endpoint intelligence

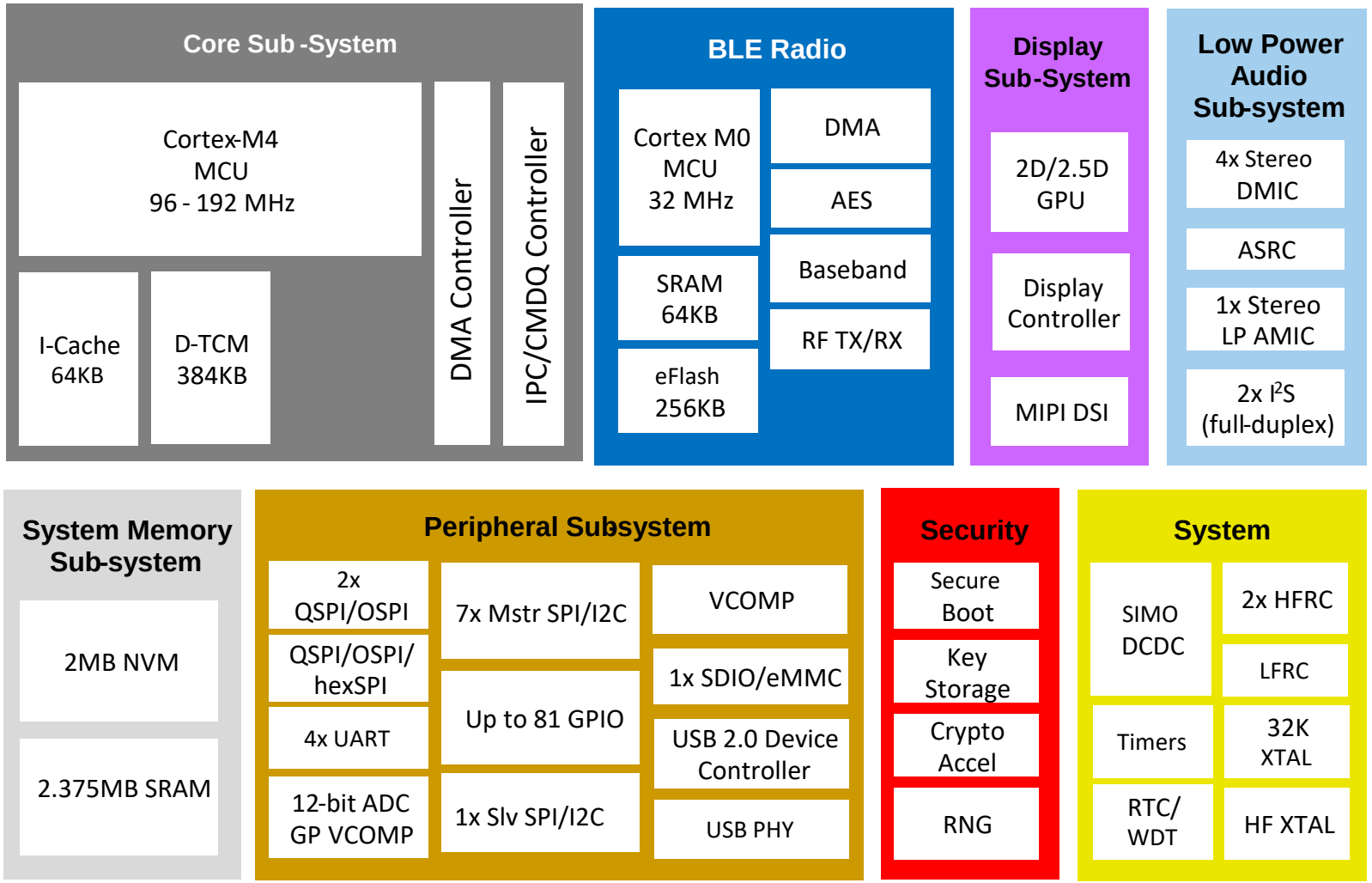
Apollo4 Plus/Blue Plus Product Training

June 2022

Apollo4 Plus/ Apollo4 Blue Plus Training Agenda

- Functional Overview
- Power Management
- Architecture and Peripheral Review
- AmbiqSuite SDK Overview
- Resources

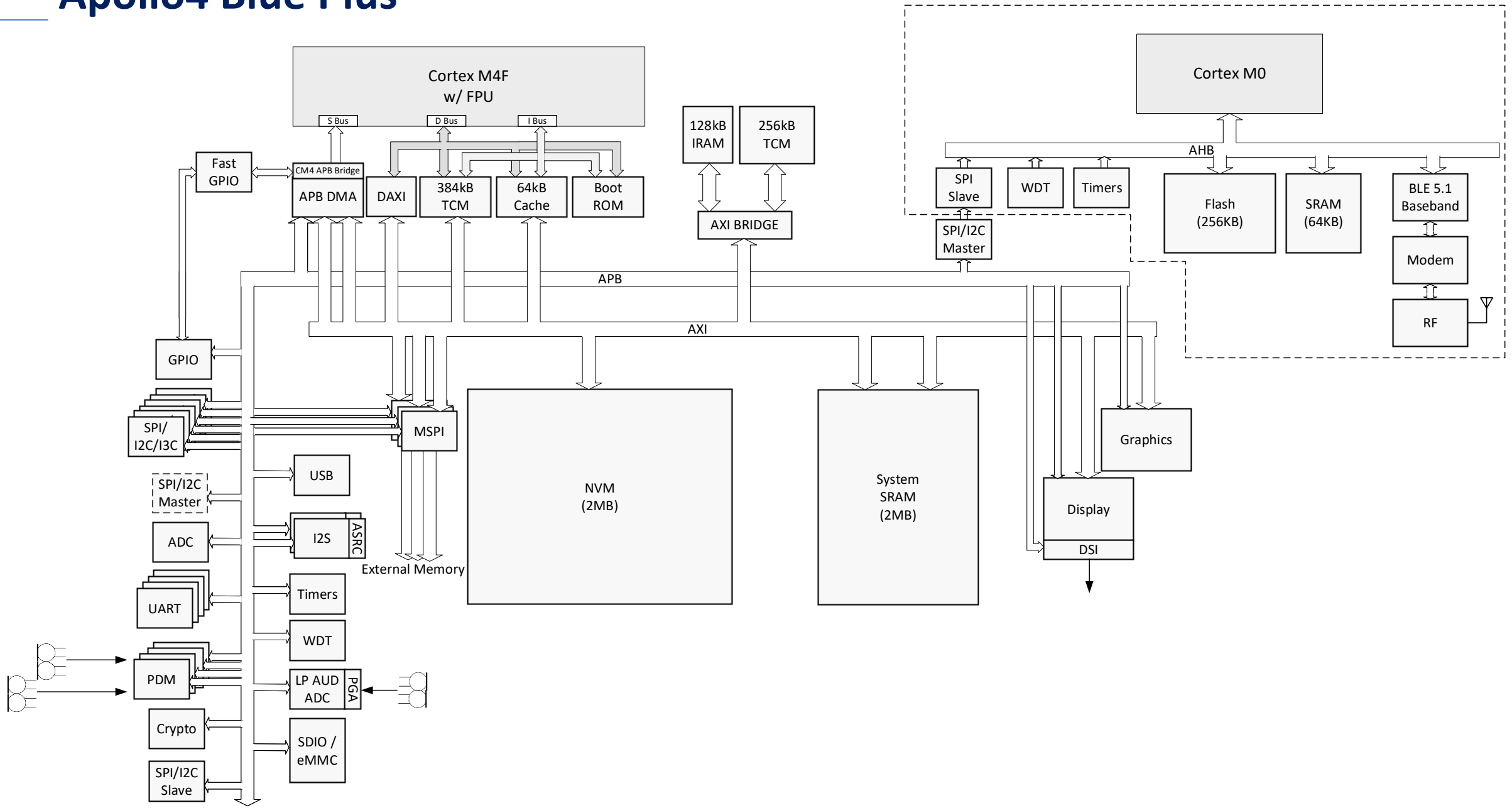
Apollo4 Blue Plus



Feature Highlights

- Achieving an unmatched 4µA/MHz executing from SRAM or MRAM with cache
- Up to 192 MHz clock frequency with TurboSPOT™
- 2D/2.5D graphics accelerator and MIPI DSI 1.2 with up to two lanes at 500Mbps per lane
- Apollo4 Blue Plus MCU is embedded with Bluetooth Low Energy 5.1 radio including Angle of Arrival (AOA) and Angle of Departure (AOD) for always-on applications
- Ultra-low power analog microphone for truly always on voice processing
- 4 PDM stereo channels, 2 full duplex I2S channels with ASRC, and an ultra-low power ADC for analog mics

Apollo4 Blue Plus



Product Comparison – Leap Forward

	Apollo3 Blue Plus	Apollo4 Blue Plus
	Cortex-M4F 48MHz/96MHz	Cortex-M4F 96MHz/192MHz
	0.75MB SRAM 16KB Code Cache	2.75MB SRAM 64KB Code Cache
	AHB (32bit)	AXI (128-bit) 32x data cache buffers
	“While Loop”: 6µA/MHz Coremark: 10.3µA/MHz Deep Sleep (no Ret): 1.2µA Deep Sleep (384K Ret): 3.7µA	“While Loop”: 4µA/MHz Coremark: 8.5µA/MHz Deep Sleep (no Ret): 7.7µA Deep Sleep (384K Ret): 11.8µA
	SW only composition	2.5D GPU + 4-layer DC 500 x 500 resolution; 60 fps Anti-aliasing Dithering Vector Graphics
	1x OSPI, 2x QuadSPI (up to 48MHz)	1x HexSPI, 2x OSPI (up to 96MHz)
	6x SPI/I2C (up to 48MHz)	8x SPI/I2C (up to 48MHz)
	N/A	USB 2.0 Device FS/HS 1x SDIO (50MB/s)
	SecureSPOT 1.0 (SW based)	SecureSPOT 2.0 (HW based)
	1x Stereo PDM	4x Stereo PDM 1x Stereo LP-AMIC 2x I2S w/ASRC
	BLE 5.0 (no optional features) Rx: -93dBm, Tx: +4dBm	BLE 5.1 (all features excl. coded PHY) Rx: -95dBm, Tx: +6dBm

ambiq

endpoint intelligence

Power Management

Power Management – Sleep

Power Management

- Apollo4 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.

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);
- Any Apollo4 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.

Power Management – Sleep

- The Binary counter example project is a good example of using LFRC as Timer count source, and going to deep sleep with that timer running. The timer then generates a periodic wake interrupt.
- Keeping timers running from either XT or LFRC add <100nA to deep sleep current (plus the current of the clock source, which is ~ 300nA for XT and < 100nA for LFRC).

Power Management – Sleep

Power Management

- All Apollo4 GPIO can be used as a wake source.
- To configure a GPIO as a wake source, you just need to enable the interrupt for that GPIO. Please see the deepsleep_wake project for simple example of this.
- All GPIO have configurable interrupts, and any GPIO that has interrupt enabled will wake the MCU from deep sleep upon GPIO interrupt.
- All Timers can be used as a wake source.

Power Management – Tips

Power Management

- The HFRC will be stopped automatically when entering deep sleep UNLESS there are peripherals or timers running that need HFRC.
  - For lowest power deep sleep, any timer or peripheral that is configured to use HFRC should be stopped before entering Deep Sleep.

Power Management – Tips

Power Management

- On Apollo4, peripherals such as UART, IOM (SPI/I2C master), IOS (SPI/I2C Slave), MSPI, ADC, I2S and PDM all have a peripheral enable/disable bits.
- All of these peripherals start disabled, which power-gates the peripheral for minimum leakage in deep sleep (exception: crypto which is enabled by default).
- Each of these peripherals must be enabled before they can be used. Upon setting the power enable bit, the peripheral will start in default state and needs to be configured.
- When going to deep sleep, any of these peripherals that is not active should be powered down to minimize deep sleep current.
- The AmbiqSuite HAL includes power_control functions that automate this process by saving config setting before powering down the peripheral, and write the saved settings to the peripheral when powering it back on.
- Note: some peripherals must be configured for minimal “active idle” power. Default settings will not be power optimized.

Power Management – Tips

Power Management

- To enable the MCU core to be put to sleep while peripherals are active, most Apollo4 peripherals have FIFOs, and many also have DMA and command queue support.
- Peripherals with FIFOs:
  - IOM I2C/SPI Master: 32Bytes TX and 32Bytes RX
  - MSPI: 32-entry FIFO (32 Bits Wide)
  - IOS I2C/SPI Slave: 256 Byte LRAM (Direct access / FIFO)
  - ADC/AUDADC: 16-deep FIFO
  - PDM: 32 Words
  - I2S: 64 entry RX and TX
  - SDIO: 2KByte buffer
  - USB: 4KByte buffer
- Peripherals with DMA support:
  - APBDMA: IOM I2C/SPI Master, MSPI, ADC, AUDADC, PDM, I2S
  - Dedicated internal DMA: SDIO, Crypto, GFX, Display Controller
- Peripherals with Command Queue support:
  - IOM I2C/SPI Master, MSPI

Power States

Power Management

- Apollo4 Supports the Following Four (4) Power Modes
  - High Performance (TurboSPOT™ or HP) Active Mode
  - Low Power (LP) Active Mode
  - Sleep Mode
  - Deep Sleep Mode
- High Performance or TurboSPOT Active Mode
  - CPU is in active state with 192MHz core clock
  - Tightly Coupled Memory (TCM) and CPU cache are also at 192MHz (CPU access remains single cycle)
  - All other frequencies in the SoC are unaffected
- Low Power Active Mode
  - Default active state
  - CPU is in active state with 96MHz core clock
- Sleep Mode
  - CPU is powered but clock gated
  - State of all other functions within the SoC are independent of the CPU
- Deep Sleep Mode
  - CPU and cache are powered down, register state in retention, cache state is optionally in retention
  - MRAM is powered down
  - TCM is still accessible to other DMA masters (powered down/retention when not accessed)
  - State of all other functions within the SoC is independent of the CPU

Power States – Active Mode

Power Management

- MCU core active with 96MHz core clock (LP mode - default)
- All MRAM and TCM start powered/enabled, but cache and all other SRAM start disabled
- 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 MCU VDD. Regardless of GPIO configuration, voltage applied to GPIO pin should not exceed VDD.
- 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
  - MRAM remains powered in standby mode
    - The LP Standby mode is recommended when long sleep/standby duration occurs or if workload can tolerate additional MRAM latency (high cache hit rates or high latency tolerant code execution)
    - MCU_CTRL:MRAMPWRCTRL:MRAMPWRCTRL asserted *and* MCU_CTRL:MRAMPWRCTRL:MRAMLPREN asserted
    - Note: LP standby mode must be de-asserted to revert back to normal standby
    - Note: MRAM accesses are still supported while MRAM in LP standby mode, however, accesses are about 5x longer than normal read accesses
  - Everything else that was active before the sleep command remains active. HFRC continues to run, but is gated from the core.

