pitch	X/Y direction	e	—	0.650	—
Pin width		b	0.250	0.300	0.350
Package edge tolerance		aaa	0.100		
HAT flatness		bbb	0.100		
Coplanarity		ddd	0.100		
Pin offset (package)		eee	0.150		
Pin offset (ball)		fff	0.080		
Pin diameter			0.300		
Pin count			676		
MD/ME			26/26		

3.3 FCBGA Type



Item		Symbol	Dimension (in mm)		
			Min	Typ	Max
Body size	X direction	D	18.900	19.000	19.100
	Y direction	E	18.900	19.000	19.100
Pin pitch	X direction	eD	0.650		
	Y direction	eE	0.650		

Total thickness		A	2.157	2.257	2.357
Hat + Adhesive		A3	1.322	1.375	1.428
Substrate thickness		c	0.602	0.672	0.742
Pin stand off		A1	0.169	0.210	0.260
Pin width		b	0.250	0.300	0.350
Package edge tolerance		aaa	0.150		
HAT flatness		ccc	0.350		
Coplanarity		ddd	0.080		
Pin offset (package)		eee	0.150		
Pin offset (ball)		fff	0.080		
Pin count		n	676		
Edge pin center to center	X direction	D1	16.250		
	Y direction	E1	16.250		
Edge pin center to package edge	X direction	gD	1.375		
	Y direction	gE	1.375		

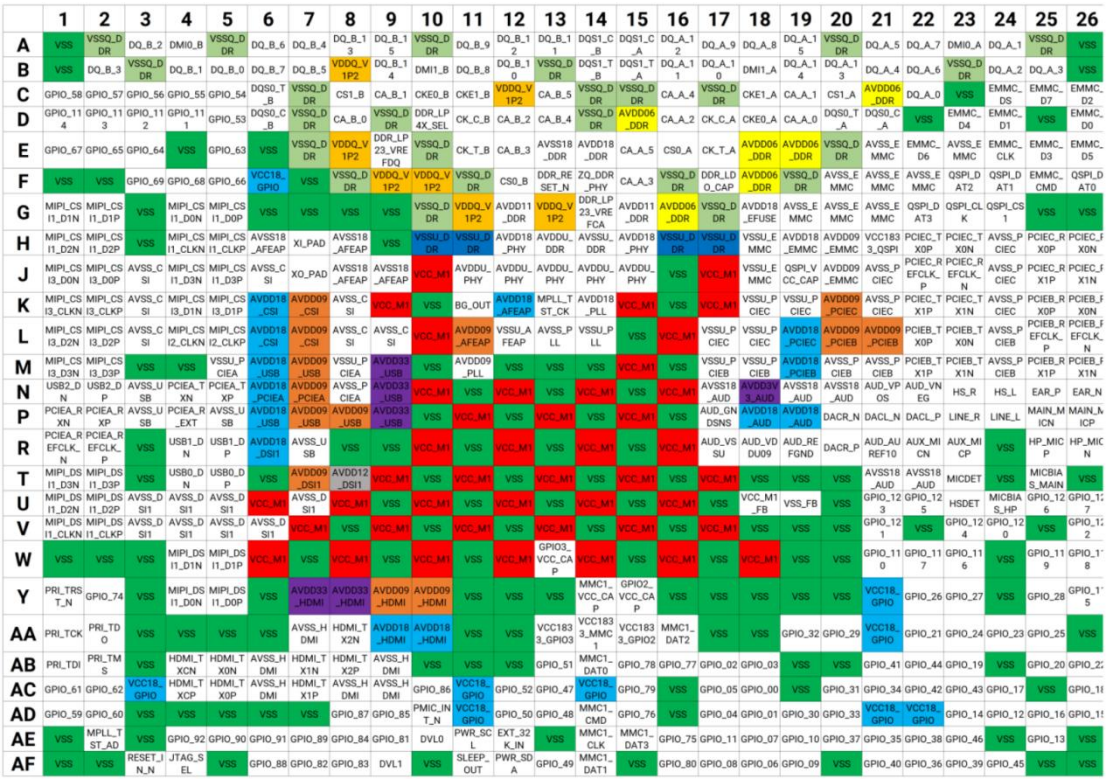
4. Pinout

4.1 Introduction

The two available packages of X1 as per **Chapter 3** are pin-to-pin.

4.2 Pinout Diagram & Description

The overall pinout diagram of X1 is depicted below.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	VSS	VSSQ_DDR	DQ_B_2	DMI0_B	VSSQ_DDR	DQ_B_6	DQ_B_4	DQ_B_13	DQ_B_15	VSSQ_DDR	DQ_B_9	DQ_B_12	DQ_B_11	DQS1_C_B	DQS1_C_A	DQ_A_12	DQ_A_9	DQ_A_8	DQ_A_15	VSSQ_DDR	DQ_A_5	DQ_A_7	DMI0_A	DQ_A_1	VSSQ_DDR	VSS
B	VSS	DQ_B_3	VSSQ_DDR	DQ_B_1	DQ_B_0	DQ_B_7	DQ_B_5	VDDQ_V1P2	DQ_B_14	DMI1_B	DQ_B_8	DQ_B_10	VSSQ_DDR	DQS1_T_B	DQS1_T_A	DQ_A_11	DQ_A_10	DMI1_A	DQ_A_14	DQ_A_13	DQ_A_4	DQ_A_6	VSSQ_DDR	DQ_A_2	DQ_A_3	VSS
C	GPIO_58	GPIO_57	GPIO_56	GPIO_55	GPIO_54	DQS0_T_B	VSSQ_DDR	CS1_B	CA_B_1	CKE0_B	CKE1_B	VDDQ_V1P2	CA_B_5	VSSQ_DDR	VSSQ_DDR	CA_A_4	VSSQ_DDR	CKE1_A	CA_A_1	CS1_A	AVDD06_DDR	DQ_A_0	VSS	EMMC_DS	EMMC_D7	EMMC_D2
D	GPIO_114	GPIO_113	GPIO_112	GPIO_111	GPIO_53	DQS0_C_B	VSSQ_DDR	CA_B_0	VSSQ_DDR	DDR_LP4X_SEL	CK_C_B	CA_B_2	CA_B_4	VSSQ_DDR	AVDD06_DDR	CA_A_2	CK_C_A	CKE0_A	CA_A_0	DQS0_T_A	DQS0_C_A	VSS	EMMC_D4	EMMC_D1	VSS	EMMC_D0
E	GPIO_67	GPIO_65	GPIO_64	VSS	GPIO_63	VSS	VSSQ_DDR	VDDQ_V1P2	DDR_LP23_VREFDQ	VSSQ_DDR	CK_T_B	CA_B_3	AVSS18_DDR	AVDD18_DDR	CA_A_5	CS0_A	CK_T_A	AVDD06_DDR	AVDD06_DDR	VSSQ_DDR	AVSS_EMMC	EMMC_D6	AVSS_EMMC	EMMC_CLK	EMMC_D3	EMMC_D5
F	VSS	VSS	GPIO_69	GPIO_68	GPIO_66	VCC18_GPIO	VSS	VSSQ_DDR	VDDQ_V1P2	VDDQ_V1P2	VSSQ_DDR	CS0_B	DDR_RESET_N	ZQ_DDR_PHY	CA_A_3	VSSQ_DDR	DDR_LDO_CAP	AVDD06_DDR	VSSQ_DDR	AVSS_EMMC	AVSS_EMMC	AVSS_EMMC	QSPI_DAT2	QSPI_DAT1	EMMC_CMD	QSPI_DAT0
G	MIPI_CSI1_D1N	MIPI_CSI1_D1P	VSS	MIPI_CSI1_D0N	MIPI_CSI1_D0P	VSS	VSS	VSS	VSS	VSSQ_DDR	VDDQ_V1P2	AVDD11_DDR	VDDQ_V1P2	DDR_LP23_VREFCA	AVDD11_DDR	AVDD06_DDR	VSSQ_DDR	AVDD18_EFUSE	AVSS_EMMC	AVSS_EMMC	AVSS_EMMC	QSPI_DAT3	QSPI_CLK	QSPI_CS1	VSS	VSS
H	MIPI_CSI1_D2N	MIPI_CSI1_D2P	VSS	MIPI_CSI1_CLKN	MIPI_CSI1_CLKP	AVSS18_AFEAP	XI_PAD	AVSS18_AFEAP	VSS	VSSU_DDR	VSSU_DDR	AVDD18_PHY	AVDDU_DDR	AVSSU_DDR	AVDD18_PHY	VSSU_DDR	VSSU_DDR	VSSU_EMMC	AVDD18_EMMC	AVDD09_EMMC	VCC1833_QSPI	PCIEC_TX0P	PCIEC_TX0N	AVSS_PCIEC	PCIEC_RX0P	PCIEC_RX0N
J	MIPI_CSI3_D0N	MIPI_CSI3_D0P	AVSS_CSI	MIPI_CSI1_D3N	MIPI_CSI1_D3P	AVSS_CSI	XO_PAD	AVSS18_AFEAP	AVSS18_AFEAP	VCC_M1	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	VSS	VCC_M1	VSSU_EMMC	QSPI_VCC_CAP	AVDD09_EMMC	AVSS_PCIEC	PCIEC_REFCLK_P	PCIEC_REFCLK_N	AVSS_PCIEC	PCIEC_RX1P	PCIEC_RX1N
K	MIPI_CSI3_CLKN	MIPI_CSI3_CLKP	AVSS_CSI	MIPI_CSI3_D1N	MIPI_CSI3_D1P	AVDD18_CSI	AVDD09_CSI	AVSS_CSI	VCC_M1	VSS	BG_OUT	AVDD18_AFEAP	MPLL_TST_CK	AVDD18_PLL	VCC_M1	VSS	VCC_M1	VSSU_PCIEC	VSSU_PCIEC	AVDD09_PCIEC	AVSS_PCIEC	PCIEC_TX1P	PCIEC_TX1N	AVSS_PCIEC	PCIEB_RX0P	PCIEB_RX0N
L	MIPI_CSI3_D2N	MIPI_CSI3_D2P	AVSS_CSI	MIPI_CSI2_CLKN	MIPI_CSI2_CLKP	AVDD18_CSI	AVDD09_CSI	AVSS_CSI	AVSS_CSI	VCC_M1	AVDD09_AFEAP	VSSU_AFEAP	AVSS_PLL	VSSU_PLL	VSS	VCC_M1	VSSU_PCIEC	VSSU_PCIEC	AVDD18_PCIEC	AVDD09_PCIEB	AVDD09_PCIEB	PCIEB_TX0P	PCIEB_TX0N	AVSS_PCIEB	PCIEB_REFCLK_P	PCIEB_REFCLK_N
M	MIPI_CSI3_D3N	MIPI_CSI3_D3P	VSS	VSS	VSSU_PCIEA	AVDD18_USB	AVDD09_USB	VSSU_PCIEA	AVDD33_USB	VSS	AVDD09_PLL	VSS	VSS	VSS	VCC_M1	VSS	VSSU_PCIEB	VSSU_PCIEB	AVDD18_PCIEB	AVSS_PCIEB	AVSS_PCIEB	PCIEB_TX1P	PCIEB_TX1N	AVSS_PCIEB	PCIEB_RX1P	PCIEB_RX1N
N	USB2_DN	USB2_DP	AVSS_USB	PCIEA_TXN	PCIEA_TXP	AVDD18_PCIEA	AVDD09_PCIEA	AVSS_PCIEA	AVDD33_USB	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	AVSS18_AUD	AVDD3V3_AUD	AVSS18_AUD	AVSS18_AUD	AUD_VPOS	AUD_VNEG	HS_R	HS_L	EAR_P	EAR_N
P	PCIEA_RXN	PCIEA_RXP	AVSS_USB	PCIEA_REXT	AVSS_USB	AVDD18_USB	AVDD09_USB	AVDD09_USB	AVDD33_USB	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	AUD_GNDSNS	AVDD18_AUD	AVDD18_AUD	DACR_N	DACL_N	DACL_P	LINE_R	LINE_L	MAIN_MICN	MAIN_MICP
R	PCIEA_REFCLK_N	PCIEA_REFCLK_P	VSS	USB1_DN	USB1_DP	AVDD18_DSI1	AVSS_USB	VSS	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	AUD_VSSU	AUD_VDDU09	AUD_REFGND	DACR_P	AUD_AUREF10	AUX_MICN	AUX_MICP	VSS	HP_MICP	HP_MICN
T	MIPI_DSI1_D3N	MIPI_DSI1_D3P	VSS	USB0_DN	USB0_DP	VSS	AVDD09_DSI1	AVDD12_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	VSS	AVSS18_AUD	AVSS18_AUD	MICDET	VSS	MICBIAS_MAIN	VSS
U	MIPI_DSI1_D2N	MIPI_DSI1_D2P	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	VCC_M1	AVSS_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1_FB	VSS_FB	VSS	GPIO_123	GPIO_125	HSDET	MICBIAS_HPS	GPIO_126	GPIO_127
V	MIPI_DSI1_CLKN	MIPI_DSI1_CLKP	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	VSS	GPIO_121	VSS	GPIO_124	GPIO_120	VSS	GPIO_122
W	VSS	VSS	VSS	MIPI_DSI1_D1N	MIPI_DSI1_D1P	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	GPIO3_VCC_CAP	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	GPIO_110	GPIO_117	GPIO_116	VSS	GPIO_119	GPIO_118
Y	PRI_TRST_N	GPIO_74	VSS	MIPI_DSI1_D0N	MIPI_DSI1_D0P	VSS	AVDD33_HDMI	AVDD33_HDMI	AVDD09_HDMI	AVDD09_HDMI	VSS	VSS	VSS	MMC1_VCC_CAP	GPIO2_VCC_CAP	VSS	VSS	VSS	VSS	VSS	VCC18_GPIO	GPIO_26	GPIO_27	VSS	GPIO_28	GPIO_15
AA	PRI_TCK	PRI_TDO	VSS	VSS	VSS	VSS	AVSS_HDMI	HDMI_TX2N	AVDD18_HDMI	AVDD18_HDMI	VSS	VSS	VCC1833_GPIO3	VCC1833_MMC1	VCC1833_GPIO2	MMC1_DAT2	VSS	VSS	GPIO_32	GPIO_29	VCC18_GPIO	GPIO_21	GPIO_24	GPIO_23	GPIO_25	VSS
AB	PRI_TDI	PRI_TMS	VSS	HDMI_TXCN	HDMI_TX0N	AVSS_HDMI	HDMI_TX1N	HDMI_TX2P	AVSS_HDMI	VSS	VSS	VSS	GPIO_51	MMC1_DAT0	GPIO_78	GPIO_77	GPIO_02	GPIO_03	VSS	VSS	GPIO_41	GPIO_44	GPIO_19	VSS	GPIO_20	GPIO_22
AC	GPIO_61	GPIO_62	VCC18_GPIO	HDMI_TXCP	HDMI_TX0P	AVSS_HDMI	HDMI_TX1P	AVSS_HDMI	AVSS_HDMI	GPIO_86	VCC18_GPIO	GPIO_52	GPIO_47	VCC18_GPIO	GPIO_79	VSS	GPIO_05	GPIO_00	VSS	GPIO_31	GPIO_34	GPIO_42	GPIO_43	GPIO_17	VSS	GPIO_18
AD	GPIO_59	GPIO_60	VSS	VSS	VSS	VSS	VSS	GPIO_87	GPIO_85	PMIC_INT_N	VCC18_GPIO	GPIO_50	GPIO_48	MMC1_CMD	GPIO_76	VSS	GPIO_04	GPIO_01	GPIO_30	GPIO_33	VCC18_GPIO	VCC18_GPIO	GPIO_14	GPIO_12	GPIO_16	GPIO_15
AE	VSS	MPLL_TST_AD	VSS	GPIO_92	GPIO_90	GPIO_91	GPIO_89	GPIO_84	GPIO_81	DVL0	PWR_SCL	EXT_32K_IN	VSS	MMC1_CLK	MMC1_DAT3	GPIO_75	GPIO_11	GPIO_07	GPIO_10	GPIO_37	GPIO_35	GPIO_38	GPIO_46	VSS	GPIO_13	VSS
AF	VSS	VSS	RESET_IN_N	JTAG_SEL	VSS	GPIO_88	GPIO_82	GPIO_83	DVL1	VSS	SLEEP_OUT	PWR_SDA	GPIO_49	MMC1_DAT1	VSS	GPIO_80	GPIO_08	GPIO_06	GPIO_09	VSS	GPIO_40	GPIO_36	GPIO_39	GPIO_45	VSS	VSS

Note. Meaning of the different colors:

- Power supplies (different voltages):
  - Brown
  - Dark Blue
  - Grey
  - Light Blue
  - Orange
  - Purple
  - Red
  - Yellow

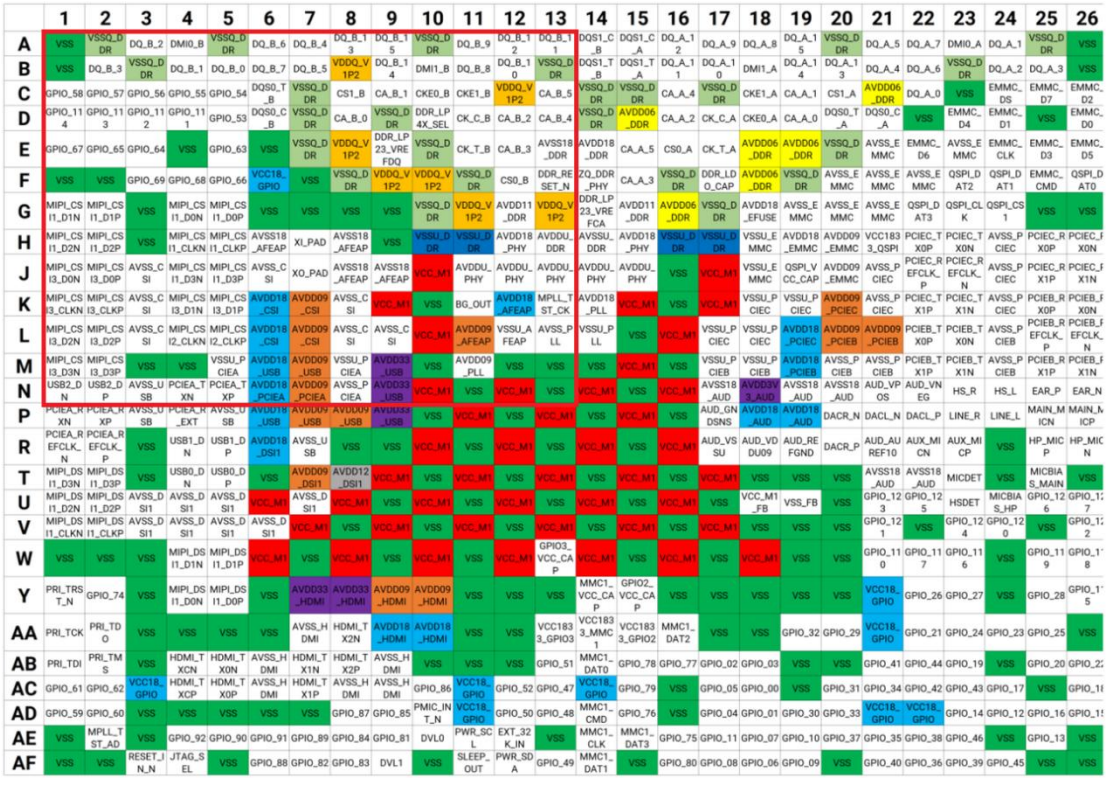
- Grounds:
  - Dark Green
  - Light Green
- Signals:
  - White

Let's consider the division into the quadrants

- (A~N, 1~13)
- (A~N, 14~26)
- (M~AF, 1~13)
- (M~AF, 14~26)

in order to provide conveniently the pinout description of X1 in the following subsections.

