endpoint intelligence

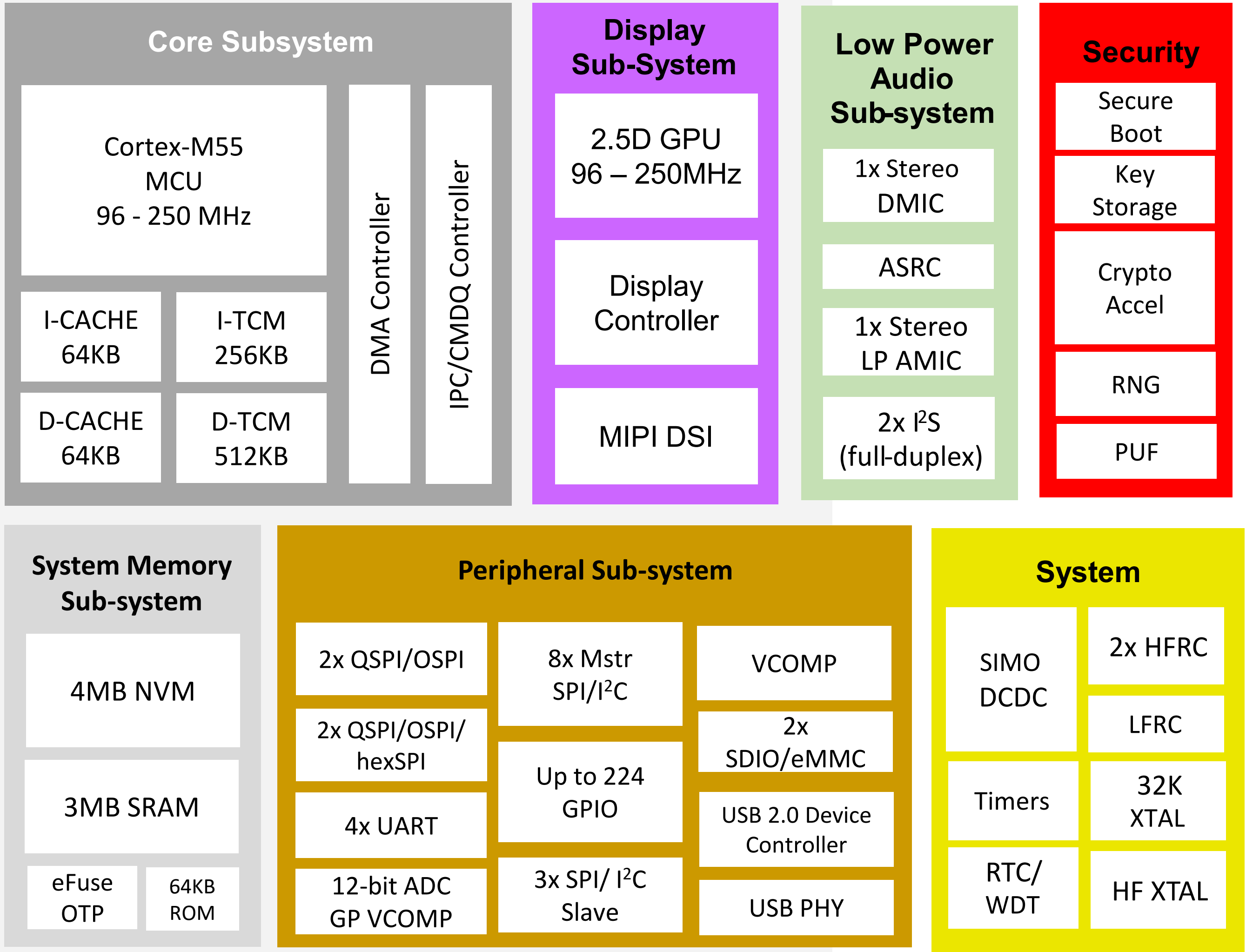
Apollo5 Family Product Training

October 2025

Apollo510 Training Agenda

- ARM M55 CPU Subsystem and Memory Subsystem (CPU, Cache, TCM, Helium, etc)
- Memory Subsystem - ARM Internal Bus (AXI, AHB, AMBA, APB, PPB, etc)
- Cache and Memory Performance
- Clock Trees and Resets
- Simobuck and LDOs
- Power Management
- Peripherals: MSPI, IOS, GPIO, UART, USB, SDIO, GPU, Display
- Security
- MRAM Recovery
- Cortex M55 Profiling
- SDK Overview
- Toolchains
- Customer Hardware

Apollo510



Feature Highlights

- Ultra low power Cortex-M55 processor with Helium vector extensions
  - Up to 250MHz clock frequency with TurboSPOT™
- 64KB 2-way instruction and 64KB 4-way data cache
- 768KB TCM Tightly Coupled Memory to the CPU
- 3.75MB total on-chip SRAM memory, 4MB MRAM NVM
- 250MHz 2.5D graphics accelerator with Vector Graphics
- Enhanced external memory interface supporting up to HexSPI (x16) operating at up to 125MHz DDR (250MT/s)
- Enhanced boot with eFuse and ROM
- MIPI DSI 1.2 with up to two lanes at 768Mbps per lane
- Up to 224 GPIO available
- 2x SDIO controller for concurrent eMMC and SDIO i/f
- Optimized deep sleep (w/RAM retention) power

Apollo510 Enhancements from Apollo4 Family

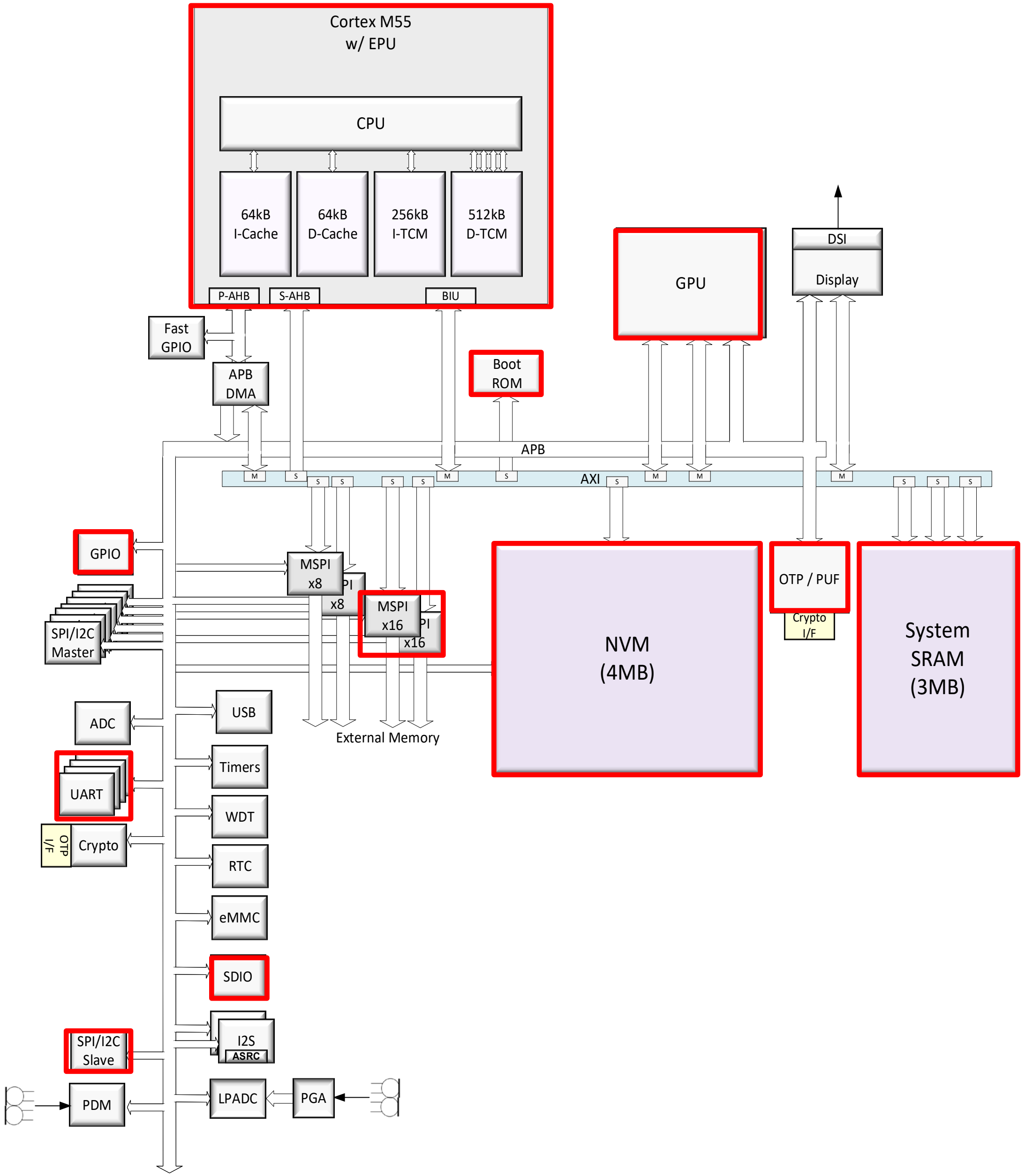
- Cortex-M55 CPU with Helium vector extensions
- 64KB I-Cache / 64-KB D-Cache
- 768KB TCM (256KB ITCM / 512KB DTCM)
- LP: 96MHz / HP: 250MHz

- Enhanced GPU performance
  - LP: 96MHz / HP: 250MHz
  - Variable burst length (up to 64B / 128B)
  - Radial/Conical Gradient Fill
  - Enhanced Vector Graphics HW Acceleration

- Enhanced boot
  - SBR (Secure Boot ROM) in ROM
  - Anti-fuse OTP (for robust trim storage)
  - MRAM Recovery from external NVM or Host

- Increased NVM and SRAM

- Enhanced Peripherals
  - MSPI Max Speed Increased (HexSPI 250MT/s)
  - USB HS with DMA
  - DMA added for UART and Full-Duplex SPI Slave
  - 2x SDIO



ambiq

Product Comparison – Leap Forward

	Apollo4 Plus	Apollo5
	Cortex-M4F 96MHz / 192MHz	Cortex-M55 with EPU 96MHz / 250MHz
	2.75MB SRAM 64KB Code Cache	3.75MB SRAM 64KB Instr / 64KB Data Cache
	AXI (128-bit) 32x data cache buffers	AXI (128-bit) (DAXI replaced with M55 Data Cache)
	“While Loop”: 13.5µ**W**/MHz Coremark: 29.4µ**W**/MHz Deep Sleep (no Ret): 25.4µ**W** Deep Sleep (2.75MB Ret): 110.4µ**W**	Coremark: 1.3x better CM/mW than AP4+ Deep Sleep (no Ret): 23µ**W** Deep Sleep (3.75MB Ret): 57µ**W**
	2.5D GPU + 4-layer DC 500 x 500 resolution; 60 fps Anti-aliasing Dithering Vector Graphics	2.5D GPU + 4-layer DC GPU: 2.5x higher performance MIPI: 768Mbps per channel (1.5Gbps total) 500 x 500 resolution; 60 fps Anti-aliasing Dithering Vector Graphics
	1x HexSPI, 2x OSPI (up to 96MHz)	2x HexSPI (up to 250MT/s), 2x OSPI (up to 96MT/s)
	8x SPI/I2C (up to 48MHz)	8x SPI/I2C (up to 48MHz)
	USB 2.0 Device FS 1x SDIO (50MB/s)	USB 2.0 Device FS/HS 2x SDIO (50MB/s)
	SecureSPOT 2.0 (HW based)	SecureSPOT 3.0 (hardened secure boot, PUF, eFuse with scramble)
	4x Stereo PDM 1x Stereo LP-AMIC 2x I2S w/ASRC	1x Stereo PDM 1x Stereo LP-AMIC 2x I2S w/ASRC

ambiq

endpoint intelligence

CPU Subsystem

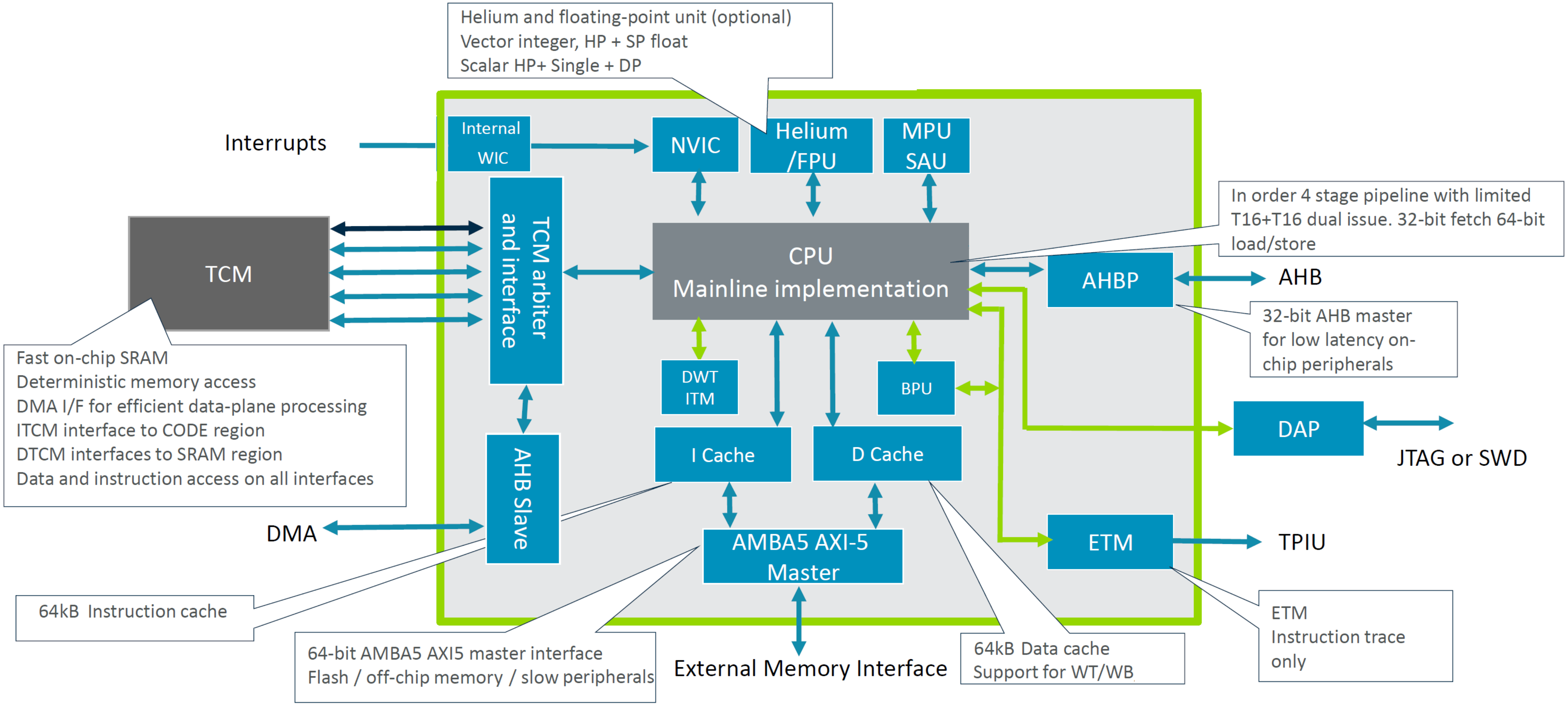
CPU Subsystem

CPU Subsystem

- Cortex-M55 with Helium vector extensions
  - Armv8.1-M instruction set
  - New four stage pipeline
  - Full in order pipeline with limited dual issue capability of 16-bit instructions
  - Full Harvard memory interface
  - Helium supports vector integer, half-precision and single-precision floating point
- TrustZone
  - MPU: 16 NS regions + 16 S regions
  - Security Attribution Unit (SAU): 8 regions
  - TCM Gate Unit (TGU)
    - ITGU: 8KB block, 32 blocks
    - DTGU: 8KB block, 64 blocks
- 64KB Instruction / 64KB Data Cache tightly coupled with CPU
- Integrated 256KB Instruction TCM / 512KB Data TCM
- LP Mode: 96MHz / HP Mode: 250MHz
- Multiple power domains
  - Core
  - EPU (Helium vector unit)
  - Cache RAMs
  - 3 TCM power banks
  - Debug
  - Always On
- Debug
  - 8 Breakpoints across all memory regions
  - Support for range based watchpoints
- Trace
  - ETM for instruction trace
  - 8 DWTs for selective data trace, event and profiling trace
  - ITM for software instrumentation trace

M55 Overview

CPU Subsystem



Memory: Cache

CPU Subsystem

- 64KB Instruction Cache (I-Cache)
  - 2-way set-associative, 256b line size, Prefetch supported
- 64KB Data Cache (D-Cache)
  - 4-way set-associative, 256b line size, Prefetch supported
- Caching is supported for all memory marked as cacheable (excluding ITCM/DTCM). Peripheral Device space is non-cacheable.
- Cache behavior configurable by memory region using MPU
  - Cache configurations include non-cacheable, write-through, write-back, etc
- Cache requires explicit management to ensure cache coherency
  - Please refer to *Apollo5_Memory_Cache_Management* document
- Cache frequency matches CPU frequency (LP or HP mode)
- Intended to provide zero wait state read access from CPU to higher latency memory and reduce power
- The cache is controlled by the M55. If the CPDLPSTATE register sets cache “OFF”, it will be powered down. If “ON”, the Cache follows the M55 core into and out of retention.
- The M55 can support multiple outstanding memory requests at the same time, using up to 4 read and 4 write AXI transaction IDs. This allows data and instruction prefetch to happen "in the background" without blocking the CPU's normal operation.