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
- 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
- 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 as long as they are using LFRC, XT, or low frequency external timer input as clock sources. These can definitely be left running in deep sleep. The XT and LFRC clocks are very low power.

ambiq

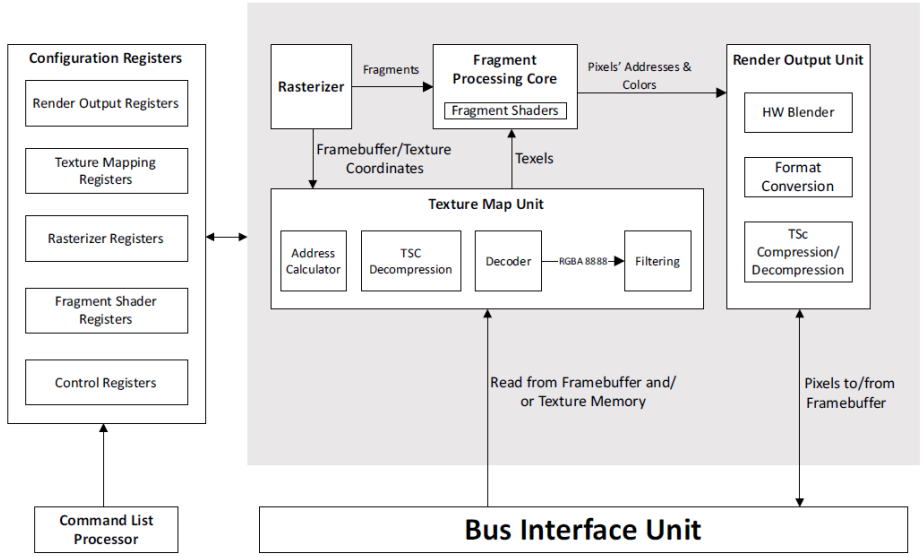
endpoint intelligence

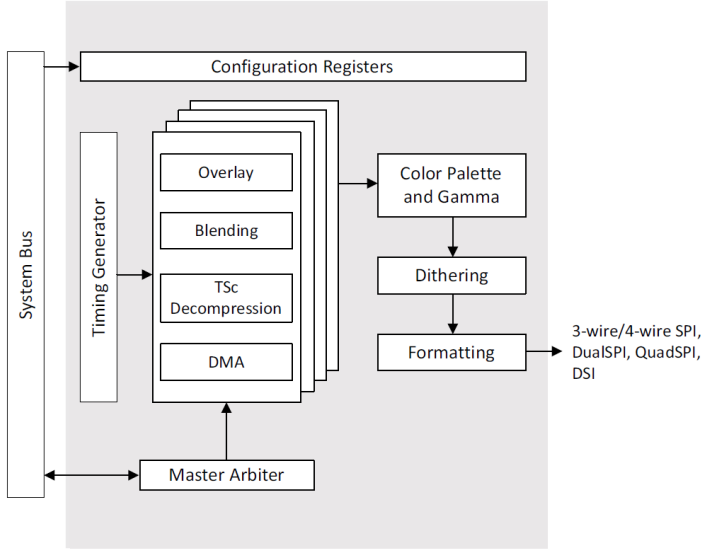
Graphics / Display Subsystem

Graphics/Display Sub-System

Display Sub-System

- GPU provides dedicated 2D/2.5D Graphics Acceleration with dedicated DMA
- 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)
  - 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

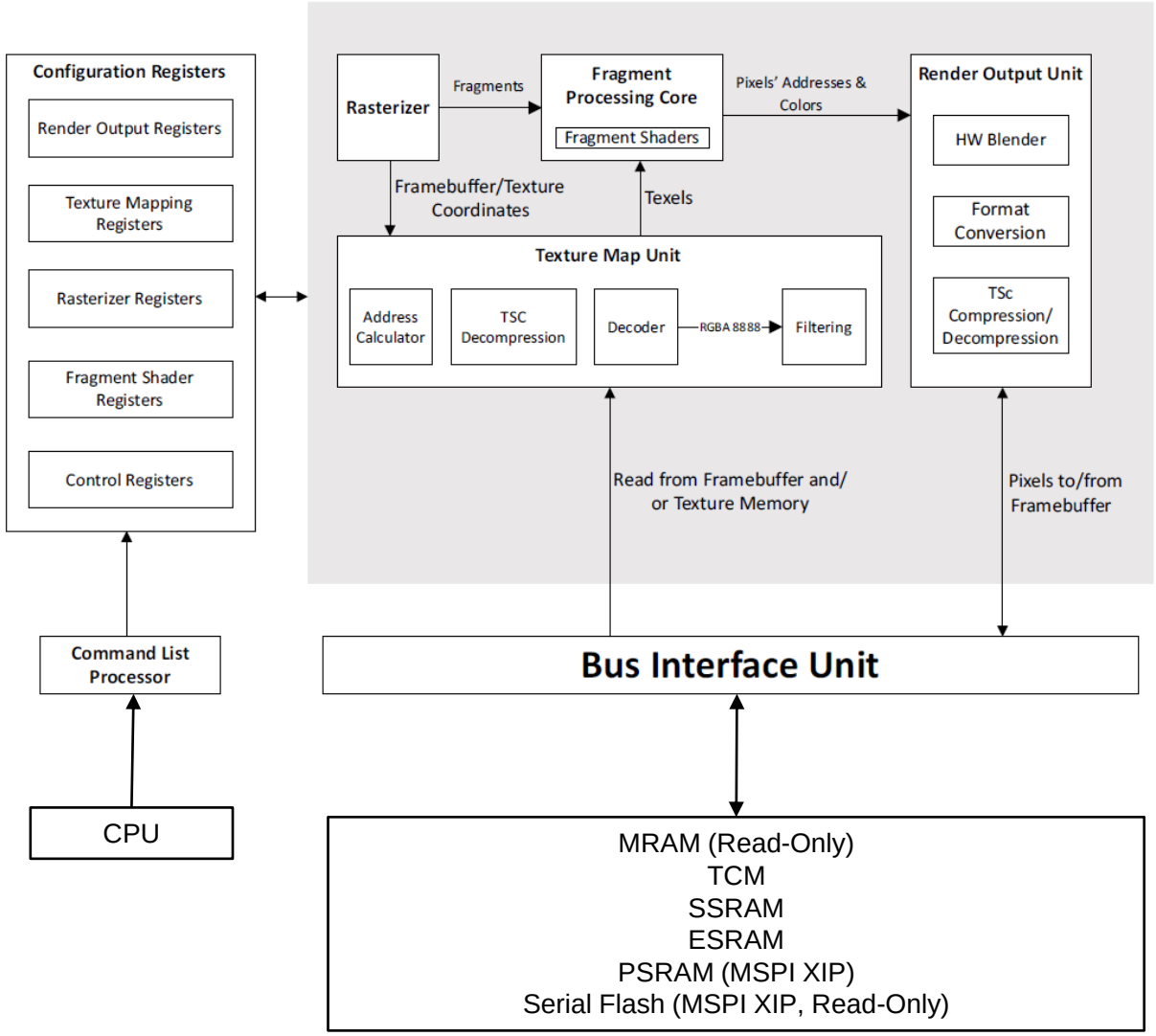




Graphics/Display Sub-System: GPU - Architecture

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 Apollo4 products
- 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)
- Support for inline frame buffer compression/decompression
  - Display controller supports inline decompression



Graphics/Display Sub-System: GPU - Graphics Features

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 Apollo4 Plus
  - Anti-Aliasing – 8x MSAA multipoint sampling on edges
  - Dithering - Uses 2x2 Bayer pattern
  - Vector Graphics

Graphics/Display Sub-System: GPU - Performance

GPU

- Example GPU Blit performance for Apollo4 versus Apollo4 Plus with texture/image assets located and Frame Buffer all located in internal SSRAM

Operations		Apollo4	Apollo4 Plus
		SSRAM	SSRAM
		RGBA8888	RGBA8888
		MPixels/sec	MPixels/sec
Asset in SSRAM	12: Blit	95.6	95.43
FB in SSRAM	13: Blit_Colorize	95.4	95.43
	14: Blit_Blend	44.5	94.95
	15: Blit_Blend_Colorize	30.12	94.95

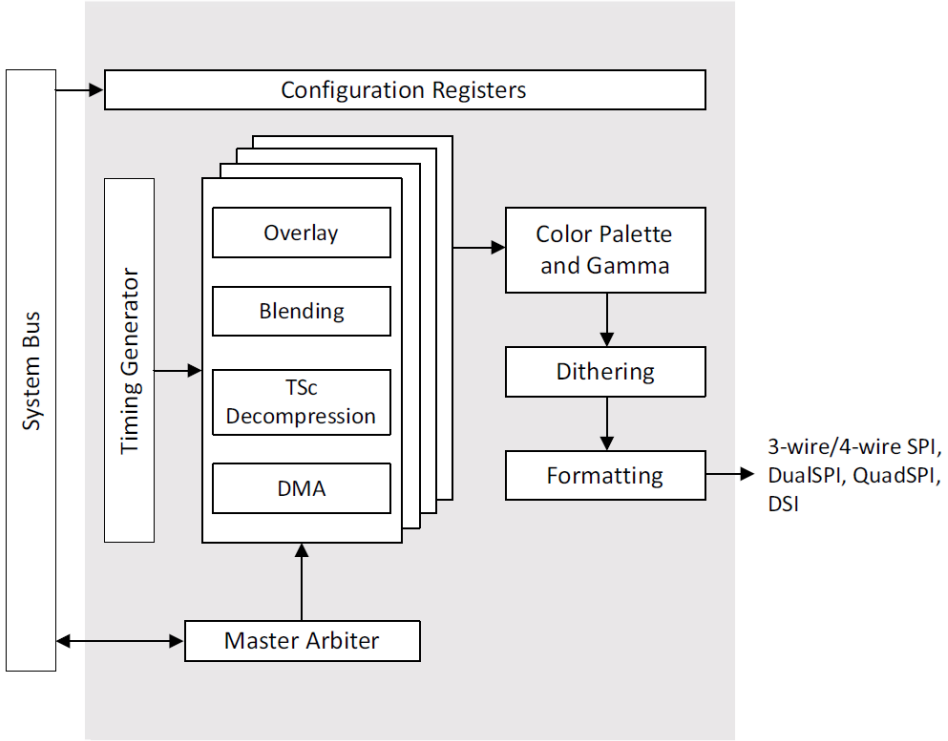
- Example GPU Blit performance for Apollo4 Plus with texture/image assets located in external PSRAM

Operations		Apollo4 Plus					
		OctalSPI			HexSPI		
		RGB565	TSC4	RGBA8888	RGB565	TSC4	RGBA8888
		MPixels/sec	MPixels/sec	MPixels/sec	MPixels/sec	MPixels/sec	MPixels/sec
Asset in PSRAM	12: Blit	37.35	41.81	18.69	61.2	63.79	30.64
FB in SSRAM	13: Blit_Colorize	37.35	41.81	18.69	61.2	63.79	30.64
	14: Blit_Blend	37.35	41.81	18.67	61.2	63.79	30.6
	15: Blit_Blend_Colorize	37.35	41.81	18.67	61.2	63.79	30.6

Graphics/Display Sub-System: Display Controller

Display Controller

- Display Interfaces:
  - MIPI DSI 1.2
    - Up to 500Mbps transfer rate (total bandwidth – not per lane)
    - Single or Dual Lane
    - Command Mode (Video mode not supported)
  - SPI, Dual-SPI or QSPI
    - Up to 48MHz
- Support for up to 4 composition layers
- Full alpha blending and scaling support
- Inline frame buffer using proprietary TSC compression Formats
  - TSC4 (4bits/pixel) TSC6 (6bits/pixel), TSC6A
- Frame Buffer may be located in TCM, SSRAM, or external PSRAM (MSPI XIP)
  - Recommend locating Frame Buffer in internal SSRAM for optimal performance if space allows



ambiq

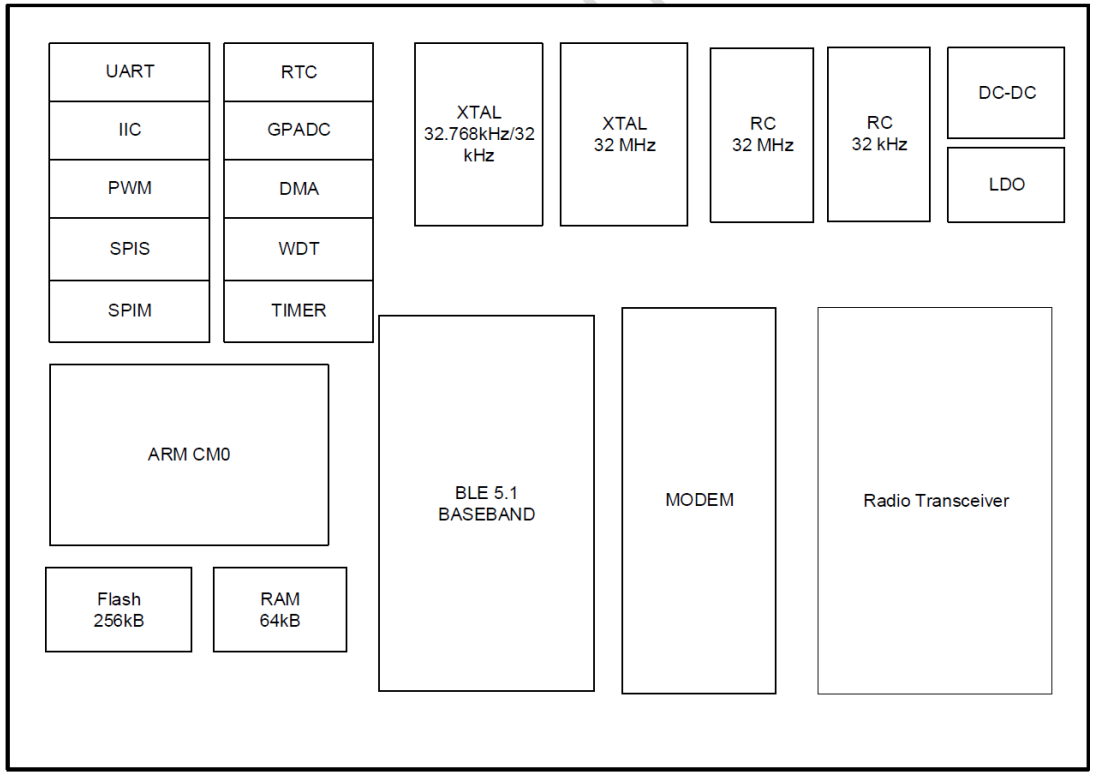
endpoint intelligence

Bluetooth Low Energy 5.1

Bluetooth Low Energy BLE 5.1

BLE

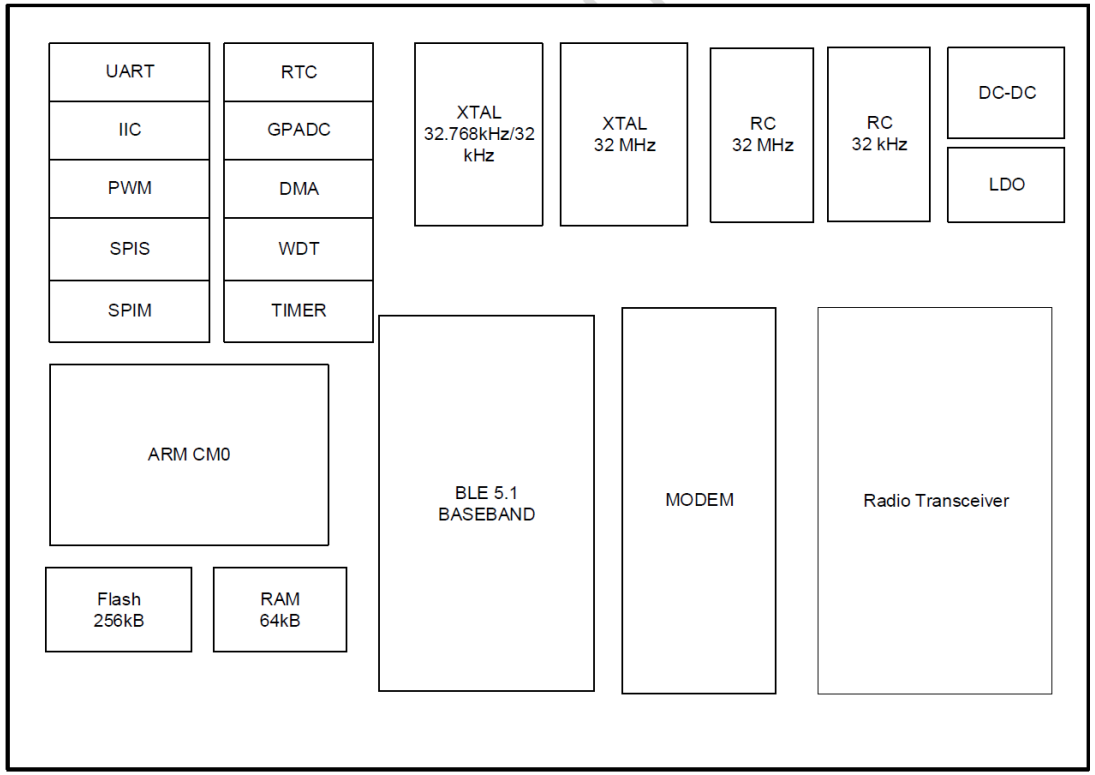
- Bluetooth 5.1 Compliant (QDID: 152224)
  - Secure connection (AES-128 HW Encryption)
  - Extended packet length support
  - Up to eight simultaneous connections
  - Supports all 5.1 features except coded PHY (long range) support
- -95 dBm sensitivity (@1Mbps)
- -10 to +6 dBm TX output power range
- Single ended output
  - Integrated Balun and antenna matching network
- 32MHz and 32KHz clocks supplied by MCU
  - Internal XTALs within Cooper IC unused
  - 32MHz XTAL startup requires MCU software to initiate