4.2.1 (A~N, 1~13)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	VSS	VSSQ_DDR	DQ_B_2	DMI0_B	VSSQ_DDR	DQ_B_6	DQ_B_4	DQ_B_13	DQ_B_15	VSSQ_DDR	DQ_B_9	DQ_B_12	DQ_B_11	DQS1_C_B	DQS1_C_A	DQ_A_12	DQ_A_9	DQ_A_8	DQ_A_15	VSSQ_DDR	DQ_A_5	DQ_A_7	DMI0_A	DQ_A_1	VSSQ_DDR	VSS
B	VSS	DQ_B_3	VSSQ_DDR	DQ_B_1	DQ_B_0	DQ_B_7	DQ_B_5	VDDQ_V1P2	DQ_B_14	DMI1_B	DQ_B_8	DQ_B_10	VSSQ_DDR	DQS1_T_B	DQS1_T_A	DQ_A_11	DQ_A_10	DMI1_A	DQ_A_14	DQ_A_13	DQ_A_4	DQ_A_6	VSSQ_DDR	DQ_A_2	DQ_A_3	VSS
C	GPIO_58	GPIO_57	GPIO_56	GPIO_55	GPIO_54	DQS0_T_B	VSSQ_DDR	CS1_B	CA_B_1	CKE0_B	CKE1_B	VDDQ_V1P2	CA_B_5	VSSQ_DDR	VSSQ_DDR	CA_A_4	VSSQ_DDR	CKE1_A	CA_A_1	CS1_A	AVDD06_DDR	DQ_A_0	VSS	EMMC_DS	EMMC_D7	EMMC_D2
D	GPIO_114	GPIO_113	GPIO_112	GPIO_111	GPIO_53	DQS0_C_B	VSSQ_DDR	CA_B_0	VSSQ_DDR	DDR_LP4X_SEL	CK_C_B	CA_B_2	CA_B_4	VSSQ_DDR	AVDD06_DDR	CA_A_2	CK_C_A	CKE0_A	CA_A_0	DQS0_T_A	DQS0_C_A	VSS	EMMC_D4	EMMC_D1	VSS	EMMC_D0
E	GPIO_67	GPIO_65	GPIO_64	VSS	GPIO_63	VSS	VSSQ_DDR	VDDQ_V1P2	DDR_LP23_VREFDQ	VSSQ_DDR	CK_T_B	CA_B_3	AVSS18_DDR	AVDD18_DDR	CA_A_5	CS0_A	CK_T_A	AVDD06_DDR	AVDD06_DDR	VSSQ_DDR	AVSS_EMMC	EMMC_D6	AVSS_EMMC	EMMC_CLK	EMMC_D3	EMMC_D5
F	VSS	VSS	GPIO_69	GPIO_68	GPIO_66	VCC18_GPIO	VSS	VSSQ_DDR	VDDQ_V1P2	VDDQ_V1P2	VSSQ_DDR	CS0_B	DDR_RESET_N	ZQ_DDR_PHY	CA_A_3	VSSQ_DDR	DDR_LDO_CAP	AVDD06_DDR	VSSQ_DDR	AVSS_EMMC	AVSS_EMMC	AVSS_EMMC	QSPI_DAT2	QSPI_DAT1	EMMC_CMD	QSPI_DAT0
G	MIPI_CSI1_D1N	MIPI_CSI1_D1P	VSS	MIPI_CSI1_D0N	MIPI_CSI1_D0P	VSS	VSS	VSS	VSS	VSSQ_DDR	VDDQ_V1P2	AVDD11_DDR	VDDQ_V1P2	DDR_LP23_VREFCA	AVDD11_DDR	AVDD06_DDR	VSSQ_DDR	AVDD18_EFUSE	AVSS_EMMC	AVSS_EMMC	AVSS_EMMC	QSPI_DAT3	QSPI_CLK	QSPI_CS1	VSS	VSS
H	MIPI_CSI1_D2N	MIPI_CSI1_D2P	VSS	MIPI_CSI1_CLKN	MIPI_CSI1_CLKP	AVSS18_AFEAP	XI_PAD	AVSS18_AFEAP	VSS	VSSU_DDR	VSSU_DDR	AVDD18_PHY	AVDDU_DDR	AVSSU_DDR	AVDD18_PHY	VSSU_DDR	VSSU_DDR	VSSU_EMMC	AVDD18_EMMC	AVDD09_EMMC	VCC1833_QSPI	PCIEC_TX0P	PCIEC_TX0N	AVSS_PCIEC	PCIEC_RX0P	PCIEC_RX0N
J	MIPI_CSI3_D0N	MIPI_CSI3_D0P	AVSS_CSI	MIPI_CSI1_D3N	MIPI_CSI1_D3P	AVSS_CSI	XO_PAD	AVSS18_AFEAP	AVSS18_AFEAP	VCC_M1	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	VSS	VCC_M1	VSSU_EMMC	QSPI_VCC_CAP	AVDD09_EMMC	AVSS_PCIEC	PCIEC_RFECLK_P	PCIEC_RFECLK_N	AVSS_PCIEC	PCIEC_RX1P	PCIEC_RX1N
K	MIPI_CSI3_CLKN	MIPI_CSI3_CLKP	AVSS_CSI	MIPI_CSI3_D1N	MIPI_CSI3_D1P	AVDD18_CSI	AVDD09_CSI	AVSS_CSI	VCC_M1	VSS	BG_OUT	AVDD18_AFEAP	MPLL_TST_CK	AVDD18_PLL	VCC_M1	VSS	VCC_M1	VSSU_PCIEC	VSSU_PCIEC	AVDD09_PCIEC	AVSS_PCIEC	PCIEC_TX1P	PCIEC_TX1N	AVSS_PCIEC	PCIEB_RX0P	PCIEB_RX0N
L	MIPI_CSI3_D2N	MIPI_CSI3_D2P	AVSS_CSI	MIPI_CSI2_CLKN	MIPI_CSI2_CLKP	AVDD18_CSI	AVDD09_CSI	AVSS_CSI	AVSS_CSI	VCC_M1	AVDD09_AFEAP	VSSU_AFEAP	AVSS_PLL	VSSU_PLL	VSS	VCC_M1	VSSU_PCIEC	VSSU_PCIEC	AVDD18_PCIEC	AVDD09_PCIEB	AVDD09_PCIEB	PCIEB_TX0P	PCIEB_TX0N	AVSS_PCIEB	PCIEB_RFECLK_P	PCIEB_RFECLK_N
M	MIPI_CSI3_D3N	MIPI_CSI3_D3P	VSS	VSS	VSSU_PCIEA	AVDD18_USB	AVDD09_USB	VSSU_PCIEA	AVDD33_USB	VBS	AVDD09_PLL	VSS	VSS	VSS	VCC_M1	VSS	VSSU_PCIEB	VSSU_PCIEB	AVDD18_PCIEB	AVSS_PCIEB	AVSS_PCIEB	PCIEB_TX1P	PCIEB_TX1N	AVSS_PCIEB	PCIEB_RX1P	PCIEB_RX1N
N	USB2_DN	USB2_DP	AVSS_USB	PCIEA_TXN	PCIEA_TXP	AVDD18_PCIEA	AVDD09_PCIEA	AVSS_PCIEA	AVDD33_USB	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	AVSS18_AUD	AVDD33_AUD	AVSS18_AUD	AVSS18_AUD	AUD_VPOS	AUD_VNEG	HS_R	HS_L	EAR_P	EAR_N
P	PCIEA_RXN	PCIEA_RXP	AVSS_USB	PCIEA_REXT	AVSS_USB	AVDD18_USB	AVDD09_USB	AVDD09_USB	AVDD33_USB	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	AUD_GNDSNS	AVDD18_AUD	AVDD18_AUD	DACR_N	DACL_N	DACL_P	LINE_R	LINE_L	MAIN_MICN	MAIN_NICP
R	PCIEA_REFCLK_N	PCIEA_REFCLK_P	VSS	USB1_DN	USB1_DP	AVDD18_DSI1	AVSS_USB	VSS	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	AUD_VSSU	AUD_VDDU09	AUD_REFGND	DACR_P	AUD_AUREF10	AUX_MICN	AUX_MICP	VSS	HP_MICP	HP_MICN
T	MIPI_DSI1_D3N	MIPI_DSI1_D3P	VSS	USB0_DN	USB0_DP	VSS	AVDD09_DSI1	AVDD12_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	VSS	AVSS18_AUD	AVSS18_AUD	MICDET	VSS	MICBIAS_MAIN	VSS
U	MIPI_DSI1_D2N	MIPI_DSI1_D2P	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	VCC_M1	AVSS_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1_FB	VSS_FB	VSS	GPIO_123	GPIO_125	HSDET	MICBIAS_HP	GPIO_126	GPIO_127
V	MIPI_DSI1_CLKN	MIPI_DSI1_CLKP	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	VSS	GPIO_121	VSS	GPIO_124	GPIO_120	VSS	GPIO_12
W	VSS	VSS	VSS	MIPI_DSI1_D1N	MIPI_DSI1_D1P	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	GPIO3_VCC_CAP	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	GPIO_110	GPIO_117	GPIO_116	VSS	GPIO_119	GPIO_118
Y	PRI_TRST_N	GPIO_74	VSS	MIPI_DSI1_D0N	MIPI_DSI1_D0P	VSS	AVDD33_HDMI	AVDD33_HDMI	AVDD09_HDMI	AVDD09_HDMI	VSS	VSS	VSS	MMC1_VCC_CAP	GPIO2_VCC_CAP	VSS	VSS	VSS	VSS	VSS	VCC18_GPIO	GPIO_26	GPIO_27	VSS	GPIO_28	GPIO_15
AA	PRI_TCK	PRI_TDO	VSS	VSS	VSS	VSS	AVSS_HDMI	HDMI_TX2N	AVDD18_HDMI	AVDD18_HDMI	VSS	VSS	VCC1833_GPIO3	VCC1833_MMC1	VCC1833_GPIO2	MMC1_DAT2	VSS	VSS	GPIO_32	GPIO_29	VCC18_GPIO	GPIO_21	GPIO_24	GPIO_23	GPIO_25	VSS
AB	PRI_TDI	PRI_TMS	VSS	HDMI_TXCN	HDMI_TX0N	AVSS_HDMI	HDMI_TX1N	HDMI_TX2P	AVSS_HDMI	VSS	VSS	VSS	GPIO_51	MMC1_DAT0	GPIO_78	GPIO_77	GPIO_02	GPIO_03	VSS	VSS	GPIO_41	GPIO_44	GPIO_19	VSS	GPIO_20	GPIO_22
AC	GPIO_61	GPIO_62	VCC18_GPIO	HDMI_TXCP	HDMI_TX0P	AVSS_HDMI	HDMI_TX1P	AVSS_HDMI	AVSS_HDMI	GPIO_86	VCC18_GPIO	GPIO_52	GPIO_47	VCC18_GPIO	GPIO_79	VSS	GPIO_05	GPIO_00	VSS	GPIO_31	GPIO_34	GPIO_42	GPIO_43	GPIO_17	VSS	GPIO_18
AD	GPIO_59	GPIO_60	VSS	VSS	VSS	VSS	VSS	GPIO_87	GPIO_85	PMIC_INT_N	VCC18_GPIO	GPIO_50	GPIO_48	MMC1_CMD	GPIO_76	VSS	GPIO_04	GPIO_01	GPIO_30	GPIO_33	VCC18_GPIO	VCC18_GPIO	GPIO_14	GPIO_12	GPIO_16	GPIO_15
AE	VSS	MPLL_TST_AD	VSS	GPIO_92	GPIO_90	GPIO_91	GPIO_89	GPIO_84	GPIO_81	DVL0	PWR_SCL	EXT_32K_IN	VSS	MMC1_CLK	MMC1_DAT3	GPIO_75	GPIO_11	GPIO_07	GPIO_10	GPIO_37	GPIO_35	GPIO_38	GPIO_46	VSS	GPIO_13	VSS
AF	VSS	VSS	RESET_IN_N	JTAG_SEL	VSS	GPIO_88	GPIO_82	GPIO_83	DVL1	VSS	SLEEP_OUT	PWR_SDA	GPIO_49	MMC1_DAT1	VSS	GPIO_80	GPIO_08	GPIO_06	GPIO_09	VSS	GPIO_40	GPIO_36	GPIO_39	GPIO_45	VSS	VSS

> **Note.** Definition of symbols used for pin type:
>
> - AO = Analog output
> - AI = Analog input
> - AIO = Analog input/output
> - G = Ground
> - I/O = Input/Output
> - P = Power
> - RO = Reference output

Pin ID	Name	Type	Power Domain	Function
A1	VSS	G	0V	Digital Core Ground
A2	VSSQ_DDR	G	0V	DDR Ground
A3	DQ_B_2	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ2 LPDDR3: DQ28
A4	DMI0_B	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Channel B DM0 LPDDR3: DQ25
A5	VSSQ_DDR	G	0V	DDR Ground
A6	DQ_B_6	AIO	lp3: 1.2V	LPDDR4X:

			lp4x: 0.6V	CHB DQ6 LPDDR3: DQ24
A7	DQ_B_4	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ4 LPDDR3: DQ30
A8	DQ_B_13	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ13 LPDDR3: DQ15
A9	DQ_B_15	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ15 LPDDR3: DQ12
A10	VSSQ_DDR	G	0V	DDR Ground
A11	DQ_B_9	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ9 LPDDR3: DQ8
A12	DQ_B_12	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ12 LPDDR3: DQ10
A13	DQ_B_11	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ11 LPDDR3:

				DQ11
B1	VSS	G	0V	Digital Core Ground
B2	DQ_B_3	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ3 LPDDR3: DQM3
B3	VSSQ_DDR	G	0V	DDR Ground
B4	DQ_B_1	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ1 LPDDR3: DQ27
B5	DQ_B_0	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ0 LPDDR3: DQ31
B6	DQ_B_7	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ7 LPDDR3: DQ29
B7	DQ_B_5	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ5 LPDDR3: DQ26
B8	VDDQ_V1P2	P	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power

B9	DQ_B_14	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ14 LPDDR3: DQ13
B10	DMI1_B	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Channel B DM1 LPDDR3: DQ14
B11	DQ_B_8	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ12 LPDDR3: DQM1
B12	DQ_B_10	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ10 LPDDR3: DQ9
B13	VSSQ_DDR	G	0V	DDR Ground
C1	GPIO_58	I/O	1.8V	General Purpose I/O 58
C2	GPIO_57	I/O	1.8V	General Purpose I/O 57
C3	GPIO_56	I/O	1.8V	General Purpose I/O 56

C4	GPIO_55	I/O	1.8V	General Purpose I/O 55
C5	GPIO_54	I/O	1.8V	General Purpose I/O 54
C6	DQS0_T_B	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Positive of CHB DQS0 LPDDR3: Positive of DQS3
C7	VSSQ_DDR	G	0V	DDR Ground
C8	CS1_B	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Active-low chip select 1 of CHB LPDDR3: N/A
C9	CA_B_1	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB CA1 LPDDR3: CA5
C10	CKE0_B	AO	lp3: 1.2V lp4x: 1.1V	LPDDR4X: clock enabling 0 of CHB LPDDR3:

				N/A
C11	CKE1_B	AO	lp3: 1.2V lp4x: 1.1V	LPDDR4X: clock enabling 1 of CHB LPDDR3: N/A
C12	VDDQ_V1P2	P	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power
C13	CA_B_5	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB CA5 LPDDR3: CA8
D1	GPIO_114	I/O	1.8V	General Purpose I/O 114
D2	GPIO_113	I/O	1.8V	General Purpose I/O 113
D3	GPIO_112	I/O	1.8V	General Purpose I/O 112
D4	GPIO_111	I/O	1.8V	General Purpose I/O 111
D5	GPIO_53	I/O	1.8V	General Purpose I/O 53

D6	DQS0_C_B	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Negative of CHB DQS0 LPDDR3: Negtive of DQS3
D7	VSSQ_DDR	G	0V	DDR Ground
D8	CA_B_0	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB CA0
D9	VSSQ_DDR	G	0V	DDR Ground
D10	DDR_lp4x_SEL	AIO	1.8V	LPDDR4X: connect to 1.8V LP234: connect to Ground
D11	CK_C_B	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: negative LPDDR differential clock of CHB LPDDR3: negative LPDDR differential clock
D12	CA_B_2	AO	lp3: 1.2V	LPDDR4X:

			lp4x: 0.6V	CHB CA2 LPDDR3: CA9
D13	CA_B_4	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA CA4 LPDDR3: CA7
E1	GPIO_67	I/O	1.8V	General Purpose I/O 67
E2	GPIO_65	I/O	1.8V	General Purpose I/O 65
E3	GPIO_64	I/O	1.8V	General Purpose I/O 64
E4	VSS	G	0V	Digital Core Ground
E5	GPIO_63	I/O	1.8V	General Purpose I/O 63
E6	VSS	G	0V	Digital Core Ground
E7	VSSQ_DDR	G	0V	DDR Ground
E8	VDDQ_V1P2	P	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power

E9	DDR_LP23_VREFDQ	P	lp3: 0.6V lp4: high-z	DQ VREF for lpddr23 , LP4/4x Keep the pin NC
E10	VSSQ_DDR	G	0V	DDR Ground
E11	CK_T_B	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: positive LPDDR differential clock of CHB LPDDR3: positive LPDDR differential clock
E12	CA_B_3	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB CA3 LPDDR3: CA6
E13	AVSS18_DDR	G	0V	DDR Ground
F1	VSS	G	0V	Digital Core Ground
F2	VSS	G	0V	Digital Core Ground
F3	GPIO_69	I/O	1.8V	General Purpose I/O

				69
F4	GPIO_68	I/O	1.8V	General Purpose I/O 68
F5	GPIO_66	I/O	1.8V	General Purpose I/O 66
F6	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
F7	VSS	G	0V	Digital Core Ground
F8	VSSQ_DDR	G	0V	DDR Ground
F9	VDDQ_V1P2	P	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power
F10	VDDQ_V1P2	P	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power
F11	VSSQ_DDR	G	0V	DDR Ground
F12	CS0_B	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: clock enabling 1 of CHB LPDDR3: N/A

F13	DDR_RESET_N	AO	lp3: 1.2V lp4x: 1.1V	LPDDR SDRAM reset
G1	MIPI_CSI1_D1N	AI	1.8V	CSI1 DATA1LANEN
G2	MIPI_CSI1_D1P	AI	1.8V	CSI1 DATA1LANEP
G3	VSS	G	0V	Digital Core Ground
G4	MIPI_CSI1_D0N	AI	1.8V	CSI1 DATA0LANEN
G5	MIPI_CSI1_D0P	AI	1.8V	CSI1 DATA0LANEP
G6	VSS	G	0V	Digital Core Ground
G7	VSS	G	0V	Digital Core Ground
G8	VSS	G	0V	Digital Core Ground
G9	VSS	G	0V	Digital Core Ground
G10	VSSQ_DDR	G	0V	DDR Ground

G11	VDDQ_V1P2	P	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power
G12	AVDD11_DDR	P	lp4x: 1.1V lp4: 1.1V lp3: 1.2V	LPDDR PHY power supply
G13	VDDQ_V1P2	P	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power
H1	MIPI_CSI1_D2N	AI	1.8V	CSI1 DATA2LANEN
H2	MIPI_CSI1_D2P	AI	1.8V	CSI1 DATA2LANEP
H3	VSS	G	0V	Digital Core Ground
H4	MIPI_CSI1_CLKN	AO	1.8V	CSI1 CKLANEN
H5	MIPI_CSI1_CLKP	AO	1.8V	CSI1 CKLANEP
H6	AVSS18_AFEAP	G	0V	DCXO Ground
H7	XI_PAD	AI	1.8V	DCXO crystal input
H8	AVSS18_AFEAP	G	0V	DCXO Ground

H9	VSS	G	0V	Digital Core Ground
H10	VSSU_DDR	G	0V	system DDR Ground
H11	VSSU_DDR	G	0V	system DDR Ground
H12	AVDD18_PHY	P	1.8V	Analog 1.8V power
H13	AVDDU_DDR	P	0.9V	LPDDR PHY PLL logical power
J1	MIPI_CSI3_D0N	AI	1.8V	CSI3 DATA0LANEN
J2	MIPI_CSI3_D0P	AI	1.8V	CSI3 DATA0LANEP
J3	AVSS_CSI	G	0V	MIPI_CSI Ground
J4	MIPI_CSI1_D3N	AI	1.8V	CSI1 DATA3LANEN
J5	MIPI_CSI1_D3P	AI	1.8V	CSI1 DATA3LANEP
J6	AVSS_CSI	G	0V	MIPI_CSI Ground

J7	XO_PAD	AO	1.8V	DCXO crystal output
J8	AVSS18_AFEAP	G	0V	DCXO Ground
J9	AVSS18_AFEAP	G	0V	DCXO Ground
J10	VCC_M1	P	0.9V	Digital Core power
J11	AVDDU_PHY	P	0.9V	LPDDR PHY core logical power
J12	AVDDU_PHY	P	0.9V	LPDDR PHY core logical power
J13	AVDDU_PHY	P	0.9V	LPDDR PHY core logical power
K1	MIPI_CSI3_CLKN	AO	1.8V	CSI3 CKLANEN for CSI3 DATALANE 0/1 when CSI3 is configured as two 2ch CSI; CSI3 CKLANEN for CSI3