endpoint intelligence

Memory Subsystem

Apollo510Memory Controller

High-performance AXI Crossbar system Fabric

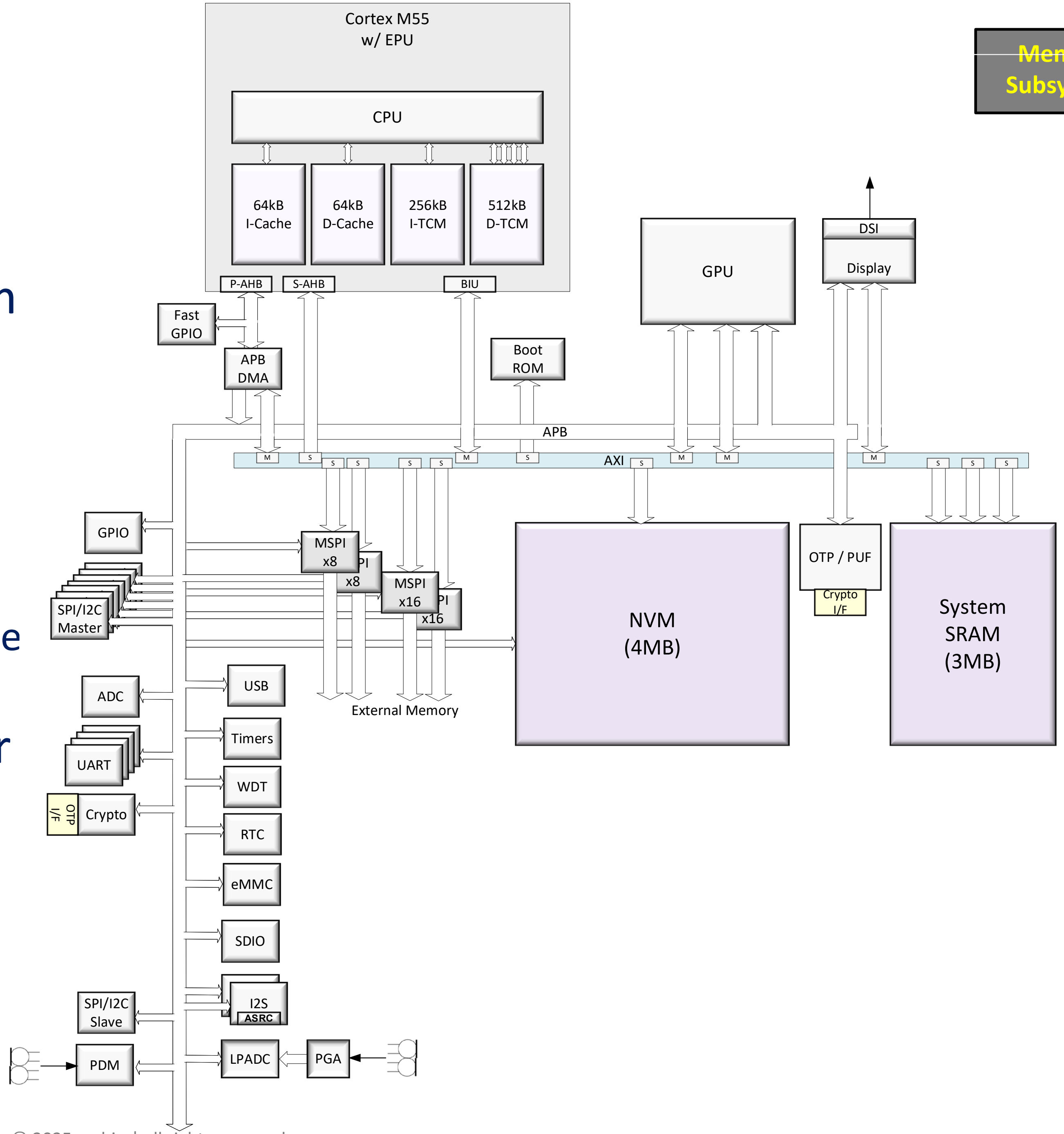
- Point-to-Point connections
- Full bandwidth available if no contention

Maximum M55 Caches (64KB)

- Processor caches provide best performance

MSPI Performance improved 2.5X over Apollo4

- 125Mhz DDR HEX (x16 wide) MSPI



Memory Subsystem

- Memory Types
  - SRAM (Tightly Coupled Memory and Shared System Memory)
  - Integrated NVM (MRAM) / External Memory via MSPI
  - Boot Loader ROM (SBR + helper)
    - Note: ROM is in separate power domain and should be powered down when not in use
    - Note: SBR is not accessible after SBR lock
  - One Time Programmable (OTP) memory
- Key Features
  - 3072 kB Shared SRAM
  - 256 kB ITCM / 512 kB DTCM
  - 4 MB MRAM
- 16 kB MRAM INFO / 4kB eFuse OTP INFO
  - 2 kB (MRAM INFO0) / 256B (eFuse INFO0)
    - For customer use
  - 8kB (MRAM INFOC) / 1kB (eFuse INFOC)
    - Contains some memory protection defaults (as well as other security parameters)
  - 6kB (MRAM INFO1) / 2.75kB (eFuse INFO1)
    - Ambiq trims usage
- External PSRAM or Flash with XiP memory mapped access via MSPI
  - 4x Memory apertures (1x256MB, 1x128MB, 2x64MB)
  - CPU XiP accesses cached by I-Cache and D-Cache

Apollo 5 Caching and Buses

Memory Subsystem

- Apollo510 with M55 has separate 64kB Instructions and Data caches.
  - Caches are part of the M55 Core delivery and functionality has been verified by ARM
- Apollo510 has 128bit (16Byte) wide 96mhz AXI-compatible system bus
  - AXI Fabric is a Crossbar with 6 manager x 10 Memory interfaces
  - Each manager (CPU, GPU, DM, DMA) can receive the full 16x96Mhz = 1.5GB/sec bandwidth in each direction unless they contend for the same memory in the same cycle
    - GPU has multiple ports
  - Maximum theoretical peak read throughput demand of 9.2GB/sec if all masters operating simultaneously
  - Bus-memory complex has more that sufficient bandwidth with headroom

Memory: NVM

Memory Subsystem

- 4MB of MRAM (2 instances of 2MB)
- Dual power domains for each instance
- Powered down via hardware FSM
  - CPU deep sleep
  - Wake on DMA read access from MRAM
- Main partition write and read protections at 16KB granularity use FLASHxPROTx registers in MCUCTRL.
- MRAM programming does *not* require erase (although page/mass erase is supported via helper function for consistency)
- MRAM supports 100K program cycles
- MRAM is sensitive to magnetic interference. Please consult magnetic resilience guidelines document and MRAM Recovery section later in this presentation.

Memory: TCM

Memory Subsystem

- 256KB Instruction TCM (ITCM)
- 512KB Data TCM (DTCM)
- TCM Banks
  - Three power domain partitions (Power and retention controlled by PWRCTRL:MEMPWREN and PWRCTRL:MEMRETCFG):
    - 32KB ITCM + 128KB DTCM
    - 128KB ITCM + 256KB DTCM
    - 256KB ITCM + 512KB DTCM
  - Always zero wait-state unless there is contention for that specific memory array with an AXI slave request
- Runs at CPU frequency (96MHz or 250MHz), and directly accessed by M55, not through I and D caches
- Accessible by other masters using DMA across AXI bus but with restrictions:
  - TCM are in M55 power domain, and cannot be accessed when M55 is off or sleeping
  - TCM accesses by DMA must not cross 4k boundaries
- Memory Latency: 0 wait-state for CPU access unless conflict with DMA access to same 32kB bank of TCM
  - If there is DMA conflict, max DMA transfer size before arbitration is 16
- Both I-TCM and D-TCM can be used interchangeably for Instruction and Data
  - I-TCM has one 32-bit interface and D-TCM has 4x 32bit interfaces
  - If locating both instructions and data in TCM for optimized performance of a small routine, then locating each in their designated TCM will enable parallel instruction and data fetches for highest performance
- TCM is useful for frequently accessed code and data to reduce cache churn, and provides deterministic low latency performance for speed or timing critical code and data

Memory: Shared SRAM (SSRAM)

Memory Subsystem

- 3072KB SRAM
- SRAM Partitioning
  - Three power partitions (controlled by PWRCTRL:MEMPWREN): Lower 1MB, Middle 1MB, Upper 1MB
  - 3 port concurrency
    - Port1: CPU Instruction/Data
    - Port2: Graphics
    - Port3: Display Controller + APB DMA
- Runs at 96MHz independent of CPU operating frequency
- Write and read protections at 16KB granularity use SSRAMxPROTx registers in MCUCTRL
- Retention at power domain granularity (controlled by PWRCTRL:MEMRETCFG)
- Memory Latencies
  - SSRAM Read access from the CPU and DC is a minimum of 10 cycles.
  - Read access from the GPU is a minimum of 10 cycles.
  - Latency will increase if there is contention for the memory section

Memory: INFO Space (OTP and MRAM)

Memory Subsystem

- 4KB Total Secure INFO Space in the OTP module using eFuses
- INFO1: 2.75 KByte contains factory preset chip trim values in eFuse OTP
- INFO0: 256 Byte for customer trim values in eFuse OTP
  - INFO0 in NVM MRAM available for development, OTP or MRAM configurable by INFOC register
- INFOC: 1 KByte for security in eFuse OTP
  - Lifecycle State
  - Ambiq key bank
  - Customer key bank
  - Secure Bootloader Configuration
  - MRAM main partition protection
  - Note: INFOC access requires crypto device to be powered
- Note: access to eFuse OTP requires OTP device to be powered (OTP is powered ON by default)
- Note: eFuse OTP device should be powered OFF by SW to save power when not in use

Memory: DMA

Memory Subsystem

- Peripheral Controllers with DMA capability:
  - UART
  - USB
  - MSPI
  - ADC & AUDADC
  - PDM
  - I2S
  - SDIO
  - Crypto
  - GFX
  - Display Controller
  - I2C Manager/Subordinate
  - SPI Manager/Subordinate
- DMA is supported from peripheral to SSRAM/TCM/PSRAM and from SSRAM/TCM/PSRAM/MRAM to peripheral
- There is no memory-to-memory DMA
- DMA transactions to/from SSRAM occur concurrently to CPU instruction/data and GPU accesses if the accesses are to different physical banks of memory. Accesses to the same physical bank are arbitrated in hardware.
- DMA to/from TCM is supported if the M55 is not in deep sleep and SW must ensure that the DMA transfer does not cross a 4KB boundary.
- Some peripheral controllers have a command queue which allows intelligent chaining of multiple DMA transfers
  - Controllers with Command Queue: SPI Manager (IOM), I2C Manager (IOM), MSPI

ambiq

endpoint intelligence

Cache and Memory Performance

M55 Cache Bandwidth

Memory Subsystem

- The M55 support both Instruction and Data cache prefetching
  - I-cache prefetching uses a single read stream, and one outstanding fetch at a time (specific algorithm is private to ARM)
  - D-cache prefetching can use 2 read streams, allowing multiple outstanding accesses in parallel (up to 6) without blocking the CPU. The prefetch engine will recognize strides in the access patterns and fetch non-sequential data cache lines.
- M55 Cache Fill Bandwidth
  - A single prefetch stream can support 658MB/sec of bandwidth from the SRAM memories and 165MB/sec from MSPI memories
  - With the dual-prefetch streams, the bandwidth can be added across the 2 streams (with some degradation for contention)
    - Two streams prefetching from the same MSPI could achieve 275MB/sec prefetch (2x165MB/sec * 85%)
    - A stream from SRAM and a stream from MSPI might achieve 800MB/sec prefetch

M55 Cache Control Details

Memory Subsystem

- Details of the M55 Caches can be found in Section 9.9 of the M55 TRM
- Data cache is 64KB, four-way set-associative. Supports write-back, write-through, and read/write allocate/no-allocate attributes
- Data Cache Transient Attribute
  - The M55 has a new “Non-Transient” memory attribute for tuning performance.
  - “Clean cache lines that are associated with Transient memory are prioritized for eviction over lines that are associated with Non-transient memory.”
  - Therefore, marking certain data as non-transient will bias that data to be retained in the cache over other transient data
- Instruction cache is 64KB, 2-way set-associative. Only supports read-allocate. Allocation controlled by inner-cacheability attributes
- M55 provides a rich set of cache maintenance operations (TRM sec 9.9.3) with: invalidate all (I-cache), invalidate by address (I & D), invalidate by set/way (D-cache), clean by address and set/way (D-cache), and clean & invalidate by address and set/way (D-cache).
- Pre-fetch –TRM sec 9.4.1 describes the prefetch support.
  - Up to 6 outstanding data linefills/prefetches supported (3 per ID stream)
  - Prefetcher recognizes access strides of -2, -1, +1 and +2

Apollo 5 MSPI Performance

Memory Subsystem

Apollo 5 MSPI Performance from the CPU

- M55 I & D caches cache all CPU accesses to MSPI XiP mapped memory
- The extended prefetch of the cache helps boost performance to 160MB/s
- The small burst size of the M55 (32-bytes) limits the performance from MSPI, so an optional **read combine** mechanism was added to combine sequential MSPI accesses to generate larger reads from MSPI to PSRAM, and further boost red performance to 275MB/s

Apollo 5 provides same DMA Bandwidth as Apollo4

- Apollo4 sustains 87-96 MB/s DMA throughput between SSRAM and PSRAM while APBDMA has a peak throughput of 192MB/s
- Apollo510 sustains 79-86 MB/s DMA throughput between SSRAM and PSRAM while APBDMA has a peak throughput of 384MB/s. The MSPI interface is the bottleneck.
- The MSPI DMA does not block out the AXI XIP/Data interface during long transfers by breaking the DMA into smaller chunks, and the bias is tunable.

Key Take Aways

Memory Subsystem

- AXI Fabric is a Crossbar
  - 6 manager x 10 Memory interfaces
  - Each manager (CPU, GPU, DM, DMA) can receive the full 16x96Mhz = 1.5GB/sec bandwidth in each direction unless they contend for the same memory in the same cycle
- MSPI XIP memory performance greatly improved over Apollo4
  - More than enough bandwidth for 640x640x32 @60fps for rendering and framebuffer
  - Significantly improved MSPI read Performance.
- M55 internal caches accesses provide the best possible CPU performance
  - Transient attribute allows for tuning between internal and external memories

ambiq

endpoint intelligence

Clocks and Resets

Clock Generation

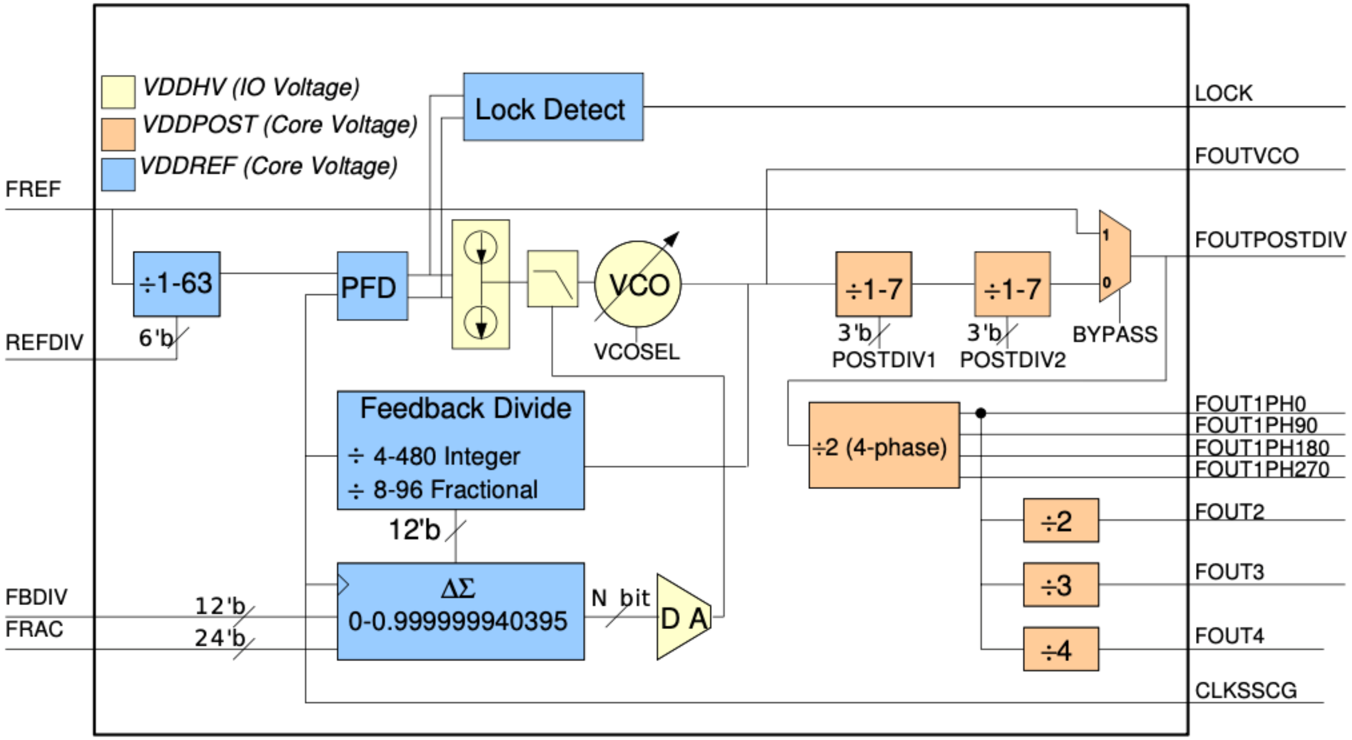
Clocks and Resets

- Independent frequency scaling for various SoC subsystems
- Ultra low power, low frequency clock generation with XTAL calibration
- Programmable I/O clock dividers
- High precision audio clock generation
- Sub modules
  - High frequency XTAL oscillator
  - Low frequency XTAL oscillator
  - 2x High frequency RC oscillator
  - Low frequency RC oscillator
  - SoC clock generation logic
  - Audio subsystem clock generation logic

Clock Name	Frequency	Tolerance	Duty Cycle (% High/ Low)	Additional Requirements / Comments
HFRC_96MHz	96 MHz	±5%	50/50	
HFRC_192MHz	192 MHz	±5%	30/70	
HFRC48	48 MHz			HFRC48 is a dedicated clock from the HFRC block (HFRC192_DIV4); selected by MCUCTRL_PLLMUXCTRL field for the PDM or AUDADC modules only.
LFRC	Nominal 900 Hz		50/50	
HFRC2_125MHz	125 MHz	±5%	50/50	
HFRC2_250MHz	250 MHz	±5%	45/55	
XTALHS[1]	32 MHz, 24 MHz and other frequencies		50/50	Startup time < 1 ms maximum, < 0.5 ms ideal.
XT_32KHz[2]	32.768 kHz	±50 ppm[3]	50/50	
EXTREFCLK	Variable depending on the external clock source			
PLLCLK	1.22 MHz to 960 MHz		50/50	PLLCLK output offered as clock for I^2S and other audio modules in HP mode, and for the USBPHY.