Bluetooth Low Energy BLE 5.1 cont'd

BLE

- Dedicated Cortex-M0 processor for BLE Controller
  - 256K Flash and 64K RAM
- BLE Controller Stack runs on BTLE controller
- HCI interface to main processor
  - HCI interface has FIFO, DMA, and command queue to minimize interruption of host processor
- Cordio-B50 BLE 5.1 Compliant Host Stack (QDID: 158224) runs on main CM4 processor
- Basic application including FreeRTOS, BLE Host Stack, plus basic BLE profile such as HRM requires approximately 80K NVM and 35K SRAM



ambiq

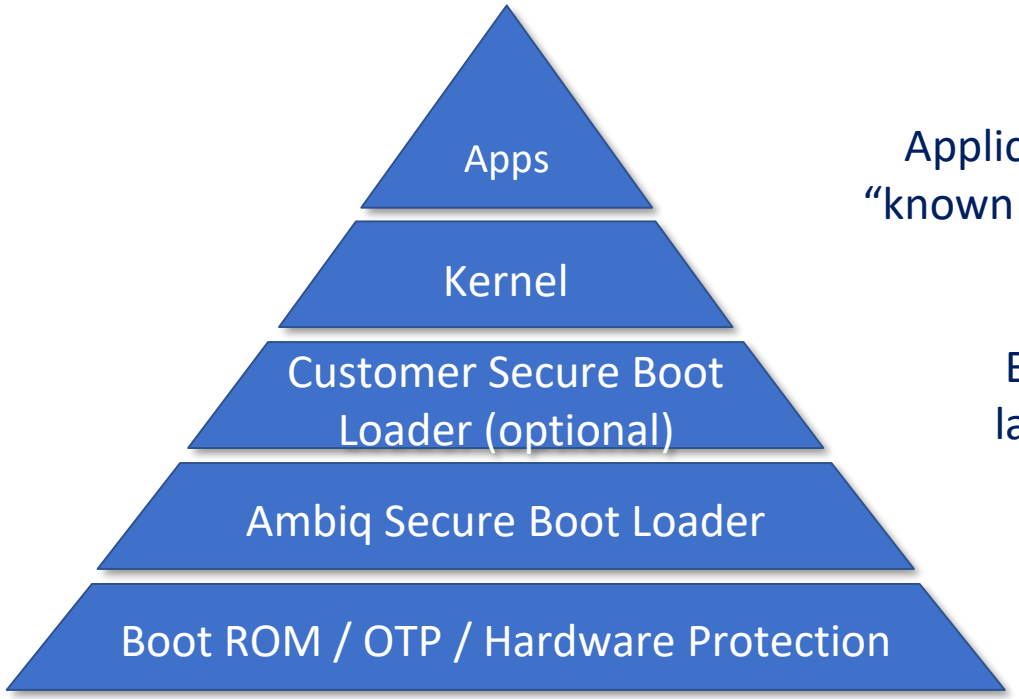
endpoint intelligence

Security

Apollo4 Security Features

- Secure Boot
- Secure OTA / Wired Update
- Secure Key Storage
- Secure Debug
- Hardware Crypto
- Hardware TRNG

Apollo4 Secure Boot

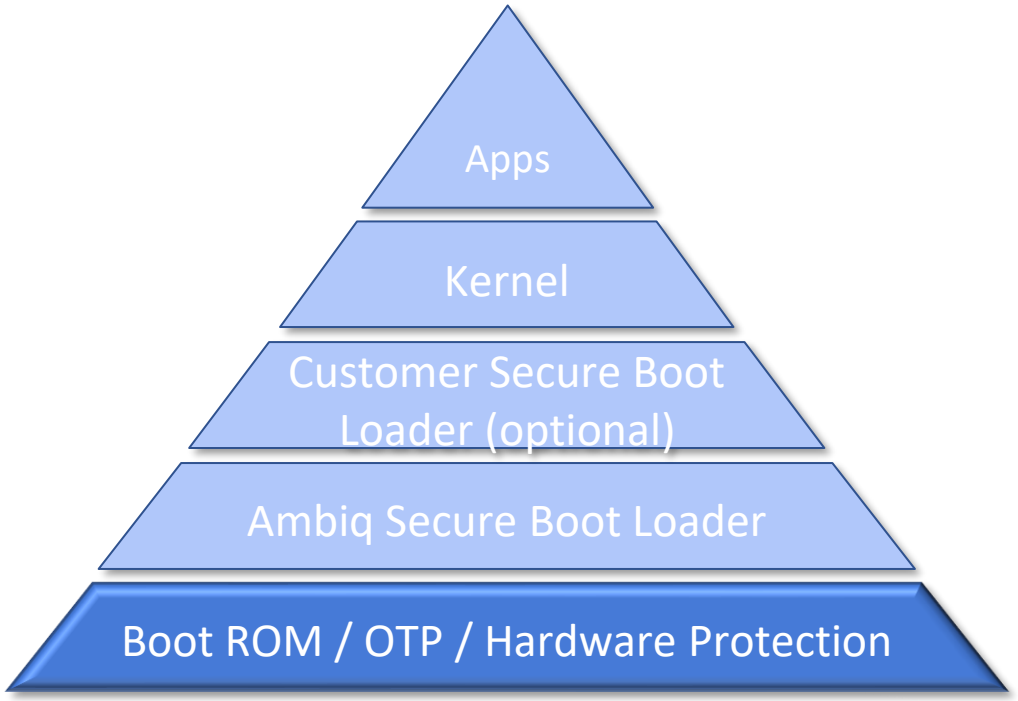


Applications run on "known good" platform

Building each foundational layer with verified firmware

Establishing Root of Trust

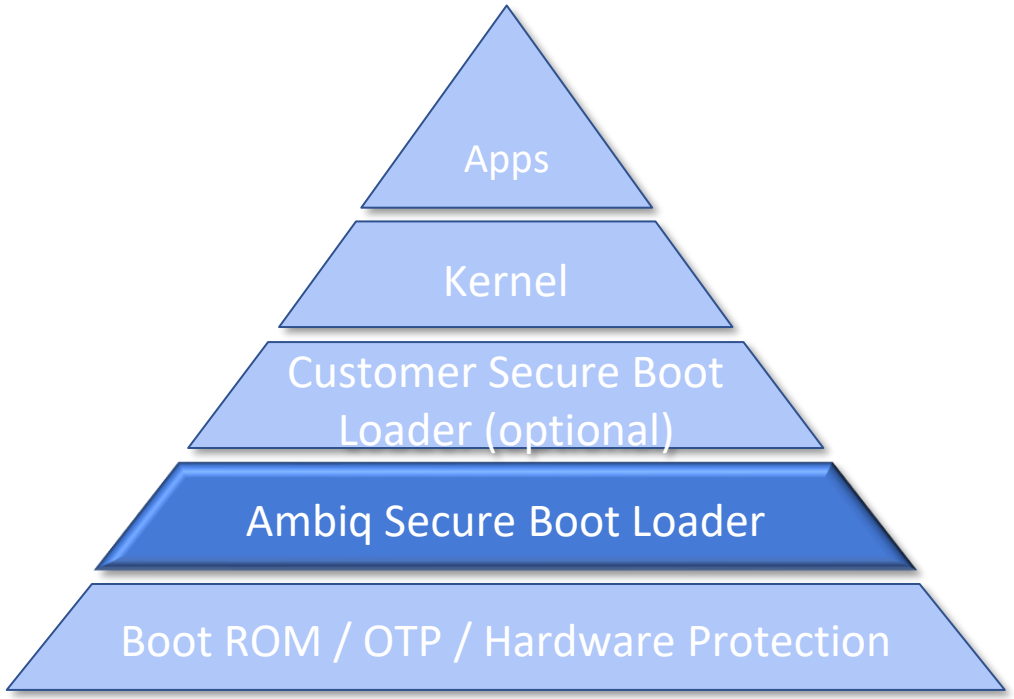
Apollo4 Secure Boot



Immutable Root of Trust

- Secure Boot ROM
- Hardware enforced
- Initiates chain of trust firmware validation using secure boot certificates
- Hardware crypto backed
- Lifecycle State management
- Establishes debug lockouts to prevent debug/tracing/interrupts during secure boot critical section execution based on secure debug certificates
- Hardware OTP key storage and security policy configuration

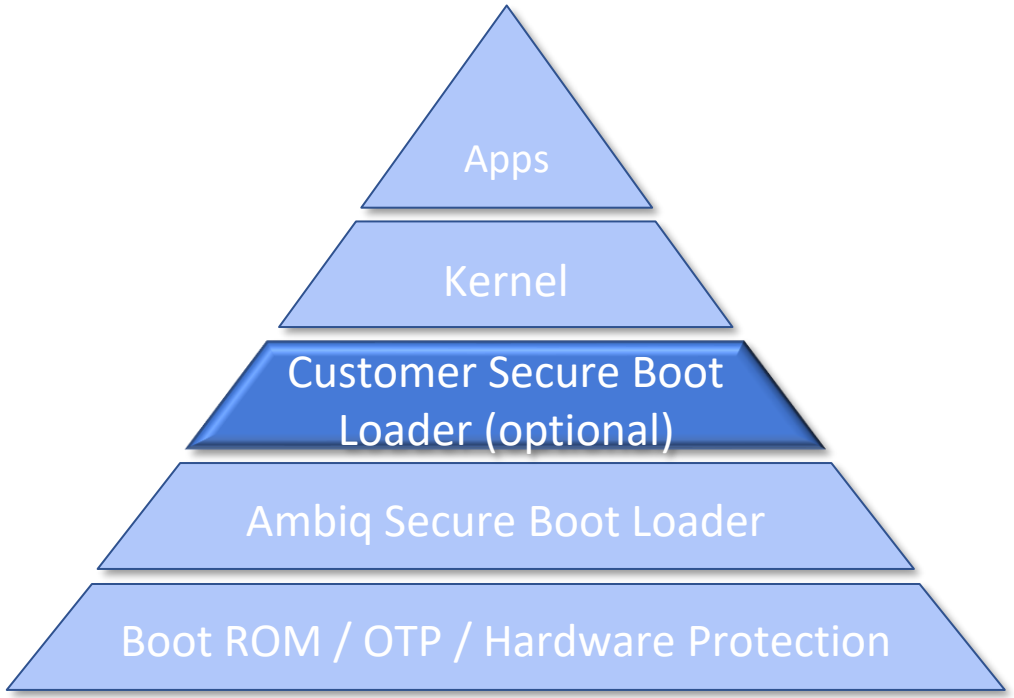
Apollo4 Secure Boot



Trusted Ambiq Firmware (SBL)

- Validated by Secure BootROM
- Processes chain of trust firmware validation using secure boot certificates
- Hardware crypto backed
- Support for secure firmware update (OTA, wired), secure firmware recovery
- Enforcement of NVM protections per security policy

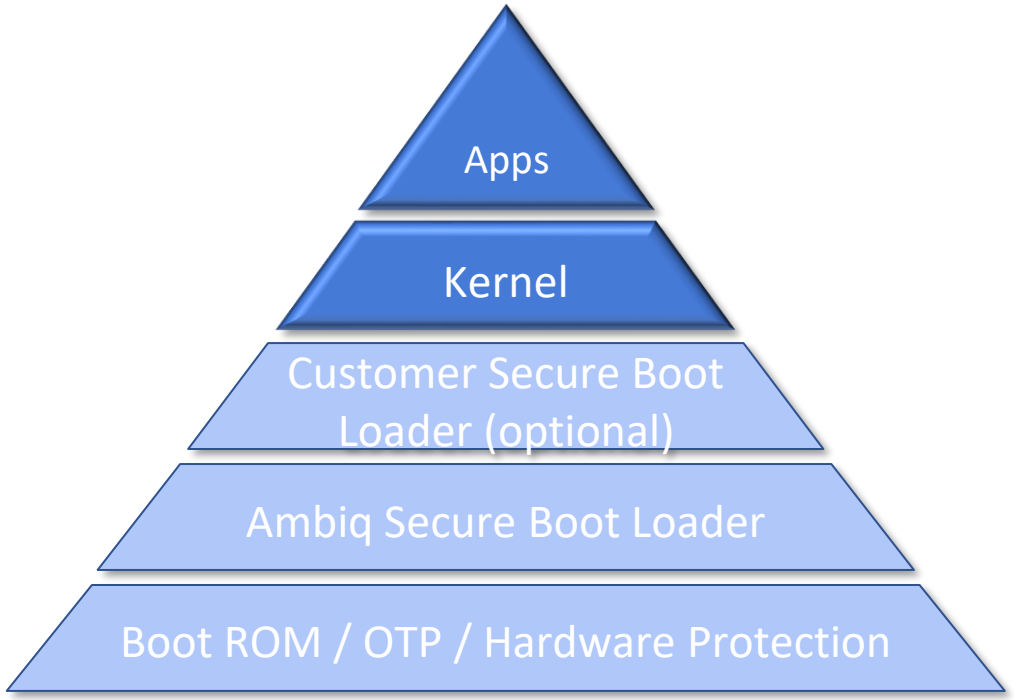
Apollo4 Secure Boot



Trusted Customer Boot Firmware

- Validated by SBL
- Processes customer main firmware validation using secure boot certificates
- Hardware crypto backed
- Enables extended features as needed by customer
- Proprietary signature validation
- SW image updates from external storage
- Custom security policy configuration (e.g. MPU)
- Enforcement of NVM protections per customer security policy

Apollo4 Secure Boot



Trusted Customer Main Firmware

- Validated by Ambiq SBL or Customer SBL

Lifecycle States

- Chip Manufacturing
  - Initial state of the part prior to silicon testing/trim/configuration by the silicon vendor (Ambiq)
- Device Manufacturing
  - This state represents the customer shipped state. Part has been trimmed/configured by Ambiq.
  - Certain facilities are still available to support customer debug and provisioning.
- Secure Enabled / Deployed
  - This state represents the part being deployed into production.
  - Device is locked/partially locked based on customer security policy
  - Customer debug is locked/unlocked based on customer security policy
- RMA
  - This state is entered when a part is returned to Ambiq and customer wishes to lock customer-specific assets.
  - Note: this state is a permanent state and part/locked assets cannot be recovered

Boot Service

- Initial execution begins in Secure Boot ROM
- Secure Boot ROM processes secure boot certificate and validates main Secure Boot Loader (SBL)
  1. Get the public key from the certificate and calculate its hash.
  2. Compare the calculated hash:
     - If it is the first certificate in the chain, compare it with the hash value stored in the OTP.
     - Otherwise, compare it with the saved hash from the previous certificate.
  3. Verify the RSA signature using the certificate's public key (RSA-3072bit[1])
  4. Verify the SW version, if exists, against the values in the OTP. Certificate must have a SW version bigger than or equal to the version saved in the OTP.
  5. Save the public key hash of the next certificate, unless it is the last certificate in the chain.
- SBL region protections enforced

Secure Debug

- Debug services dependent on Lifecycle State (LCS) and debug policy
- Secure BootROM processes debug certificate(s) as follows:
  - In CM LCS
    - The certificates are verified by the RSA signature, as done in the Secure Boot.
  - In DM LCS
    - If the certificate is signed with customer key, the Secure Debug does not verify the root key and the certificates are only verified by the RSA signature, as in Secure Boot. This means that any valid certificate may be used.
    - If the certificate is signed with Ambiq key, the Secure Debug behavior is as defined in Secure LCS.
  - In Secure LCS
    - The first certificate is verified against the OTP.
    - The unique device ID is required to create the debug certificate. It is only validated in this LCS.
- Once debug certificate is verified, the appropriate debug features are enabled based on the security policy defined in OTP