				DATALANE 0/1/2/3 when CSI3 is configured as 4ch CSI
K2	MIPI_CSI3_CLKP	AO	1.8V	CSI3 CKLANEP for CSI3 DATALANE 0/1 when CSI3 is configured as two 2ch CSI; CSI3 CKLANEP for CSI3 DATALANE 0/1/2/3 when CSI3 is configured as 4ch CSI
K3	AVSS_CSI	G	0V	MIPI_CSI Ground
K4	MIPI_CSI3_D1N	AI	1.8V	CSI3 DATA1LANEN
K5	MIPI_CSI3_D1P	AI	1.8V	CSI3 DATA1LANEP
K6	AVDD18_C	P	1.8V	MIPI_CSI analog

	SI			power
K7	AVDD09_CSI	P	0.9V	MIPI_CSI digtial power
K8	AVSS_CSI	G	0V	MIPI_CSI Ground
K9	VCC_M1	P	0.9V	Digital Core power
K10	VSS	G	0V	Digital Core Ground
K11	BG_OUT	AO	1.8V	Bandgap output
K12	AVDD18_AFEAP	P	1.8V	1.8V power for DCXO
K13	MPLL_TST_CK	AIO	1.8V	Analog testpin
L2	MIPI_CSI3_D2P	AI	1.8V	CSI3 DATA2LANEP
L3	AVSS_CSI	G	0V	MIPI_CSI Ground
L4	MIPI_CSI2_CLKN	AO	1.8V	CKLANEN for CSI3 DATALANE 2/3 when CSI3 is configured as two 2ch

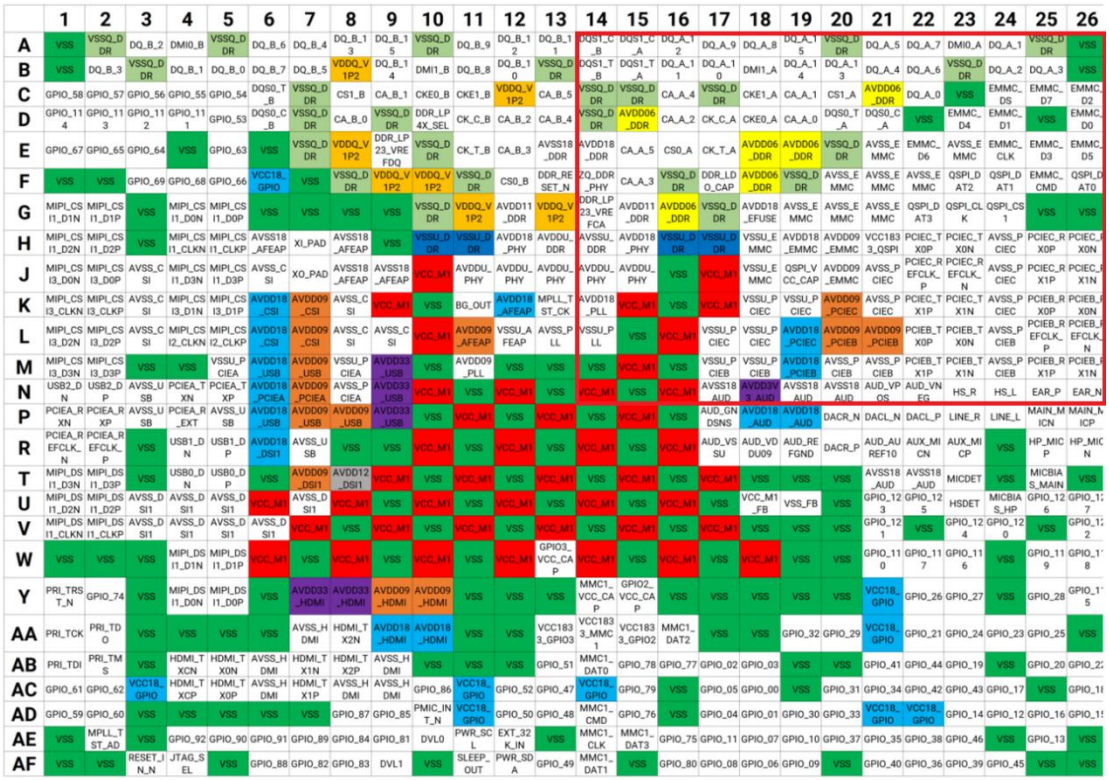
				CSI; Disabled when CSI3 is configured as 4ch CSI
L5	MIPI_CSI2_CLKP	AO	1.8V	CKLANEP for CSI3 DATALANE 2/3 when CSI3 is configured as two 2ch CSI; Disabled when CSI3 is configured as 4ch CSI
L6	AVDD18_CSI	P	1.8V	MIPI_CSI analog power
L7	AVDD09_CSI	P	0.9V	MIPI_CSI digtial power
L8	AVSS_CSI	G	0V	MIPI_CSI Ground
L9	AVSS_CSI	G	0V	MIPI_CSI Ground
L10	VCC_M1	P	0.9V	Digital Core power
L11	AVDD09_AF	P	0.9V	0.9V power

	EAP			for DCXO
L12	VSSU_AFEAP	G	0V	DCXO Ground
L13	AVSS_PLL	G	0V	Analog Core Ground
M1	MIPI_CSI3_D3N	AI	1.8V	CSI3 DATA3LANEN
M2	MIPI_CSI3_D3P	AI	1.8V	CSI3 DATA3LANEP
M3	VSS	G	0V	Digital Core Ground
M4	VSS	G	0V	Digital Core Ground
M5	VSSU_PCIEA	G	0V	PCIEA Ground
M6	AVDD18_USB	P	1.8V	USB2.0 1.8V power
M7	AVDD09_USB	P	0.9V	USB2.0 digital power
M8	VSSU_PCIEA	G	0V	PCIEA Ground
M9	AVDD33_USB	P	3.3V	USB2.0 3.3V power

M10	VSS	G	0V	Digital Core Ground
M11	AVDD09_PLL	P	0.9	System PLL power supply
M12	VSS	G	0V	Digital Core Ground
M13	VSS	G	0V	Digital Core Ground
N1	USB2_DN	AIO	3.3V	USB2.0_2 D- differential data line
N2	USB2_DP	AIO	3.3V	USB2.0_2 D+ differential data line
N3	AVSS_USB	G	0V	USB2.0 Ground
N4	PCIEA_TXN	AO	1.8V	PCIEA TXLANEN
N5	PCIEA_TXP	AO	1.8V	PCIEA TXLANEP
N6	AVDD18_PCIEA	P	1.8V	PCIEA analog power
N7	AVDD09_P	P	0.9V	PCIEA

	CIEA			digital power
N8	AVSS_PCIEA	G	0V	PCIEA Ground
N9	AVDD33_USB	P	3.3V	USB2.0 3.3V power
N10	VCC_M1	P	0.9V	Digital Core power
N11	VSS	G	0V	Digital Core Ground
N12	VCC_M1	P	0.9V	Digital Core power
N13	VSS	G	0V	Digital Core Ground

4.2.2 (A~N, 14~26)



Note. Definition of symbols used for pin type:

- AO = Analog output
- AI = Analog input
- AIO = Analog input/output
- G = Ground
- I/O = Input/Output
- P = Power
- RO = Reference output

Pin ID	Name	Type	Power Domain	Function
A14	DQS1_C_B	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Negative of CHB DQS1 LPDDR3: Negtive of

				DQS1
A15	DQS1_C_A	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Negative of CHA DQS1 LPDDR3: Negtive of DQS0
A16	DQ_A_12	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ12 LPDDR3: DQM0
A17	DQ_A_9	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ9 LPDDR3: DQ7
A18	DQ_A_8	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ8 LPDDR3: DQ5
A19	DQ_A_15	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ15 LPDDR3: DQ3
A20	VSSQ_DDR	G	0V	DDR Ground
A21	DQ_A_5	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ5 LPDDR3:

				DQ21
A22	DQ_A_7	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ7 LPDDR3: DQ17
A23	DMI0_A	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Channel A DM0 LPDDR3: DQ22
A24	DQ_A_1	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ1 LPDDR3: DQ16
A25	VSSQ_DDR	G	0V	DDR Ground
A26	VSS	G	0V	Digital Core Ground
B14	DQS1_T_B	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Positive of CHB DQS1 LPDDR3: Positive of DQS1
B15	DQS1_T_A	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Positive of CHA DQS1 LPDDR3: Positive of

				DQS0
B16	DQ_A_11	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ11 LPDDR3: DQ4
B17	DQ_A_10	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ10 LPDDR3: DQ6
B18	DMI1_A	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Channel A DM1 LPDDR3: DQ2
B19	DQ_A_14	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ14 LPDDR3: DQ1
B20	DQ_A_13	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ13 LPDDR3: DQ0
B21	DQ_A_4	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ4 LPDDR3: DQ18
B22	DQ_A_6	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ6

				LPDDR3: DQ23
B23	VSSQ_DDR	G	0V	DDR Ground
B24	DQ_A_2	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ2 LPDDR3: DQ19
B25	DQ_A_3	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHB DQ3 LPDDR3: DQM2
B26	VSS	G	0V	Digital Core Ground
C14	VSSQ_DDR	G	0V	DDR Ground
C15	VSSQ_DDR	G	0V	DDR Ground
C16	CA_A_4	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA CA4 LPDDR3: CA3
C17	VSSQ_DDR	G	0V	DDR Ground
C18	CKE1_A	AO	lp3: 1.2V lp4x: 1.1V	LPDDR4X: clock enabling 1 of

				CHA LPDDR3: clock enabling 1
C19	CA_A_1	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA CA1 LPDDR3: CA2
C20	CS1_A	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Active-low chip select 1 of CHA LPDDR3: Active-low chip select 1
C21	AVDD06_DDR	P	lp4x: 0.6V lp4: TBD/lp3: TBD	LPDDR4X IO power
C22	DQ_A_0	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA DQ0 LPDDR3: DQ20
C23	VSS	G	0V	Digital Core Ground
C24	EMMC_DS	I/O	1.8V	eMMC data strobe
C25	EMMC_D7	I/O	1.8V	eMMC data7

C26	EMMC_D2	I/O	1.8V	eMMC data2
D14	VSSQ_DDR	G	0V	DDR Ground
D15	AVDD06_DDR	P	lp4x: 0.6V lp4: TBD lp3: TBD	LPDDR4X IO power
D16	CA_A_2	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA CA2
D17	CK_C_A	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: negative LPDDR differential clock of CHA LPDDR3: N/A
D18	CKE0_A	AO	lp3: 1.2V lp4x: 1.1V	LPDDR4X: clock enabling 0 of CHA LPDDR3: clock enabling 0
D19	CA_A_0	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA CA0 LPDDR3: CA4
D20	DQS0_T_A	AIO	lp3: 1.2V	LPDDR4X: Positive of

			lp4x: 0.6V	CHA DQS0 LPDDR3: Positive of DQS2
D21	DQS0_C_A	AIO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Negative of CHA DQS0 LPDDR3: Negative of DQS2
D22	VSS	G	0V	Digital Core Ground
D23	EMMC_D4	I/O	1.8V	eMMC data4
D24	EMMC_D1	I/O	1.8V	eMMC data1
D25	VSS	G	0V	Digital Core Ground
D26	EMMC_D0	I/O	1.8V	eMMC data0
E14	AVDD18_DDR	P	1.8V	LPDDR PHY PLL 1.8V power
E15	CA_A_5	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA CA5 LPDDR3: CA1
E16	CS0_A	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: Active-low chip select 0

				of CHA LPDDR3: Active-low chip select 0
E17	CK_T_A	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: positive LPDDR differential clock of CHA LPDDR3: N/A
E18	AVDD06_DDR	P	lp4x: 0.6V lp4: TBD lp3: TBD	LPDDR4X IO power
E19	AVDD06_DDR	P	lp4x: 0.6V lp4: TBD lp3: TBD	LPDDR4X IO power
E20	VSSQ_DDR	G	0V	DDR Ground
E21	AVSS_EMMC	G	0V	eMMC Ground
E22	EMMC_D6	I/O	1.8V	eMMC data6
E23	AVSS_EMMC	G	0V	eMMC Ground
E24	EMMC_CLK	I/O	1.8V	eMMC Clock
E25	EMMC_D3	I/O	1.8V	eMMC data3

E26	EMMC_D5	I/O	1.8V	eMMC data5
F14	ZQ_DDR_PHY	AIO	lp3: 1.2V lp4x: 0.6V	DDR ZQ calibration
F15	CA_A_3	AO	lp3: 1.2V lp4x: 0.6V	LPDDR4X: CHA CA3 LPDDR3: CA0
F16	VSSQ_DDR	G	0V	DDR Ground
F17	DDR_LDO_CAP	RO	0.7~0.9V	External LDO output ball; Connect to a 100nF capacitor on PCB board
F18	AVDD06_DDR	P	lp4x: 0.6V lp4: TBD lp3: TBD	LPDDR4X IO power
F19	VSSQ_DDR	G	0V	DDR Ground
F20	AVSS_EMMC	G	0V	eMMC Ground
F21	AVSS_EMMC	G	0V	eMMC Ground
F22	AVSS_EMM	G	0V	eMMC

	C			Ground
F23	QSPI_DAT2	I/O	1.8V/3.3V	QSPI data2
F24	QSPI_DAT1	I/O	1.8V/3.3V	QSPI data1
F25	EMMC_CMD	I/O	1.8V	eMMC command
F26	QSPI_DAT0	I/O	1.8V/3.3V	QSPI data0
G14	DDR_LP23_VREFCA	P	lp3: 0.6V lp4: high-z	CA VREF for lpddr23, LP4/4x Keep the pin NC
G15	AVDD11_DDR	P	lp4x: 1.1V lp4: 1.1V lp3: 1.2V	LPDDR PHY power supply
G16	AVDD06_DDR	P	lp4x: 0.6V lp4: TBD lp3: TBD	LPDDR4X IO power
G17	VSSQ_DDR	G	0V	DDR Ground
G18	AVDD18_EFUSE	P	1.8V	ANAGRP
G19	AVSS_EMMC	G	0V	eMMC Ground
G20	AVSS_EMMC	G	0V	eMMC Ground

G21	AVSS_EMMC	G	0V	eMMC Ground
G22	QSPI_DAT3	I/O	1.8V/3.3V	QSPI data3
G23	QSPI_CLK	I/O	1.8V/3.3V	QSPI CLK
G24	QSPI_CS1	I/O	1.8V/3.3V	QSPI CS
G25	VSS	G	0V	Digital Core Ground
G26	VSS	G	0V	Digital Core Ground
H14	AVSSU_DDR	G	0V	DDR Ground
H15	AVDD18_PHY	P	1.8V	Analog 1.8V power
H16	VSSU_DDR	G	0V	System DDR Ground
H17	VSSU_DDR	G	0V	System DDR Ground
H18	VSSU_EMMC	G	0V	eMMC Ground
H19	AVDD18_EMMC	P	1.8V	eMMC analog power
H20	AVDD09_E	P	0.9V	eMMC

	MMC			digtial power
H21	VCC1833_QSPI	P	1.8V/3.3V	QSPI IO power
H22	PCIEC_TX0P	AO	1.8V	PCIEC TX0LANEP
H23	PCIEC_TX0N	AO	1.8V	PCIEC TX0LANEN
H24	AVSS_PCIEC	G	0V	PCIEC Ground
H25	PCIEC_RX0P	AI	1.8V	PCIEC RX0LANEP
H26	PCIEC_RX0N	AI	1.8V	PCIEC RX0LANEN
J14	AVDDU_PHY	P	0.9V	LPDDR PHY core logical power
J15	AVDDU_PHY	P	0.9V	LPDDR PHY core logical power
J16	VSS	G	0V	Digital Core Ground
J17	VCC_M1	P	0.9V	Digital Core power
J18	VSSU_EMMC	G	0V	eMMC Ground

J19	QSPI_VCC_CAP	RO	1.8V	QSPI 1.8V LDO cap
J20	AVDD09_EMMC	P	0.9V	eMMC digtial power
J21	AVSS_PCIEC	G	0V	PCIEC Ground
J22	PCIEC_REFCLK_P	AIO	1.8V	PCIEC CKLANEP
J23	PCIEC_REFCLK_N	AIO	1.8V	PCIEC CKLANEN
J24	AVSS_PCIEC	G	0V	PCIEC Ground
J25	PCIEC_RX1P	AI	1.8V	PCIEC RX1LANEP
J26	PCIEC_RX1N	AI	1.8V	PCIEC RX1LANEN
K14	AVDD18_PLL	P	1.8	System PLL power supply
K15	VCC_M1	P	0.9V	Digital Core power
K16	VSS	G	0V	Digital core Ground
K17	VCC_M1	P	0.9V	Digital Core power

K18	VSSU_PCIEC	G	0V	PCIEC Ground
K19	VSSU_PCIEC	G	0V	PCIEC Ground
K20	AVDD09_PCIEC	P	0.9V	PCIEC digital power
K21	AVSS_PCIEC	G	0V	PCIEC Ground
K22	PCIEC_TX1P	AO	1.8V	PCIEC TX1LANEP
K23	PCIEC_TX1N	AO	1.8V	PCIEC TX1LANEN
K24	AVSS_PCIEC	G	0V	PCIEC Ground
K25	PCIEB_RX0P	AI	1.8V	PCIEB RX0LANEP
K26	PCIEB_RX0N	AI	1.8V	PCIEB RX0LANEN
L14	VSSU_PLL	G	0V	System PLL Ground
L15	VSS	G	0V	Digital core Ground
L16	VCC_M1	P	0.9V	Digital Core power
L17	VSSU_PCIE	G	0V	PCIEC

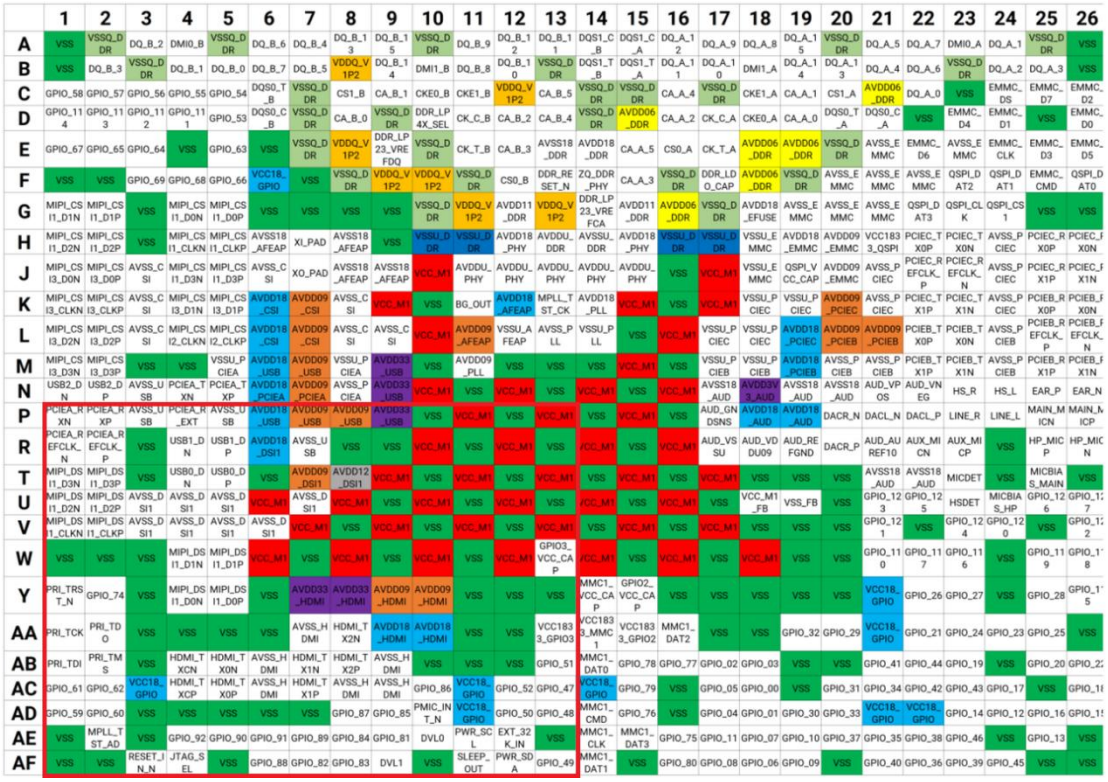
	C			Ground
L18	VSSU_PCIEC	G	0V	PCIEC Ground
L19	AVDD18_PCIEC	P	1.8V	PCIEC analog power
L20	AVDD09_PCIEB	P	0.9V	PCIEB digital power
L21	AVDD09_PCIEB	P	0.9V	PCIEB digital power
L22	PCIEB_TX0P	AO	1.8V	PCIEB TX0LANEP
L23	PCIEB_TX0N	AO	1.8V	PCIEB TX0LANEN
L24	AVSS_PCIEB	G	0V	PCIEB Ground
L25	PCIEB_REFCLK_P	AIO	1.8V	PCIEB CKLANEP
L26	PCIEB_REFCLK_N	AIO	1.8V	PCIEB CKLANEN
M14	VSS	G	0V	Digital Core Ground
M15	VCC_M1	P	0.9V	Digital Core power
M16	VSS	G	0V	Digital Core