Programmable Fractional PLL

- Generates a precision clock for the audio subsystem
- Reference clock is HF OSC or EXTREF clock
- 24-bit fractional accuracy
- Ultra-low jitter output clocks
- Supports all audio sampling rates (48KHz, 44.1KHz, 16KHz, etc.)



Reset Controller

Clocks and Resets

- Four levels of MCU-Level Reset
  - SYSRESETn – triggered by AIRCR.SYSRESETREQ, an M55 warm reset
  - POR = POR Software Reset = ARESET = WatchDog = RSTn, an M55 cold reset
  - POI = POI Software Reset
  - POA = Power-on Reset
- Most resets trigger POR reset, which is the one of the shallowest Resets
- POI reset is the deepest digital reset affecting HW subsystems and triggers INFO space settings to be reloaded from Flash
- POA is the very deepest reset, digital and analog, and is only triggered by voltage going below POA voltage (~1.3V – See data sheet for exact value)

Reset Level	CPU, GPIO and Peripherals (Except XT, LFRC, RTC and STIMER)	Debug Subsystem	CLKGEN (XT, LFRC, etc.), Reloads INFO Settings (Trims, Options and Security)	RTC and STIMER
SYSRESETn	x			
POR[1]	x	x		
POI[2]	x	x	x	
POA (Power-on Reset)	x	x	x	x

1. Doesn't affect CLKGEN except for setting the CPU clock to 96 MHz.
2. Reloads CLKGEN_OCTRL_OSEL bit from trim and resets CLKGEN features such as registers, CPU clock, HFADJ and HF2ADJ - everything else is reset only by POA.

Reset Actions

- Reset can be caused by:

  SW POI Request, Brownout, Reset Pin, Watchdog Timer, SW POR Request, SYSRESETREQ, SW TPIU Request, CDBGRSTREQ, SW Debug Reset, SW IWIC Reset

- Features with individual reset:

  MCU (Cold Reset or Warm Reset), Debug-AHB/EPP Buffer, PMU, Peripherals, Debug, TPIU, IWIC, Power Control

Action	MCU Cold Reset	MCU Warm Reset	RTC/STimer Reset	D-AHB/EPPB Reset	PMU Reset	Async Reset	Debug Reset	TPIU Reset	IWIC Reset	Power Ctrl Reset
POA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
POI, SW POI, Brownout	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
Reset Pin, Watchdog, SW POR[1]	Y	Y	N	Y	Y[2]	Y	Y	N	Y	N
SYSRESETREQ	N	Y	N	N	N	Y	N	N	Y	N
TPIU SW Disable, SW TPIU	N	N	N	N	N	N	N	Y	N	N
CDBGRSTREQ	N	N	N	N	N	N	Y	Y	N	N
SW DBG Reset	N	N	N	N	N	N	Y	N	N	N
SW IWIC Reset	N	N	N	N	N	N	N	N	Y	N

ambiq

endpoint intelligence

Simobuck and LDOs

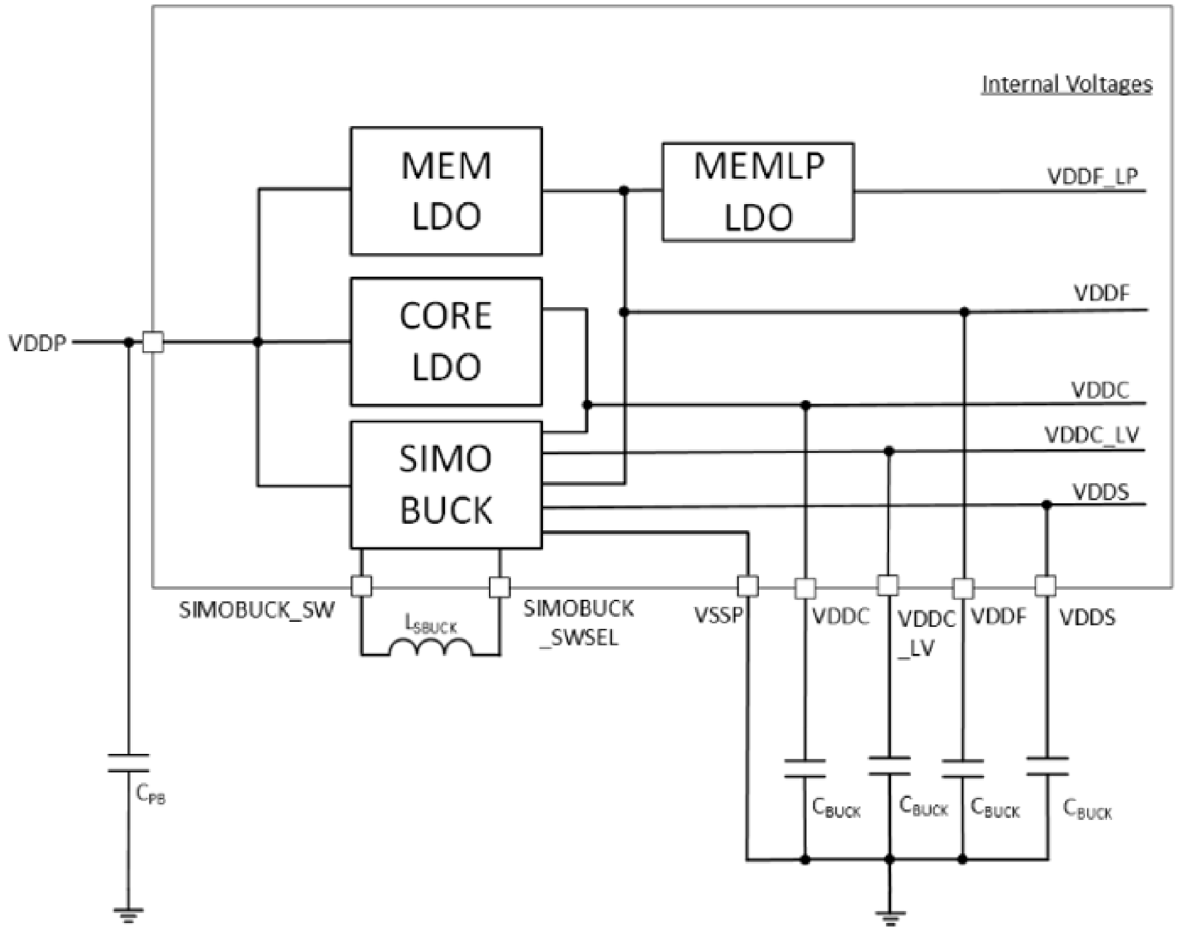
Power Architecture

SIMOBuck and LDOs

- PMU consists of:
  - Single Inductor Multiple Output SIMOBUCK
  - LDOs for internal
  - Power switches for muxing or gating internal domains
- Both BUCKs and LDOs have 2 modes of operation :
  - ACT -> can provide mAs of current but quiescent current is high
  - LP-> can source 100’s of uA with much reduced quiescent current
- When MCU is in Deepsleep with no peripherals enabled, regulators go the LP mode
- MCU Awake in LP/HP modes, or in Deepsleep with peripheral enabled then regulators go to ACT

SIMOBUCK & LDOs

SIMOBuck and LDOs



- Simobuck can generate 4 regulated rails (VDDF,VDDC,VDDC_LV & VDDS)
- LDOs (CORE&MEM) generate 2 unique regulated rails (VDDF shorts to VDDS, VDDC to VDDC_LV)
- MCU powers up in LDO mode, after initialization simobuck can be enabled
- MEMLPLDO and ANALDO (not shown) generate internal voltage rails for analog use

Apollo5 PMU Internal Rail Details

SIMOBuck and LDOs

- Internal blocks powered by the four SIMOBuck rails

	CPU LP Mode			CPU HP Mode		
	CPU Only	Standard Peripherals on	GPU or SDIO on	CPU Only	Standard Peripherals on	GPU or SDIO on
CPU	VDDC_LV	VDDC_LV	VDDC_LV	VDDC	VDDF	VDDF
AXI/APB, DISP controller	VDDC	VDDC	VDDC	VDDC	VDDC	VDDC
GPU	VDDC	VDDC	VDDC	VDDF	VDDF	VDDF
TCM/Cache/SSRAM core, MSPI, SDIO, DISP Interface, Always on logic	VDDF	VDDF	VDDF	VDDF	VDDF	VDDF
TCM/Cache Periphery	VDDC	VDDF	VDDF	VDDF	VDDF	VDDF
SSRAM Periphery	VDDC	VDDF	VDDF	VDDC	VDDF	VDDF
Memory Retention	VDDS	VDDS	VDDS	VDDS	VDDS	VDDS

ambiq

endpoint intelligence

Power Management

Power Management – Sleep

Power Management

- Apollo510 sleep modes are very flexible.
- Entering Normal or Deep Sleep does not automatically change the clock source for timers, nor does it stop or disable any peripherals or interrupt sources.
- You can configure the peripherals and GPIO in any way you want and they will continue to operate in normal or deep sleep. Of course, keeping peripherals active will result in higher deep sleep current, so for lowest power deep sleep all peripherals not actively in use should be powered down.
- [New]
  - Arm and Ambiq variants of normal sleep
    - In Ambiq sleep, when the processor goes to sleep, clocks are turned off at the source (CLKIN is disabled) and that may lead to HFRC/2 being disabled if there are no other clients of HFRC/2
    - In ARM sleep, CLKIN stays enabled.   The M55 will do internal clock gating to save power. ARM sleep is less power efficient than Ambiq sleep, but may have a shorter wakeup time
    - ARM sleep vs Ambiq sleep is controlled by SLEEPMODE field of PWRCTRL_CPUPWRCTRL register.

ambiq

Power Management – Sleep

Power Management

- It is straightforward to enter sleep modes:
  - am_hal_sysctrl_sleep(AM_HAL_SYSCTRL_SLEEP_DEEP);
  - am_hal_sysctrl_sleep(AM_HAL_SYSCTRL_SLEEP_NORMAL);
    - ARM sleep vs Ambiq sleep is controlled by the SLEEPMODE field of the PWRCTRL_CPUPWRCTRL register.
- Any Apollo510 interrupt can be used as a wake trigger from normal or deep sleep.
- Common wake interrupt sources used in deep sleep include:
  - GPIO, Ctimer, Stimer, RTC Alarm, Comparator, ADC Window compare, I2C/SPI slave write, FIFO threshold interrupts for any serial interface, etc.
- Ambiq to ARM sleep transitions:
  - Even if the processor starts normal sleep in Ambiq sleep, it will transition behind the scenes to ARM sleep when an interrupt is received.
  - Reason for this is the M55 does not output the wakeup mask during normal sleep (only in deep sleep). Transitioning to ARM sleep allows the iWIC and the NVIC to synch up and determine if a real wakeup event has been received. If yes, wakeup sequence is initiated. If no, CM55 stays in ARM sleep until a wakeup interrupt is received.
  - Ambiq sleep was implemented as a power optimization over ARM sleep due to the external clock gating. It is expected that the Ambiq sleep power advantages will be realized a majority of the time.

Power States

Power Management

- High Performance (or TurboSPOT) Active Mode
  - CPU is in active state with 250MHz core clock
  - Tightly Coupled Memory (TCM) and CPU caches (if on) are also at 250MHz (CPU access remains single cycle)
  - Cache can be turned off, EPU can be in retention to conserve power
  - All other frequencies in the SoC are unaffected
- Low Power Active Mode
  - Default active state
  - CPU is in active state with 96MHz core clock
  - Cache can be turned off, EPU can be in retention to conserve power
- Sleep Mode (ARM "ON" or “FUNC_RET” Mode)
  - CPU is powered but clock gated
    - Ambiq Sleep – CM55 is externally clock gated
    - ARM Sleep – CM55 is internally clock gated
  - TCM is accessible
  - State of all other functions within the SoC are independent of the CPU
- Deep Sleep Mode (ARM "FULL_RET” Mode)
  - CPU and cache are powered down, register state in retention, cache state is optionally in retention
  - MRAM is powered down
  - TCM is powered down/retention
    - TCM is NOT accessible when the M55 is in deep sleep, or if the TCM is explicitly powered off.
  - State of all other functions within the SoC is independent of the CPU

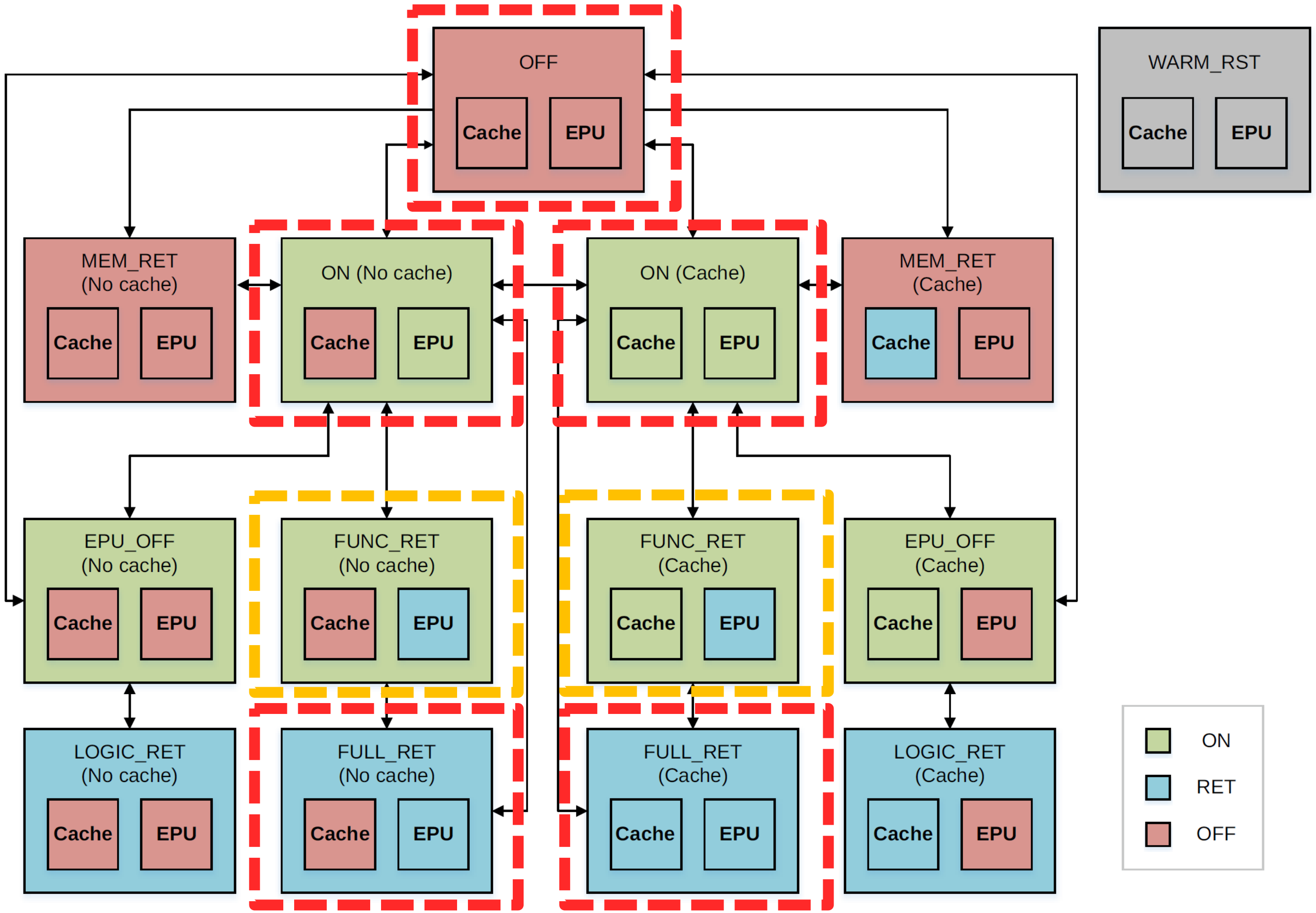
Support Power States

Supported power states

Optional supported power states

Use P-Channel to communicate between CPU and PMU.