Secure Firmware Update

- Secure OTA (via Bluetooth) and Secure Wired Update supported via Ambiq Secure Boot Loader
- Supports Authentication, Encryption, Rollback Prevention
- Could use Ambiq or Customer keys – secure access through hardware crypto
- All firmware components can be updated (except Secure BootROM)

Secure OTP

- OTP region is partitioned into
  - Ambiq Key Storage OTP
  - Ambiq Security Policy OTP
  - Customer Key Storage OTP
  - Customer Security Policy OTP
- OTP region is implemented using special NVM partition separate from main partition
  - Hardware enforced access control to ensure key storage is only accessible to crypto hardware
  - Hardware enforced irreversible locks to ensure, once provisioned and locked, cannot be modified (true OTP)
    - Separate locks for each OTP section
    - Locking of a region is accessible to only the respective owner (based on LCS) – e.g. Customer Key Storage OTP can only be locked by customer

Security Keys

- Platform Key
  - Hardware key
  - Used for provisioning during LCS=CM and LCS=DM
  - Not valid/accessible in LCS=Secure. Only accessible by hardware crypto during secure boot, otherwise locked
- Device Root Key (HUK)
  - Generated at device initialization in hardware using the TRNG (unique per device)
  - Used for derivation of other keys
- Ambiq ROT Public Key / Public Key Hash
  - Used for SBL validation
- Customer Public Key / Public Key Hash
  - Used for customer firmware validation
- Ambiq Provision Key
  - Accessible to Ambiq only during LCS=CM
  - 128-bit AES key used for device provisioning
- Ambiq Encryption Key
  - Accessible to Ambiq only during LCS=CM
  - 128-bit AES key used to decrypt SW images
- Customer Provision Key
  - Accessible to Customer only during LCS=DM
  - 128-bit AES key used for device provisioning
- Customer Encryption Key
  - Accessible to Customer only during LCS=DM
  - 128-bit AES key used to decrypt SW images

Hardware Crypto

- Hardware accelerated crypto functions
  - Symmetric Key
    - AES 128/192/256
    - DES/TDES 64/128/192
    - AES MAC 128/192/256
    - AES-CCM 128/192/256
  - Public Key
    - RSA PKCS#1 2048/3072/4096
    - ECC
    - DH
  - Hash / HMAC
    - SHA1, SHA2 (SHA224, SHA256, SHA384, SHA512)
    - MD5
- TRNG
- Secure Key Management

Additional Support/Information

- The following documents are available to customers with NDA
  - Apollo4-SoC-Security-Users-Guide
  - Apollo4-and-Apollo4-Blue-Secure-Update-Users-Guide
  - Apollo4-OEM-Provisioning-Update-and-Tools-Users-Guide
- Apollo4 OEM Getting Started Training Deck

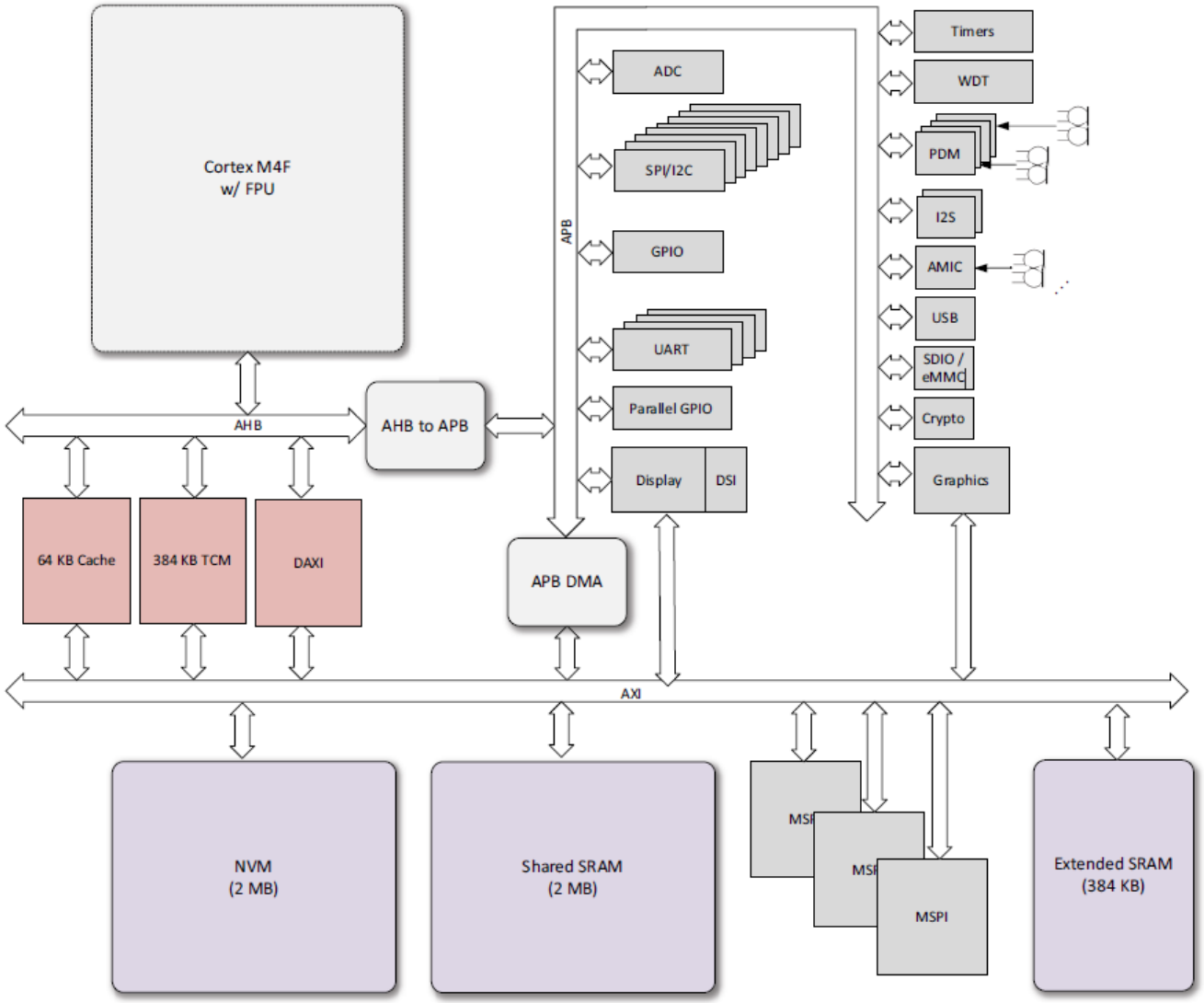
ambiq

endpoint intelligence

Memory Subsystem

Apollo4 Plus Memory Controller

Memory Subsystem



ambiq

Memory Subsystem

- Memory Types
  - SRAM (Tightly Coupled Memory and Shared System Memory)
  - Integrated NVM (MRAM) / External Memory via MSPI
  - Boot Loader ROM
  - One Time Programmable (OTP) memory
- Key Features
  - 2048 kB Shared SRAM
  - 384 kB TCM
  - 384 kB Extended SRAM
  - 2 MB MRAM
  - 64 kB NVM cache (2-way set-associative/Direct Mapped 128-bit line size)
- 16 kB OTP
  - 2 kB for customer use, including NVM protection fields
  - NVM Protection specified in 16 kB Chunks
  - 128 OTP bits specify Write Protected Chunks
  - 128 OTP bits specify Read Protected Chunks
  - A Chunk is Execute Only if Both Corresponding Protection Bits Specified
  - OTP bits Specify Debugger Lock Out State
  - OTP bits Can Protect SRAM Contents From Debugger Inspection
- External flash with XiP (via MSPI) and cache (instruction space only) support (up to 3x64 Mb apertures)

Memory: NVM

Memory Subsystem

- 2MB of MRAM
- Single separate power domain
- Powered down via hardware FSM
  - CPU deep sleep
  - Wake on DMA read access from MRAM
  - Support for Low Power Standby mode (significantly reduces standby power but increases read latency)
- Main partition write and read protections at 16KB granularity
- 2KByte Customer Trim region
  - Separate from 2MB main partition
  - Reads/writes handled through helper functions
  - Usage is customer defined (except lower 100bytes used for pre-defined configs)
- MRAM programming does *not* require erase (although page/mass erase is supported via helper function for consistency)
- MRAM is sensitive to magnetic interference. Please consult magnetic resilience guidelines document

Memory: Cache

Memory Subsystem

- 64KB Cache
  - 2-way set-associative or Direct Mapped
  - 512/1024/2048/4096 entry, 128b line size
- Unified ICode and DCode cache controller Configurable to cache Instruction, Data, or Both
- Caching is supported for the following
  - Entire 2MB internal MRAM (instruction and data)
  - 3x64MB external memory aperture (via MSPI – only instruction space)
  - SSRAM and Extended SRAM (instruction only)
- Intended to provide zero wait state read access from CPU to higher latency memory and reduce power
- Cache configuration controlled via CPU:CACHECFG register (cache domains must be enabled via PWRCTRL:MEMPWREN before use
  - Note: low power configuration using CACHECFG:CONFIG=0xC and disabling CACHEB2 domain in PWRCTRL:MEMPWREN
- Retention controlled by PWRCTRL:MEMRETCFG
- Two regions can be designated as non-cacheable or upper/lower cache locked

Memory: TCM

Memory Subsystem

- 384KB SRAM
- DTCM Banks
  - Three power domain partitions (controlled by PWRCTRL:MEMPWREN): 8KB, 120KB, 256KB
  - Always zero wait-state unless there is contention for that specific memory array with a DMA requestor
- Runs at CPU frequency (96MHz or 192MHz)
- Retention at power domain granularity (controlled by PWRCTRL:MEMRETCFG)
- Memory Latencies:

Memory / Storage Element	Cache Access	Accessed Memory	Access Type	DMA Bank Conflict	Cache State[1]	Wait States	Notes
TCM	No	Data	Read/ Write	No	-	0	
				Yes	-	N_{TCM}	- Dependent on DMA activity and conflict to same 32KB bank of memory - Max transfer size is 16 before arbitration

Memory: Shared SRAM (SSRAM)

- 2048KB SRAM
- SRAM Partitioning
  - Two power partitions (controlled by PWRCTRL:MEMPWREN): 1MB, 1MB
  - 2 port concurrency
    - Port1: Graphics + Display Controller
    - Port2: CPU Instruction/Data + DMA
- Runs at 96MHz independent of CPU operating frequency
- Retention at power domain granularity (controlled by PWRCTRL:MEMRETCFG)
- Memory Latencies

Memory / Storage Element	Cache Access	Accessed Memory	Access Type	DMA Bank Conflict	Cache State[1]	Wait States	Notes
SSRAM	Yes	Data	Write	-	Available	0	
				-	Unavail-able	N_{WR}	- 32-entry, 128b line buffer (LRU allocated) - Sub-128b writes are accumulated into buffer - Writes are flushed based on time, evic-tion or flush command - Dependent on bus load - Unloaded is 0 cycles
			Read	-	Hit	0	
				-	Miss	N_{RD}	- Critical word (miss) - Hits serviced from 4 entry 128b line buf-fer - Dependent on bus load - Unloaded is 7 cycles (LP) and 13 cycles (HP)
	Yes	Instruction	Read	-	Hit	0	
				-	Miss	N_{MISS}	- Critical word (miss) - Dependent on bus load - Unloaded is 7 cycles (LP) and 13 cycles (HP)

Memory: Extended SRAM

Memory Subsystem

- 384KB SRAM
- Single power partition (controlled by PWRCTRL:MEMPWREN): 384KB
- Runs at 96MHz independent of CPU operating frequency
- Retention at power domain granularity (controlled by PWRCTRL:MEMRETCFG)
- Memory Latencies

Memory / Storage Element	Cache Access	Accessed Memory	Access Type	DMA Bank Conflict	Cache State[1]	Wait States	Notes
SSRAM	Yes	Data	Write	-	Available	0	
				-	Unavail-able	N_{WR}	- 32-entry, 128b line buffer (LRU allocated) - Sub-128b writes are accumulated into buffer - Writes are flushed based on time, evic-tion or flush command - Dependent on bus load - Unloaded is 0 cycles
			Read	-	Hit	0	
				-	Miss	N_{RD}	- Critical word (miss) - Hits serviced from 4 entry 128b line buf-fer - Dependent on bus load - Unloaded is 7 cycles (LP) and 13 cycles (HP)
	Yes	Instruction	Read	-	Hit	0	
				-	Miss	N_{MISS}	- Critical word (miss) - Dependent on bus load - Unloaded is 7 cycles (LP) and 13 cycles (HP)

Memory: NVM OTP – INFO Space

Memory Subsystem

- 16KB of Secure “OTP”
- 6 KBytes contain factory preset chip trim values (INFO1)
- 2 KBytes for customer trim values (INFO0)
- 8 KBytes for security OTP
  - Lifecycle State
  - Ambiq key bank
  - Customer key bank
  - Secure Bootloader Configuration
  - MRAM main partition protection

Memory: DMA

Memory Subsystem

- Peripheral Controllers with DMA capability:
  - I2C Master
  - SPI Master
  - MSPI
  - ADC
  - AUDADC
  - PDM
  - I2S
  - SDIO
  - Crypto
  - GFX
  - Display Controller
- DMA is supported from peripheral to SRAM and from SRAM/MRAM to peripheral
- DMA transactions to/from TCM/SSRAM/Extended SRAM occur concurrently to CPU instruction/data accesses as long as the accesses are to different physical banks of memory. Accesses to the same physical bank are arbitrated in hardware.
- Some peripheral controllers have a command queue which allows intelligent chaining of multiple DMA transfers
- Controllers with Command Queue:
  - SPI Master
  - I2C Master
  - MSPI

Data-AXI (DAXI) Module

- Thirty-two 128-bit line buffers that are used for read caching and write buffering. Each line will be marked as one of five states:
  - I: INVALID. Buffer is invalidated
  - S: SHARED. Data read from system memory that will be cached for subsequent reads.
  - W: WRITE: Buffer is allocated as a write buffer. Writes will accumulate in buffer until line is flushed. Byte enables indicate which bytes have been written for write-back.
  - WL: WRITELOAD. Write buffer is reloading remainder of line to satisfy a load.
  - M: MODIFIED. Read buffer that has local write modifications that need to be flushed
- LRU/Aging counter to manage line allocation and temporal flushes
- Flush on sleep and sideband flush controls
- Asynchronous RAXI system interface with 2-entry address FIFO (A), 2-entry write data FIFO (W), 1-entry read data FIFO (R) and 1-entry write response FIFO (B)

Data-AXI (DAXI) Module con't

- DAXI provides read caching and write buffering to the system memory interfaces (SSRAM, Extended SRAM, MSPI)
- Each line buffer is always in one of five states:
  - ALID: The invalid state basically indicates the line buffer is not in use and its data is invalid and the buffer is available for allocation. INVALID lines are the first buffers used for re-allocation.
  - SHARED: The line contains data read from the AXI bus and reads will be serviced from the line. SHARED lines are converted to MODIFIED lines on a write to its address. SHARED lines are allocated only if no INVALID lines are available.
  - WRITE: Lines in the WRITE state have write data but do not have valid read data. They are simply buffered writes and allow the DAXI to accumulate as many writes as possible before committing them to the AXI subsystem. WRITE lines can be converted to MODIFIED on a read operation by the CPU to the line's address.
  - WRITE LOAD: Lines that have write data and are in the process of reloading a line for a read operation that will be converted to MODIFIED state.
  - MODIFIED: Modified lines contain cached read data as well as hold partial (or full) updates from write operations. MODIFIED lines must be written back to system memory before re-allocation.