				Ground
M17	VSSU_PCIEB	G	0V	PCIEB Ground
M18	VSSU_PCIEB	G	0V	PCIEB Ground
M19	AVDD18_PCIEB	P	1.8V	PCIEB analog power
M20	AVSS_PCIEB	G	0V	PCIEB Ground
M21	AVSS_PCIEB	G	0V	PCIEB Ground
M22	PCIEB_TX1P	AO	1.8V	PCIEB TX1LANEP
M23	PCIEB_TX1N	AO	1.8V	PCIEB TX1LANEN
M24	AVSS_PCIEB	G	0V	PCIEB Ground
M25	PCIEB_RX1P	AI	1.8V	PCIEB RX1LANEP
M26	PCIEB_RX1N	AI	1.8V	PCIEB RX1LANEN
N14	VCC_M1	P	0.9V	Digital Core power
N15	VSS	G	0V	Digital Core

				Ground
N16	VCC_M1	P	0.9V	Digital Core power
N17	AVSS18_AUD	G	0V	Audio Ground
N18	AVDD3V3_AUD	P	3.3V	3.3V power for earphone driver
N19	AVSS18_AUD	G	0V	Audio Ground
N20	AVSS18_AUD	G	0V	Audio Ground
N21	AUD_VPOS	P	1.8V	Positive voltage for headphone driver
N22	AUD_VNEG	P	-1.8V	Negative voltage for headphone driver
N23	HS_R	AO	+/-1.8V	Headphone R-channel driver
N24	HS_L	AO	+/-1.8V	Headphone L-channel driver
N25	EAR_P	AO	3.3V	Earphone

				driver P
N26	EAR_N	AO	3.3V	Earphone driver N

4.2.3 (P~AF, 1~13)



Note. Definition of symbols used for pin type:

- AO = Analog output
- AI = Analog input
- AIO = Analog input/output
- G = Ground
- I/O = Input/Output
- P = Power
- RO = Reference output

Pin ID	Name	Type	Power Domain	Function
P1	PCIEA_RXN	AI	1.8V	PCIEA RXLANEN
P2	PCIEA_RXP	AI	1.8V	PCIEA RXLANEP
P3	AVSS_USB	G	0V	USB2.0 Ground
P4	PCIEA_R_EXT	AO	1.8V	PCIEA External calibration resistor
P5	AVSS_USB	G	0V	USB2.0 Ground
P6	AVDD18_USB	P	1.8V	USB2.0 1.8V power
P7	AVDD09_USB	P	0.9V	USB2.0 digital power
P8	AVDD09_USB	P	0.9V	USB2.0 digital power
P9	AVDD33_USB	P	3.3V	USB2.0 3.3V power
P10	VSS	G	0V	Digital Core Ground
P11	VCC_M1	P	0.9V	Digital Core power

P12	VSS	G	0V	Digital Core Ground
P13	VCC_M1	P	0.9V	Digital Core power
R1	PCIEA_REFCLK_N	AIO	1.8V	PCIEA CKLANEN
R2	PCIEA_REFCLK_P	AIO	1.8V	PCIEA CKLANEP
R3	VSS	G	0V	Digital core Ground
R4	USB1_DN	AIO	3.3V	USB2.0_1 D- differential data line
R5	USB1_DP	AIO	3.3V	USB2.0_1 D+ differential data line
R6	AVDD18_DSI1	P	1.8V	DSI analog power
R7	AVSS_USB	G	0V	USB2.0 Ground
R8	VSS	G	0V	Digital Core Ground
R9	VSS	G	0V	Digital Core Ground

R10	VCC_M1	P	0.9V	Digital Core power
R11	VSS	G	0V	Digital Core Ground
R12	VCC_M1	P	0.9V	Digital Core power
R13	VSS	G	0V	Digital Core Ground
T1	MIPI_DSI1_D3N	AO	1.2V	DSI DATA3LANEN
T2	MIPI_DSI1_D3P	AO	1.2V	DSI DATA3LANEP
T3	VSS	G	0V	Digital core ground
T4	USB0_DN	AIO	3.3V	USB2.0_0 D- differential data line
T5	USB0_DP	AIO	3.3V	USB2.0_0 D+ differential data line
T6	VSS	G	0V	Digital core ground
T7	AVDD09_D	P	0.9V	DSI digital

	SI1			power
T8	AVDD12_DSI1	P	1.2V	DSI driver power
T9	VCC_M1	P	0.9V	Digital Core power
T10	VSS	G	0V	Digital Core ground
T11	VCC_M1	P	0.9V	Digital Core power
T12	VSS	G	0V	Digital Core ground
T13	VCC_M1	P	0.9V	Digital Core power
U1	MIPI_DSI1_D2N	AO	1.2V	DSI DATA2LANEN
U2	MIPI_DSI1_D2P	AO	1.2V	DSI DATA2LANEP
U3	AVSS_DSI1	G	0V	DSI Ground
U4	AVSS_DSI1	G	0V	DSI Ground
U5	AVSS_DSI1	G	0V	DSI Ground
U6	VCC_M1	P	0.9V	Digital Core power
U7	AVSS_DSI1	G	0V	DSI Ground

U8	VCC_M1	P	0.9V	Digital Core power
U9	VSS	G	0V	Digital Core ground
U10	VCC_M1	P	0.9V	Digital Core power
U11	VSS	G	0V	Digital Core ground
U12	VCC_M1	P	0.9V	Digital Core power
U13	VSS	G	0V	Digital Core ground
V1	MIPI_DSI1_CLKN	AO	1.2V	DSI CKLANEN
V2	MIPI_DSI1_CLKP	AO	1.2V	DSI CKLANEP
V3	AVSS_DSI1	G	0V	DSI Ground
V4	AVSS_DSI1	G	0V	DSI Ground
V5	AVSS_DSI1	G	0V	DSI Ground
V6	AVSS_DSI1	G	0V	DSI Ground
V7	VCC_M1	P	0.9V	Digital Core power
V8	VSS	G	0V	Digital Core ground

V9	VCC_M1	P	0.9V	Digital Core power
V10	VSS	G	0V	Digital Core ground
V11	VCC_M1	P	0.9V	Digital Core power
V12	VSS	G	0V	Digital Core ground
V13	VCC_M1	P	0.9V	Digital Core power
W1	VSS	G	0V	Digital Core ground
W2	VSS	G	0V	Digital Core ground
W3	VSS	G	0V	Digital Core ground
W4	MIPI_DSI1_D1N	AO	1.2V	DSI DATA1LANEN
W5	MIPI_DSI1_D1P	AO	1.2V	DSI DATA1LANEP
W6	VCC_M1	P	0.9V	Digital Core power
W7	VSS	G	0V	Digital Core ground

W8	VCC_M1	P	0.9V	Digital Core power
W9	VSS	G	0V	Digital Core ground
W10	VCC_M1	P	0.9V	Digital Core power
W11	VSS	G	0V	Digital Core ground
W12	VCC_M1	P	0.9V	Digital Core power
W13	GPIO3_VCC_CAP	RO	1.8V	GPIO3 1.8V LDO cap
Y1	PRI_TRST_N	I/O	1.8V	JTAG reset
Y2	GPIO_74	I/O	1.8V	General Purpose I/O 74
Y3	VSS	G	0V	Digital Core ground
Y4	MIPI_DSI1_D0N	AO	1.2V	DSI DATA0LANEN
Y5	MIPI_DSI1_D0P	AO	1.2V	DSI DATA0LANEP
Y6	VSS	G	0V	Digital Core

				ground
Y7	AVDD33_HDMI	P	3.3V	HDMI 3.3V power
Y8	AVDD33_HDMI	P	3.3V	HDMI 3.3V power
Y9	AVDD09_HDMI	P	0.9V	HDMI digtial power
Y10	AVDD09_HDMI	P	0.9V	HDMI digtial power
Y11	VSS	G	0V	Digital Core ground
Y12	VSS	G	0V	Digital Core ground
Y13	VSS	G	0V	Digital Core ground
AA1	PRI_TCK	I/O	1.8V	JTAG clock
AA2	PRI_TDO	I/O	1.8V	JTAG output data
AA3	VSS	G	0V	Digital Core ground
AA4	VSS	G	0V	Digital Core ground
AA5	VSS	G	0V	Digital Core ground
AA6	VSS	G	0V	Digital Core

				ground
AA7	AVSS_HDMI	G	0V	HDMI Ground
AA8	HDMI_TX2N	AO	1.8V	HDMI data2n
AA9	AVDD18_HDMI	P	1.8V	HDMI 1.8V power
AA10	AVDD18_HDMI	P	1.8V	HDMI 1.8V power
AA11	VSS	G	0V	Digital Core ground
AA12	VSS	G	0V	Digital Core ground
AA13	VCC1833_GPIO3	P	1.8V/3.3V	GPIO3 IO power
AB1	PRI_TDI	I/O	1.8V	JTAG input data
AB2	PRI_TMS	I/O	1.8V	JTAG mode selection
AB3	VSS	G	0V	Digital Core ground
AB4	HDMI_TXCN	AO	1.8V	HDMI clkn
AB5	HDMI_TX0N	AO	1.8V	HDMI data0n

AB6	AVSS_HDMI	G	0V	HDMI Ground
AB7	HDMI_TX1N	AO	1.8V	HDMI data1n
AB8	HDMI_TX2P	AO	1.8V	HDMI data2p
AB9	AVSS_HDMI	G	0V	HDMI Ground
AB10	VSS	G	0V	Digital Core ground
AB11	VSS	G	0V	Digital Core ground
AB12	VSS	G	0V	Digital Core ground
AB13	GPIO_51	I/O	1.8V/3.3V	General purpose I/O 51
AC1	GPIO_61	I/O	1.8V	General Purpose I/O 61
AC2	GPIO_62	I/O	1.8V	General Purpose I/O 62
AC3	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power

AC4	HDMI_TXCP	AO	1.8V	HDMI clkp
AC5	HDMI_TX0P	AO	1.8V	HDMI data0p
AC6	AVSS_HDMI	G	0V	HDMI Ground
AC7	HDMI_TX1P	AO	1.8V	HDMI data1p
AC8	AVSS_HDMI	G	0V	HDMI Ground
AC9	AVSS_HDMI	G	0V	HDMI Ground
AC10	GPIO_86	I/O	1.8V	General Purpose I/O 86
AC11	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
AC12	GPIO_52	I/O	1.8V/3.3V	General Purpose I/O 52
AC13	GPIO_47	I/O	1.8V/3.3V	General Purpose I/O 47
AD1	GPIO_59	I/O	1.8V	General Purpose I/O 59

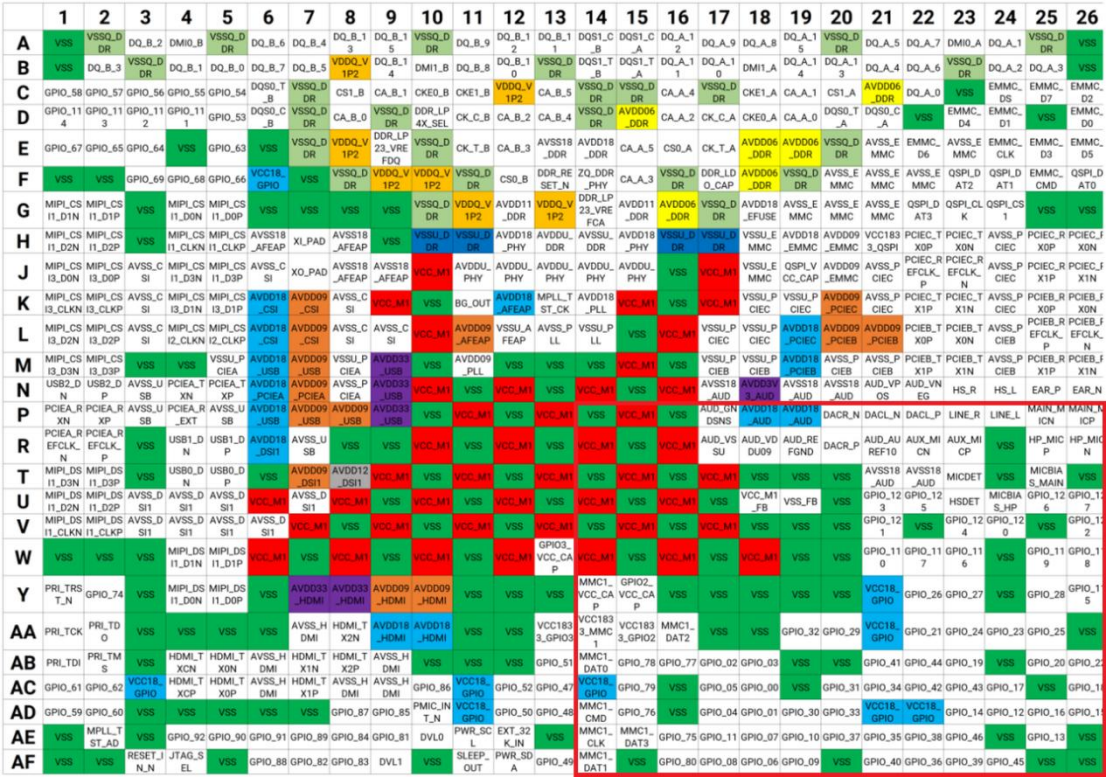
AD2	GPIO_60	I/O	1.8V	General Purpose I/O 60
AD3	VSS	G	0V	Digital Core ground
AD4	VSS	G	0V	Digital Core ground
AD5	VSS	G	0V	Digital Core ground
AD6	VSS	G	0V	Digital Core ground
AD7	VSS	G	0V	Digital Core ground
AD8	GPIO_87	I/O	1.8V	General Purpose I/O 87
AD9	GPIO_85	I/O	1.8V	General Purpose I/O 85
AD10	PMIC_INT_N	I/O	1.8V	PMIC interrupt
AD11	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
AD12	GPIO_50	I/O	1.8V/3.3V	General Purpose I/O 50

AD13	GPIO_48	I/O	1.8V/3.3V	General Purpose I/O 48
AE1	VSS	G	0V	Digital Core ground
AE2	MPLL_TST_AD	AIO	1.8V	Analog testpin
AE3	VSS	G	0V	Digital Core ground
AE4	GPIO_92	I/O	1.8V	General Purpose I/O 92
AE5	GPIO_90	I/O	1.8V	General Purpose I/O 90
AE6	GPIO_91	I/O	1.8V	General Purpose I/O 91
AE7	GPIO_89	I/O	1.8V	General Purpose I/O 89
AE8	GPIO_84	I/O	1.8V	General Purpose I/O 84
AE9	GPIO_81	I/O	1.8V	General Purpose I/O 81

AE10	DVL0	I/O	1.8V	Hardware dynamic voltage regulation signal0
AE11	PWR_SCL	I/O	1.8V	PMIC I2C bus clock
AE12	EXT_32K_IN	I/O	1.8V	32K clock input
AE13	VSS	G	0V	Digital Core ground
AF1	VSS	G	0V	Digital Core ground
AF2	VSS	G	0V	Digital Core ground
AF3	RESET_IN_N	I/O	1.8V	Reset input
AF4	JTAG_SEL	I/O	1.8V	Primary JTAG selection
AF5	VSS	G	0V	Digital Core ground
AF6	GPIO_88	I/O	1.8V	General Purpose I/O 88
AF7	GPIO_82	I/O	1.8V	General Purpose I/O

				82
AF8	GPIO_83	I/O	1.8V	General Purpose I/O 83
AF9	DVL1	I/O	1.8V	Hardware dynamic voltage regulation signal1
AF10	VSS	G	0V	Digital Core ground
AF11	SLEEP_OUT	I/O	1.8V	VCXO enabling
AF12	PWR_SDA	I/O	1.8V	PMIC I2C bus data/address
AF13	GPIO_49	I/O	1.8V/3.3V	General Purpose I/O 49