Optional supported power states can be disabled by REG_PWRCTRL_CPUPWRCTRL.DISFUNCRETMODE



Power States – Active Mode

Power Management

- MCU core active with 96MHz core clock (LP mode - default)
- All MRAM, eFuse OTP, ROM, TCM and cache start powered/enabled, but SSRAM starts powered down
  - Default trims enable all SSRAM banks
- All Pins start with GPIO buffers disabled (High-Z) except for the two SWD pins
  - GPIO20 and GPIO21 are configured as SWDIO and SWDCK with no internal pull-up/down (i.e. floating) so these either need external pull-up/down as recommended in design guide or need to be reconfigured by software. All other pins can be left unconnected.
  - Note that all GPIO pins have ESD diodes that connect the GPIO pin to VDDHn of their respective IO domain. Regardless of GPIO configuration, voltage applied to GPIO pin should not exceed VDDHn.
- XT and RTC clock start enabled at initial power-up. However, the standard bsp init code in the SDK stops both.
  - Unlike nearly all other registers, the state of XT and RTC are not cleared by any reset other than power-on reset (to allow RTC to operate through brownout or software resets without losing time). Therefore, it is prudent to explicitly initialize these to desired state after reset regardless of default power-on-reset state. (Low power init function disables XT and RTC)
- Peripherals (I2C/SPI, MSPI, UART, ADC, PDM, etc.) start powered-off (power-gated)

Power States – Sleep Mode

Power Management

- When normal sleep mode is entered:
  - Clock is gated from the CPU Core
    - With the M55, there are two sleep modes: ARM Sleep keeps the main core clock running but locally gated, while in AMBIQ Sleep the main core clock is gated. Certain capabilities are impacted depending on the mode selected (run but SYSTICK). The default is AMBIQ Sleep.
  - MRAM remains powered in standby mode
    - On Apollo510, there are 2 2MB MRAM instances. There are separate power domains for each allowing SW to power down the unused macro if applicable per workload to save standby power.
  - Everything else that was active before the sleep command remains active. HFRC continues to run but is gated from the core.
  - It is recommended that SW power off the ROM and eFuse OTP domains whenever possible to save power. Of course, these need to be powered up whenever used.

Power States – Deep Sleep Mode

Power Management

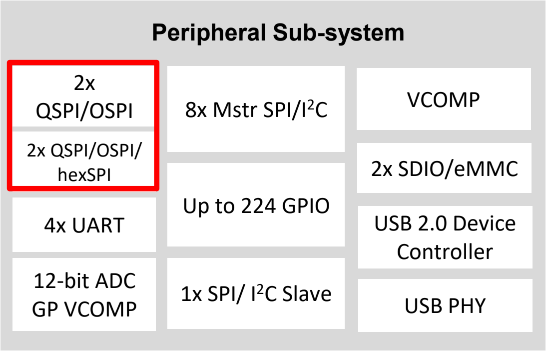
- When Deep sleep is entered, anything that is not being used will be powered down:
- All GPIO retain state, and any GPIO can be configured to generate an interrupt which will wake the MCU from deep sleep
- MRAM is powered down (note: both MRAM instances are powered down at same time)
- SRAM transitions to a low-power retention state
  - Deep Sleep retention setting determines how much of the SRAM is retained.
- If cache is enabled, it is retained or not based on cache deep sleep retention configuration
  - Cache retention is handled by the HW under SW direction via the P/Q interface and CPDLPSTATE CPU register. If cache is configured OFF in the CPDLPSTATE register, it will be powered down. If it is ON, it will follow the state of the M55, going into retention when M55 is in deep sleep.
- Any clocks (HFRC, XT, and LFRC) not being used by peripherals or timers will be stopped (HFRC is additionally powered down if not being selected for any clock source). Conversely, any clock that is being used will continue running.
- Any peripheral that is powered on will stay on. For lowest power deep sleep, this is the most critical item, to power down as many peripherals as possible.
  - Note this does not include RTC, CTimers, or STimer which are very low power if they are using LFRC, XT, or low frequency external timer input as clock sources. These can be left running in deep sleep. The XT and LFRC clocks are very low power.

ambiq

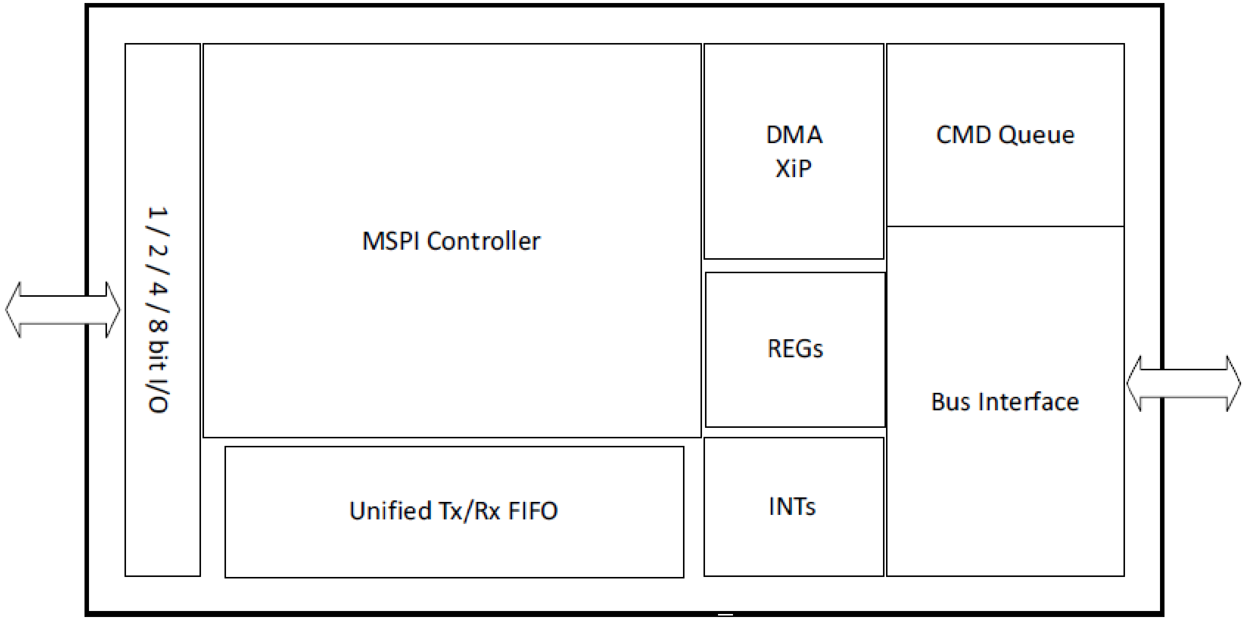
endpoint intelligence

MSPI

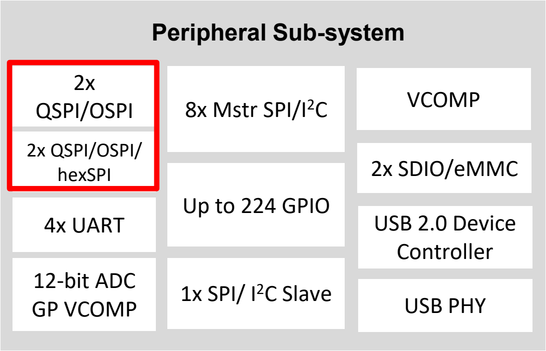
MSPI



- Apollo510 MSPI Interface
  - 2x OctalSPI - 1/2/4/8-bit MSPI interface
  - 2x HexSPI - 1/2/4/8/16-bit MSPI interface
- Support for 1.2V and 1.8V I/O voltages
- Support for DCX signal for displays
- XiP supported
- DMA with peripheral-to-memory and memory-to-peripheral support
- MSPI1/MSPI2: Up to 96 MHz interface in SDR mode; 48 MHz in DDR mode (96MT/s)
- MSPI0/MSPI3: Up to 125 MHz DDR (250MT/s)
- All four SPI modes supported
- Command Queue Support



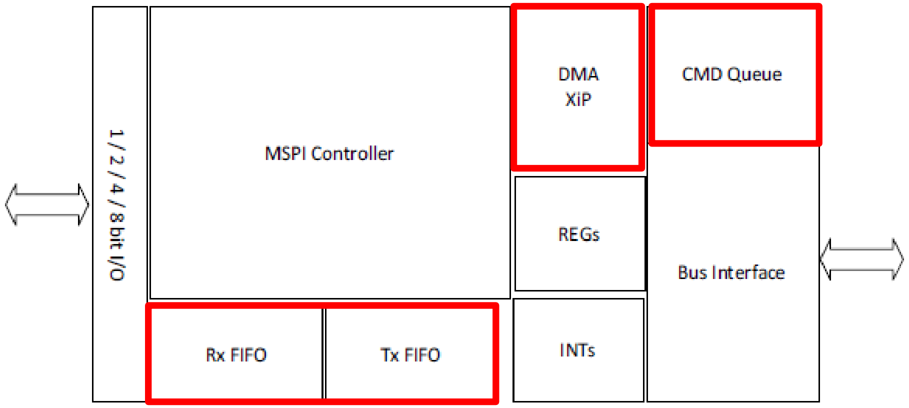
MSPI (cont’d)



- Unified 32-entry FIFO (32 bits wide) for reads and writes
- DMA Support
  - Simple DMA model where software sets internal (SRAM or flash) address and external device address, transfer direction, and transfer size
  - MSPI DMA controller automatically handles sequencing of instructions and address to serial flash device.
  - Software configures registers to specify device read/write command bytes and address bytes (1 to 4)
- Command Queue
  - Software can construct a buffer of operations and MSPI will execute the series of operations autonomously
  - Used in combination with DMA

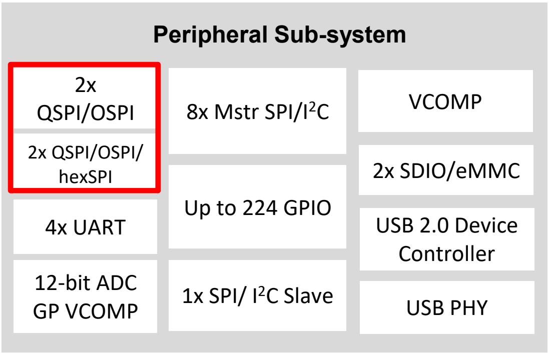
Table 363: FLASH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
READINSTR								WRITEINSTR								RSVD					XIPMIXED			XIPSENDI	XIPSENDA	XIPENTURN	XIPBIGENDIAN	XIPACK		RSVD	XIPEN

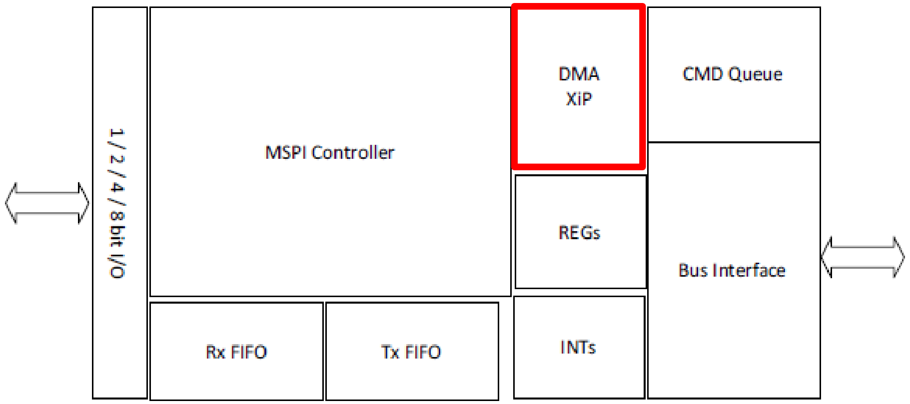


ambiq

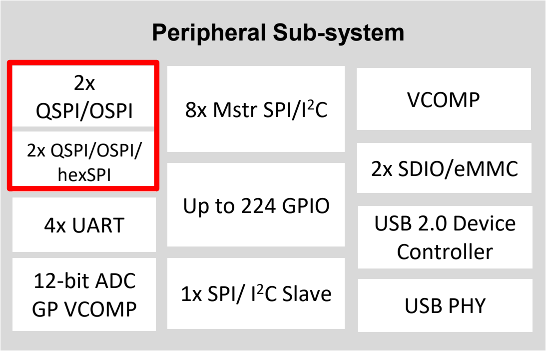
MSPI (cont’d)



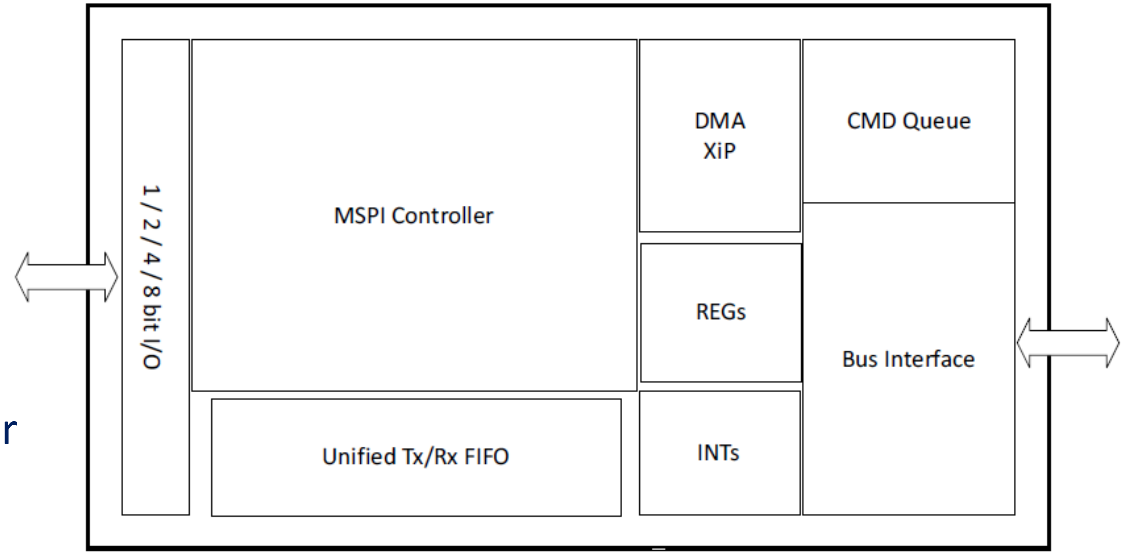
- MSPI supports Execute in Place (XIP) Operations
- Instruction and Data accesses to MSPI connected devices can be cached in CPU I-Cache and D-cache, respectively
- XIP and DMA/PIO operations can be interleaved. MSPI controller will allow current operation to complete before performing the XIP operation.



Serial Communication: MSPI cont'd



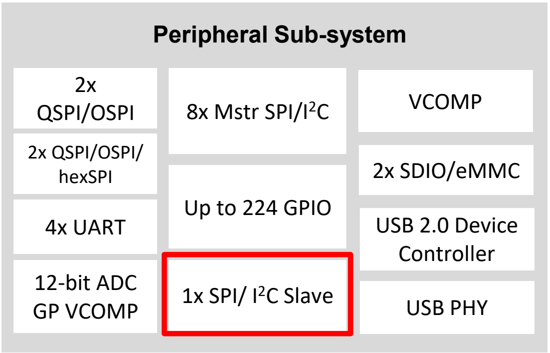
- The AXI addressable aperture is increased to 512MB.
  - MSPI0 – 256MB
  - MSPI1/2 – 64MB
  - MSPI3 – 128MB
- Programmable CE Latency – To relax the timing requirement between memory CE and CLOCK, user can insert up to 4 clock cycles between CE and CLOCK via CELATENCY register
- Separated power domains for 4 MSPI instances
- CPU Read Burst Combining Operation - Combining is only advantageous for the CPU as the other masters (GFX, DC, and DMA) can generate larger bursts on their own to achieve higher bandwidth.