Data-AXI (DAXI) Module con't

- From the invalid state, there are two or three common paths that its state will take:
  - On a read, the DAXI would issue a read to the primary AXI bus and the buffer would be allocated (SHARED state) to hold the returned 128-bit data. Data would remain cached until a flush (in which case the line is invalidated) or until it is reallocated for a different read or write operation.
  - On a write, the DAXI would allocate the line buffer to hold write data (WRITE state). Byte enables would indicate which bytes have been written. In the case of adjacent or subsequent writes to that line, the DAXI will continue updating bytes as the ARM issues them (think of the case of the ARM sequentially writing bytes or words of data). If the ARM reads any data from this line, the DAXI will convert the buffer to WRITE LOAD state, issue a read operation on the AXI bus, merge the incoming read data with existing write data and then convert the buffer to the MODIFIED state.
  - On a read followed by a write operation from the ARM, a buffer would start out as SHARED (from the read) and then transition to MODIFIED after the write operation. It would then be flushed before being reallocated.

Data-AXI (DAXI) Module con’t

- Control logic in the DAXI will generally try to maintain a minimum of one buffer in the invalid or shared state at all times to ensure that write operations can be immediately serviced
- Generally, the DAXI will allocate buffers as follows:
  - The DAXI selects a buffer in the INVALID state if one is available, picking the INVALID entry closest to entry zero.
  - If an INVALID entry was not available, the DAXI selects a SHARED buffer in the LRU group if one is available, picking the SHARED entry closest to entry zero.
  - If neither was available, the DAXI will wait for one to become available.
    - In this case the “hardware stalled allocation flush” mechanism described below will make one available.
    - WRITE and MODIFIED buffers are not eligible for allocation.
    - Buffers in the 2nd LRU group, 3rd LRU group and MRU group are not eligible for allocation.

Data-AXI (DAXI) Module con’t

- Each line also maintains an LRU group state
- Line buffers are divided into 4 LRU group populations (MRU, 3rd LRU, 2nd LRU, LRU). A buffer's LRU state is set to MRU* on allocate or non-MRU hit. If adding a buffer to the MRU group would exceed the MRU group population limit, all other entries are demoted. If LRU population is fully depleted, all other entries are demoted.
- Promotion/Demotion
  - Updated entry -> MRU*
  - LRU entries -> LRU
  - 2nd LRU entries -> LRU
  - 3rd LRU entries -> 2nd LRU
  - MRU entries -> 3rd LRU
- *If only 3 buffers enabled, update as 3rd LRU; if only 2 buffers enabled, update as 2nd LRU

Data-AXI (DAXI) Module con't

- Hardware Reset
  - Reset invalidates all line buffers. Reset invalidates buffered writes in the DAXI regardless of outstanding transactions. Outstanding writes may not be completed.
  - Use a SW DAXIFLUSHWRITE and then poll for DAXIREADY to ensure writes complete before a reset.
- Hardware aging
  - An aging counter is a grey-coded counter that increments every N CPU clock cycles (N is programmable). Every time a buffer is allocated, or a read/write transaction utilized a buffer, it records the current age in the buffer. Lines will automatically self-flush after the aging counter reaches a count of two more than the originally stored counter value. This helps ensure that stale data is routinely flushed from the DAXI to preemptively make room for new allocations. Older stale data may have a lower probability for future reuse.
  - The AGINGCOUNTER setting essentially sets a temporal window in which any buffer that is unused during the time period will be invalidated or flushed. Young buffers remain live for reuse for a number of active DAXI cycles. The number of DAXI cycles is a minimum of AGINGCOUNTER active DAXI cycles and a maximum 3*AGINGCOUNTER active DAXI cycles. The diagram below demonstrates the concept by showing the progression of the aging counter through its four states at the top (each state indicated by a unique color) and operations on each of the four buffers as colored circles.

Data-AXI (DAXI) Module con't

- Hardware flush level
  - The DAXI has a FLUSHLEVEL configuration that indicates how aggressively the DAXI will flush buffers in order to ensure that resources are always available for the CPU.
  - When set to 1 and 3 or more buffers are enabled, the DAXI will attempt to maintain three free buffers
  - When set to 0 and 3 or more buffers are enabled, the DAXI will attempt to maintain two free buffers
  - When set to 1 and 2 buffers are enabled, the DAXI will attempt to maintain two free buffers
  - When set to 0 and 2 buffers are enabled, the DAXI will attempt to maintain one free buffers
  - Not applicable when only 1 buffer is enabled
  - Buffers are considered "free" when INVALID, SHARED or in the process of being flushed. This mechanism flushes a MODIFIED or WRITE buffer when there less than FLUSHLEVEL enabled buffers free. This is pre-emptive, before an allocation is required. A flush writes back cached modified data. MODIFIED buffers downgrade to shared allowing additional reads. WRITE buffers downgrade to INVALID. Pre-emptively flushing with this mechanism allows quick future reallocation for another line.
- Hardware stalled allocation flush
  - In additional to being flushed by the pre-emptive aging and flush level mechanisms, the DAXI control logic will dynamically flush/invalidate buffers on demand in cases where the DAXI needs to re-allocate the buffer to support the current CPU operations. A write check or read check may that misses in the DAXI will need to allocate a new entry.
  - Flush when an allocation is required and there are no INVALID buffers and no SHARED buffers in the LRU group and no buffers in the process of being flushed.

Data-AXI (DAXI) Module con't

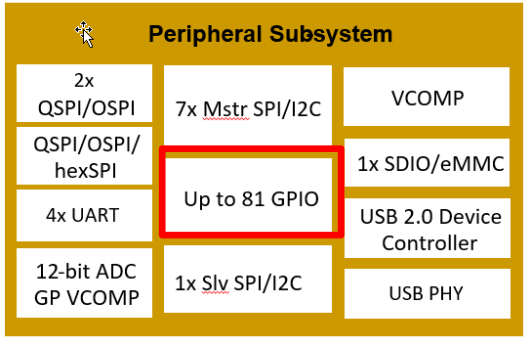
- Picking a buffer for flush level and stalled allocation flushed
  - Same selection process picks entry 31 if it is in the LRU group, (MODIFIED or WRITE) and not already in the process of flushing
  - Else pick entry 30 if it is in the LRU group, (MODIFIED or WRITE) and not already in the process of flushing
  - Else pick entry 0 if it is in the LRU group, (MODIFIED or WRITE) and not already in the process of flushing
  - If no entries are available to flush, wait for demotion of modified or write entries into LRU group
- Software flush
  - Software must use the DAXI sync controls in some situations (like writing to a buffer used for DMA or self-modifying code) to ensure coherency when interacting with other agents in the hardware. The DAXICTRL register has 2 actions and 1 status bit that provide a software interface to the hardware mechanism.
    - DAXIINVALIDATE
      - Writing a 1 to this bitfield invalidates any SHARED data buffers
      - SHARED → INVALID
    - DAXIFLUSHWRITE
      - Writing a 1 to this bitfield forces a flush of WRITE or MODIFIED buffers
      - MODIFIED -> SHARED
      - WRITE -> IINVALID
    - DAXIREADY
      - DAXI status indicating flush of WRITE or MODIFIED buffers is in progress when 0
      - Flush is done and DAXI is ready when it return 1

ambiq

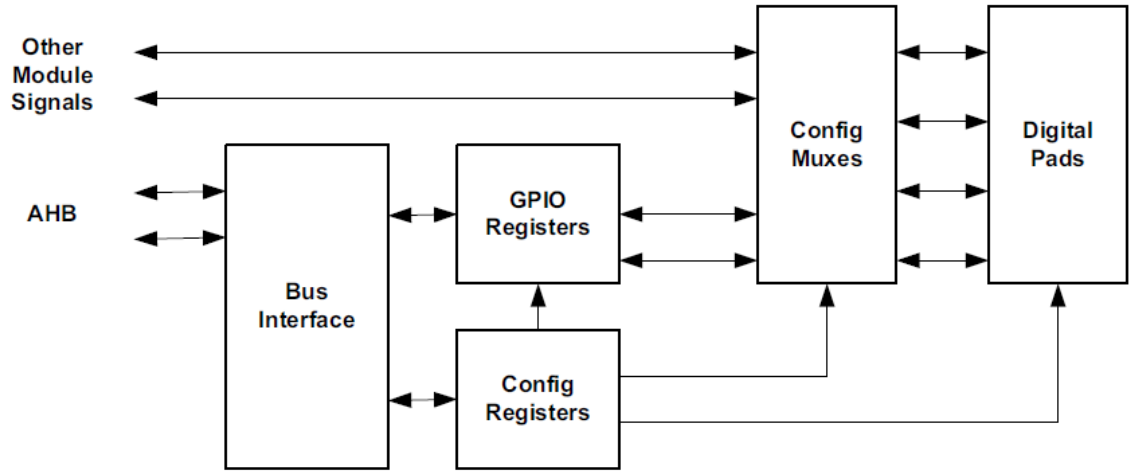
endpoint intelligence

Peripheral Subsystem

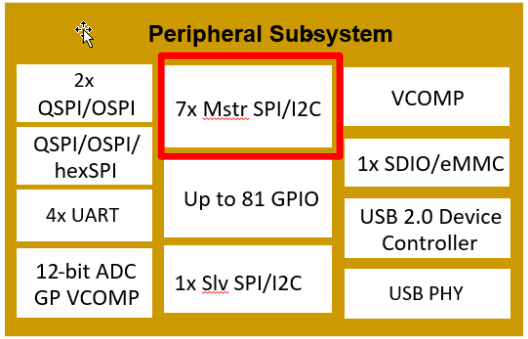
General Purpose I/Os



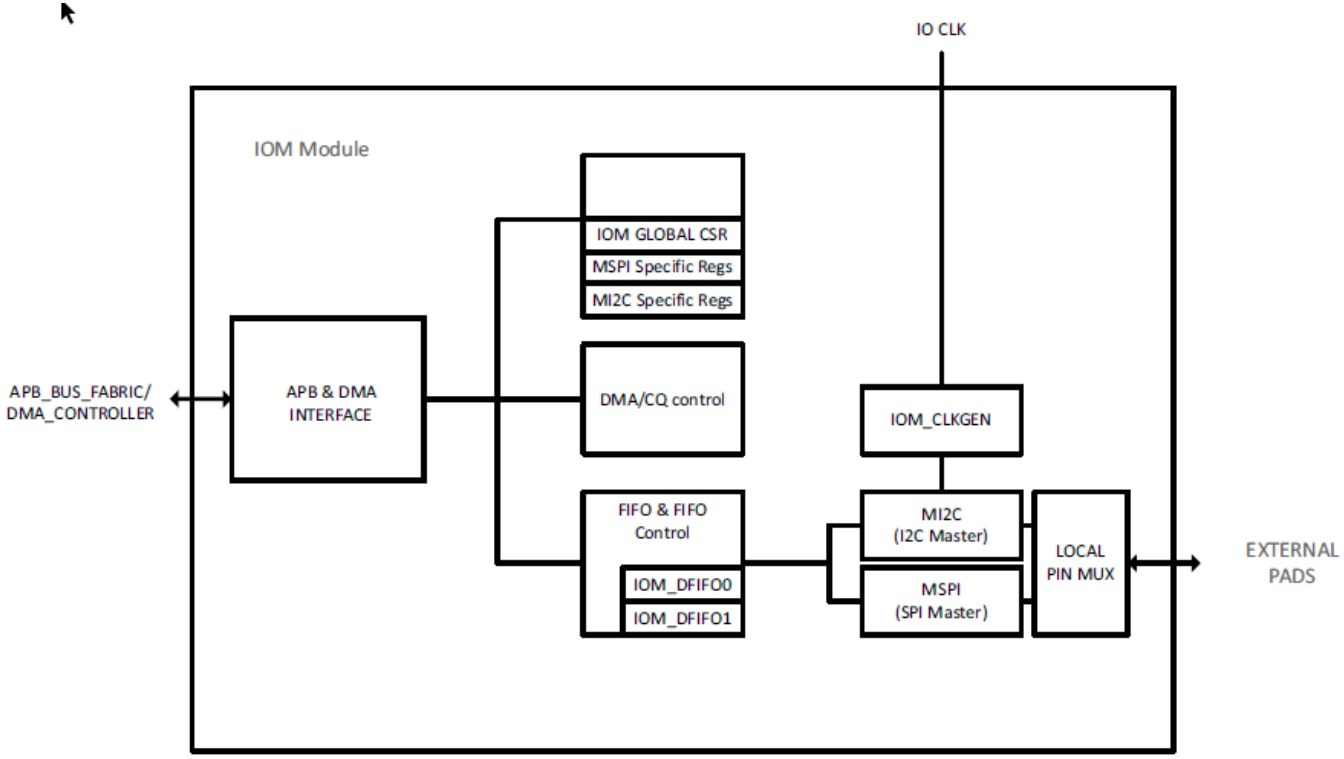
- Apollo4 Blue Plus: 81 GPIO in BGA SiP package
- Apollo4 Plus: 105 GPIO in BGA package
- Configurable
  - Tristate, Open Drain, or Push Pull
  - Configurable drive strength or 2, 4, 8, or 12mA
  - Many pin function mapping options
- Integrated pull up/down



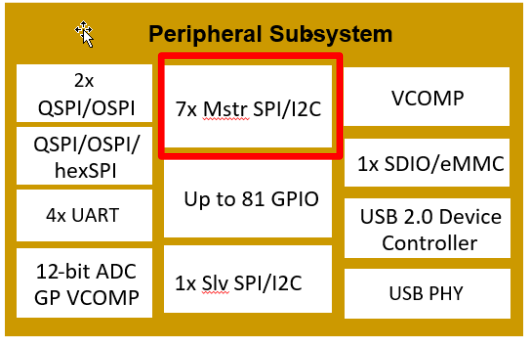
I/O Masters (IOM)



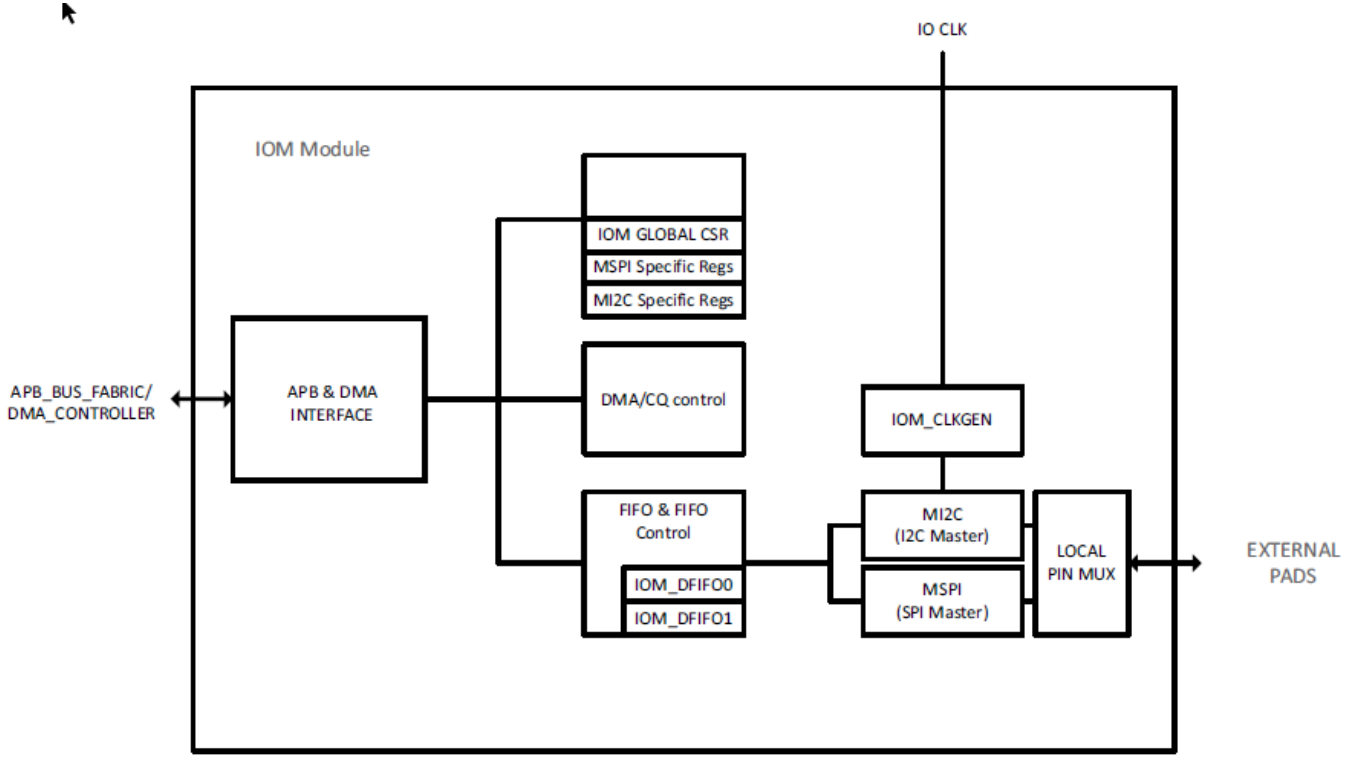
- Each IOM contains independent Read and Write FIFOs of 32Bytes
- Each IOM has a DMA controller that supports IOM to SRAM, and SRAM/Flash to IOM transfers
- Each IOM supports command queues which can schedule multiple transactions to occur without CPU intervention.
  - Used in combination with DMA.
- CPU can go to sleep while IOM is performing direct reads/writes to/from FIFO or transferring data by way of DMA