4.2.4 (P~AF, 14~26)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	VSS	VSSQ_DDR	DQ_B_2	DMI0_B	VSSQ_DDR	DQ_B_6	DQ_B_4	DQ_B_13	DQ_B_15	VSSQ_DDR	DQ_B_9	DQ_B_12	DQ_B_11	DQS1_C_B	DQS1_C_A	DQ_A_12	DQ_A_9	DQ_A_8	DQ_A_15	VSSQ_DDR	DQ_A_5	DQ_A_7	DMI0_A	DQ_A_1	VSSQ_DDR	VSS
B	VSS	DQ_B_3	VSSQ_DDR	DQ_B_1	DQ_B_0	DQ_B_7	DQ_B_5	VDDQ_V1P2	DQ_B_14	DMI1_B	DQ_B_8	DQ_B_10	VSSQ_DDR	DQS1_T_B	DQS1_T_A	DQ_A_11	DQ_A_10	DMI1_A	DQ_A_14	DQ_A_13	VSSQ_DDR	DQ_A_4	DQ_A_6	VSSQ_DDR	DQ_A_2	DQ_A_3	VSS
C	GPIO_58	GPIO_57	GPIO_56	GPIO_55	GPIO_54	DQS0_T_B	VSSQ_DDR	CS1_B	CA_B_1	CKE0_B	CKE1_B	VDDQ_V1P2	CA_B_5	VSSQ_DDR	VSSQ_DDR	CA_A_4	VSSQ_DDR	CKE1_A	CA_A_1	CS1_A	AVDD06_DDR	DQ_A_0	VSS	EMMC_DS	EMMC_D7	EMMC_D2
D	GPIO_114	GPIO_113	GPIO_112	GPIO_111	GPIO_53	DQS0_C_B	VSSQ_DDR	CA_B_0	VSSQ_DDR	DDR_LP4X_SEL	CK_C_B	CA_B_2	CA_B_4	VSSQ_DDR	AVDD06_DDR	CA_A_2	CK_C_A	CKE0_A	CA_A_0	DQS0_T_A	DQS0_C_A	VSS	EMMC_D4	EMMC_D1	VSS	EMMC_D0
E	GPIO_67	GPIO_65	GPIO_64	VSS	GPIO_63	VSS	VSSQ_DDR	VDDQ_V1P2	DDR_LP23_VREFDQ	VSSQ_DDR	CK_T_B	CA_B_3	AVSS18_DDR	AVDD18_DDR	CA_A_5	CS0_A	CK_T_A	AVDD06_DDR	AVDD06_DDR	VSSQ_DDR	AVSS_EMMC	EMMC_D6	AVSS_EMMC	EMMC_CLK	EMMC_D3	EMMC_D5
F	VSS	VSS	GPIO_69	GPIO_68	GPIO_66	VCC18_GPIO	VSS	VSSQ_DDR	VDDQ_V1P2	VDDQ_V1P2	VSSQ_DDR	CS0_B	DDR_RESET_N	ZQ_DDR_PHY	CA_A_3	VSSQ_DDR	DDR_LDO_CAP	AVDD06_DDR	VSSQ_DDR	AVSS_EMMC	AVSS_EMMC	AVSS_EMMC	QSPI_DAT2	QSPI_DAT1	EMMC_CMD	QSPI_DAT0
G	MIPI_CSI1_D1N	MIPI_CSI1_D1P	VSS	MIPI_CSI1_D0N	MIPI_CSI1_D0P	VSS	VSS	VSS	VSS	VSSQ_DDR	VDDQ_V1P2	AVDD11_DDR	VDDQ_V1P2	DDR_LP23_VREFCA	AVDD11_DDR	AVDD06_DDR	VSSQ_DDR	AVDD18_EFUSE	AVSS_EMMC	AVSS_EMMC	AVSS_EMMC	QSPI_DAT3	QSPI_CLK	QSPI_CS1	VSS	VSS
H	MIPI_CSI1_D2N	MIPI_CSI1_D2P	VSS	MIPI_CSI1_CLKN	MIPI_CSI1_CLKP	AVSS18_AFEAP	XI_PAD	AVSS18_AFEAP	VSS	VSSU_DDR	VSSU_DDR	AVDD18_PHY	AVDDU_DDR	AVSSU_DDR	AVDD18_PHY	VSSU_DDR	VSSU_DDR	VSSU_EMMC	AVDD18_EMMC	AVDD09_EMMC	VCC1833_QSPI	PCIEC_TX0P	PCIEC_TX0N	AVSS_PCIEC	PCIEC_RX0P	PCIEC_RX0N
J	MIPI_CSI3_D0N	MIPI_CSI3_D0P	AVSS_CSI	MIPI_CSI1_D3N	MIPI_CSI1_D3P	AVSS_CSI	XO_PAD	AVSS18_AFEAP	AVSS18_AFEAP	VCC_M1	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	AVDDU_PHY	VSS	VCC_M1	VSSU_EMMC	QSPI_VCC_CAP	AVDD09_EMMC	AVSS_PCIEC	PCIEC_REFCLK_P	PCIEC_REFCLK_N	AVSS_PCIEC	PCIEC_RX1P	PCIEC_RX1N
K	MIPI_CSI3_CLKN	MIPI_CSI3_CLKP	AVSS_CSI	MIPI_CSI3_D1N	MIPI_CSI3_D1P	AVDD18_CSI	AVDD09_CSI	AVSS_CSI	VCC_M1	VSS	BG_OUT	AVDD18AFEAP	MPLL_TST_CK	AVDD18_PLL	VCC_M1	VSS	VCC_M1	VSSU_PCIEC	VSSU_PCIEC	AVDD09_PCIEC	AVSS_PCIEC	PCIEC_TX1P	PCIEC_TX1N	AVSS_PCIEC	PCIEB_RX0P	PCIEB_RX0N
L	MIPI_CSI3_D2N	MIPI_CSI3_D2P	AVSS_CSI	MIPI_CSI2_CLKN	MIPI_CSI2_CLKP	AVDD18_CSI	AVDD09_CSI	AVSS_CSI	AVSS_CSI	VCC_M1	AVDD09_AFEAP	VSSU_AFEAP	AVSS_PLL	VSSU_PLL	VSS	VCC_M1	VSSU_PCIEC	VSSU_PCIEC	AVDD18_PCIEC	AVDD09_PCIEB	AVDD09_PCIEB	PCIEB_TX0P	PCIEB_TX0N	AVSS_PCIEB	PCIEB_RX0P	PCIEB_EFCLK_N
M	MIPI_CSI3_D3N	MIPI_CSI3_D3P	VSS	VSS	VSSU_PCIEA	AVDD18_USB	AVDD09_USB	VSSU_PCIEA	AVDD33_USB	VSS	AVDD09_PLL	VSS	VSS	VSS	VCC_M1	VSS	VSSU_PCIEB	VSSU_PCIEB	AVDD18_PCIEB	AVSS_PCIEB	AVSS_PCIEB	PCIEB_TX1P	PCIEB_TX1N	AVSS_PCIEB	PCIEB_RX1P	PCIEB_EFCLK_N
N	USB2_DN	USB2_DP	AVSS_USB	PCIEA_TXN	PCIEA_TXP	AVDD18_PCIEA	AVDD09_PCIEA	AVSS_PCIEA	AVDD33_USB	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	AVSS18_AUD	AVDD33_AUD	AVSS18_AUD	AVSS18_AUD	AUD_VPOS	AUD_VNEG	HS_R	HS_L	EAR_P	EAR_N
P	PCIEA_RXN	PCIEA_RXP	AVSS_USB	PCIEA_REXT	AVSS_USB	AVDD18_USB	AVDD09_USB	AVDD09_USB	AVDD33_USB	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	AUD_GNDSNS	AVDD18_AUD	AVDD18_AUD	DACR_N	DACL_N	DACL_P	LINE_R	LINE_L	MAIN_MICN	MAIN_MICP
R	PCIEA_REFCLK_N	PCIEA_REFCLK_P	VSS	USB1_DN	USB1_DP	AVDD18_DSI1	AVSS_USB	VSS	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	AUD_VSSU	AUD_VDDU09	AUD_REFGND	DACR_P	AUD_AUREF10	AUX_MICN	AUX_MICP	VSS	HP_MICP	HP_MICN
T	MIPI_DSI1_D3N	MIPI_DSI1_D3P	VSS	USB0_DN	USB0_DP	VSS	AVDD09_DSI1	AVDD12_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	VSS	AVSS18_AUD	AVSS18_AUD	MICDET	VSS	MICBIAS_MAIN	VSS
U	MIPI_DSI1_D2N	MIPI_DSI1_D2P	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	VCC_M1	AVSS_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1_FB	VSS_FB	VSS	GPIO_123	GPIO_125	HSDET	MICBIAS_HP	GPIO_126	GPIO_117
V	MIPI_DSI1_CLKN	MIPI_DSI1_CLKP	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	AVSS_DSI1	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	VSS	GPIO_121	VSS	GPIO_124	GPIO_120	VSS	GPIO_112
W	VSS	VSS	VSS	MIPI_DSI1_D1N	MIPI_DSI1_D1P	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	GPIO3_VCC_CAP	VCC_M1	VSS	VCC_M1	VSS	VCC_M1	VSS	VSS	GPIO_110	GPIO_117	GPIO_116	VSS	GPIO_119	GPIO_118
Y	PRI_TRST_N	GPIO_74	VSS	MIPI_DSI1_D0N	MIPI_DSI1_D0P	VSS	AVDD33_HDMI	AVDD33_HDMI	AVDD09_HDMI	AVDD09_HDMI	VSS	VSS	VSS	MMC1_VCC_CAP	GPIO2_VCC_CAP	VSS	VSS	VSS	VSS	VSS	VCC18_GPIO	GPIO_26	GPIO_27	VSS	GPIO_28	GPIO_115
AA	PRI_TCK	PRI_TDO	VSS	VSS	VSS	VSS	AVSS_HDMI	HDMI_TX2N	AVDD18_HDMI	AVDD18_HDMI	VSS	VSS	VCC1833_GPIO3	VCC1833_MMC1	VCC1833_GPIO2	MMC1_DAT2	VSS	VSS	GPIO_32	GPIO_29	VCC18_GPIO	GPIO_21	GPIO_24	GPIO_23	GPIO_25	VSS
AB	PRI_TDI	PRI_TMS	VSS	HDMI_TXCN	HDMI_TX0N	AVSS_HDMI	HDMI_TX1N	HDMI_TX2P	AVSS_HDMI	VSS	VSS	VSS	GPIO_51	MMC1_DAT0	GPIO_78	GPIO_77	GPIO_02	GPIO_03	VSS	VSS	GPIO_41	GPIO_44	GPIO_19	VSS	GPIO_20	GPIO_22
AC	GPIO_61	GPIO_62	VCC18_GPIO	HDMI_TXCP	HDMI_TX0P	AVSS_HDMI	HDMI_TX1P	AVSS_HDMI	AVSS_HDMI	GPIO_86	VCC18_GPIO	GPIO_52	GPIO_47	VCC18_GPIO	GPIO_79	VSS	GPIO_05	GPIO_00	VSS	GPIO_31	GPIO_34	GPIO_42	GPIO_43	GPIO_17	VSS	GPIO_18
AD	GPIO_59	GPIO_60	VSS	VSS	VSS	VSS	VSS	GPIO_87	GPIO_85	PMIC_INT_N	VCC18_GPIO	GPIO_50	GPIO_48	MMC1_CMD	GPIO_76	VSS	GPIO_04	GPIO_01	GPIO_30	GPIO_33	VCC18_GPIO	VCC18_GPIO	GPIO_14	GPIO_12	GPIO_16	GPIO_15
AE	VSS	MPLL_TST_AD	VSS	GPIO_92	GPIO_90	GPIO_91	GPIO_89	GPIO_84	GPIO_81	DVL0	PWR_SCL	EXT_32K_IN	VSS	MMC1_CLK	MMC1_DAT3	GPIO_75	GPIO_11	GPIO_07	GPIO_10	GPIO_37	GPIO_35	GPIO_38	GPIO_46	VSS	GPIO_13	VSS
AF	VSS	VSS	RESET_IN_N	JTAG_SEL	VSS	GPIO_88	GPIO_82	GPIO_83	DVL1	VSS	SLEEP_OUT	PWR_SDA	GPIO_49	MMC1_DAT1	VSS	GPIO_80	GPIO_08	GPIO_06	GPIO_09	VSS	GPIO_40	GPIO_36	GPIO_39	GPIO_45	VSS	VSS

> **Note.** Definition of symbols used for pin type:
>
> - AO = Analog output
> - AI = Analog input
> - AIO = Analog input/output
> - G = Ground
> - I/O = Input/Output
> - P = Power
> - RO = Reference output

Pin ID	Name	Type	Power Domain	Function
P14	VSS	G	0V	Digital Core Ground
P15	VCC_M1	P	0.9V	Digital Core power

P16	VSS	G	0V	Digital Core Ground
P17	AUD_GNDSNS	G	0V	Headphone sense_Ground
P18	AVDD18_AUD	P	1.8V	1.8V power for audio
P19	AVDD18_AUD	P	1.8V	1.8V power for audio
P20	DACR_N	AO	1.8V	Audio DAC R-channel output -N
P21	DACL_N	AO	1.8V	Audio DAC L-channel output -N
P22	DACL_P	AO	1.8V	Audio DAC L-channel output -P
P23	LINE_R	AI	1.8V	Linein R-channel
P24	LINE_L	AI	1.8V	Linein L-channel
P25	MAIN_MICN	AI	1.8V	Board main -mic input -N
P26	MAIN_MICP	AI	1.8V	Board main -mic input -P

R14	VCC_M1	P	0.9V	Digital Core power
R15	VSS	G	0V	Digital Core Ground
R16	VCC_M1	P	0.9V	Digital Core power
R17	AUD_VSSU	G	0V	Audio Ground
R18	AUD_VDDU09	P	0.9V	0.9V power for audio
R19	AUD_REFGND	G	0V	Audio Reference Ground
R20	DACR_P	AO	1.8V	Audio DAC R-channel output -P
R21	AUD_AUREF10	RO	1.8V	Audio reference voltage
R22	AUX_MICN	AI	1.8V	Board aux -mic input -N
R23	AUX_MICP	AI	1.8V	Board aux -mic input -P
R24	VSS	G	0V	Digital core ground
R25	HP_MICP	AI	1.8V	Headset

				-mic input -N
R26	HP_MICN	AI	1.8V	Headset -mic input -P
T14	VSS	G	0V	Digital Core ground
T15	VCC_M1	P	0.9V	Digital Core power
T16	VSS	G	0V	Digital Core ground
T17	VCC_M1	P	0.9V	Digital Core power
T18	VSS	G	0V	Digital Core ground
T19	VSS	G	0V	Digital Core ground
T20	VSS	G	0V	Digital Core ground
T21	AVSS18_AUD	G	0V	Audio Ground
T22	AVSS18_AUD	G	0V	Audio Ground
T23	MICDET	AI	1.8V	Headset detect
T24	VSS	G	0V	Digital Core ground

T25	MICBIAS_MAIN	AO	3.3V	MicBias voltage for Main mic&AUX mic
T26	VSS	G	0V	Digital core ground
U14	VCC_M1	P	0.9V	Digital Core power
U15	VSS	G	0V	Digital Core ground
U16	VCC_M1	P	0.9V	Digital Core power
U17	VSS	G	0V	Digital Core ground
U18	VCC_M1_FB	P	0.9V	Digital Core power FeedBack
U19	VSS_FB	G	0V	Digital Core ground FeedBack
U20	VSS	G	0V	Digital core ground
U21	GPIO_123	I/O	1.8V	General Purpose I/O 123
U22	GPIO_125	I/O	1.8V	General

				Purpose I/O 125
U23	HSDET	AI	1.8V	plug-in detect
U24	MICBIAS_HP	AO	3.3V	micbias voltage for HP mic
U25	GPIO_126	I/O	1.8V	General Purpose I/O 126
U26	GPIO_127	I/O	1.8V	General Purpose I/O 127
V14	VSS	G	0V	Digital Core ground
V15	VCC_M1	P	0.9V	Digital Core power
V16	VSS	G	0V	Digital Core ground
V17	VCC_M1	P	0.9V	Digital Core power
V18	VSS	G	0V	Digital Core ground
V19	VSS	G	0V	Digital Core ground
V20	VSS	G	0V	Digital Core

				ground
V21	GPIO_121	I/O	1.8V	General Purpose I/O 121
V22	VSS	G	0V	Digital Core ground
V23	GPIO_124	I/O	1.8V	General Purpose I/O 124
V24	GPIO_120	I/O	1.8V	General Purpose I/O 120
V25	VSS	G	0V	Digital Core ground
V26	GPIO_122	I/O	1.8V	General purpose I/O 122
W14	VCC_M1	P	0.9V	Digital Core power
W15	VSS	G	0V	Digital Core ground
W16	VCC_M1	P	0.9V	Digital Core power
W17	VSS	G	0V	Digital Core ground
W18	VCC_M1	P	0.9V	Digital Core

				power
W19	VSS	G	0V	Digital Core ground
W20	VSS	G	0V	Digital Core ground
W21	GPIO_110	I/O	1.8V	General Purpose I/O 110
W22	GPIO_117	I/O	1.8V	General Purpose I/O 117
W23	GPIO_116	I/O	1.8V	General Purpose I/O 116
W24	VSS	G	0V	Digital Core ground
W25	GPIO_119	I/O	1.8V	General Purpose I/O 119
W26	GPIO_118	I/O	1.8V	General Purpose I/O 118
Y14	MMC1_VCC_CAP	RO	1.8V	SD card 1.8V LDO cap
Y15	GPIO2_VCC_CAP	RO	1.8V	GPIO2 1.8V LDO cap

Y16	VSS	G	0V	Digital Core ground
Y17	VSS	G	0V	Digital Core ground
Y18	VSS	G	0V	Digital Core ground
Y19	VSS	G	0V	Digital Core ground
Y20	VSS	G	0V	Digital Core ground
Y21	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
Y22	GPIO_26	I/O	1.8V	General Purpose I/O 26
Y23	GPIO_27	I/O	1.8V	General Purpose I/O 27
Y24	VSS	G	0V	Digital Core ground
Y25	GPIO_28	I/O	1.8V	General Purpose I/O 28
Y26	GPIO_115	I/O	1.8V	General Purpose I/O 115

AA14	VCC1833_MMC1	P	1.8V/3.3V	SD card IO power
AA15	VCC1833_GPIO2	P	1.8V/3.3V	GPIO2 IO power
AA16	MMC1_DAT2	I/O	1.8V/3.3V	SD card data 2
AA17	VSS	G	0V	Digital Core ground
AA18	VSS	G	0V	Digital Core ground
AA19	GPIO_32	I/O	1.8V	General Purpose I/O 32
AA20	GPIO_29	I/O	1.8V	General Purpose I/O 29
AA21	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
AA22	GPIO_21	I/O	1.8V	General Purpose I/O 21
AA23	GPIO_24	I/O	1.8V	General Purpose I/O 24
AA24	GPIO_23	I/O	1.8V	General Purpose I/O

				23
AA25	GPIO_25	I/O	1.8V	General Purpose I/O 25
AA26	VSS	G	0V	Digital Core ground
AB14	MMC1_DAT 0	I/O	1.8V/3.3V	SD card data 0
AB15	GPIO_78	I/O	1.8V/3.3V	General Purpose I/O 78
AB16	GPIO_77	I/O	1.8V/3.3V	General Purpose I/O 77
AB17	GPIO_02	I/O	1.8V	General Purpose I/O 02
AB18	GPIO_03	I/O	1.8V	General Purpose I/O 03
AB19	VSS	G	0V	Digital Core ground
AB20	VSS	G	0V	Digital Core ground
AB21	GPIO_41	I/O	1.8V	General Purpose I/O 41

AB22	GPIO_44	I/O	1.8V	General Purpose I/O 44
AB23	GPIO_19	I/O	1.8V	General Purpose I/O 19
AB24	VSS	G	0V	Digital Core ground
AB25	GPIO_20	I/O	1.8V	General Purpose I/O 20
AB26	GPIO_22	I/O	1.8V	General Purpose I/O 22
AC14	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
AC15	GPIO_79	I/O	1.8V/3.3V	General Purpose I/O 79
AC16	VSS	G	0V	Digital Core ground
AC17	GPIO_05	I/O	1.8V	General Purpose I/O 05
AC18	GPIO_00	I/O	1.8V	General Purpose I/O 00

AC19	VSS	G	0V	Digital Core ground
AC20	GPIO_31	I/O	1.8V	General Purpose I/O 31
AC21	GPIO_34	I/O	1.8V	General Purpose I/O 34
AC22	GPIO_42	I/O	1.8V	General Purpose I/O 42
AC23	GPIO_43	I/O	1.8V	General Purpose I/O 43
AC24	GPIO_17	I/O	1.8V	General Purpose I/O 17
AC25	VSS	G	0V	Digital Core ground
AC26	GPIO_18	I/O	1.8V	General Purpose I/O 18
AD14	MMC1_CMD	I/O	1.8V/3.3V	SD card command
AD15	GPIO_76	I/O	1.8V/3.3V	General Purpose I/O 76

AD16	VSS	G	0V	Digital Core ground
AD17	GPIO_04	I/O	1.8V	General Purpose I/O 04
AD18	GPIO_01	I/O	1.8V	General Purpose I/O 01
AD19	GPIO_30	I/O	1.8V	General Purpose I/O 30
AD20	GPIO_33	I/O	1.8V	General Purpose I/O 33
AD21	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
AD22	VCC18_GPIO	P	1.8V	GPIO1/4/5/PMIC I/O power
AD23	GPIO_14	I/O	1.8V	General Purpose I/O 14
AD24	GPIO_12	I/O	1.8V	General Purpose I/O 12
AD25	GPIO_16	I/O	1.8V	General Purpose I/O

				16
AD26	GPIO_15	I/O	1.8V	General Purpose I/O 15
AE14	MMC1_CLK	I/O	1.8V/3.3V	SD card clock
AE15	MMC1_DAT 3	I/O	1.8V/3.3V	SD card data 3
AE16	GPIO_75	I/O	1.8V/3.3V	General Purpose I/O 75
AE17	GPIO_11	I/O	1.8V	General Purpose I/O 11
AE18	GPIO_07	I/O	1.8V	General Purpose I/O 07
AE19	GPIO_10	I/O	1.8V	General Purpose I/O 10
AE20	GPIO_37	I/O	1.8V	General Purpose I/O 37
AE21	GPIO_35	I/O	1.8V	General Purpose I/O 35
AE22	GPIO_38	I/O	1.8V	General Purpose I/O

				38
AE23	GPIO_46	I/O	1.8V	General Purpose I/O 46
AE24	VSS	G	0V	Digital Core ground
AE25	GPIO_13	I/O	1.8V	General Purpose I/O 13
AE26	VSS	G	0V	Digital Core ground
AF14	MMC1_DAT 1	I/O	1.8V/3.3V	SD card data 1
AF15	VSS	G	0V	Digital Core ground
AF16	GPIO_80	I/O	1.8V/3.3V	General Purpose I/O 80
AF17	GPIO_08	I/O	1.8V	General Purpose I/O 08
AF18	GPIO_06	I/O	1.8V	General Purpose I/O 06
AF19	GPIO_09	I/O	1.8V	General Purpose I/O 09

AF20	VSS	G	0V	Digital Core ground
AF21	GPIO_40	I/O	1.8V	General Purpose I/O 40
AF22	GPIO_36	I/O	1.8V	General Purpose I/O 36
AF23	GPIO_39	I/O	1.8V	General Purpose I/O 39
AF24	GPIO_45	I/O	1.8V	General Purpose I/O 45
AF25	VSS	G	0V	Digital Core ground
AF26	VSS	G	0V	Digital Core ground

4.3 I/O Pin Parameters

4.3.1 For 1.8V I/O Pins

Power Domain	Symbol	Description	Min	Typ	Max
1.8V Input	Vih	High level input	VCC×0.7V	1.8V	VCC+0.2V

	Vil	Low level input	-0.3V	0V	VCCx0.3V
	Rpu	Pull up resister	55kOhm	79K Ohm	121kOhm
	Rpd	Pull down resister	51kOhm	87k Ohm	169kOhm
	Iil	Input leakage current Pad in input mode			10uA
1.8V Ouput	Voh	High level output	VCC-0.2V		
	Vol	Low level output			0.2V
	Iol DCS[1:0]= 00 01 10 11	Low level output current when Vpad=0.2V	 13mA 25mA 37mA 49mA		
	Ioh DCS[1:0]= 00 01 10 11	High level output current when Vpad=VCC-0.2V	 11mA 21mA 32mA 42mA		

4.3.2 For 3.3V I/O Pins

Power Domain	Symbol	Description	Min	Typ	Max
3.3V Input	Vih	High level input	2V		VCC+0.3V
	Vil	Low level input	-0.3V	0V	0.8V
	Rpu	Pull up resister	26kOhm	47kOhm	72kOhm
	Rpd	Pull down resister	27kOhm	54kOhm	267kOhm
	Iil	Input leakage current			10uA
3.3V Ouput	Voh	High level output	2.4V		
	Vol	Low level output			0.4V
	Iol DS[2:0]= 000 001 010 011 100 101 110 111	Low level output current when Vpad=0.4V	 7mA 10mA 14mA 18mA 21mA 24mA 28mA 31mA		
	Ioh DS[2:0]= 000 001	High level output current when Vpad=VCC-0.5V	 7mA 10mA		

	010		13mA		
	011		16mA		
	100		19mA		
	101		23mA		
	110		26mA		
	111		29mA		

4.4 Multiplexed Signal/Pin Functions

The **Function 0** through 7 signals is assigned to the I/O pins of X1.