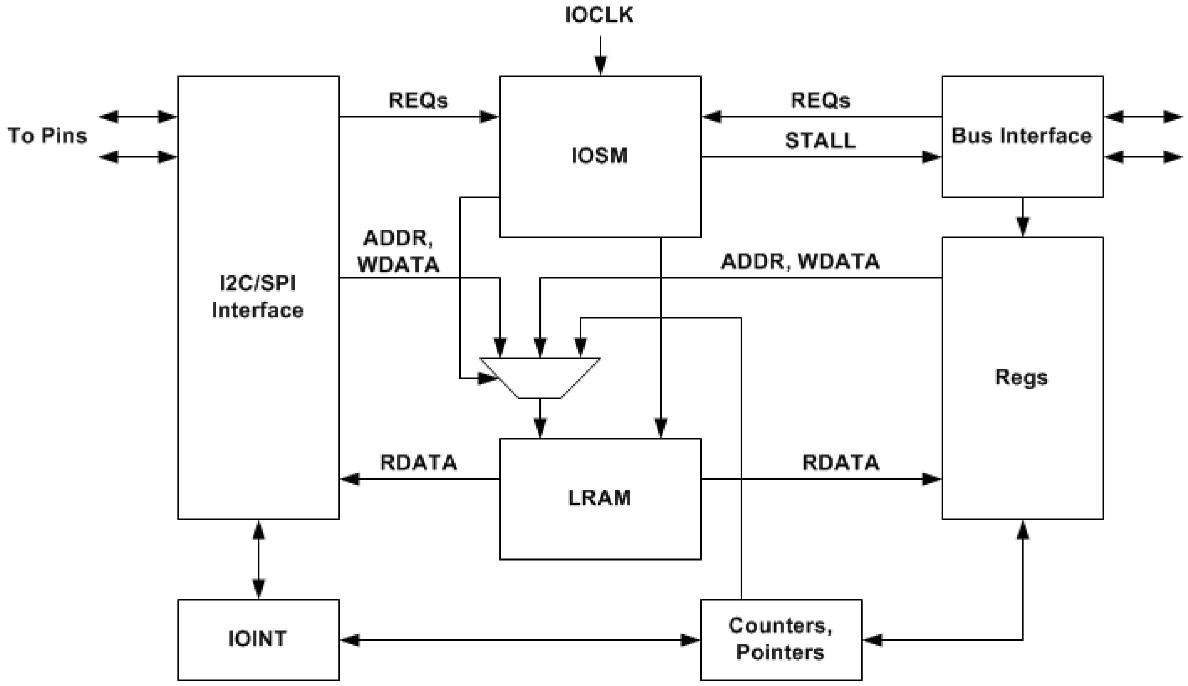
endpoint intelligence

I/O Subordinate (IOS)

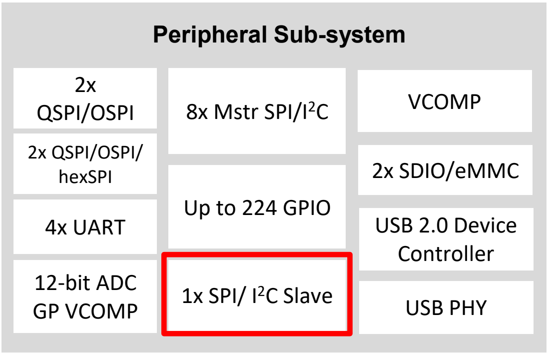
IOS



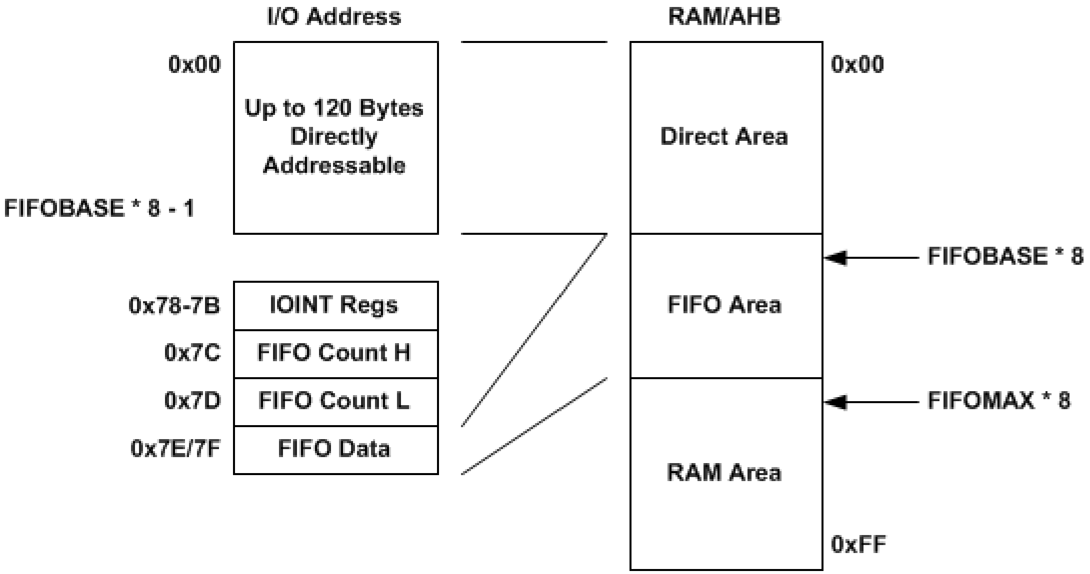
- 2 IOS
  - One Half-duplex IOS supports I2C or SPI (3-wire or 4-wire)
  - One Full-Duplex IOS supports 4-wire SPI
- I2C mode:
  - configurable 7 and 10-bit addressing
  - interface freq. up to 1.2 MHz
- SPI mode:
  - Supports all polarity/phase combinations
  - interface freq. up to 48 MHz
    - With MISO access time of up to 14ns.
- IOSFD (I/O Subordinate Full Duplex)
  - The Full-Duplex IOSFD consists of two IOS module, called IOSFD0 and IOSFD1.
  - IOSFD0 and IOSFD1 are equivalent IOS modules and collectively used as a full duplex SPI (IOSFD).
  - Only 4-wire SPI is supported for IOSFD



IOS (cont'd)



- IOS LRAM
  - IOS0 contains 256 Bytes or LRAM
  - IOSFD0 and IOSFD1 contain 64 Bytes of LRAM each
- Broken up into 3 sections:
  - Directly addressable RAM up to 120 Bytes for which can be accessed by external IOM host while MCU is asleep
  - Status and Config Registers & FIFO. FIFO size up to 256 Bytes for IOS0, and 64 bytes each for IOSFD0 and IOSFD1.
  - Any LRAM not configured as either direct-access or FIFO is left as RAM accessible by Apollo SW only
- DMA
  - All IOS instances support DMA
  - IOSFD has Full-Duplex DMA support
  - DMA can only support IO access through FIFO Data register



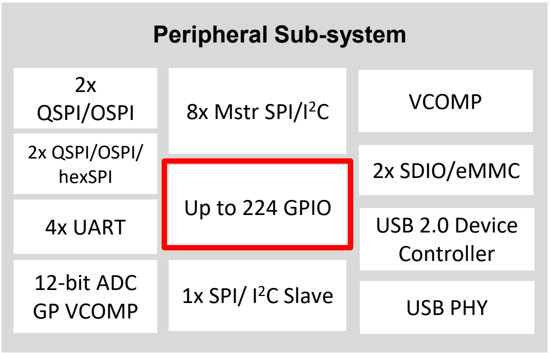
ambiq

ambiq

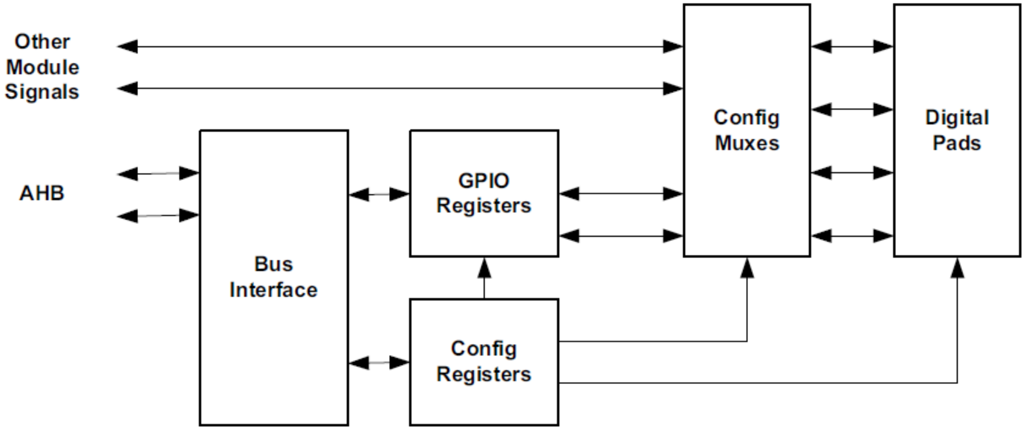
endpoint intelligence

GPIO

General Purpose I/Os

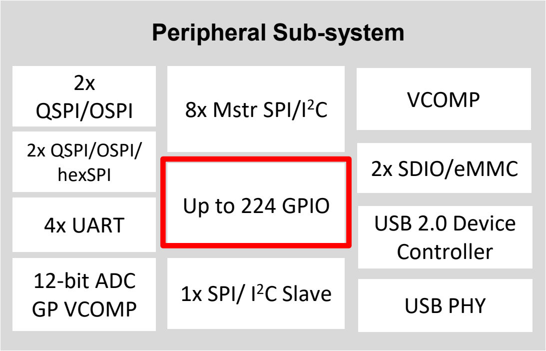


- Apollo510 supports up to 224 GPIOs
  - BGA package supports 183 GPIO
  - CSP package supports 144 GPIOs (still under development)
- Configurable
  - Tristate, Open Drain, or Push Pull
  - Configurable drive strength of 2, 4, 8, or 12mA
    - High Speed (HS) Pins have 8 drive strength levels
  - Many pin function mapping options
- Integrated pull up/down on all GPIO
  - Note: High Speed I/O (VDDH3, VDDH4, VDDH1 mainly) have limited PU/PD options and Slew/DriveStrength are combined
- 6 I/O voltage domains (All domains must be powered)
  - VDDH: Primary voltage domain, 1.71 – 2.2v
  - VDDH1: Trace port, 1.71 – 3.63v
  - VDDH2: Secondary I/O voltage domain, 1.71 – 3.63v
  - VDDH3: MSPI0 I/O voltage domain, 1.14 – 2.2v
  - VDDH4: MSPI3 I/O voltage domain, 1.14 – 2.2v
  - VDDH5: Display I/O voltage domain, 1.71 – 3.63v
- 16 IRQs supported
  - 7 IRQs (+1 reserved) represent OR of a bank of 32 GPIO IRQs
  - 7 IRQs (+1 reserved) represent a MASKED OR of a bank of 32 GPIO IRQs (intended to isolate specific GPIO IRQ for easier INT handler)
  - All IRQs can be used as wake source, and all GPIOs are wake capable



GPIO Pin Function Mapping

- Apollo510 GPIO have flexible pin function mapping
- Refer to Datasheet GPIO chapter or Apollo5 Pin Mapping Spreadsheet for details



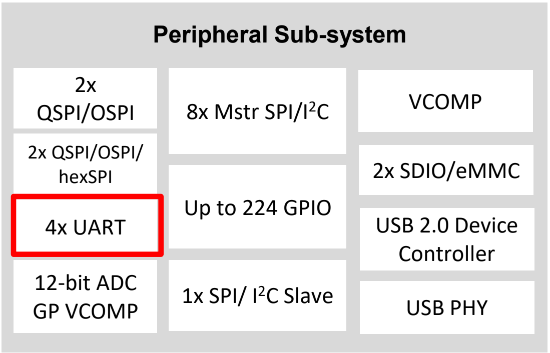
	A	B	C	F	G	H	I	J	K	L	M	N	O	P	Q	S	T	U	V	W	X	Z
1								Pad Function Select Number (PADnFNCSEL)													Package	
2	CSP Pin	BGA Pin	Apollo5 Pad	Apollo5 Function Used	Pad FNCSEL #	Pad	I/O Voltage Domain	0	1	2	3	4	5	6	7	9	10	11	12	13	BGA PKG	CSP PKG
57	E8	A9	PAD19			19	0	ADCSE0			GPIO19	UART2CTS	UART3CTS	CT19	NCE19	I2S1_SDIN		FPIO19			X	X
58	H13	P4	PAD20	SWDCK	0	20	0	SWDCK	TRIG1		GPIO20	UART0TX	UART1TX	CT20	NCE20			FPIO20			X	X
59	J12	N5	PAD21	SWDIO	0	21	0	SWDIO	TRIG2		GPIO21	UART0RX	UART1RX	CT21	NCE21			FPIO21			X	X
60	B8	D9	PAD22			22	0	M7SCL	M7SCK	SWO	GPIO22	UART2TX	UART3TX	CT22	NCE22	VCMPO		FPIO22			X	X
61	A7	C9	PAD23			23	0	M7SDAWIR3	M7MOSI	SWO	GPIO23	UART2RX	UART3RX	CT23	NCE23	VCMPO		FPIO23			X	X
62	A8	B9	PAD24			24	0	M7MISO	TRIG3	SWO	GPIO24	UART0RTS	UART1RTS	CT24	NCE24	MNCE0_0	MNCE0_1	FPIO24			X	X
63	N10	T6	PAD25			25	0	M2SCL	M2SCK		GPIO25		UART1TX	CT25	NCE25			FPIO25			X	X
64	M10	R6	PAD26			26	0	M2SDAWIR3	M2MOSI		GPIO26		UART1RX	CT26	NCE26	VCMPO		FPIO26			X	X
65	N9	P6	PAD27			27	0	M2MISO	TRIG0	MNCE3_0	GPIO27		UART1CTS	CT27	NCE27			FPIO27			X	X
66	H12	N4	PAD28	SWO	0	28	0	SWO	VCMPO		GPIO28	UART2CTS		CT28	NCE28			FPIO28			X	X
67	J11	M4	PAD29			29	0	TRIG0	VCMPO		GPIO29	UART1CTS		CT29	NCE29			FPIO29			X	X
68	K11	P3	PAD30			30	0	TRIG1	VCMPO		GPIO30	UART0TX		CT30	NCE30			FPIO30			X	X
69	N14	N2	PAD31			31	0	M3SCL	M3SCK	I2S0_CLK	GPIO31	UART2TX	UART2CTS	CT31	NCE31	VCMPO		FPIO31			X	X
70	M13	M2	PAD32			32	0	M3SDAWIR3	M3MOSI	I2S0_DATA	GPIO32	UART0RX	UART3CTS	CT32	NCE32	I2S0_SDOUT		FPIO32			X	X
71	M14	L2	PAD33			33	0	M3MISO	CLKOUT	I2S0_WS	GPIO33	UART2RX	UART2RTS	CT33	NCE33	DISP_TE	MNCE1_0	FPIO33			X	X
72	L14	N1	PAD34			34	0	M4SCL	M4SCK	SWO	GPIO34	UART0TX	UART2RX	CT34	NCE34	VCMPO	I2S1_CLK	FPIO34			X	X
73	L13	M1	PAD35			35	0	M4SDAWIR3	M4MOSI	SWO	GPIO35	UART2TX	UART3TX	CT35	NCE35	I2S1_SDOUT	I2S1_DATA	FPIO35			X	X
74	K14	L1	PAD36			36	0	M4MISO	TRIG0	MNCE3_0	GPIO36	UART0RX	UART1RX	CT36	NCE36	MNCE1_0	I2S1_WS	FPIO36			X	X
75	F2	G16	PAD37			37	3	MSPI0_10	TRIG1	32KHzXT	GPIO37	UART2RX	UART3RX	CT37	NCE37			FPIO37			X	X
76	G1	G15	PAD38			38	3	MSPI0_11	TRIG2	SWTRACECLK	GPIO38	UART0RTS	UART2RTS	CT38	NCE38			FPIO38			X	X
77	F4	G14	PAD39			39	3	MSPI0_12	TRIG3	SWTRACE0	GPIO39	UART2RTS	UART3RTS	CT39	NCE39			FPIO39			X	X
78	D2	G13	PAD40			40	3	MSPI0_13	TRIG1	SWTRACE1	GPIO40	UART0CTS	UART1CTS	CT40	NCE40			FPIO40			X	X
79	E4	F14	PAD41			41	3	MSPI0_14	TRIG0	SWTRACE2	GPIO41	UART0TX	UART1TX	CT41	NCE41	SWO		FPIO41			X	X
80	H1	J14	PAD42			42	3	MSPI0_15	TRIG2	SWTRACE3	GPIO42	UART2TX	UART3TX	CT42	NCE42			FPIO42			X	X
81	G2	H16	PAD43			43	3	MSPI0_16	TRIG3	SWTRACECTL	GPIO43	UART0RX	UART1RX	CT43	NCE43			FPIO43			X	X

ambiq

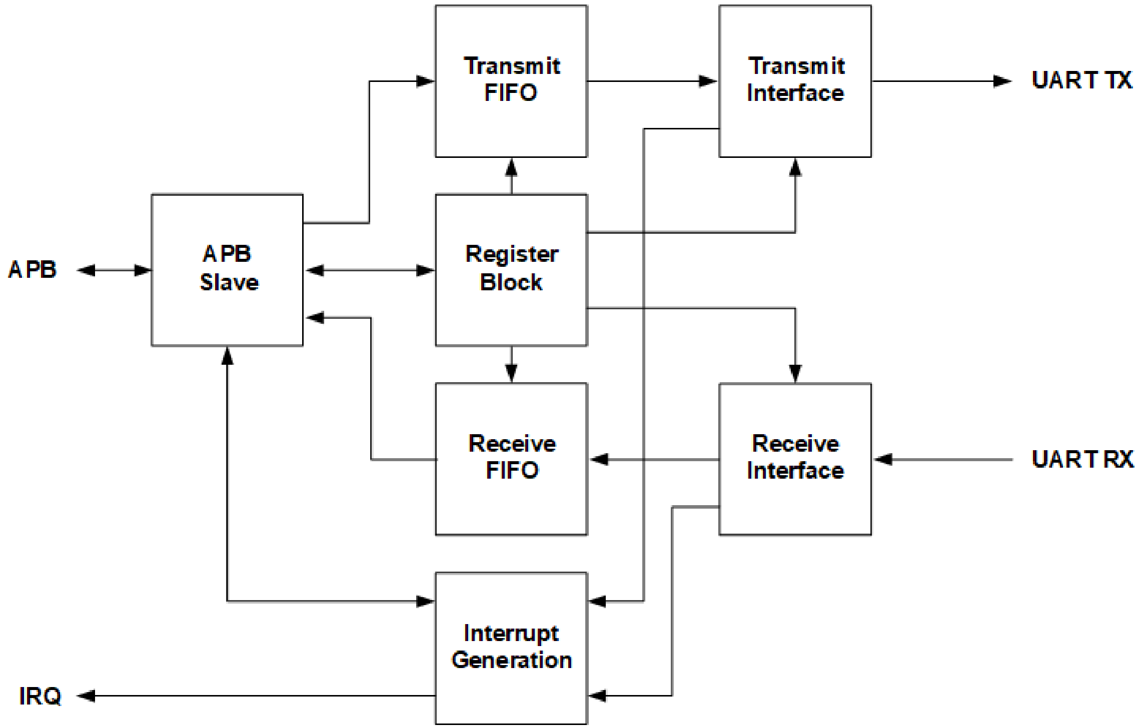
endpoint intelligence

UART

UART



- 4 UARTs
  - MCU can enter sleep mode during transfers
  - 32Byte Transmit and 32Byte receive FIFOs reduce MCU active time
- Configurable baud rate generator
  - Maximum rate of 3Mps
- Highly programmable
  - Data size, parity, and stop bit length
  - Hardware flow control
  - Full-duplex and half-duplex modes
- Loopback functionality for diagnostics and testing
- Support for full-duplex or half duplex communication
- Support DMA transfer (Half-Duplex)