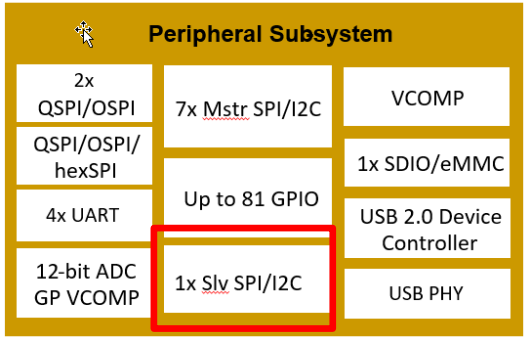
I/O Masters (IOM) Continued



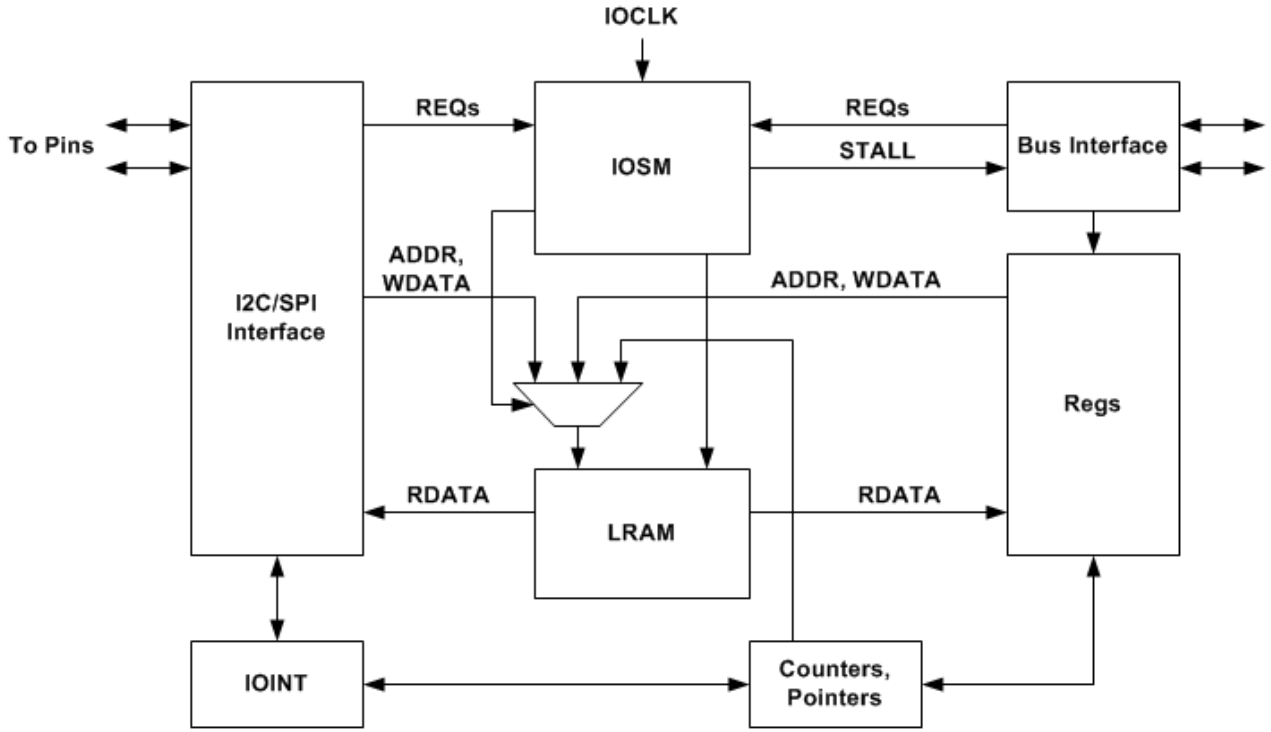
- Non-Blue: 8x IOM (SPI/I2C)
- Blue: 7x IOM (SPI/I2C) where 1 SPI used for C2C with BLE radio die
- I2C mode
  - Supports 7 and 10-bit addressing
  - Multi-master arbitration
  - 100KHz, 400KHz, and 1MHz
  - Integrated pull-up resistors
- SPI mode
  - 4 Chip Selects per IOM
  - 3 and 4-wire modes
  - Up to 48 MHz
  - Supports all 4 standard SPI modes
  - Standard embedded address (command byte) operations
  - Also supports raw read/write operations



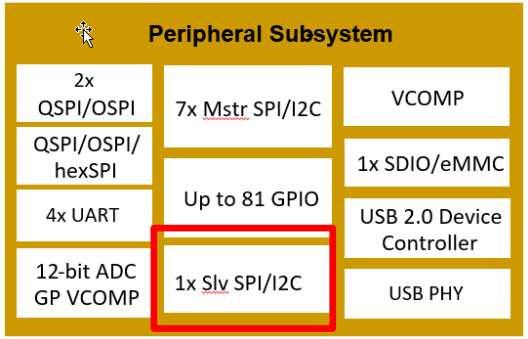
I/O Slave



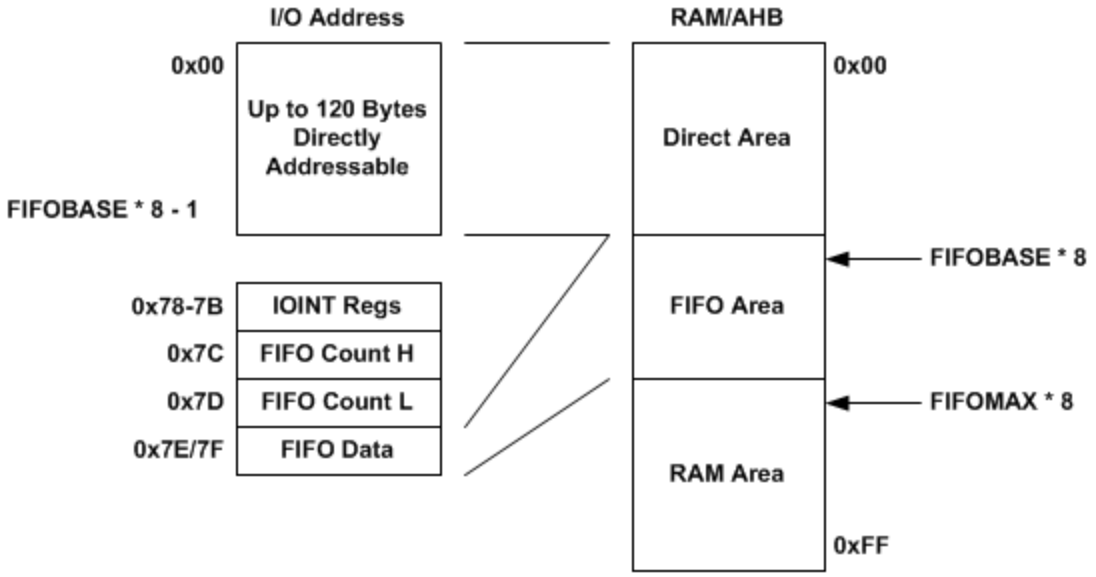
- 1 IOS (I/O Slave)
  - I2C or SPI mode
- I2C mode:
  - configurable 7 and 10-bit addressing
  - interface freq. up to 1.2 MHz
- SPI mode:
  - Supports all polarity/phase combinations
  - interface freq. up to 16 MHz
    - FIFO Mode restricted to >15MHz or <500KHz



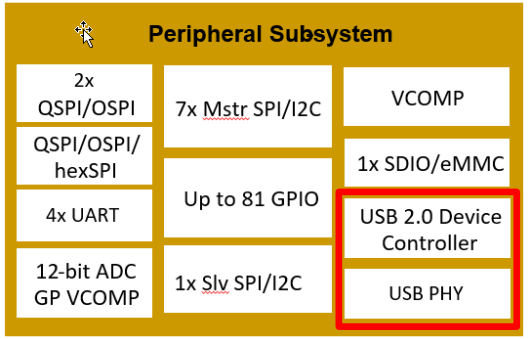
I/O Slave Con'd



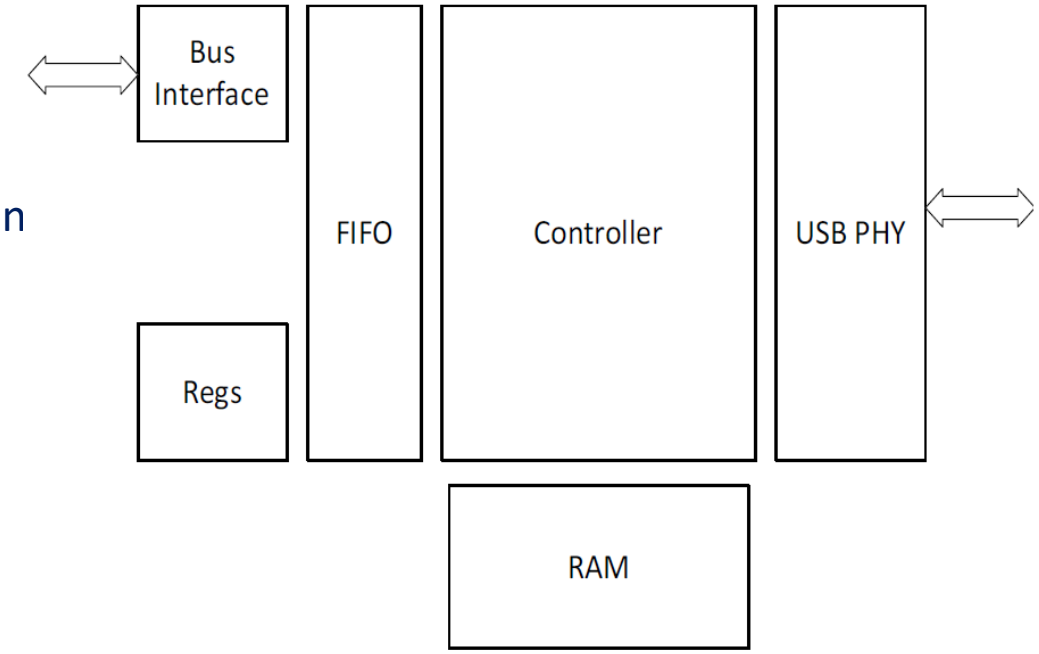
- IOS LRAM is 256 bytes
- Broken up into 3 sections:
  - Directly addressable RAM up to 120 Bytes which can be accessed by external IOM host while MCU is asleep
  - Status and Config Registers & FIFO. FIFO size up to 256 Bytes
  - Any LRAM not configured as direct access or FIFO is left as RAM accessible by Apollo SW only



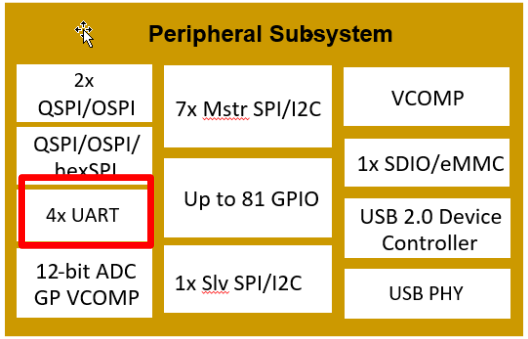
Universal Serial Bus (USB)



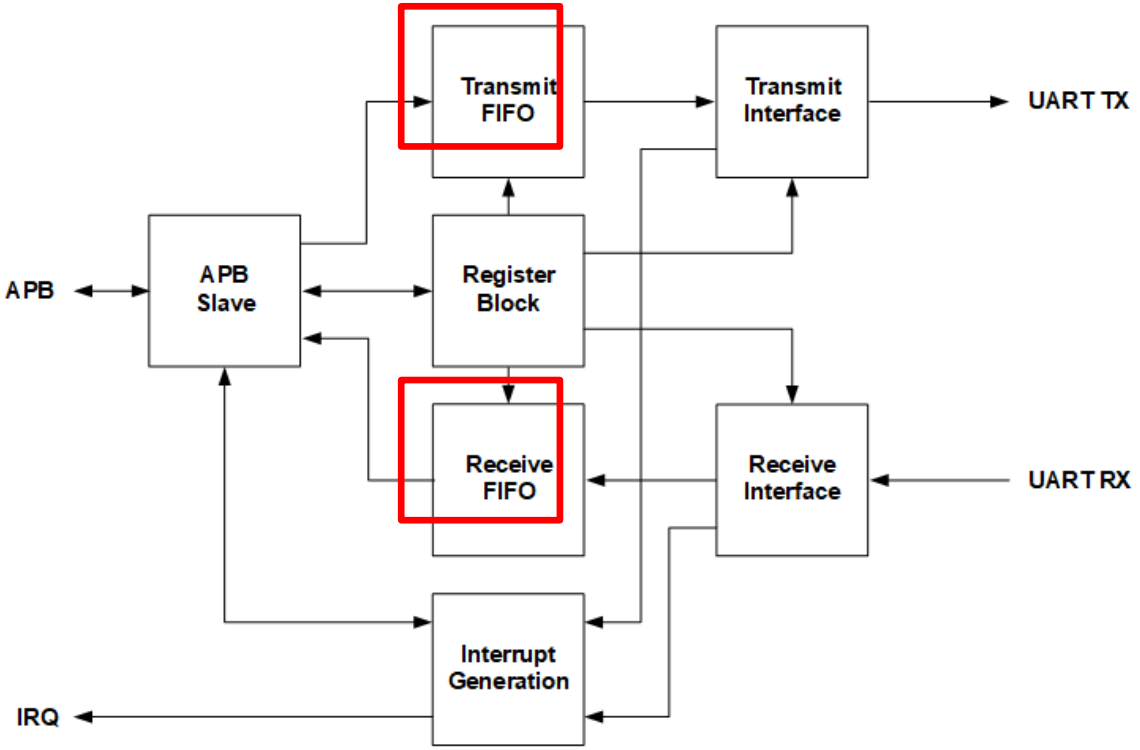
- USB 2.0 FS/HS PHY device with support for low-power mode
- Battery charging detection (BC1.2 & vendor-specific)
- Interrupt driven weak battery/good battery algorithm for advanced power saving
- Crystal-less operation
- On-die pull-ups/downs and termination (no external calibration resistors, pull-ups-downs required
- Dynamic FIFO sizing: 4 kB total FIFO
- 5 IN/OUT endpoints plus 1 control
- Soft connect/disconnect
- Suspend mode
- Note: USB controller does not have DMA so all data transactions require heavy CPU involvement