Most I/O pins of X1 are multi-function allowing them to be configured for one of several available functions using Multi-Function Pin Registers (MFPRs). Additionally, some functions can be configured to be present on several different pins.

The assigned signals are organized by their functions (e.g. power supply, clock, etc.) which are arranged in groups according to their interfaces (e.g. JTAG, SPIx, etc.) as per description in the following subsections (sorted alphabetically for user convenience).

> **Note.** Definition of symbols used for signal/pin type:
>
> - I = Input
> - O = Output
> - I/O = Input/Output
> - OD = Open-Drain
> - RO = Reference output

4.4.1 JTAG

4.4.1.1 Primary

Signal/Pin		Description
Name	**Type**	
PRI_TCK	I	Primary JTAG interface 1 test clock. Used for all transfers on the JTAG test interface.
PRI_TDI	I	Primary JTAG interface 1 test data input. Used to send data from the JTAG controller to the X1 processor. This pin has an internal pullup resistor.
PRI_TDO	O	Primary JTAG Interface 1 test data output Used to return data from the X1 processor to the JTAG controller.
PRI_TMS	I	Primary JTAG Interface 1 test mode select. Used to select the test mode required from the JTAG controller. This pin has an internal pullup resistor.
PRI_TRSTn	I	Primary JTAG Interface 1 test reset. Used for IEEE 1194.1 test reset.
VCXO_OUT	O	24 MHz VCXO output clock
VCXO_REQ	I	OCLK1 request

4.4.1.2 Secondary

Signal/Pin		Description
Name	**Type**	

SEC2_TCK	I	Secondary JTAG Interface 2 test clock. Used for all transfers on the JTAG test interface.
SEC2_TDI	I	Secondary JTAG Interface 2 test data input. Used to send data from the JTAG controller to the X1 processor. This pin has an internal pullup resistor.
SEC2_TDO	O	Secondary JTAG Interface 2 test data output. Used to return data from the X1 processor to the JTAG controller.
SEC2_TMS	I	Secondary JTAG Interface 2 test mode select. Used to select the test mode required from the JTAG controller. This pin has an internal pullup resistor.
SEC2_TRSTn	I	Secondary JTAG Interface 2 test reset. Used for IEEE 1194.1 test reset.

4.4.2 Keypad Controller

Signal/Pin		**Description**
Name	**Type**	
KP_DK[4: 0]	I	Keypad direct key inputs [4: 0]
KP_MKIN[3: 0]	I	Keypad matrix key inputs [3: 0]
KP_MKOUT[3: 0]	O	Keypad matrix key outputs [3: 0]

4.4.3 Miscellaneous

Signal/Pin		Description
Name	**Type**	
MPLL_TST_CK		PLL test pin
MN_CLK_OUT	O	Fractional (M/N) divided clock. Main PMU general purpose M/N fractional clock divider clock output. CLK_REQ must be set as Function 0 and pulled high for the 13 MHz clock to be output on GPIO[122] (MN_CLK_OUT).
Sleep_OUT	O	PMIC sleep setting

4.4.4 SPIx

Signal/Pin		Description
Name	**Type**	
SPIx_FRM	I/O	Synchronous serial port frame 0/2. The serial frame sync can be configured as an output (master mode operation) or an input (slave mode operation).
SPIx_RXD	I	Synchronous serial port receive data 0/2. Serial data latched using the bit clock.

SPIx_SCLK	I/O	Synchronous serial port clock 0/2. The serial bit clock can be configured as an output (master mode operation) or an input (slave mode operation).
SPIx_TXD	O	Synchronous serial port transmit data 0/2. Serial data driven out synchronously with the bit clock.

4.4.5 TWSI

4.4.5.1 Dedicated

Signal/Pin		**Description**
Name	**Type**	
PWR_SDA	I/O	TWSI serial data/address signal
PWR_SCL	I/O	TWSI serial clock line signal

4.4.5.2 Common

Signal/Pin		**Description**
Name	**Type**	
I2Cx_SCL	I/O,OD	TWSIx clock
I2Cx_SDA	I/O,OD	TWSIx data

4.4.6 UARTx

Signal/Pin		Description
Name	**Type**	
UARTx_CTSn	I	UARTx clear-to-send
UARTx_RTSn	O	UARTx request-to-send
UARTx_RXD	I	UARTx receive data
UARTx_TXD	O	UARTx transmit data

4.4.7 USB

Signal/Pin		Description
Name	**Type**	
USBx_N	I/O	USB D±
USBx_P	I/O	
VBUS_ON	I	USB VBUS present indicator

4.5 Multi-Function I/O Pin Assignments

All functions that are assigned to a pin as its primary functions are tabled below.

Group	PadName	DefaultPulling	PadEdgeDetected	Function 0	Function 1	Function 2	Function 3	Function 4	Function 5	Function 6
QSPI	QSPI_DAT3	DOWN	ENABLE	QSPI_DAT[3]/strap[3]	GPIO[98]		UART1_TXD <secure domain>			
	QSPI_DAT2	DOWN	ENABLE	QSPI_DAT[2]/strap[2]	GPIO[99]		UART1_RXD <secure domain>			
	QSPI	DOW	ENA	QSPI_DAT[1]/s	GPIO[100]		UART1_CTS <secure	UART4_TXD		

	_DAT1	N	BLE	trap[1]			domain>			
	QSPI_DAT0	DOWN	ENABLE	QSPI_DAT[0]/strap[0]	GPIO[101]		UART1_RTS <secure domain>	UART4_RXD		
	QSPI_CLK	DOWN	ENABLE	QSPI_CLK	GPIO[102]		UART5_TXD			
	QSPI_CS1	UP	ENABLE	QSPI_CS1	GPIO[103]		UART5_RXD			
SD/MMC	MMC1_DA	UP	ENABLE	MMC1_DAT[3]	R_I2S2_SCLK	SEC2_TMS	UART0_TXD	GPIO[104]	PWM0	

	T3									
	MMC1_DAT2	UP	ENABLE	MMC1_DAT[2]	R_I2S2_LRCK	SEC2_TDI	UART0_RXD	GPIO[105]	PWM1	
	MMC1_DAT1	UP	ENABLE	MMC1_DAT[1]	R_I2S2_TXD	SEC2_TDO		GPIO[106]	PWM2	
	MMC1_DAT0	UP	ENABLE	MMC1_DAT[0]	R_I2S2_RXD	SEC2_TRSTn		GPIO[107]	PWM3	
	MMC1_CMD	UP	ENABLE	MMC1_CMD	UART0_TXD	CPU_SEL	R_UART0_TXD	GPIO[108]	PWM4	

	MMC1_CLK	DOWN	ENABLE	MMC1_CLK	R_I2S2_SYSCLK	SEC2_TCK		GPIO[109]	PWM5	
PMIC	RESET_IN_N	UP	NO	RESET_IN_N						
	EXT_32K_IN	DOWN	NO	EXT_32K_IN						
	PWR_SCL	UP	ENABLE	PWR_SCL	GPIO[93]					
	PW	U	EN	PWR_SD	GPIO[9					

	R_SDA	P	ABLE	A	4]				
	SLEEP_OUT	NO	ENABLE	SLEEP_OUT	GPIO[95]				
	DVL0	DOWN	ENABLE	DVL0	GPIO[96]		VCXO_REQ		
	DVL1	DOWN	ENABLE	DVL1	GPIO[97]	IR_RX	VCXO_OUT		
	PMIC_INT_	UP	ENABLE	PMIC_INT_N					

N									
GPIO[81]	UP	ENABLE	GPIO[81]	R_I2S3_SCLK	UART3_TXD	UART4_CTS_N	MN_CLK	AP_I2C5_SCL	
GPIO[82]	UP	ENABLE	GPIO[82]	R_I2S3_LRCK	UART3_RXD	UART4_RTS_N	UART8_TXD	AP_I2C5_SDA	
GPIO[83]	UP	ENABLE	GPIO[83]	R_I2S3_TXD	UART3_CTS_N	UART4_TXD	UART8_RXD	AP_I2C6_SCL	
GPIO[84]	UP	ENABLE	GPIO[84]	R_I2S3_RXD	UART3_RTS_N	UART4_RXD	AP_I2C2_SCL		
GPIO[85]	UP	ENABLE	GPIO[85]	R_I2S3_SYSCLK	UART6_CTS_N	MN_CLK2	AP_I2C2_SDA		

	GPIO[86]	UP	ENABLE	GPIO[86]	HDMI_TX_HSCL	UART6_TXD	DCLK <SPI_LCD>	UART7_CTS_N		
	GPIO[87]	UP	ENABLE	GPIO[87]	HDMI_TX_HSDA	UART6_RXD	DCX/DOUT1 <SPI_LCD>	UART7_RTS_N		
	GPIO[88]	DOWN	ENABLE	GPIO[88]	HDMI_TX_HCEC	UART7_TXD	DIN <SPI_LCD>	PWM6		
	GPIO[89]	DOWN	ENABLE	GPIO[89]	HDMI_TX_PDP	UART7_RXD	DOUT0 <SPI_LCD>	VCXO_REQ		
	GPIO[90]	DOWN	ENABLE	GPIO[90]/strap[4]		UART6_RTS_N	CS<SPI_LCD>	VCXO_OUT	AP_I2C6_SDA	
	GPI	U	EN	GPIO	MN_C	VCXO_O	DSI_TE	R_I2C0		

	O[91]	P	ABLE	[91]	LK2	UT		_SCL		
	GPIO[92]	UP	ENABLE	GPIO[92]	MN_CLK	PWM7		R_I2C0_SDA		
	JTAG_SEL	DOWN	NO	JTAG_SEL						
GPIO1	GPIO[0]	DOWN	ENABLE	GPIO[0]	GMAC0_RXDV	UART6_TXD	PWM8			
	GPIO[1]	DOWN	ENABLE	GPIO[1]	GMAC0_RX_D0	UART6_RXD	PWM9			
	GPIO[	DOW	ENA	GPIO[2]	GMAC0_RX_D1	UART6_CTS_N	PWM10			

	2]	N	BLE							
	GPIO[3]	DOWN	ENABLE	GPIO[3]	GMAC0_RX_CLK	UART6_RTS_N	PWM11			
	GPIO[4]	DOWN	ENABLE	GPIO[4]	GMAC0_RX_D2	UART7_TXD	PWM12			
	GPIO[5]	DOWN	ENABLE	GPIO[5]	GMAC0_RX_D3	UART7_RXD	PWM13			
	GPIO[6]	DOWN	ENABLE	GPIO[6]	GMAC0_TX_D0	UART7_CTS_N	PWM14			
	GPIO[7]	DOWN	ENABL	GPIO[7]	GMAC0_TX_D1	UART7_RTS_N	PWM15			

		E							
GPIO[8]	DOWN	ENABLE	GPIO[8]	GMAC0_TX	UART8_TXD	–			
GPIO[9]	DOWN	ENABLE	GPIO[9]	GMAC0_TX_D2	UART8_RXD	PWM16			
GPIO[10]	DOWN	ENABLE	GPIO[10]	GMAC0_TX_D3	UART8_CTS_N	PWM17			
GPIO[11]	DOWN	ENABLE	GPIO[11]	GMAC0_TX_EN	UART8_RTS_N	PWM18			
GPIO[12]	DOWN	ENABLE	GPIO[12]	GMAC0_MDC	UART9_TXD	VCXO_OUT			

	GPIO[13]	DOWN	ENABLE	GPIO[13]	GMAC0_MDIO	UART9_RXD	PWM19			
	GPIO[14]	DOWN	ENABLE	GPIO[14]	GMAC0_INT_N		PWM0			
	GPIO[15]	UP	ENABLE	GPIO[15]	MMC2_DATA3	PCIe0_PERSTN		PCIe1_PERSTN		
	GPIO[16]	UP	ENABLE	GPIO[16]	MMC2_DATA2	PCIe0_WAKEN	VCXO_REQ	PCIe1_WAKEN		
	GPIO[17]	UP	ENABLE	GPIO[17]	MMC2_DATA1	PCIe0_CLKREQN	VCXO_OUT	PCIe1_CLKREQN		
	GPI	U	EN	GPIO	MMC2_DATA	UART3_T		PCIe2_PERST		

O[18]	P	ABLE	[18]	0	XD		N		
GPIO[19]	UP	ENABLE	GPIO[19]	MMC2_CMD	UART3_RXD		PCIe2_WAKEN		
GPIO[20]	UP	ENABLE	GPIO[20]	MMC2_CLK	UART3_CTS_N	MN_CLK	PCIe2_CLKREQN		
GPIO[21]	DOWN	ENABLE	GPIO[21]	UART2_TXD	UART3_RTS_N	32K_OUT			
GPIO[22]	DOWN	ENABLE	GPIO[22]	UART2_RXD	PWM2		PWM0		
GPIO[23	DOWN	ENAB	GPIO[23]	UART2_CTS_N	UART4_TXD	MN_CLK	PWM1		

]		LE							
GPIO[24]	DOWN	ENABLE	GPIO[24]	UART2_RTS_N	UART4_RXD	I2S1_SYSCLK	PWM2		
GPIO[25]	DOWN	ENABLE	GPIO[25]	I2S1_SCLK	UART5_TXD		PWM3		
GPIO[26]	DOWN	ENABLE	GPIO[26]	I2S1_LRCK	UART5_RXD				
GPIO[27]	DOWN	ENABLE	GPIO[27]	I2S1_TXD	UART5_CTS_N				
GPIO[28]	DOWN	ENABLE	GPIO[28]	I2S1_RXD	UART5_RTS_N		32K_OUT		

	GPIO[29]	DOWN	ENABLE	GPIO[29]	GMAC1_RXDV	UART1_TXD <secure domain>	PWM1	PCIe0_PERSTN		
	GPIO[30]	DOWN	ENABLE	GPIO[30]	GMAC1_RX_D0	UART1_RXD <secure domain>	PWM2	PCIe0_WAKEN		
	GPIO[31]	DOWN	ENABLE	GPIO[31]	GMAC1_RX_D1	UART1_CTS_N <secure domain>	32K_OUT	PCIe0_CLKREQN		
	GPIO[32]	DOWN	ENABLE	GPIO[32]	GMAC1_RX_CLK	UART1_RTS_N <secure domain>	MN_CLK	PCIe1_PERSTN		
	GPIO[33]	DOWN	ENABLE	GPIO[33]	GMAC1_RX_D2	UART4_TXD	PWM3	PCIe1_WAKEN		
	GPI	DO	EN	GPIO	GMAC1_RX_	UART4_	PWM4	PCIe1_CLKRE		

	O[34]	WN	ABLE	[34]	D3	RXD		QN		
	GPIO[35]	DOWN	ENABLE	GPIO[35]	GMAC1_TX_D0	UART4_CTS_N	PWM5	PCIe2_PERSTN		
	GPIO[36]	DOWN	ENABLE	GPIO[36]	GMAC1_TX_D1	UART4_RTS_N	PWM6	PCIe2_WAKEN		
	GPIO[37]	DOWN	ENABLE	GPIO[37]	GMAC1_TX	PWM7	–	PCIe2_CLKREQN		
	GPIO[38]	UP	ENABLE	GPIO[38]	GMAC1_TX_D2	AP_I2C3_SCL <secure domain>	R_I2S3_SCLK	PWM8		
	GPIO[39	UP	ENAB	GPIO[39]	GMAC1_TX_D3	AP_I2C3_SDA <secure domain>	R_I2S3_LRCK	PWM9		

]		LE							
GPIO[40]	UP	ENABLE	GPIO[40]	GMAC1_TX_EN	AP_I2C4_SCL	R_I2S3_TXD	PWM10		
GPIO[41]	UP	ENABLE	GPIO[41]	GMAC1_MDC	AP_I2C4_SDA	R_I2S3_RXD	PWM11		
GPIO[42]	DOWN	ENABLE	GPIO[42]	GMAC1_MDIO	UART5_TXD	R_I2S3_SYSCLK	PWM12		
GPIO[43]	DOWN	ENABLE	GPIO[43]	GMAC1_INT_N	UART5_RXD		PWM13		
GPIO[44]	DOWN	ENABLE	GPIO[44]	MN_CLK	UART5_CTS_N	R_IR_RX	PWM14		

	GPIO[45]	DOWN	ENABLE	GPIO[45]	GMAC0_CLK_REF	UART5_RTS_N		PWM15		
	GPIO[46]	DOWN	ENABLE	GPIO[46]	GMAC1_CLK_REF			PWM16		
	GPIO[110]	DOWN	ENABLE	GPIO[110]	R_CAN_TX0	R_UART1_TXD	UART9_CTS_N	PCIe0_PERSTN	ONE_WIRE	
	GPIO[115]	DOWN	ENABLE	GPIO[115]	R_CAN_RX0	R_UART1_RXD	UART9_RTS_N	PCIe0_WAKEN		
	GPIO[116]	DOWN	ENABLE	GPIO[116]	R_PWM1	R_UART1_CTS_N	UART9_TXD	PCIe0_CLKREQN	VCXO_REQ[1]	
	GPI	DO	EN	GPIO	R_PW	R_UART	UART9_	PCIe2_CLKRE	VCXO	

	O[117]	WN	ABLE	[117]	M2	1_RTS_N	RXD	QN	_CLK_OUT	
	GPIO[118]	UP	ENABLE	GPIO[118]	AP_I2C7_SCL (CAM)	AP_I2C6_SCL	I2S0_SCLK	R_PWM8	KP_MKIN[0]	
	GPIO[119]	UP	ENABLE	GPIO[119]	AP_I2C7_SDA (CAM)	AP_I2C6_SDA	I2S0_LRCK	R_PWM9	KP_MKOUT[0]	
	GPIO[120]	DOWN	ENABLE	GPIO[120]	CAM_MCLK2		I2S0_TXD	R_PWM6	KP_MKIN[1]	
	GPIO[121]	DOWN	ENABLE	GPIO[121]	CAMERA2_RST	VBUS_ON2	I2S0_RXD	R_PWM7	KP_MKOUT[1]	
	GPIO[12	DOWN	ENAB	GPIO[122]	CAMERA2_PDN	USB_ID2	I2S0_SYSCLK		KP_MKIN[2]	

2]		LE							
GPIO[123]	DOWN	ENABLE	GPIO[123]	DRIVE_VBUS2_ISO	KP_DKIN[0]	KP_MKIN[0]			
GPIO[124]	DOWN	ENABLE	GPIO[124]	DRIVE_VBUS1_ISO	KP_DKIN[1]	KP_MKOUT[0]			
GPIO[125]	DOWN	ENABLE	GPIO[125]	VBUS_ON0	KP_DKIN[2]	KP_MKIN[1]			
GPIO[126]	DOWN	ENABLE	GPIO[126]	USB_ID0	KP_DKIN[3]	KP_MKOUT[1]			
GPIO[127]	DOWN	ENABLE	GPIO[127]	DRIVE_VBUS0_ISO	KP_DKIN[4]	KP_MKIN[2]			