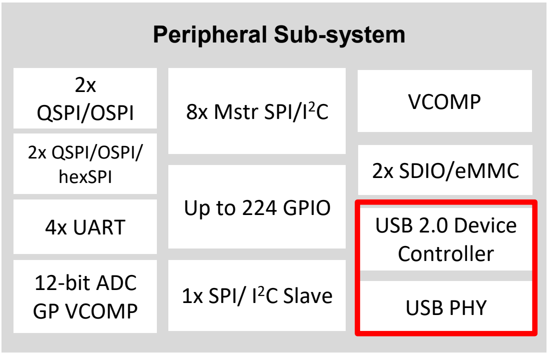


ambiq

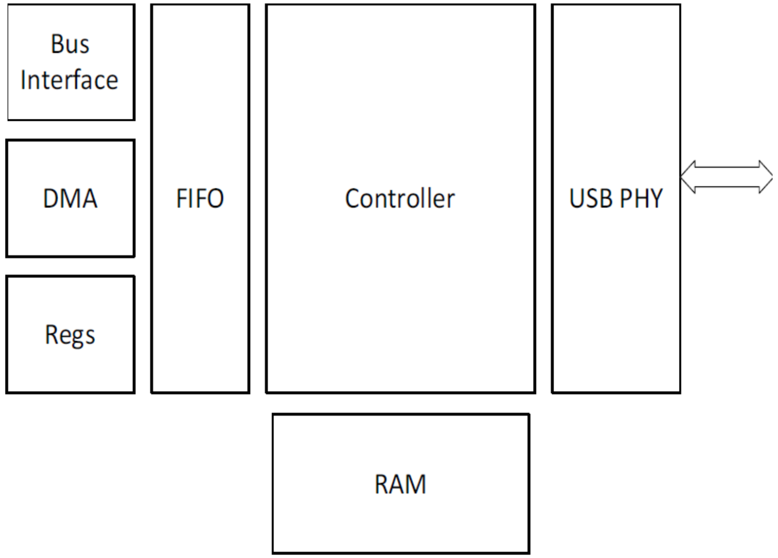
endpoint intelligence

USB

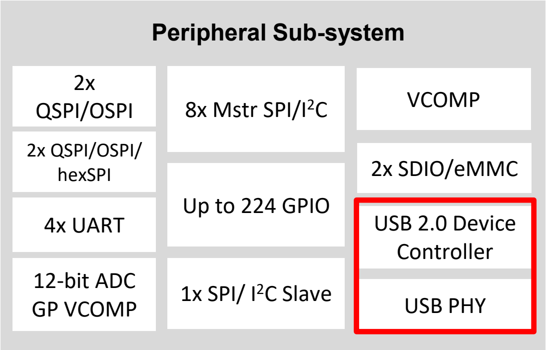
Universal Serial Bus (USB)



- USB 2.0 FS/HS PHY device with support for low-power mode
- Battery charging detection (BC1.2 & vendor-specific), with Interrupt driven weak battery/good battery algorithm for advanced power saving
- Crystal-less operation for FS mode
- USB HS requires an external HS crystal or clock input. Any frequency HS crystal allows use of PLL to generate USB clock, while 24MHz or 48MHz crystal or external clock can be directly selected as USB clock source
- On-die pull-ups/downs and termination (no external calibration resistors, pull-ups-downs required for USB operation
  - If USB 3.3v supply is provided locally while not connected to USB host, then 2M pull-down should be added to the D+ and D- lines to minimize leakage
- Dynamic FIFO sizing: 4 kB total FIFO
- 5 IN/OUT endpoints plus 1 control
- Soft connect/disconnect, Suspend mode
- Support concurrent DMA for all 5 IN/OUT endpoints



USB (cont’d)



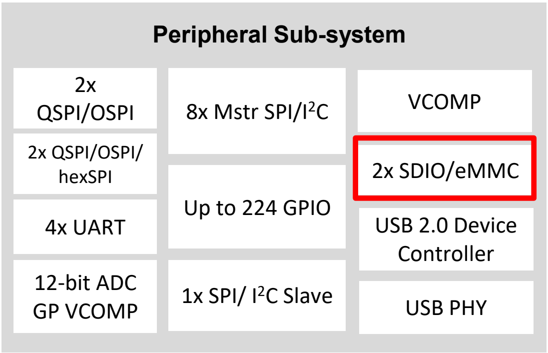
- Use of external low-jitter Crystal/Clock or new internal System PLL as clock source for USBPHYREFCLK addresses High-Speed USB clock jitter issue seen in Apollo4
  - Note: There is only 1 System PLL available in Apollo510. If the PLL is used for USB, it cannot be used for audio applications, and vice versa.
- Support double packet buffering for each endpoint to further improve performance by removing the need to wait for DMA/MCU to load/unload the FIFO before starting next transaction with host.
- USB Stack & HAL Upgrades:
  - USB HAL supports PIO and two DMA modes
    - DMA mode 0 interrupts CPU for each packet
    - DMA mode 1(Auto-DMA) interrupts CPU for each transfer for higher throughput
  - HAL handling for High-throughput DMA will be transparent to application layer while providing performance improvement according to the transfer size requested.
  - Enlarged MSC buffer to utilize DMA Mode 1 Auto-DMA for improving throughput and reducing CPU workload during large data transfer
  - TinyUSB stack (v0.18.0 in SDK 5.0.1 release)
  - Ambiq custom CDC Stack modification for application to assign its own buffer to CDC stack to eliminate unnecessary data copy operations. HAL will directly load/unload FIFO from/to application buffer.

ambiq

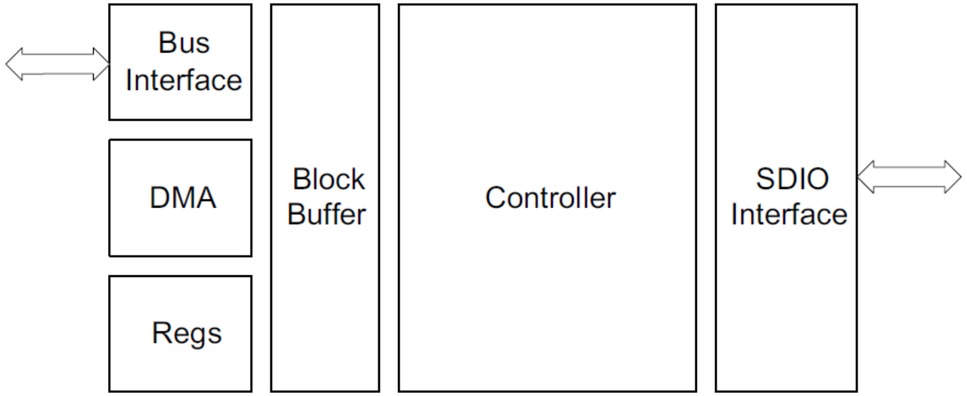
endpoint intelligence

SDIO

Secure Digital Input Output (SDIO)



- 2x SDIO controller instances
- SDIO card specification Version 3.0
- Host clock rate variable between 0 and 96 MHz
- Up to 50MBytes per second data rate using 4 parallel data lines (SDR50/DDR50 mode)
- Transfers the data in 1-bit and 4-bit SD modes(SDR50 or DDR50)
- Supports 8-bit eMMC data transfer in HS200 mode (up to 96MHz)
- Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
- Variable-length data transfers
- Supports Read wait Control, Suspend/Resume operation
- Supports up to 2 KB block buffering as well as dedicated DMA controller support to provide maximum host offload
- Programmable DLL to allow for timing tuning for optimal windowing
- Enhanced SDIO HAL to support different cards (eMMC, SD card and SDIO card) with card type auto-detection
- New drivers for RS9116 SDIO WIFI device and EMMC RPMB support included in latest SDK



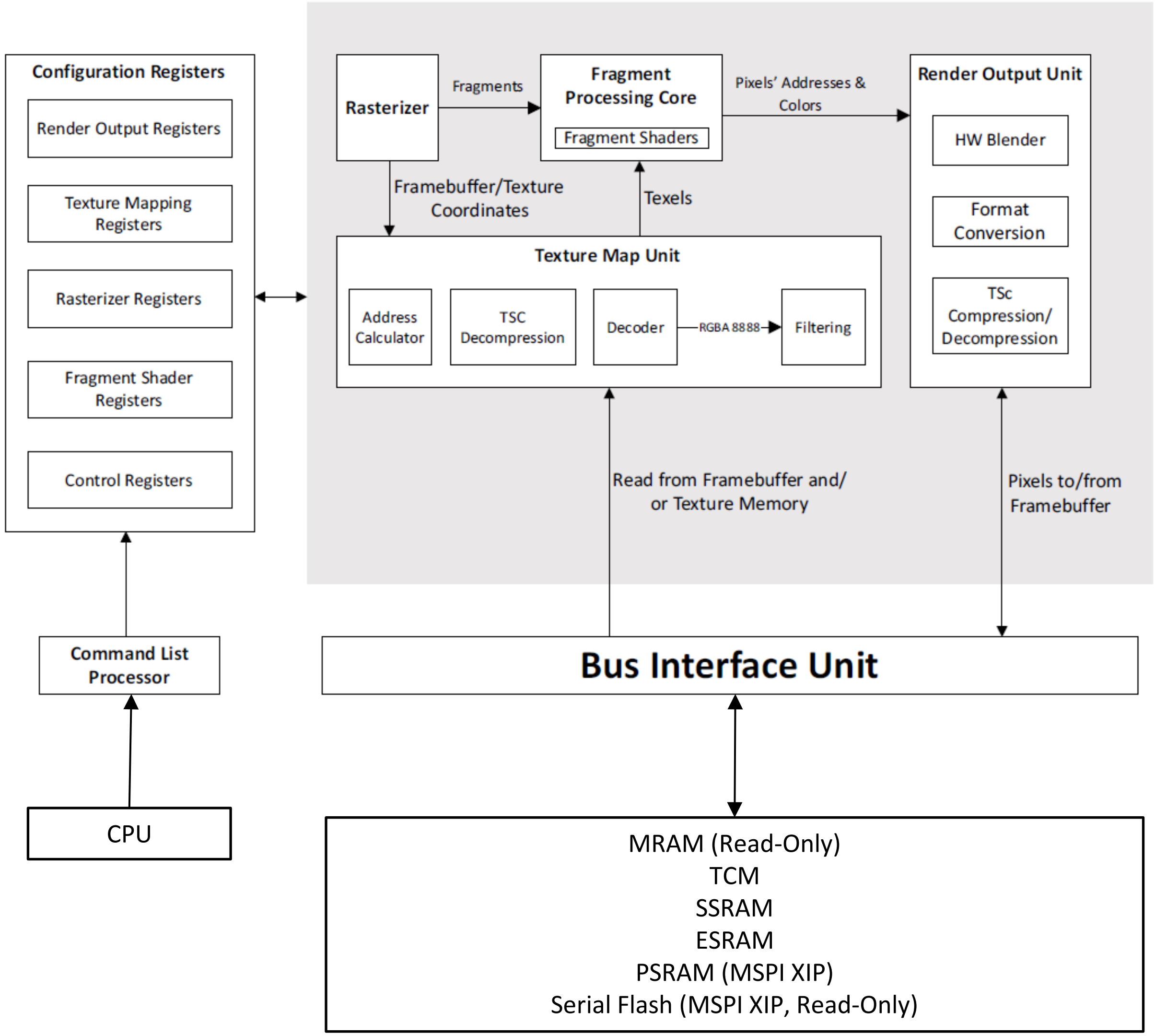
endpoint intelligence

Graphics / Display Subsystem

Graphics/Display Sub-System: GPU - Architecture

GPU

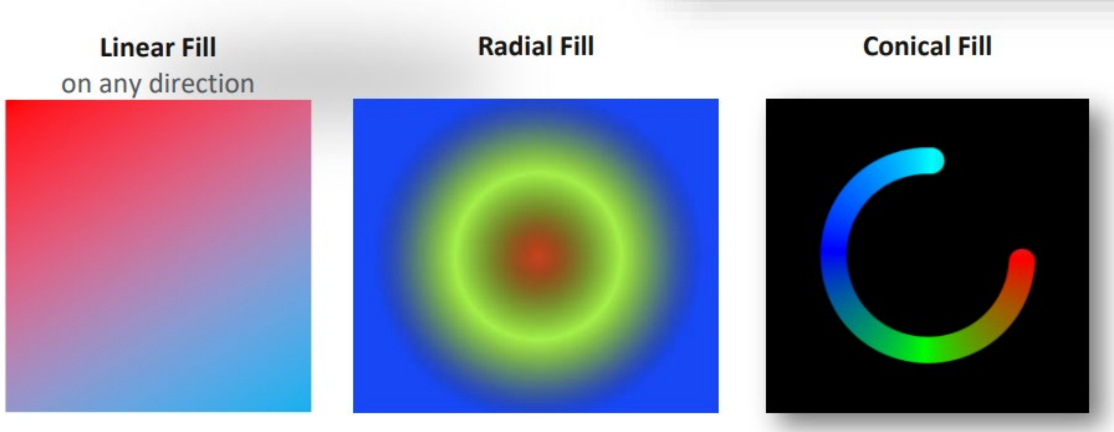
- GPU uses Command Lists to minimize CPU overhead
- GPU can directly access memory-mapped regions over Bus Interface, including TCM, SSRAM, and XIP mapped external MSPI PSRAM or Flash
  - MRAM as source is not supported on Apollo510products
- Support for inline texture decompression to minimize memory bandwidth/capacity requirements
  - Offline tool to compress textures/images in TSC formats TSC4 (4bits/pixel) and TSC6/6A (6bits/pixel) and TSC12/12A
- Support for inline frame buffer compression/decompression
  - Display controller supports inline decompression



Graphics Sub-System (Key enhancements on Apollo5 from Apollo4)

GPU

- GPU has two operating modes: LP and HP
  - LP (96MHz) is the default operating mode
  - HP (250MHz) can be selected by SW (via PWRCTRL regs)
  - Operating mode must be selected *before* GPU is powered up and cannot be changed until powered down.
- Vector Graphics acceleration: New Blocks offloading VG SW
  - Vertex Transformation
  - Bezier Tessellation
  - Bezier Draw
- Variable Burst Length: Boost transformations
  - Support up to 128B
  - Burst length configured as part of command list
- Gradient: Linear / Radial / Conical
- TSC High quality Compression
  - 12bpp with or without alpha
  - Frame-buffer: Real time on-the-fly (compression & decompression)
  - Texture:
    - Alpha support (TSc6a & TSc12a)
    - Real-time decompression,
    - Off-line compression: PixPresso



ambiq

Graphics/Display Sub-System: GPU – Raster Graphics

GPU

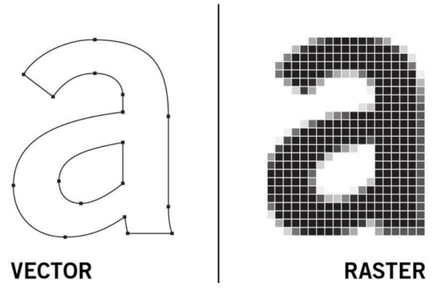
- 2D Primitives: More complex objects are constructed by multiple primitive objects
- Single pixels, Lines on any direction, Rectangles, Quadrilaterals, Triangles
  - Gradient Fills
  - Shaded Triangles (Gouraud Shader)
- Blit Support
- Rotation (any angle), mirroring, Stretch (independently on X and Y),
- Source/Destination color keying, Format conversions on the fly
- Font Rendering
- Alpha Blending
- Stencil: Masking with stencil maps can provide textured effects