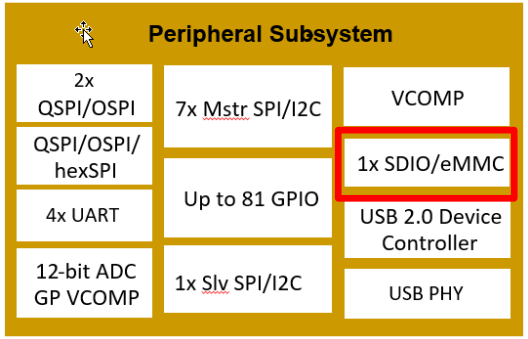
UART



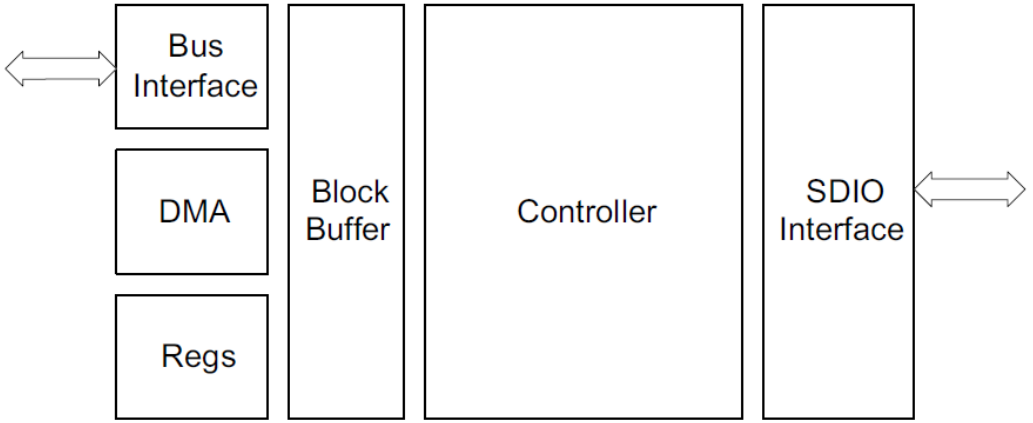
- 4 UARTs
  - MCU can enter sleep mode during transfers
  - 32Byte Transmit and 32Byte receive FIFOs reduce MCU active time
- Configurable baud rate generator
  - Maximum rate of 2.6Mps
- Highly programmable
  - Data size, parity, and stop bit length
  - Hardware flow control
  - Full-duplex and half-duplex modes
- Loopback functionality for diagnostics and testing
- Support for full-duplex or half duplex communication



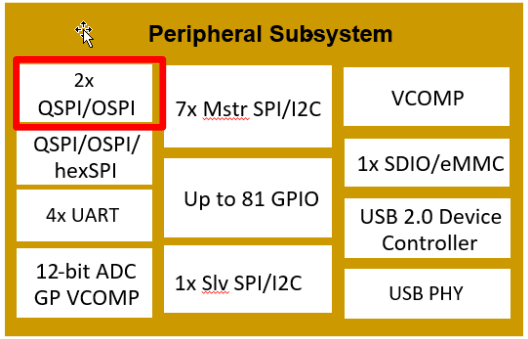
Secure Digital Input Output (SDIO)



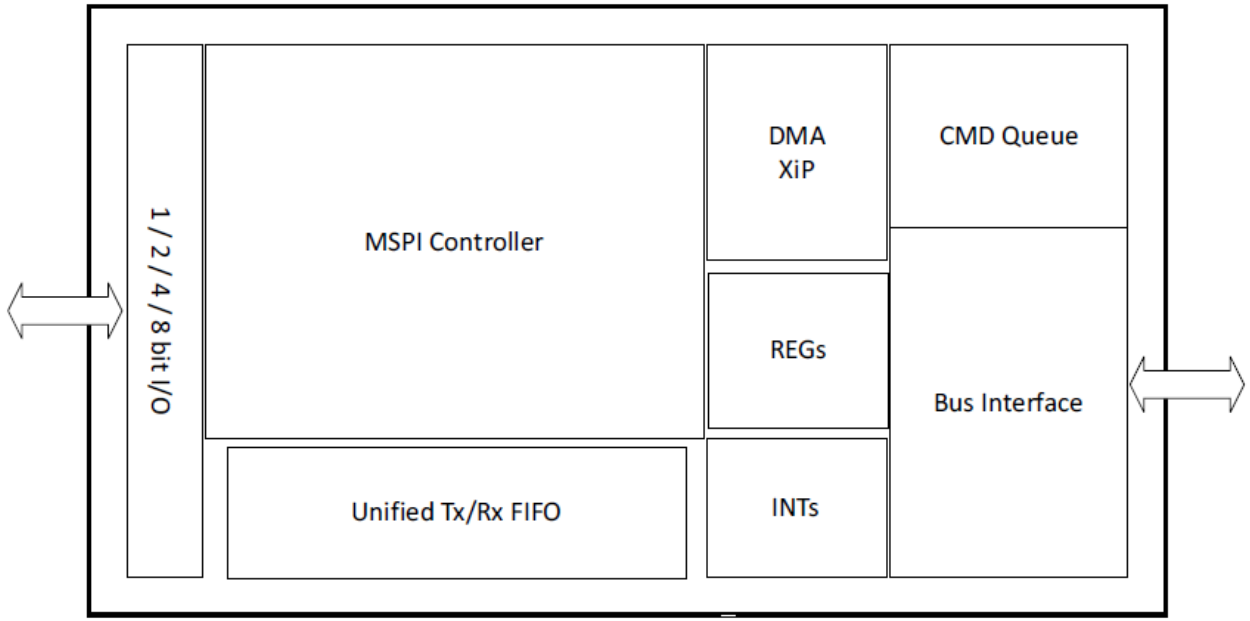
- SDIO card specification Version 3.0
- Host clock rate variable between 0 and 96 MHz
- Up to 50MBytes per second data rate using 4 parallel data lines (SDR50/DDR50 mode)
- Transfers the data in 1 bit and 4 bit SD modes
- Transfers the data in SDR50 or DDR50 modes.
- Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
- Variable-length data transfers
- Performs Read wait Control, Suspend/Resume operation SDIO CARD.
- 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
- Supports a programmable DLL to allow for timing tuning for optimal windowing.



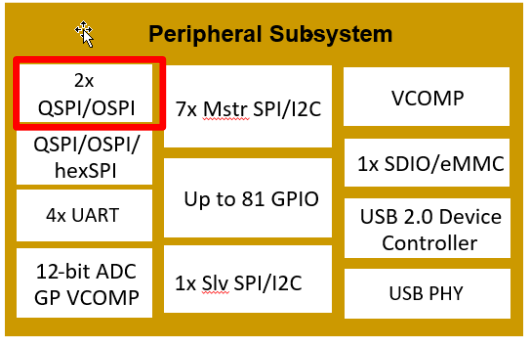
MSPI



- Apollo4 Interface
  - 2x - 1/2/4/8-bit SPI interface
  - 1x - 1/2/4/8-bit or HexSPI interface
- Apollo4 Blue Plus-KBR
  - 2x MSPI support 1/2/4/8/ interfaces
  - HexSPI Not Supported
- Apollo4 Blue Plus-KXR
  - HexSPI supported
  - 1x SPI interface is not available
- Support for DCX signal for displays
- XiP supported
- DMA with peripheral-to-memory and peripheral-to-peripheral support
- Up to 96 MHz interface in SDR mode; 48 MHz in DDR mode
- All four SPI modes supported
- Command Queue Support



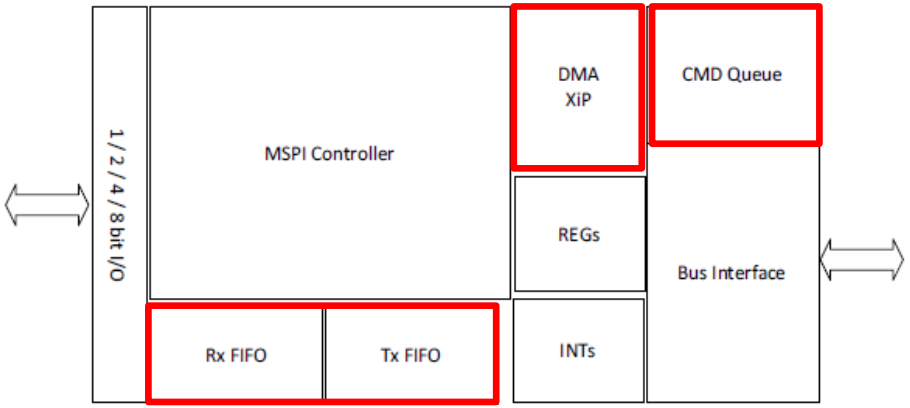
Serial Communication: MSPI cont'd



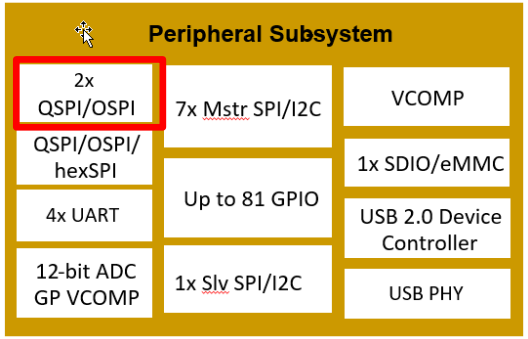
- Unified 32-entry FIFO (32 bits wide) for reads and writes
- DMA Support
  - Simple DMA model where software sets internal (SRAM or flash) address and external device address, transfer direction, and transfer size
  - MSPI DMA controller automatically handles sequencing of instructions and address to 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

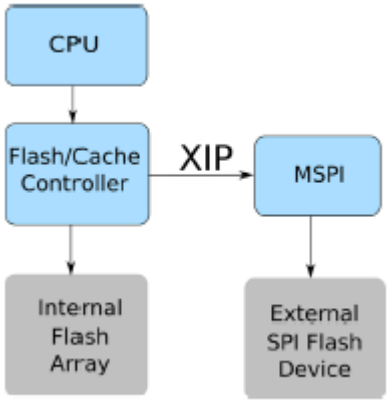
31:30	29:28	27:26	25:24	23:22	21:20	19:18	17:16	15:14	13:12	11:10	09:08	07	06	05	04	03:02	01	00
READINSTR				WRITEINSTR				RSVD		XIPMIXED		XIPSENDI	XIPSENDA	XIPENTURN	XIPBIGENDIAN	XIPACK	RSVD	XIPEN

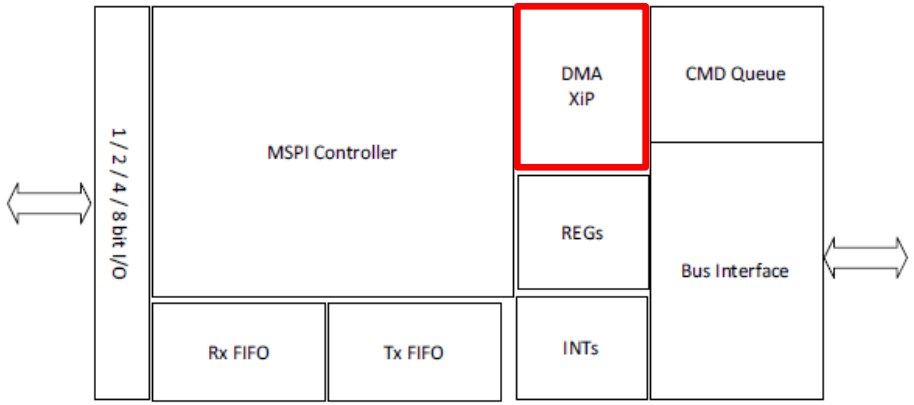


Serial Communication: MSPI cont'd

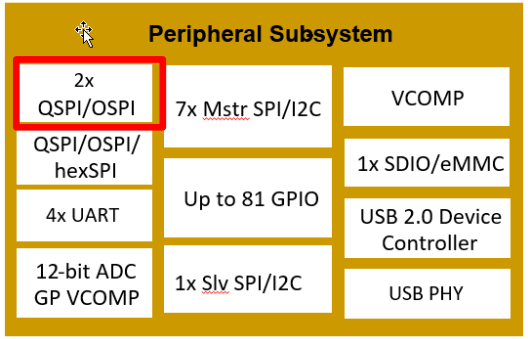


- MSPI supports Execute in Place (XIP) Operations
- Instruction accesses to MSPI connected devices can be cached
- Data accesses to MSPI connected devices get mapped through DAXI
- XIP and DMA/PIO operations can be interleaved. MSPI controller will allow current operation to complete before performing the XIP operation.

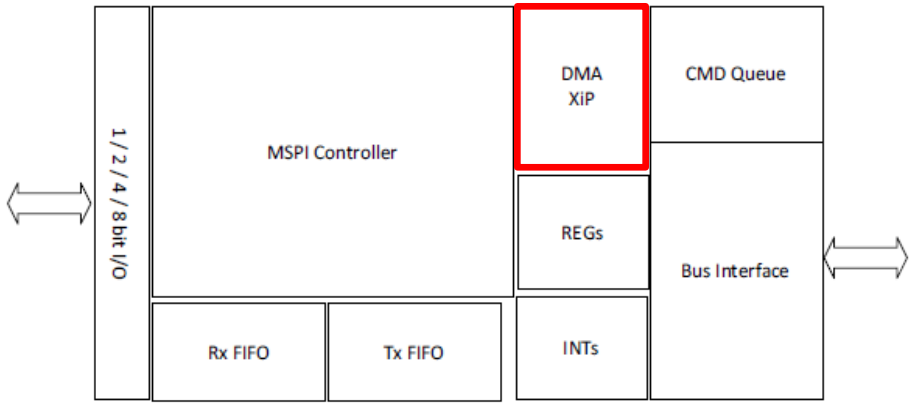




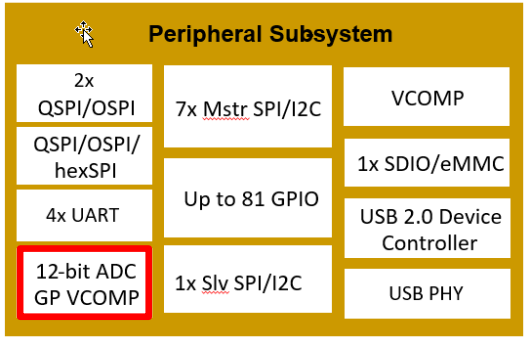
Serial Communication: MSPI cont'd



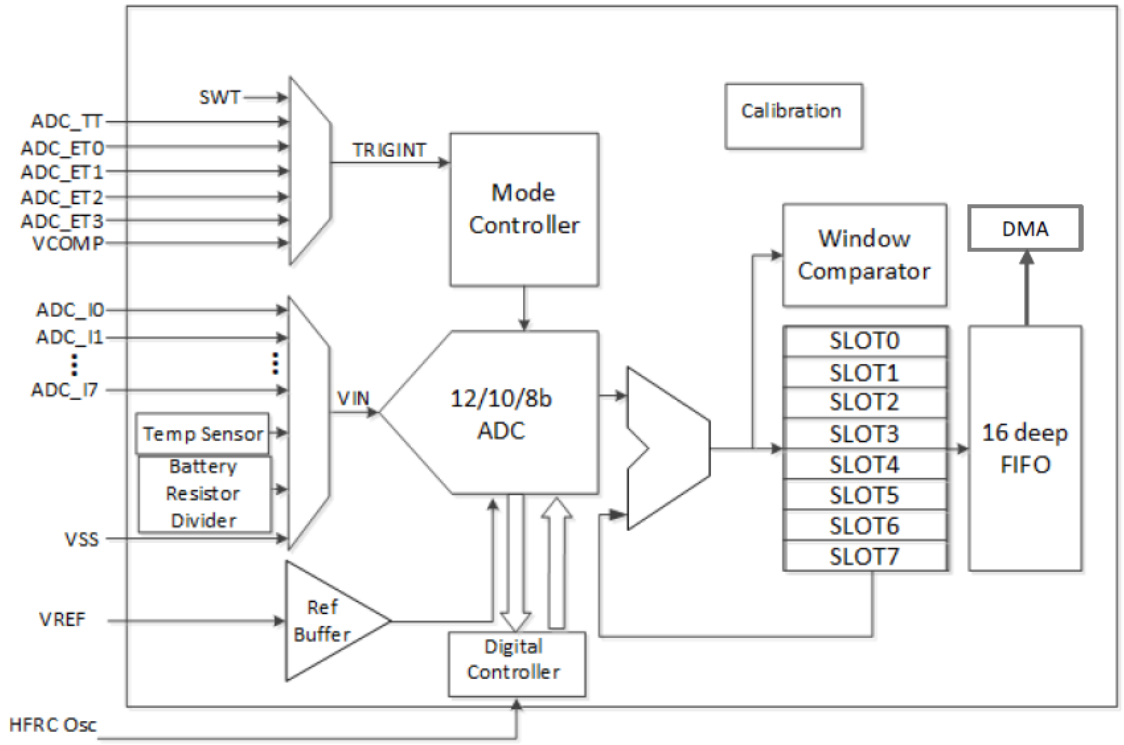
- MSPI also support read/write memory-mapping into SRAM space (XIPMM)
- XIPMM mapped memory is NOT cached (all accesses go through DAXI)
- Note: On Apollo4, XIP and XIPMM are mapped to same address range and only instruction accesses to MSPI address region are cached
- Also note that writes to XIPMM do not flush cached data to the same address, so care must be taken if same area is mapped as XIP and XIPMM