GPIO2	GPIO[75]	UP	ENABLE	GPIO[75]	SPI2_SCLK <secure domain>	SPI3_SCLK	CAN_TX0	UART8_TXD	AP_I2C4_SCL	
	GPIO[76]	UP	ENABLE	GPIO[76]	SPI2_FRM <secure domain>	SPI3_FRM	CAN_RX0	UART8_RXD	AP_I2C4_SDA	
	GPIO[77]	UP	ENABLE	GPIO[77]	SPI2_TXD <secure domain>	SPI3_TXD	AP_I2C3_SCL <secure domain>	UART8_CTS_N	R_PWM0	KP_MKOUT[2]
	GPIO[78]	UP	ENABLE	GPIO[78]	SPI2_RXD <secure domain>	SPI3_RXD	AP_I2C3_SDA <secure domain>	UART8_RTS_N	R_PWM1	KP_MKIN[3]
	GPIO[79]	DOWN	ENABLE	GPIO[79]	IR_RX	R_PWM2				KP_MKOUT[3]
	GPI	DO	EN	GPIO	MMC_Card_d	R_PWM3	UART0_	R_UAR		

	O[80]	WN	ABLE	[80]	etect		RXD	T0_RXD		
GPIO3	GPIO[47]	UP	ENABLE	GPIO[47]	R_UART0_TXD	R_CAN_TX0	R_PWM8	AP_I2C3_SCL<secure domain>	ONE_WIRE	
	GPIO[48]	UP	ENABLE	GPIO[48]	R_UART0_RXD	R_CAN_RX0	R_IR_RX	AP_I2C3_SDA<secure domain>	KP_MKOUT[2]	
	GPIO[49]	UP	ENABLE	GPIO[49]	R_SPI_SCLK	R_UART1_CTS_N	R_PWM4	R_I2C0_SCL	KP_MKIN[3]	
	GPIO[50]	UP	ENABLE	GPIO[50]	R_SPI_FRM	R_UART1_RTS_N	R_PWM5	R_I2C0_SDA	KP_MKOUT[3]	
	GPIO[51	UP	ENAB	GPIO[51]	R_SPI_TXD	R_UART1_TXD	R_PWM6	AP_I2C4_SCL		

	]		LE							
	GPIO[52]	UP	ENABLE	GPIO[52]	R_SPI_RXD	R_UART1_RXD	R_PWM7	AP_I2C4_SDA		
GPIO4	GPIO[53]	DOWN	ENABLE	GPIO[53]	CAM_MCLK0	PWM17	PCIe0_CLKREQN	UART3_TXD		
	GPIO[54]	UP	ENABLE	GPIO[54]	AP_I2C0_SCL (CAM)	CAN_TX0	PCIe0_PERSTN	UART3_RXD	AP_I2C5_SCL	
	GPIO[55]	UP	ENABLE	GPIO[55]	AP_I2C0_SDA (CAM)	CAN_RX0	PCIe0_WAKEN	UART3_CTS_N	AP_I2C5_SDA	
	GPIO[56]	UP	ENABLE	GPIO[56]	AP_I2C1_SCL (CAM)	UART6_TXD	PCIe1_PERSTN	UART3_RTS_N	AP_I2C6_SCL	

	GPIO[57]	UP	ENABLE	GPIO[57]	AP_I2C1_SDA (CAM)	UART6_RXD	PCIe1_WAKEN	PWM18	AP_I2C6_SDA	
	GPIO[58]	DOWN	ENABLE	GPIO[58]	CAM_MCLK1	I2S0_SYSCLK	PCIe1_CLKREQN	IR_RX		
	GPIO[111]	DOWN	ENABLE	GPIO[111]	CAMERA0_RST	I2S0_SCLK	PCIe2_PERSTN	UART4_TXD		
	GPIO[112]	DOWN	ENABLE	GPIO[112]	CAMERA1_RST	I2S0_LRCK	PCIe2_WAKEN	UART4_RXD		
	GPIO[113]	DOWN	ENABLE	GPIO[113]	CAMERA0_PDN	I2S0_TXD	PCIe2_CLKREQN	UART4_CTS_N		
	GPI	DO	EN	GPIO	CAMERA1_P	I2S0_RX	DSI_TE	UART4_		

	O[114]	WN	ABLE	[114]	DN	D		RTS_N		
	GPIO[63]	DOWN	ENABLE	GPIO[63]	DRIVE_VBUS0_ISO	R_I2S2_SYSCLK		PWM19	KP_DKIN[0]	
	GPIO[64]	DOWN	ENABLE	GPIO[64]	VBUS_ON0	R_I2S2_SCLK	SPI2_SCLK <secure domain>	R_PWM0	KP_DKIN[1]	
	GPIO[65]	UP	ENABLE	GPIO[65]	USB_ID0	R_I2S2_LRCK	SPI2_FRM <secure domain>	R_PWM1	KP_DKIN[2]	
	GPIO[66]	DOWN	ENABLE	GPIO[66]	DRIVE_VBUS1_ISO	R_I2S2_TXD	SPI2_TXD <secure domain>	R_PWM2	KP_DKIN[3]	
	GPIO[67	DOWN	ENAB	GPIO[67]	DRIVE_VBUS2_ISO	R_I2S2_RXD	SPI2_RXD <secure domain>	R_PWM3	KP_DKIN[4]	

	]		LE							
	GPIO[68]	DOWN	ENABLE	GPIO[68]	VBUS_ON2	UART0_TXD	AP_I2C2_SCL	R_PWM4		
	GPIO[69]	UP	ENABLE	GPIO[69]	USB_ID2	UART0_RXD	AP_I2C2_SDA	R_PWM5		
GPIO5	GPIO[59]	UP	ENABLE	GPIO[59]	HDMI_TX_HSCL	SPI3_SCLK	UART1_TXD <secure domain>	PCIe1_PERSTN		
	GPIO[60]	UP	ENABLE	GPIO[60]	HDMI_TX_HSDA	SPI3_FRM	UART1_RXD <secure domain>	PCIe1_WAKEN		
	GPIO[61]	UP	ENABLE	GPIO[61]	HDMI_TX_HCEC	SPI3_TXD	UART1_CTS_N <secure domain>	PCIe1_CLKREQN		

	GPIO[62]	UP	ENABLE	GPIO[62]	HDMI_TX_PDP	SPI3_RXD	UART1_RTS_N <secure domain>	PCIe2_PERSTN		
	PRI_TDI	UP	NO	PRI_TDI	GPIO[70]	AP_I2C2_SCL	DCLK <SPI_LCD>	UART5_TXD		
	PRI_TMS	UP	NO	PRI_TMS	GPIO[71]	AP_I2C2_SDA	DCX/DOUT1 <SPI_LCD>	UART5_RXD		
	PRI_TCK	DOWN	NO	PRI_TCK	GPIO[72]	UART9_TXD	DIN<SPI_LCD>	UART5_CTS_N		
	PRI_TDO	UP	NO	PRI_TDO	GPIO[73]	UART9_RXD	DOUT0 <SPI_LCD>	UART5_RTS_N		
	PRI_	UP	NO	PRI_TRS						

	TRSTn			Tn						
	GPIO[74]	UP	ENABLE	GPIO[74]		PWM9	CS<SPI_LCD>	PCIe2_WAKEN		
EMMC5.1	EMMC_D0			EMMC_D0	GPIO[93]					
	EMMC_D1			EMMC_D1	GPIO[94]					
	EMMC_D2			EMMC_D2	GPIO[95]					

	EMMC_D3			EMMC_D3	GPIO[96]					
	EMMC_D4			EMMC_D4	GPIO[97]					
	EMMC_D5			EMMC_D5	GPIO[98]					
	EMMC_D6			EMMC_D6	GPIO[99]					
	EMMC_			EMMC_D7	GPIO[100]					

	D7									
	EMMC_DS			EMMC_DS	GPIO[101]					
	EMMC_CLK			EMMC_CLK	GPIO[102]					
	EMMC_CMD			EMMC_CMD	GPIO[103]					

4.6 Power Supply Pins

Pin Name	Domain Name	Domain Voltage	Description
AUD_VDDU09	AUDIO	0.9V	0.9V power for audio

AUD_VNEG	AUDIO	-1.8V	Negative voltage for headphone driver
AUD_VPOS	AUDIO	1.8V	Positive voltage for headphone driver
AVDD18_AUD	AUDIO	1.8V	1.8V power for audio
AVDD3V3_AUD	AUDIO	3.3V	3.3V power for earphone driver
VCC_M1	CORE	0.9V	Digital core power
AVDD09_CSI	CSI	0.9V	MIPI_CSI digital power
AVDD18_CSI	CSI	1.8V	MIPI_CSI analog power
AVDD09_AFEAP	DCXO	0.9V	0.9V power for DCXO
AVDD18_AFEAP	DCXO	1.8V	1.8V power for DCXO
AVDD06_DDR	DDR	lp4x: 0.6V lp4: TBD lp3: TBD	LPDDR4X IO power
AVDD11_DDR	DDR	lp4x:1.1V lp4:1.1V lp3: 1.2V	LPDDR PHY power supply
AVDD18_DDR	DDR	1.8V	LPDDR PHY PLL 1.8V power

AVDD18_PHY	DDR	1.8V	Analog 1.8V power
AVDDU_DDR	DDR	0.9V	LPDDR PHY PLL logical power
AVDDU_PHY	DDR	0.9V	LPDDR PHY core logical power
DDR_LDO_CAP	DDR	0.7~0.9V	External LDO output ball. Connect to a 100nF capacitor on PCB board.
DDR_LP23_VREFCA	DDR	lp3:0.6V lp4: high-z	CA VREF for lpddr23. LP4/4x, Keep the pin NC.
DDR_LP23_VREFDQ	DDR	lp3: 0.6V lp4: high-z	DQ VREF for lpddr23. LP4/4x, keep the pin NC.
VDDQ_V1P2	DDR	lp3: 1.2V lp4x: 0.6V	LPDDR3 IO power
AVDD09_DSI1	DSI	0.9V	DSI digital power
AVDD12_DSI1	DSI	1.2V	DSI driver power
AVDD18_DSI1	DSI	1.8V	DSI analog power
AVDD18_EFUSE	EFUSE	1.8V	ANAGRP

AVDD09_EMMC	EMMC	0.9V	eMMC digital power
AVDD18_EMMC	EMMC	1.8V	eMMC analog power
VCC18_GPIO	GPIO1/4/5/PMIC	1.8V	GPIO1/4/5/PMIC I/O power
VCC1833_GPIO2	GPIO2	1.8V/3.3V	GPIO2 IO power
VCC1833_GPIO3	GPIO3	1.8V/3.3V	GPIO3 IO power
AVDD09_HDMI	HDMI	0.9V	HDMI digital power
AVDD18_HDMI	HDMI	1.8V	HDMI 1.8V power
AVDD33_HDMI	HDMI	3.3V	HDMI 3.3V power
AVDD09_PCIEA	PCIEA	0.9V	PCIEA digital power
AVDD18_PCIEA	PCIEA	1.8V	PCIEA analog power
AVDD09_PCIEB	PCIEB	0.9V	PCIEB digital power
AVDD18_PCIEB	PCIEB	1.8V	PCIEB analog power
AVDD09_	PCIEC	0.9V	PCIEC digital power

PCIEC			
AVDD18_PCIEC	PCIEC	1.8V	PCIEC analog power
AVDD09_PLL	PLL	0.9V	System PLL power supply
AVDD18_PLL	PLL	1.8V	System PLL power supply
VCC1833_QSPI	QSPI	1.8V/3.3V	QSPI IO power
VCC1833_MMC1	SD card	1.8V/3.3V	SD card IO power
AVDD09_USB	USB2.0	0.9V	USB2.0 digital power
AVDD18_USB	USB2.0	1.8V	USB2.0 1.8V power
AVDD33_USB	USB2.0	3.3V	USB2.0 3.3V power

4.7 Multi-Function Pin Registers (MFPRs)

In X1 are defined and implemented Multi-Function Pin Registers (MFPRs). In particular, there are 129 MFPR in total, starting from the base address 0xD401_E000 with a stride of 0x4, as tabled below.

MFPR ID	**Address**	**Offset**
GPIO_00	0xD401E004	0x4
GPIO_01	0xD401E008	0x8

GPIO_02	0xD401E00C	0xC
GPIO_03	0xD401E010	0x10
GPIO_04	0xD401E014	0x14
GPIO_05	0xD401E018	0x18
GPIO_06	0xD401E01C	0x1C
GPIO_07	0xD401E020	0x20
GPIO_08	0xD401E024	0x24
GPIO_09	0xD401E028	0x28
GPIO_10	0xD401E02C	0x2C
GPIO_11	0xD401E030	0x30
GPIO_12	0xD401E034	0x34
GPIO_13	0xD401E038	0x38
GPIO_14	0xD401E03C	0x3C
GPIO_15	0xD401E040	0x40
GPIO_16	0xD401E044	0x44
GPIO_17	0xD401E048	0x48
GPIO_18	0xD401E04C	0x4C
GPIO_19	0xD401E050	0x50
GPIO_20	0xD401E054	0x54

GPIO_21	0xD401E058	0x58
GPIO_22	0xD401E05C	0x5C
GPIO_23	0xD401E060	0x60
GPIO_24	0xD401E064	0x64
GPIO_25	0xD401E068	0x68
GPIO_26	0xD401E06C	0x6C
GPIO_27	0xD401E070	0x70
GPIO_28	0xD401E074	0x74
GPIO_29	0xD401E078	0x78
GPIO_30	0xD401E07C	0x7C
GPIO_31	0xD401E080	0x80
GPIO_32	0xD401E084	0x84
GPIO_33	0xD401E088	0x88
GPIO_34	0xD401E08C	0x8C
GPIO_35	0xD401E090	0x90
GPIO_36	0xD401E094	0x94
GPIO_37	0xD401E098	0x98
GPIO_38	0xD401E09C	0x9C
GPIO_39	0xD401E0A0	0xA0

GPIO_40	0xD401E0A4	0xA4
GPIO_41	0xD401E0A8	0xA8
GPIO_42	0xD401E0AC	0xAC
GPIO_43	0xD401E0B0	0xB0
GPIO_44	0xD401E0B4	0xB4
GPIO_45	0xD401E0B8	0xB8
GPIO_46	0xD401E0BC	0xBC
GPIO_47	0xD401E0C0	0xC0
GPIO_48	0xD401E0C4	0xC4
GPIO_49	0xD401E0C8	0xC8
GPIO_50	0xD401E0CC	0xCC
GPIO_51	0xD401E0D0	0xD0
GPIO_52	0xD401E0D4	0xD4
GPIO_53	0xD401E0D8	0xD8
GPIO_54	0xD401E0DC	0xDC
GPIO_55	0xD401E0E0	0xE0
GPIO_56	0xD401E0E4	0xE4
GPIO_57	0xD401E0E8	0xE8
GPIO_58	0xD401E0EC	0xEC

GPIO_59	0xD401E0F0	0xF0
GPIO_60	0xD401E0F4	0xF4
GPIO_61	0xD401E0F8	0xF8
GPIO_62	0xD401E0FC	0xFC
GPIO_63	0xD401E100	0x100
GPIO_64	0xD401E104	0x104
GPIO_65	0xD401E108	0x108
GPIO_66	0xD401E10C	0x10C
GPIO_67	0xD401E110	0x110
GPIO_68	0xD401E114	0x114
GPIO_69	0xD401E118	0x118
PRI_TDI	0xD401E11C	0x11C
PRI_TMS	0xD401E120	0x120
PRI_TCK	0xD401E124	0x124
PRI_TDO	0xD401E128	0x128
GPIO_74	0xD401E12C	0x12C
GPIO_75	0xD401E130	0x130
GPIO_76	0xD401E134	0x134
GPIO_77	0xD401E138	0x138

GPIO_78	0xD401E13C	0x13C
GPIO_79	0xD401E140	0x140
GPIO_80	0xD401E144	0x144
GPIO_81	0xD401E148	0x148
GPIO_82	0xD401E14C	0x14C
GPIO_83	0xD401E150	0x150
GPIO_84	0xD401E154	0x154
GPIO_85	0xD401E158	0x158
QSPI_DAT0	0xD401E168	0x168
QSPI_DAT1	0xD401E16C	0x16C
QSPI_DAT2	0xD401E170	0x170
QSPI_DAT3	0xD401E174	0x174
QSPI_CS1	0xD401E178	0x178
QSPI_CLK	0xD401E17C	0x17C
MMC1_DAT3	0xD401E1B8	0x1B8
MMC1_DAT2	0xD401E1BC	0x1BC
MMC1_DAT1	0xD401E1C0	0x1C0
MMC1_DAT	0xD401E1C4	0x1C4

0		
MMC1_CMD	0xD401E1C8	0x1C8
MMC1_CLK	0xD401E1CC	0x1CC
GPIO_110	0xD401E1D0	0x1D0
PWR_SCL	0xD401E1D4	0x1D4
PWR_SDA	0xD401E1D8	0x1D8
VCXO_EN	0xD401E1DC	0x1DC
DVL0	0xD401E1E0	0x1E0
DVL1	0xD401E1E4	0x1E4
PMIC_INT_N	0xD401E1E8	0x1E8
GPIO_86	0xD401E1EC	0x1EC
GPIO_87	0xD401E1F0	0x1F0
GPIO_88	0xD401E1F4	0x1F4
GPIO_89	0xD401E1F8	0x1F8
GPIO_90	0xD401E1FC	0x1FC
GPIO_91	0xD401E200	0x200
GPIO_92	0xD401E204	0x204
GPIO_111	0xD401E20C	0x20C

GPIO_112	0xD401E210	0x210
GPIO_113	0xD401E214	0x214
GPIO_114	0xD401E218	0x218
GPIO_115	0xD401E21C	0x21C
GPIO_116	0xD401E220	0x220
GPIO_117	0xD401E224	0x224
GPIO_118	0xD401E228	0x228
GPIO_119	0xD401E22C	0x22C
GPIO_120	0xD401E230	0x230
GPIO_121	0xD401E234	0x234
GPIO_122	0xD401E238	0x238
GPIO_123	0xD401E23C	0x23C
GPIO_124	0xD401E240	0x240
GPIO_125	0xD401E244	0x244
GPIO_126	0xD401E248	0x248
GPIO_127	0xD401E24C	0x24C

4.7.1 MFPR Functional Description

4.7.1.1 I/O PAD Parameter Definition

The input thresholds of Buffer Mode of I/O PADs are tabled below.

ST1:ST0==2'b00				
Input Threshold	**Min**	**Typ**	**Max**	**Unit**
VT	0.75	0.91	1.09	V
VT PU	0.74	0.90	1.08	V
VT PD	0.76	0.92	1.10	V

Instead, the input thresholds of Schmitt Trigger Mode of I/O PADs are tabled below.

ST1:ST0==2'b01				
Input Threshold	**Min**	**Typ**	**Max**	**Unit**
VT+	0.82	0.97	1.13	V
VT-	0.72	0.85	1.02	V
VT+PU	0.81	0.96	1.12	V
VT-PU	0.71	0.84	1.01	V
VT+PD	0.82	0.98	1.14	V
VT-PD	0.73	0.86	1.03	V

ST1:ST0==2'b10/2'b11				
Input Threshold	**Min**	**Typ**	**Max**	**Unit**

VT+	0.87	1.04	1.19	V
VT-	0.69	0.80	0.95	V
VT+PU	0.86	1.03	1.18	V
VT-PU	0.68	0.79	0.94	V
VT+PD	0.88	1.05	1.20	V
VT-PD	0.69	0.81	0.96	V

4.7.1.2 MFPR Field Description

Bit(s)	Field	Type	Reset	Description
31:16	RSVD	RO	0	This field is reserved for future use
15	PULL SEL	RW	0x1	This field selects between two sets of controls for the pull-up and pull-down functionality as follows: • 0: The pull-up and pull-down resistors are controlled by the selected alternate function for the pin ◦ 1: The pull-up and pull-down resistors are controlled by the <PULLUP EN> and <PULLDN EN> fields in this register, overriding the function indicated by the selected alternate function. During low-power states, this field is overridden to 1 and controlled by the <PULLUP EN> and <PULLDN EN> fields.