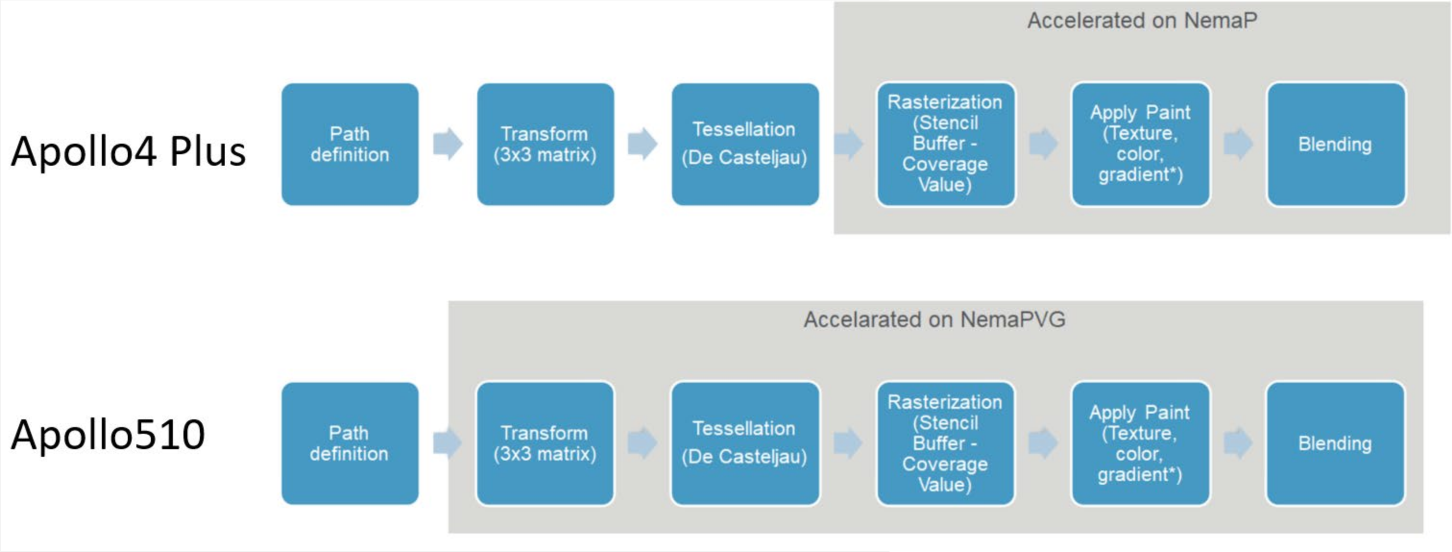
- 2D Color Keying
- Image Scaling
- Projective Rendering to provide illusion of depth
- Post Processing Effects
  - Blur, Sharpen, Edge Detection, Emboss
- Frame Buffer Compression – Real-time on-the-fly
  - Fixed rate, Lossy 4bpp, 6bpp with/without alpha
  - Real time on-the-fly
- Texture Compression - Real Time decompression, Off-line Tool for compression
  - Fixed rate, Lossy 4bpp, 6bpp with Alpha support
- Advanced GPU features of Apollo5
  - Anti-Aliasing – 8x MSAA multipoint sampling on edges
  - Dithering - Uses 2x2 Bayer pattern
  - Vector Graphics
  - Radial/Conical Gradient Fill Accel

Graphics Sub-System: GPU – Vector Graphics

GPU

- NemaVG APIs
- Vector Font Support
  - Raster fonts are images, vector fonts are paths/vertices
  - TTF is a vector font format, can be resized without losing quality
  - Offline TTF converter
- TSVG support
  - Scalable Vector Graphics (SVG) is an XML based vector image format for 2D graphics
  - TSVG is Think Silicon's binary representation of SVG files
  - Offline SVG to TSVG converter





Applications/Use cases

- Resolution independent rendering
- Fonts
- Maps
- Rounded geometries (Cubic/Quadratic Bezier curves, etc.)

Path Segment Types

- Lines
- Quadratic Beziers
- Cubic Beziers
- Arcs

Path Rendering

- Fill
  - Even/Odd
  - Non-Zero
- Stroking

Paint

- Color Fill
- Gradient
  - Linear / Radial / Conical
- Texture (Pattern)

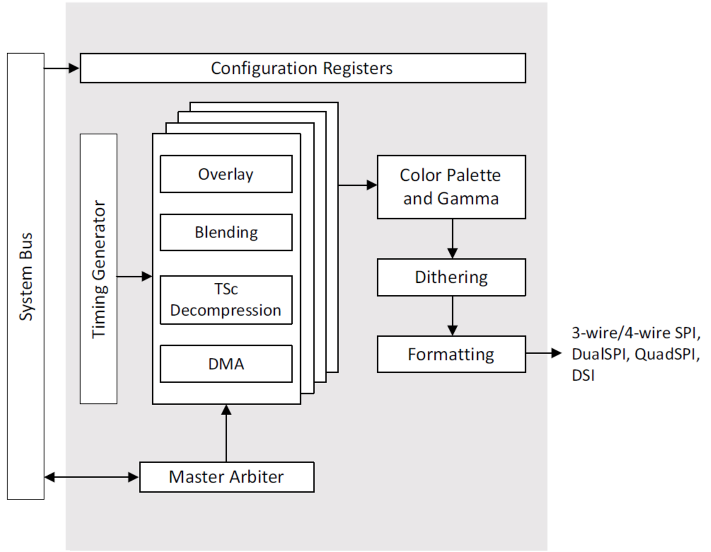
Render Quality

- Antialiasing
  - Per edge 8xMSAA (Hardware Accelerated almost zero cost)

Graphics/Display Sub-System

Display Sub-System

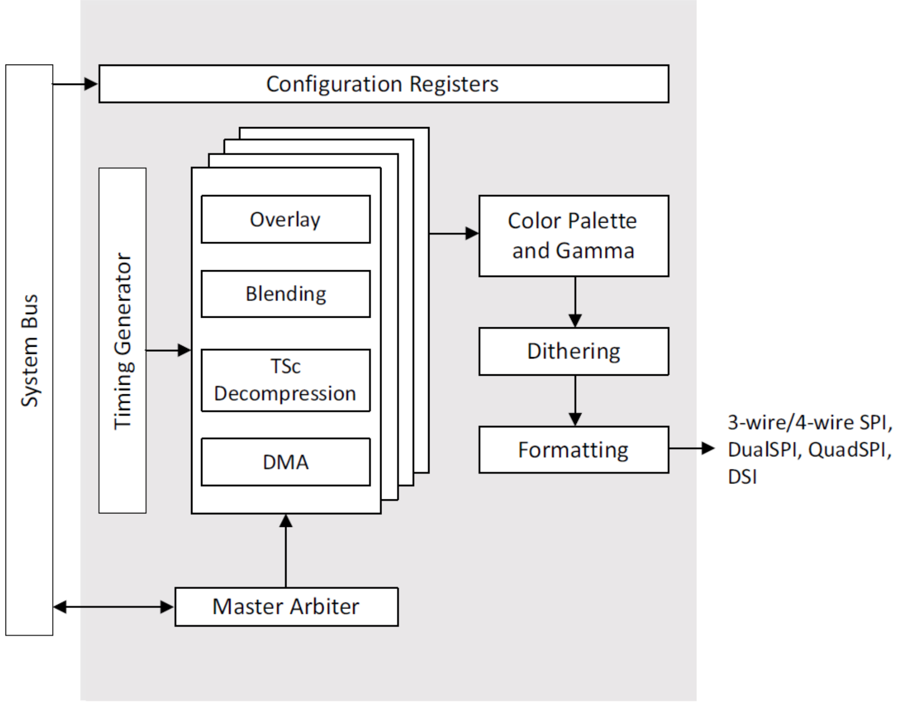
- Display Controller has DMA support to send data to display with minimal CPU overhead
- GPU and Display Controller are separate peripherals which can be powered on and configured independently
  - Frame Buffer can be located in internal SSRAM or in external PSRAM, and directly accessed by both GPU and Display Controller
  - GPU and Display Controller both can support Frame Buffer compression (TSC4 or TSC6/6a or TSC12/12a)
  - For non-standard display interfaces not supported by Display Controller, GPU can still be used to composite the frame buffer, and use alternate peripheral to send data to display



Graphics/Display Sub-System: Display Controller

Display Controller

- Display Interfaces:
  - MIPI DSI 1.2
    - Up to 768Mbps transfer rate (per lane, total 1.5Gbps bandwidth)
    - Single or Dual Lane
    - Command and Video mode
  - SPI, Dual-SPI or QSPI Up to 48MHz
  - DPI, up to 48MHz
  - DBI, up to 48MHz write, 12MHz read
  - MiP/JDI with FF support up to 48MHz
- DC layers (Frame-Buffers):
  - Support for up to 4 composition layers with alpha blending and scaling
  - Inline TSC4, 6/6A, 12/12A frame buffer, decompression on-the-fly
  - Frame Buffer may be in TCM, SSRAM, or external PSRAM (MSPI XIP)
    - Recommend locating Frame Buffer in internal SSRAM for optimal performance if space allows
    - If TCM is used for frame buffer, the CPU cannot be put in deep-sleep.



ambiq

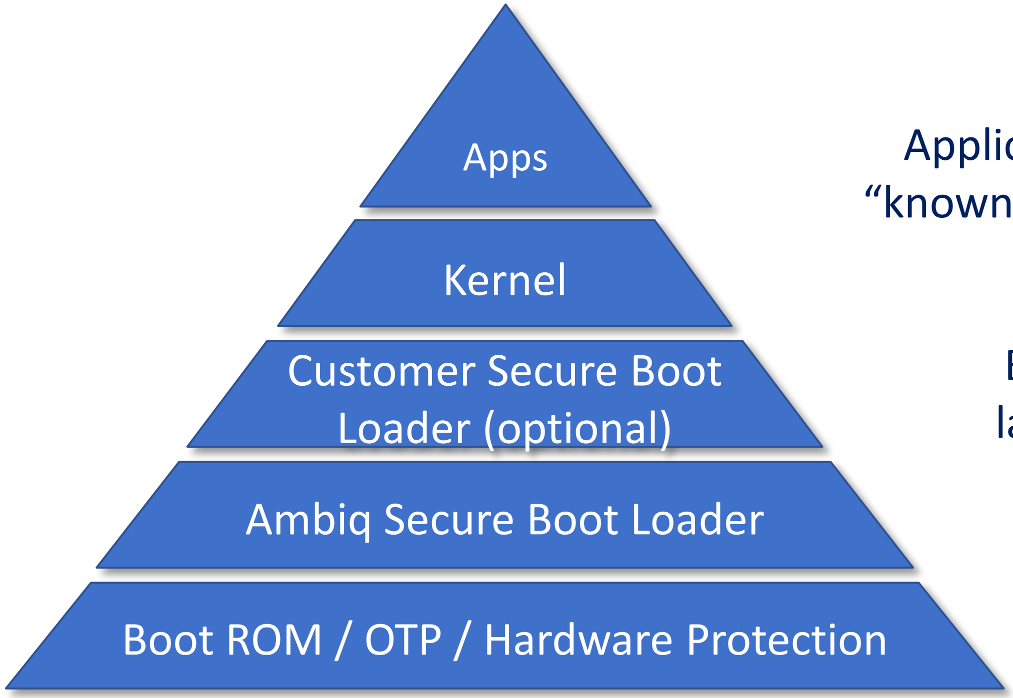
endpoint intelligence

Apollo5 Series Security

Apollo510Security Features

- Builds on and hardens Apollo4 Security Features
  - Secure Boot
  - Secure OTA / Wired Update
  - Secure Key Storage
  - Secure Debug
  - Hardware Crypto
  - Hardware TRNG
  - Lifecycle States
- eFuse OTP for Hardware Trims & Security config
  - Shuffle and address scramble for anti-tamper protection
- Secure Boot ROM (SBR) in metal ROM
- Secure BootLoader (SBL) in NVM
  - SBL update handled by SBR
- Physically Unclonable Function (PUF)
- TrustZone-M (coming soon)
- Secure MRAM Recovery
- Supports different Security SKU options for future
  - Ambiq-Secure (Default) - First 64K of MRAM is reserved for Ambiq SBL
  - No pre-installed Ambiq-SBL
    - Non-Secure
    - OEM-Secure

Apollo510Secure Boot



Applications run on “known good” platform

Building each foundational layer with verified firmware

Establishing Root of Trust

Boot Service – SBR and SBL

- Secure Boot ROM (boot ROM + SBR)
  - Authentication of Ambiq SBL using SecureBoot certificate to continue to boot flow
  - Provisioning, LifeCycle Management, Secure Debug Certificate Processing
  - Trim Updates (INFO1), SBL Update, MRAM (SBL) Recovery
- Ambiq SBL (Secure Bootloader)
  - Verification of Customer images, Transfer of control to OEM firmware
  - (Secure) Updates (OTA), Wired update, INFO0 updates
  - Other Value-added features
    - Cert chain update
    - Key Revocation
  - MRAM (OEM image) Recovery

Default (Shipped from Ambiq) Parts

- LCS – Device Manufacturing (DM)
- Un-provisioned OEM INFOC
  - OEM INFOC fields are set to 0
- Device boots in non-secure mode
  - Main image assumed at default (0x410000) address
  - Debugger is open after bootloader exits
    - Debugger not allowed during bootloader
- By default, Wired Host connection is enabled with following settings
  - UART only
    - Using UART0, Pad 55 as Rx, and Pad 30 as Tx
    - Baudrate of 115200
    - SBL will timeout after 500 msec
- INFO0 can be programmed
  - SDK provides tools to generate (create_info0.py), and program INFO0 (jlink-prog-info0.txt, jlink-prog-info0-otp.txt)

HW Crypto and PUF

ARM CryptoCell-312 HW Crypto

- HW acceleration of many Symmetric and Asymmetric crypto functions, including:
  - AES, CRC32, Diffie-Hellman, ECC key generation, SHA1/224/256, RSA PKCS#1, RSA key generation
- Asset Protection
  - Secure asset provisioning, Image verification, and security lifecycle management
- TRNG
- Crypto SW support through ARM mbedTLS and cc312 libraries

PUF - Physically Unclonable Function

- OTP antifuses used for INFO, INFO0, and INFO1
- PUFuid® - 1 kilobit unique keys programmed into PUF array in OTP
  - Can be used as a source for a unique ID, root key or entropy source, which is different for each chip.
- PUFtrng® - True random number generator
  - Leverages 1 kilobits of PUF data for initial seeding

Trustzone-M support in Apollo510 (coming soon)

- CM55 enabled for TrustZone support
  - Security Attributes of Instruction address determines the security state of the processor
- Support outside of the core is limited
  - All other masters (DMA, GPU, DC) during runtime are treated as non-secure
- Secure Memory – Configurable using SAU
  - BootROM helper function region should be classified as Secure
  - Portions of ITCM, DTCM and SSRAM can be classified as Secure
  - Secure memory cannot be accessed by other masters – configured separately
- Protecting Secure Memory from external Nonsecure Access
  - TGU – TCM Gate Units (for ITCM, DTCM)
  - SSRAMPROT – SSRAM Protection (for SSRAM)

- Secure Peripherals on APB - Configurable using SAU & NVIC
- Recommended configuration for Secure Peripherals:
  - CLKGEN
  - CLKMGR (SW)
  - MCUCTRL
  - MRAM
  - PWRCTRL
  - RSTGEN
  - RTC
  - SPOTMGR (SW)
  - SSC*
  - STIMER
  - WDT

(*) New peripheral SSC (System Security Control) for global controls

endpoint intelligence

MRAM Recovery in Apollo510

MRAM Recovery to mitigate MRAM corruption

- Issue:
  - MRAM may be corrupted due to cumulative or instantaneous magnetic interference, or due to power instabilities at power-up or during MRAM programming
- Effects:
  - Ambiq Secure Boot Loader (SBL) and all customer code are susceptible to MRAM corruption under these conditions
- Mitigation in Apollo5 series devices:
  - MRAM reference cells, Ambiq Trims, and SBR located in ROM or OTP, protected from MRAM corruption
  - Customer INFO1 Trims/Security configuration and key assets may be located in OTP
    - MRAM INFO1 should only be used during development
    - OTP INFO1 should be used for production so the configurations and assets are protected
  - MRAM recovery mechanism which leverages the SBR ROM and OTP configuration settings to reload Ambiq SBL and then uses SBL to reload customer bootloader from external recovery image
    - Recovery images for SBL and customer bootloader can be located in eMMC, serial NOR (MSPI), or external host over wired interface (UART or IOS SPI)

Apollo510MRAM Recovery

- Triggered by:
  - SBR detecting corruption in SBL
  - SBL detecting corruption of the main customer application based on image authentication
    - only if device is configured for SecureBoot otherwise SBL does not perform checks
  - Customer's SBL detects corruption (and initiates recovery)
  - Manually triggered (by Application request or configurable GPIO pin)
- Recovery Options/Methods:
  - Generic MSPI driver to NOR flash (Serial NAND Flash not supported)
  - eMMC device to provide recovery images.
  - UART or SPI wired interface from Main Applications processor or from PC
- There will be multiple Recovery Images:
  - Ambiq Recovery Image + Certificates (~64KB)
  - Customer Recovery Image (customer bootloader) + Certificates (limited to 256KB)
  - Additional Customer recovery Steps/Images – Customer specific,
    - Additional Customer Main MRAM Assets (Main Image + Certificates + Other Assets)

ambiq

endpoint intelligence

Cortex M55 Profiling

M55 Performance Monitoring Unit

- 8 16-bit event monitoring counters
- Counters can be chained to form 4 32-bit counters
  - Counters can also be extended by SW with minimal SW overhead, to support up to 8 32bit (or larger) counters
- Configurable interrupts for counter overflow
- 180 PMU events
- More details can be found in the ARM CM55 TRM r0p2 chapter 15

Table 7-1: PMU events

Event number	Event mnemonic	PMU event bus bit	Event name
0x0000	SW_INCR	0	Instruction architecturally executed, condition code check pass, software increment
0x0001	L1I_CACHE_REFILL	1	L1 instruction cache linefill
0x0003	L1D_CACHE_REFILL	2	L1 data cache linefill
0x0004	L1D_CACHE	3	L1 data cache access
0x0006	LD_RETIRED	4	Instruction architecturally executed, condition code check pass, load
0x0007	ST_RETIRED	5	Instruction architecturally executed, condition code check pass, store
0x0008	INST_RETIRED	6	Instruction architecturally executed.
0x0009	EXC_TAKEN	7	Exception taken.
0x000A	EXC_RETURN	8	Instruction architecturally executed, condition code check pass, exception return.
0x000C	PC_WRITE_RETIRED	9	Instruction architecturally executed, condition code check pass, software change of the PC.

Utils and Example Support

- \utils\am_hal_pmu.*
  - am_util_pmu_enable/disable
  - am_util_pmu_init
  - am_util_pmu_get_profiling
- Examples:
  - \boards\apollo5_eb\examples\graphics\nemagfx_vg_test
  - \boards\apollo5_eb\examples\power\coremark_pro
- Event Definitions:
  - \CMSIS\ARM\Include\pmu_armv8.h

```c
#ifdef USE_PMU_PROFILING
    am_util_pmu_config_t pmu_config;
    am_util_pmu_profiling_t pmu_profiling;

    pmu_config.ui32Counters = VALID_PMU_COUNTERS;               // Enable all valid event counters

    pmu_config.ui32EventType[0] = ARM_PMU_CPU_CYCLES;
    pmu_config.ui32EventType[1] = ARM_PMU_CHAIN;                // Chain an odd-numbered counter with a preceding even-numbered counter to form a 32-bit counter.

    pmu_config.ui32EventType[2] = ARM_PMU_MEM_ACCESS;
    pmu_config.ui32EventType[3] = ARM_PMU_CHAIN;                // Chain an odd-numbered counter with a preceding even-numbered counter to form a 32-bit counter.

    pmu_config.ui32EventType[4] = ARM_PMU_L1D_CACHE_MISS_RD;
    pmu_config.ui32EventType[5] = ARM_PMU_CHAIN;                // Chain an odd-numbered counter with a preceding even-numbered counter to form a 32-bit counter.

    pmu_config.ui32EventType[6] = ARM_PMU_L1D_CACHE_RD;
    pmu_config.ui32EventType[7] = ARM_PMU_CHAIN;                // Chain an odd-numbered counter with a preceding even-numbered counter to form a 32-bit counter.

    //
    // By chaining counters in pairs,the counter range can be increased by halving the number of counters.
    //
    am_util_pmu_init(&pmu_config);
#endif
```

```c
#ifdef USE_PMU_PROFILING
        //
        // Enable performance monitoring unit
        //
        am_util_pmu_enable();
#endif
```

```c
#ifdef USE_PMU_PROFILING
        //
        // Stop Events Counters & Collect Profilings.
        //
        am_util_pmu_get_profiling(&pmu_config,&pmu_profiling);
        //
        // Disable performance monitoring unit
        //
        am_util_pmu_disable();
#endif
```

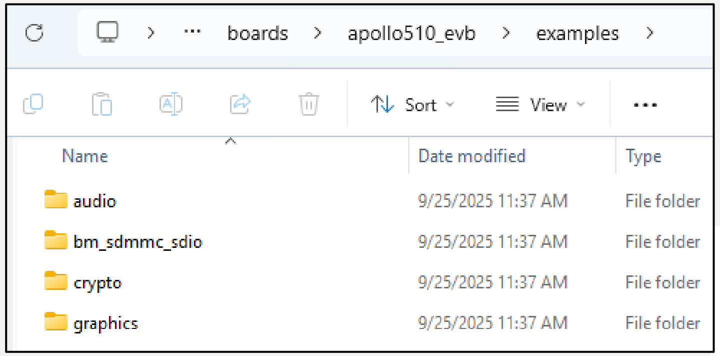
ambiq

endpoint intelligence

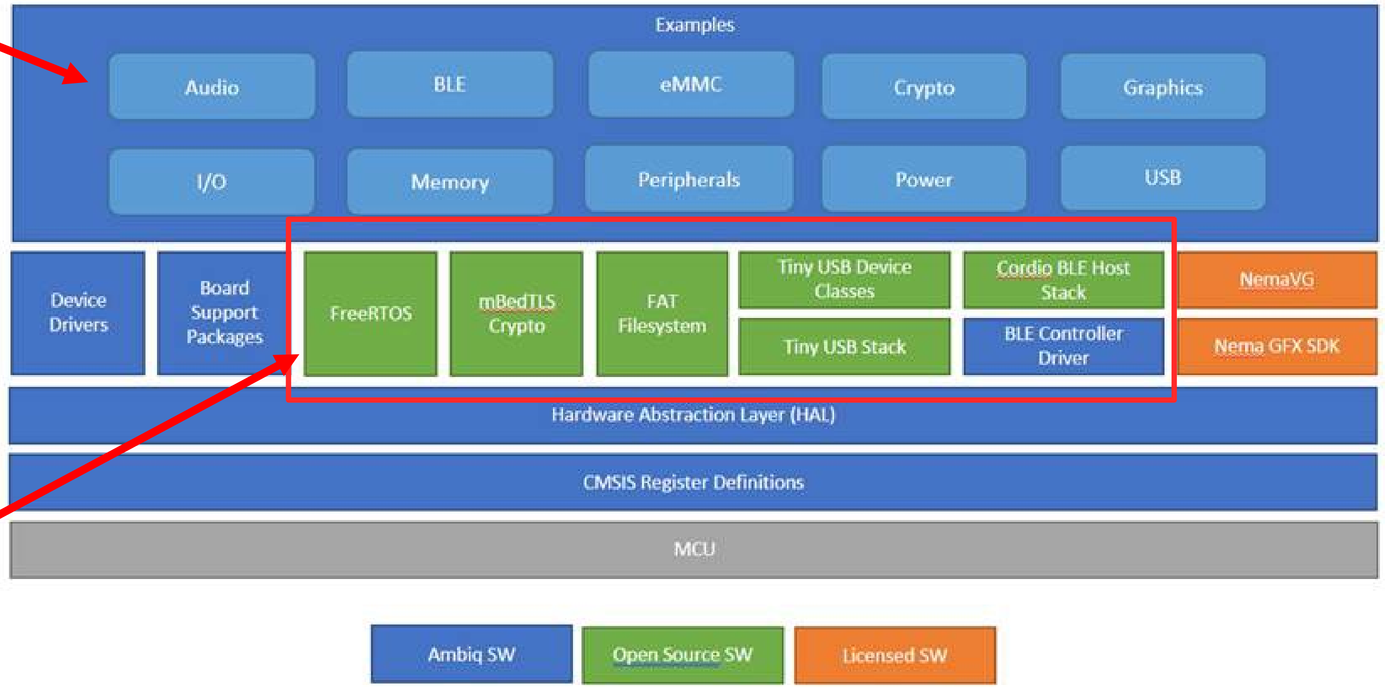
SDK Overview

AmbiqSuite SDK

SDK



Apollo510Family Software Stack



No customer license fees required

Open source license agreements included in SDK zip file

AmbiqSuite SDK Rev 5.0.1

SDK

- Register files (CMSIS compatible)
- HAL source code
- BSP source code for each Target
  - Apollo510_evb
- API documentation
- Development Tools supported:
  - SEGGER J-Link Software (v8.12g or later)
  - Keil uVision5 (ARM6 compiler; v5.42 or later)
  - GCC (v13.2.1 or later)
  - IAR (9.60.4 or later)

- Software examples
  - Audio (audadc, pdm, tdm loopback)
  - Crypto (PUFtrng®, mbedtls)
  - EMMC and SDIO
  - Graphics (lvgl, nemadc and nemagfx)
  - Interfaces (legacy, mspi)
  - MRAM programming
  - Peripheral Usage (adc, counter, rtc, timers and watchdog)
  - Power (coremark, deepsleep)
  - USB (tinyusb cdc and hid)
  - Keil, GCC, and IAR project files

ambiq

AmbiqSuite SDK Directory Layout

SDK

Boards

```
└── boards
    └── apollo510_evb
        ├── bsp
        └── examples
```

BSP

```
└── bsp
    ├── Makefile
    ├── am_bsp.c
    ├── am_bsp.h
    ├── am_bsp_pins.c
    ├── am_bsp_pins.h
    ├── bsp_pins.src
    ├── gcc
    ├── iar
    └── keil6
```

Examples

```
└── examples
    ├── audio
    │   ├── audadc_power_cycling
    │   ├── audadc_rtt_stream
    │   ├── i2s_loopback
    │   ├── pdm_fft
    │   ├── pdm_rtt_stream
    │   ├── pdm_to_i2s
    │   └── tdm_loopback
    ├── bm_sdmmc_sdio
    │   ├── emmc_bm_fatfs
    │   ├── emmc_raw_block_read_write
    │   └── emmc_scatter_raw_read_write
    ├── crypto
    │   ├── crypto_self_test
    │   ├── ecdsa_sign_verify
    │   ├── rng_puf_test
    │   ├── rng_test
    │   └── rsa_sign_verify
    ├── graphics
    │   ├── lvgl_benchmark
    │   ├── lvgl_cellular_menu
```

Top Level Directory

```
└── AmbiqSuite_SDK5.0.1
    ├── CMSIS
    ├── boards
    ├── devices
    ├── docs
    ├── makedefs
    ├── mcu
    ├── pack
    ├── third_party
    ├── tools
    └── utils
```

MCU
Hardware Abstraction Layer

```
mcu
└── apollo510
    ├── hal
    │   └── mcu
    └── regs
```

CMSIS Layer

```
cmsis
├── ARM
│   ├── Include
│   │   ├── arm_math.h
│   │   ├── arm_math_memory.h
│   │   ├── arm_math_types.h
│   │   ├── arm_math_v1.10.0.h
│   │   ├── cachel1_armv7.h
│   │   ├── cmsis_armcc.h
│   │   ├── cmsis_armclang.h
│   │   ├── cmsis_armclang_ltm.h
│   │   ├── cmsis_compiler.h
│   │   ├── cmsis_gcc.h
│   │   ├── cmsis_iccarm.h
│   │   ├── cmsis_version.h
│   │   ├── core_armv81mml.h
│   │   ├── core_cm4.h
│   │   ├── core_cm55.h
│   │   ├── dsp
│   │   ├── mpu_armv7.h
│   │   ├── mpu_armv8.h
│   │   ├── nn
│   │   ├── pac_armv81.h
│   │   └── pmu_armv8.h
│   └── Lib
│       └── ARM
├── AmbiqMicro
│   ├── Include
│   │   ├── apollo510.h
│   │   └── system_apollo510.h
│   └── Source
│       └── system_apollo510.c
└── version_cmsis_info.txt
```

endpoint intelligence

ToolChains

Compiler Support

- GCC 13.2.1*
  - Open Source
  - Gnu C Compiler for m55
  - Windows, Linux Support
  - Full C, C++ language support
  - CMSIS 5.9 compatible
  - Preliminary m55 Helium Support
  - Solid Performance for traditional applications.
- Keil MDK 5.42*
  - License Required
  - Arm Compiler 6 for M55
  - Windows, Linux Support
  - Full C, C++ language support
  - CMSIS 5.9 compatible
  - Full M55 Helium Support
  - Best performance for math heavy applications
- IAR EWARM: 9.60.4* also supported

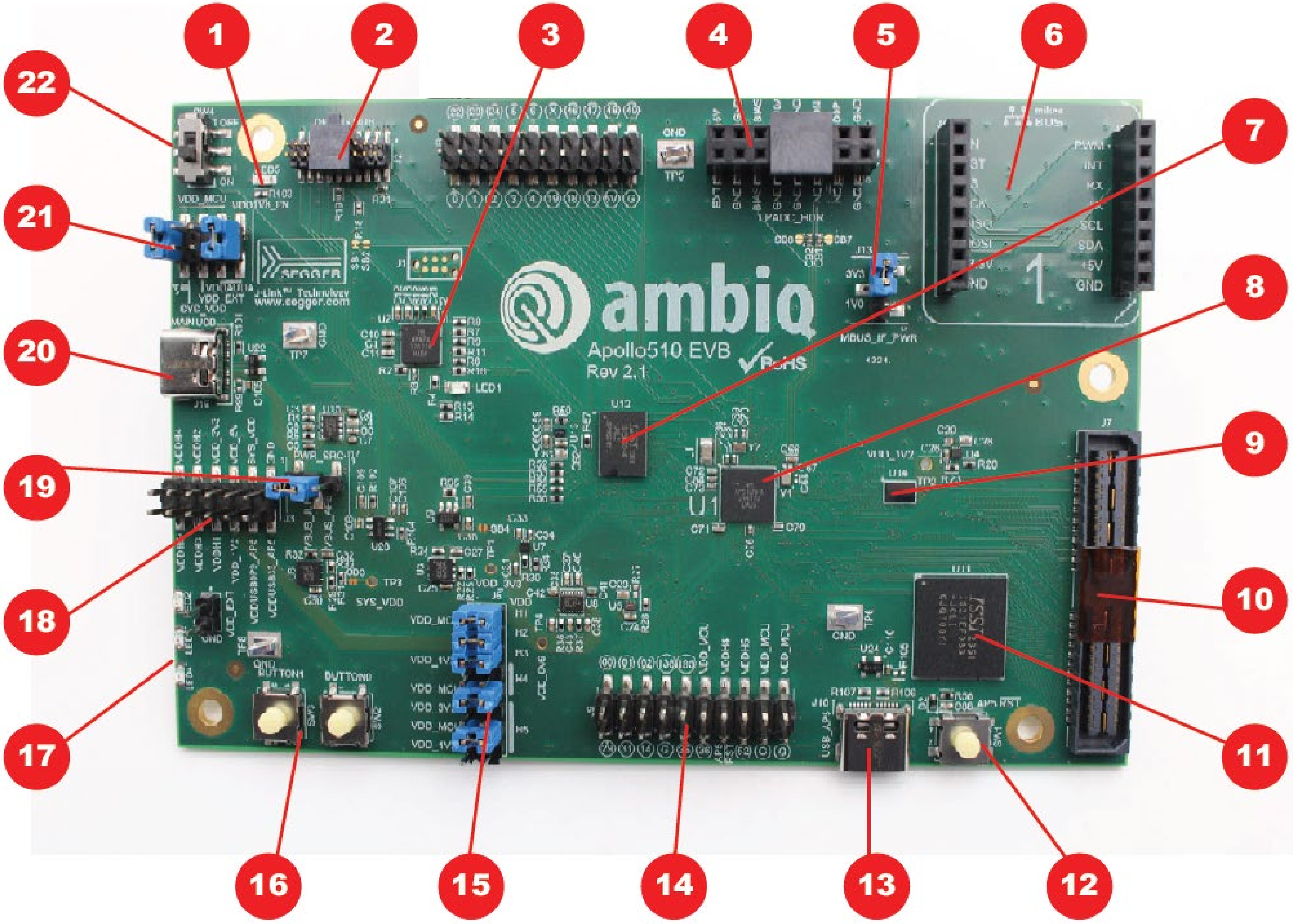
* Tool versions supported in SDK5.0.1 (Oct 2025). Please check release notes of SDK for latest tool version support

ambiq

endpoint intelligence

Evaluation Boards

Apollo510 Evaluation Board – Available now



1. Power LED
2. Debug in Connector
3. JLINK Controller
4. Voice-on-SPOT (VOS) Connector
5. MikroBUS I/F Voltage Header
6. MikroBUS Conectors
7. MSPI1 - IS25WX064-JHL
8. Apollo510 BGA MCU
9. MSPI0 - APS512XXB-AOB5NI-WA
10. High Speed Connector
11. eMMC0 - IS21EF08G-JCLI
12. Reset Switch
13. Apollo5 USB Connector
14. Dual-Row Headers (2x)
15. VDDHx Source Selection Header
16. User Switches (2x)
17. User LEDs
18. Power Test Points Header
19. Power Source Header
20. JLINK USB Comm
21. Power Config Header
22. Power Switch

endpoint intelligence

Q&A

endpoint intelligence

Thank **You**!