				In these low-power states, this field is effectively 1, although the register value is not changed (refer to low-power (sleep) mode operation for more information).
14	PULLUP EN	RW	0x0	This field controls the output function while the <PULL SEL> field is set to 1 (or is effectively 1) as follows: • 0: The internal pull-up resistor of the pin is disabled • 1: The internal pull-up resistor of the pin is enabled The address and reset value is on a pin-by-pin basis. Do not rely on the reset value of this field. It must be configured by software to the desired settings.
13	PULLDN EN	RW	0x0	This field controls the output function while <PULL SEL> is set to 1 (or is effectively 1) as follows: • 0: The internal pull-down resistor of the pin is disabled ◦ 1: The internal pull-down resistor of the pin is enabled The address and reset value is on a pin-by-pin basis . Do not rely on the reset value of this field. It must be configured by software to the desired settings.
12:11	DRIVE[1:0]	RW	0x2	This field defines the drive strength and slew rate for this pin (in functional mode when the pin is driving HIGH or LOW value) as follows:

				• 2'b00: SLOW • 2'b01: SLOW ◦ 2'b10: MEDIUM • 2'b11: FAST They are the DS1 and DS0 bit of the drive strength in the current table.
10	DRIVE[2]	RW	0x0	This is the DS2 bit to program for higher level of driving strength in the current table. The address and reset value is on a pin-by-pin basis. Do not rely on the reset value of this field. It must be configured by software to the desired settings. For Medium (all GPIOs except for SD card), it is 010. For Fast (SD card I/O), it is 110.
9:8	ST[1:0]	RW	0x0	This field controls the Schmitt trigger input threshold as follows: ◦ 2'b00: buffer input, threshold is 0.9v • 2'b01/10/11: enabled the Schmitt trigger with larger hysteresis for VT- and VT+ threshold (refer to **Section 4.7.1.1**)
7	SLE	RW	0x0	This field enables/disables the slew rate output control as follows: ◦ 1'b1: Enabled • 1'b0: Disabled Enabling the slew rate output control will slow down the output ramp for EMI considerations.

6	EDGE_CLEAR	RW	0x1	This field enable/disable the edge-detection logic as follows: ◦ 1'b0: Enabled and ready to detect an edge ▪ 1'b1: Disabled and no edge is detected This is an enable for the <EDGE_FALL_EN> and <EDGE_RISE_EN> control fields. This field is only present when a pin has been defined as potentially waking up on an edge. If the device is not configured in this manner, this field is not present (i.e. reserved) and writing to it has no effect (refer to **Section 4.5** for more information about which MFPRs include or not include these bits).
5	EDGE_FALL_EN	RW	0x0	This field enables/disable to detect a falling edge as follows: ▪ 1'b0: Disabled ▪ 1'b1: Enable To detect a falling edge on this pin, ◦ The pin needs not be an output ▪ This field must be set to 1 ◦ The <EDGE_CLEAR> field must be set to 0 This field is only present when a pin has been defined as potentially waking up on an edge. If the device is not configured in this

				manner, this field is not present (i.e. reserved) and writing to it has no effect (refer to **Section 4.5** for more information about which MFPRs include or not include these bits).
4	EDGE_RISE_EN	RW	0x0	This field enables/disable to detect a rising edge as follows: ◦ 1'b0: Disables ◦ 1'b1: Enabled To detect a rising edge on this pin,, • The pin need not be an output ◦ This field must be set to 1 ◦ The <EDGE_CLEAR> field must be set to 0 This field is only present when a pin has been defined as potentially waking up on an edge. If the device is not configured in this manner, this field is not present (i.e. reserved). and writing to it has no effect (refer to **Section 4.5** for more information about which MFPRs include or not include these bits).
3	SPU	RW	0x0	This field enables/disables a strong pull resistor as follows: ◦ 1'b0: Disabled ◦ 1'b1: Enabled This field is used for I2C or SD card PADs which require a strong pull resistor.

2:0	AF SEL	RW	0x0	This field is used for the selection of an alternate function for a pin between eight possible options as follows: • 0x0: Alternate function 0 (always as the primary at reset) ◦ 0x1: Alternate function 1 ◦ 0x2: Alternate function 2 ◦ 0x3: Alternate function 3 • 0x4: Alternate function 4 ◦ 0x5: Alternate function 5 ◦ 0x6: Alternate function 6 • 0x7: Alternate function 7

5. Electrical Characteristics

5.1 Pin AC/DC Operating Conditions

Item	**Symbol/Pin**	**Min**	**Typ**	**Max**	**Unit**	**Note**
Digital Power	VCC_M1	0.85	0.9	1.0	V	
PLL	AVDD09_PLL	0.855	0.9	0.945	V	
	AVDD18_PLL	1.71	1.8	1.89	V	
OSC	AVDD09_AFEAP	0.855	0.9	0.945	V	

	AVDD18_AFEAP	1.71	1.8	1.89	V	
PCIeC	AVDD18_PCIEC	1.71	1.8	1.89	V	
	AVDD09_PCIEC	0.855	0.9	0.945	V	
PCIeB	AVDD18_PCIEB	1.71	1.8	1.89	V	
	AVDD09_PCIEB	0.855	0.9	0.945	V	
PCIeA	AVDD18_PCIEA	1.71	1.8	1.89	V	
	AVDD09_PCIEA	0.855	0.9	0.945	V	
USB IO	AVDD33_USB	3.135	3.3	3.465	V	
USB PHY	AVDD18_USB	1.71	1.8	1.89	V	
	AVDD09_USB	0.855	0.9	0.945	V	
MIPI DSI PHY	AVDD09_DSI1	0.855	0.9	0.945	V	
	AVDD18_DSI1	1.71	1.8	1.89	V	
MIPI DSI IO	AVDD12_	1.14	1.2	1.26	V	

	DSI1					
MIPI CSI PHY	AVDD09_CSI	0.855	0.9	0.945	V	
	AVDD18_CSI	1.71	1.8	1.89	V	
HDMI	AVDD09_HDMI	0.855	0.9	0.945	V	
	AVDD18_HDMI	1.71	1.8	1.89	V	
	AVDD33_HDMI	3.135	3.3	3.465	V	
eMMC	VDD09_EMMC	0.855	0.9	0.945	V	
	V18_EMMC	1.71	1.8	1.89	V	
QSPI	VCC1833_QSPI	1.71	1.8	1.89	V	Dual power domain
		3.135	3.3	3.465	V	
SD	VCC1833_MMC1	1.71	1.8	1.89	V	Dual power domain
		3.135	3.3	3.465	V	
DDR PHY	AVDD18_PHY	1.71	1.8	1.89	V	
	AVDD18_DDR	1.71	1.8	1.89	V	

	AVDD11_DDR	1.045	1.1	1.155	V	LP4/4X
		1.14	1.2	1.26	V	LP3
	AVDDU_PHY	0.855	0.9	0.945	V	
	AVDDU_DDR	0.855	0.9	0.945	V	
DDR IO	AVDD06_DDR	0.57	0.6	0.63	V	
	VDDQ_V1P2	1.14	1.2	1.26	V	
eFuse	AVDD18_EFUSE	1.71	1.8	1.89	V	
Audio Logic	AUD_VDDU09	0.855	0.9	0.945	V	
Audio Power NEG	AUD_VNEG	-1.71	-1.8	-1.89	V	
Audio Power POS	AUD_VPOS	1.71	1.8	1.89	V	
Audio Analog	AVDD18_AUD	1.71	1.8	1.89	V	
	AVDD3V3_AUD	3.135	3.3	3.465	V	
GPIO	VCC18_GPIO	1.71	1.8	1.89	V	

GIOP3	VCC1833_GPIO3	1.71	1.8	1.89	V	Dual power domain
		3.135	3.3	3.465	V	
GIOP2	VCC1833_GPIO2	1.71	1.8	1.89	V	Dual power domain
		3.135	3.3	3.465	V	

5.2 Absolute Max Ratings

5.2.1 For Pins

Item	**Symbol/Pin**	**Min**	**Max**	**Unit**
Digital Power	VCC_M1	-0.1	1.035	V
PLL	AVDD09_PLL	-0.1	1.035	V
	AVDD18_PLL	-0.1	2.07	V
OSC	AVDD09_AFEAP	-0.1	1.035	V
	AVDD18_AFEAP	-0.1	2.07	V
PCIeC	AVDD18_PCIEC	-0.1	2.07	V
	AVDD09_PCIEC	-0.1	1.035	V
PCIeB	AVDD18_PCIEB	-0.1	2.07	V

	AVDD09_PCIEB	-0.1	1.035	V
PCIeA	AVDD18_PCIEA	-0.1	2.07	V
	AVDD09_PCIEA	-0.1	1.035	V
USB IO	AVDD33_USB	-0.1	3.795	V
USB PHY	AVDD18_USB	-0.1	2.07	V
	AVDD09_USB	-0.1	1.035	V
MIPI DSI IO	AVDD12_DSI1	-0.1	1.38	V
MIPI DSI PHY	AVDD09_DSI1	-0.1	1.035	V
	AVDD18_DSI1	-0.1	2.07	V
MIPI CSI PHY	AVDD09_CSI	-0.1	1.035	V
	AVDD18_CSI	-0.1	2.07	V
HDMI	AVDD09_HDMI	-0.1	1.035	V
	AVDD18_HDMI	-0.1	2.07	V
	AVDD33_HDMI	-0.1	3.795	V
eMMC	VDD09_EMMC	-0.1	1.035	V
	V18_EMMC	-0.1	2.07	V
QSPI	VCC1833_QSPI	-0.1	2.07	V
		-0.1	3.795	V

SD	VCC1833_MMC1	-0.1	2.07	V
		-0.1	3.795	V
DDR PHY	AVDD18_PHY	-0.1	2.07	V
	AVDD18_DDR	-0.1	2.07	V
	AVDD11_DDR	-0.1	1.265	V
	AVDD11_DDR	-0.1	1.38	V
	AVDDU_PHY	-0.1	1.035	V
	AVDDU_DDR	-0.1	1.035	V
DDR IO	AVDD06_DDR	-0.1	0.69	V
	VDDQ_V1P2	-0.1	1.38	V
eFuse	AVDD18_EFUSE	-0.1	2.07	V
Audio Logic	AUD_VDDU09	-0.1	1.035	V
Audio Power NEG	AUD_VNEG	N/A	-2.07	V
Audio Power POS	AUD_VPOS	-0.1	2.07	V
Audio Analog	AVDD18_AUD	-0.1	2.07	V
	AVDD3V3_AUD	-0.1	3.795	V
GPIO	VCC18_GPIO	-0.1	2.07	V
GPIO3	VCC1833_GPIO3	-0.1	2.07	V
		-0.1	3.795	V

GPIO2	VCC1833_GPIO2	-0.1	2.07	V
		-0.1	3.795	V

5.2.2 For Packages

Item	Symbol	Min	Max	Unit
Operating Temperature (Industrial Standard)	Ta	-40	+85	°C
Junction Temperature	Tj	N/A	125	℃
Storage Temperature	Tstg	-40	125	℃

5.3 Pin Max Currents

Item	**Symbol/Pin**	**Max**	**Unit**
Digital Power	VCC_M1	10000	mA
PLL	AVDD09_PLL	5	mA
	AVDD18_PLL	5	mA
OSC	AVDD09_AFEAP	5	mA
	AVDD18_AFEAP	5	mA
PCIeC	AVDD18_PCIEC	50	mA
	AVDD09_PCIEC	100	mA

PCIeB	AVDD18_PCIEB	50	mA
	AVDD09_PCIEB	100	mA
PCIeA	AVDD18_PCIEA	50	mA
	AVDD09_PCIEA	100	mA
USB IO	AVDD33_USB	90	mA
USB PHY	AVDD18_USB	90	mA
	AVDD09_USB	15	mA
MIPI DSI PHY	AVDD09_DSI1	20	mA
	AVDD18_DSI1	50	mA
MIPI DSI IO	AVDD12_DSI1	50	mA
MIPI CSI PHY	AVDD09_CSI	70	mA
	AVDD18_CSI	100	mA
HDMI	AVDD09_HDMI	10	mA
	AVDD18_HDMI	10	mA
	AVDD33_HDMI	10	mA
eMMC	VDD09_EMMC	50	mA
	V18_EMMC	50	mA
QSPI	VCC1833_QSPI	150	mA
SD	VCC1833_MMC1	150	mA

DDR PHY	AVDD18_PHY	200	mA
	AVDD18_DDR	20	mA
	AVDD11_DDR	100	mA
	AVDDU_PHY	100	mA
	AVDDU_DDR	100	mA
DDR IO	AVDD06_DDR	100	mA
	VDDQ_V1P2	600	mA
eFuse	AVDD18_EFUSE	150	mA
Audio Logic	AUD_VDDU09	1	mA
Audio Power NEG	AUD_VNEG	102	mA
Audio Power POS	AUD_VPOS	102	mA
Audio Analog	AVDD18_AUD	10	mA
	AVDD3V3_AUD	100	mA

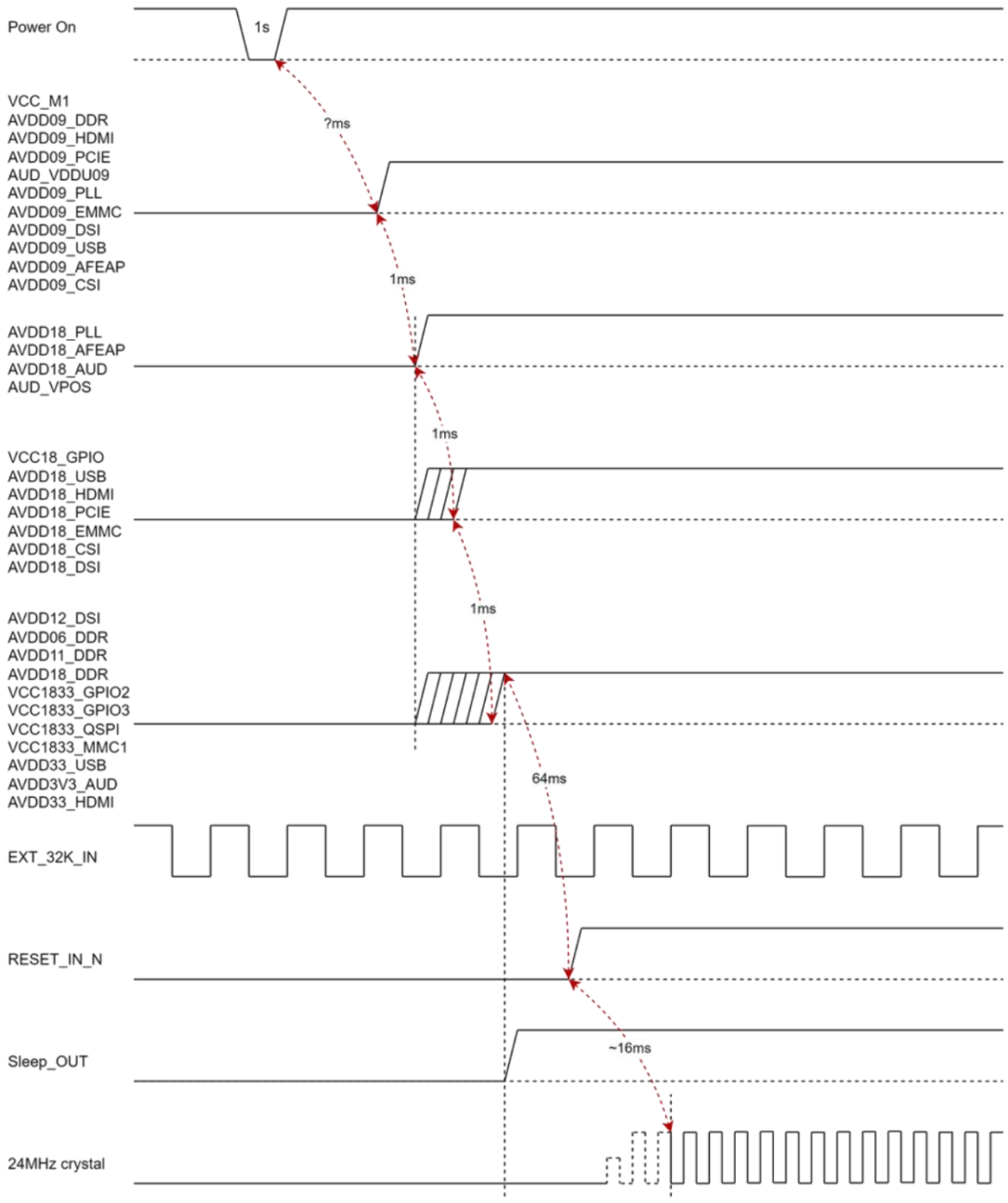
5.4 Power On/Off Sequence

5.4.1 Power On Sequence

- A short pressure (i.e. 1 second) of the power button will turn on the X1 processor automatically if it was off before (cold start)
- The Power Management IC (PMIC) will turn on firstly the core logic then the external I/O to ensure proper initialization
- PMIC will asserts a Power-On-Reset (POR) to initialize the system and ensure a

defined starting state

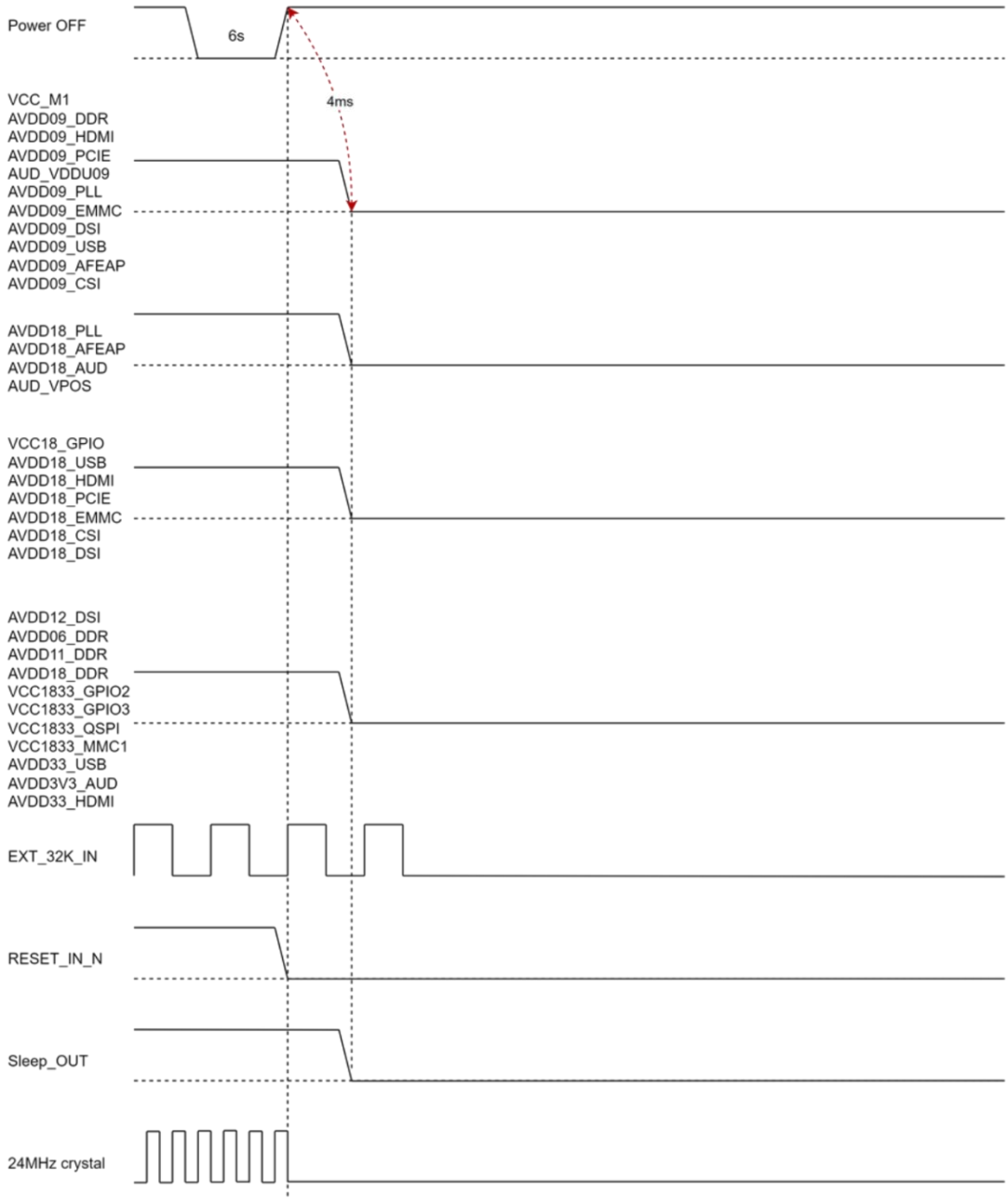
The order of the involved pins with state change during the power on sequence is depicted below.



5.4.2 Power Off Sequence

- A long pressure (i.e. 6 seconds) of the power button will turn off the X1 processor.

The order of the involved pins with state change during the power off sequence is depicted below.



5.5 Power Consumption

5.5.1 In Typical Application Scenarios

To be defined soon.

5.5.2 In Particular Application Scenarios

To be defined soon.