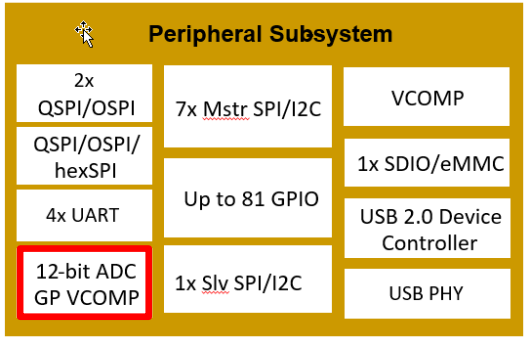
Analog-to-Digital Converter (ADC)



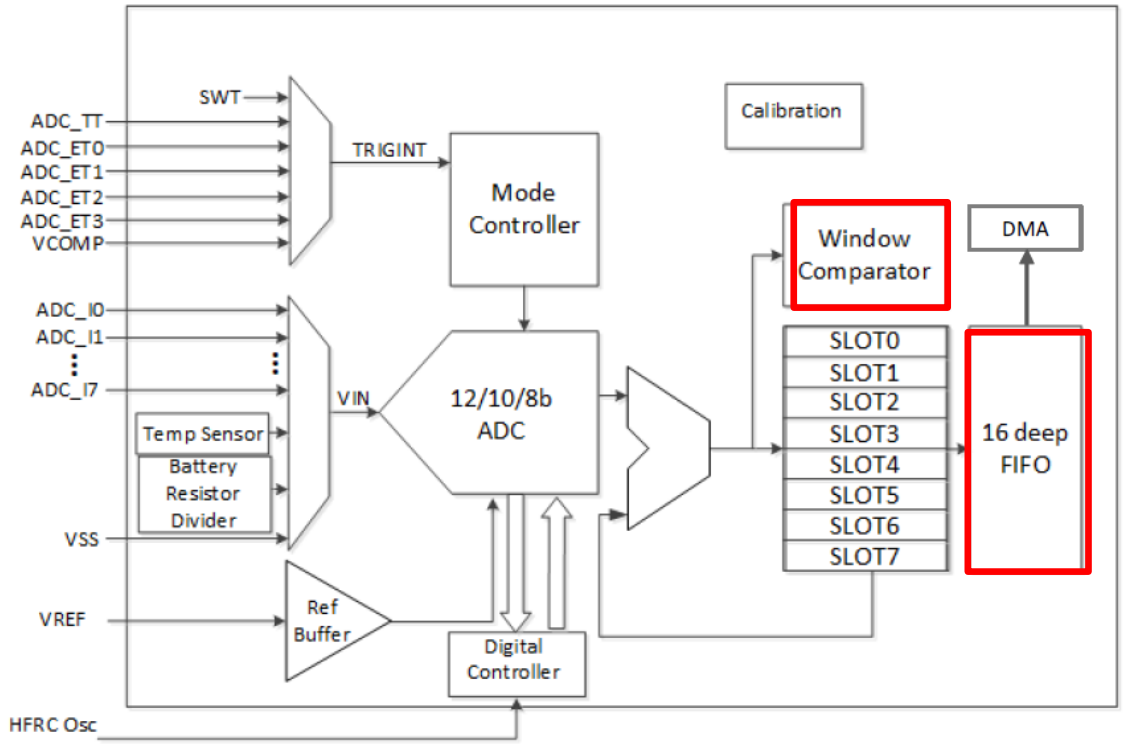
- 12-bit SAR architecture (successive approximation register)
- 8 Single-ended and 2 Differential external inputs
- 3 internal inputs: VSS, Temp Sensor, and VDD/3
- On-chip bandgap reference voltage (1.2V)
- Single shot, repeating single shot, scan, and repeating scan modes
- User-selectable clock source for variable sampling rates



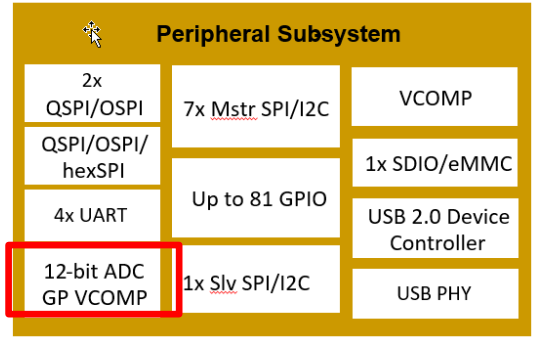
ADC Continued



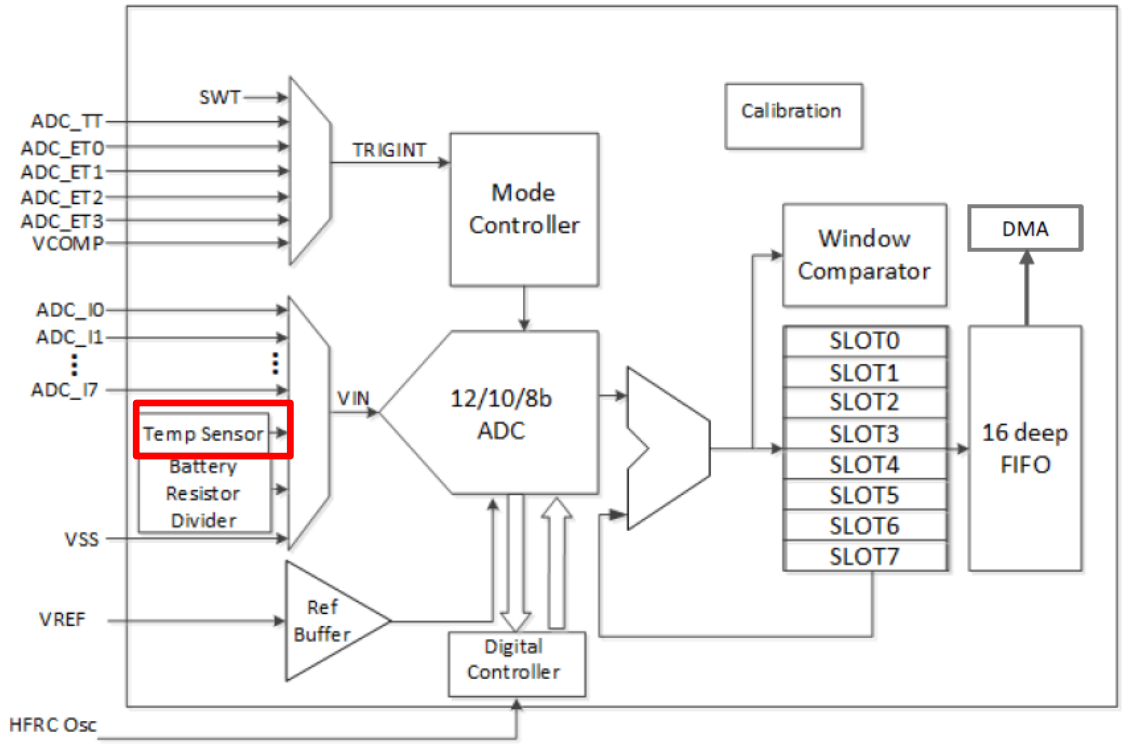
- Multiple conversion triggers: External pins, TIMER, SW trigger
- Window comparator for monitoring voltage excursions into or out of selectable thresholds
- 16-entry FIFO and DMA for storing measurement results and maximizing MCU sleep time
- Multiple low power modes
  - Selectively shut down portions of the ADC between conversions
- Sample rate up to 1.2Msps at 12bit and up to 2.8Msps at 8bit
- Built-in accumulate and scale module
  - Automatically accumulate up to 128 samples (Per-slot 21bit accumulation register)
  - Scaling capability for automated averaging



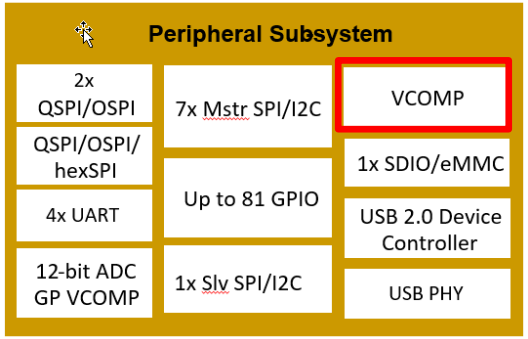
Temperature Sensor



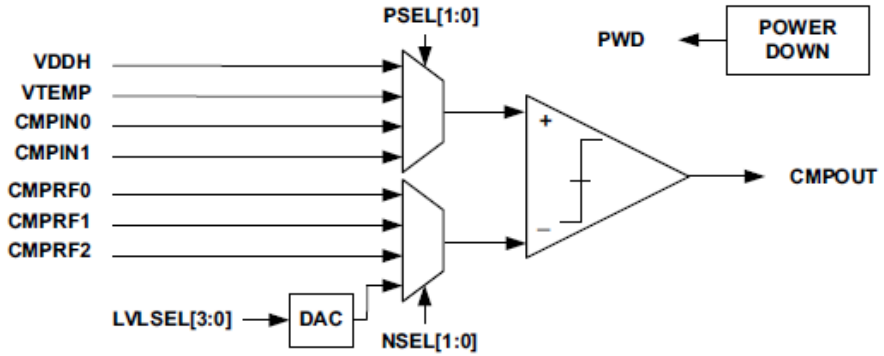
- Built-in sensor, accessible via ADC channel or Comparator
- ± 3 ˚C accuracy (If calibrated)
- 3.8mV/C sensor slope
- 3 calibration values needed
  - Offset
  - Voltage
  - Temp. during calibration



Voltage Comparator



- Interrupt and register access to comparator output
- If V+ rises above or falls below threshold: generates interrupt
- Programmable compare threshold:
  - 4-bit DAC
  - 3 different external pins
- Power consumption TBD (it is ~ 2uA)



0P58V = 0x0 - Set Reference input to 0.58 Volts.
0P77V = 0x1 - Set Reference input to 0.77 Volts.
0P97V = 0x2 - Set Reference input to 0.97 Volts.
1P16V = 0x3 - Set Reference input to 1.16 Volts.
1P35V = 0x4 - Set Reference input to 1.35 Volts.
1P55V = 0x5 - Set Reference input to 1.55 Volts.
1P74V = 0x6 - Set Reference input to 1.74 Volts.
1P93V = 0x7 - Set Reference input to 1.93 Volts.
2P13V = 0x8 - Set Reference input to 2.13 Volts.
2P32V = 0x9 - Set Reference input to 2.32 Volts.
2P51V = 0xA - Set Reference input to 2.51 Volts.
2P71V = 0xB - Set Reference input to 2.71 Volts.
2P90V = 0xC - Set Reference input to 2.90 Volts.
3P09V = 0xD - Set Reference input to 3.09 Volts.
3P29V = 0xE - Set Reference input to 3.29 Volts.
3P48V = 0xF - Set Reference input to 3.48 Volts.

ambiq

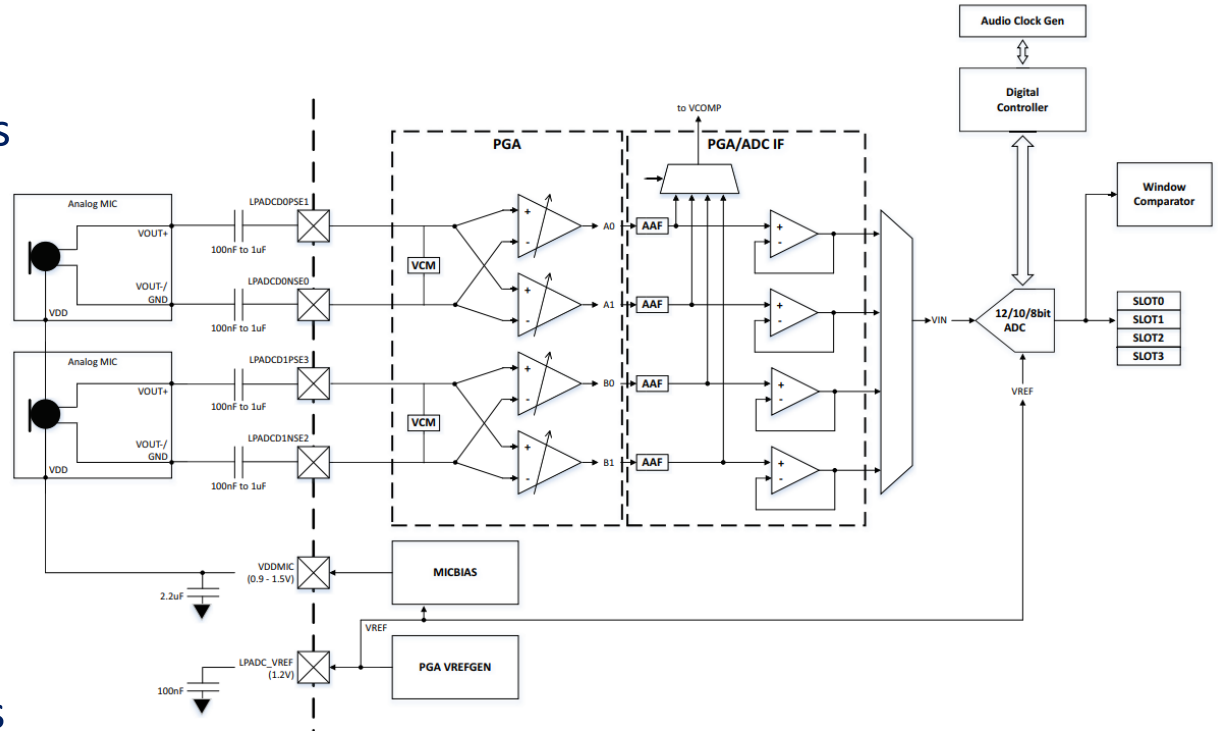
endpoint intelligence

Audio Subsystem

Low Power Analog Audio Interface

Audio Subsystem

- 4 channels of Programmable Gain Amplifiers
  - Programmable gain for AC-coupled audio inputs (20 Hz - 20 kHz) to drive AUDADC
  - Audio inputs may be microphone or line inputs
    - Single Ended (SD)
    - Pseudo Differential (PD)
    - Fully Differential (FD)
  - Full Scale Voltage
    - SE/PD: 0.5 Vrms
    - FD: 1 Vrms
  - Gain steps supported: 0-30 dB in 0.5 dB increments, obtained from two stages
  - Preamp gain steps: 0 (bypass), 12, 15, 18
  - PGA gain steps: 0-11.5 dB in 0.5 dB increments
- Set input common-mode for active and sleep mode operation
- Implicit 2/3 attenuation to fit 1.2 V ADC full scale



12-bit 4-channel Audio ADC (AUDADC)

Audio Subsystem

- Reconfigurable Successive Approximation Register (SAR) ADC
- 4 dedicated single-ended input channels from four PGA sources
- Input Range: 0 V to 1.2 V
- Configurable automatic low power control between scans
- Configurable for 12 / 10 / 8 bit ADC Precision Modes
- Sampling rate up to 2.0 MS/s (12-bit Mode) & 2.8 MS/s (8-bit Mode)
- Configurable sampling time
- Uses 1.2 V external reference with internal buffer

Stereo Low Power Analog Microphone

- MICBIAS provides user-programmable regulated (0.9 V to 1.5 V) supply to analog MEMS microphones
- Performance Summary
  - 200 µA max load current with 2.2 µF capacitor
  - 560 nA quiescent current
  - Typical PSR (from VDDAUD)
  - 34 dB @ 1 kHz
  - 15 dB @ 20 kHz
  - Startup < 1 ms

Digital Microphones (PDM to PCM Converter)

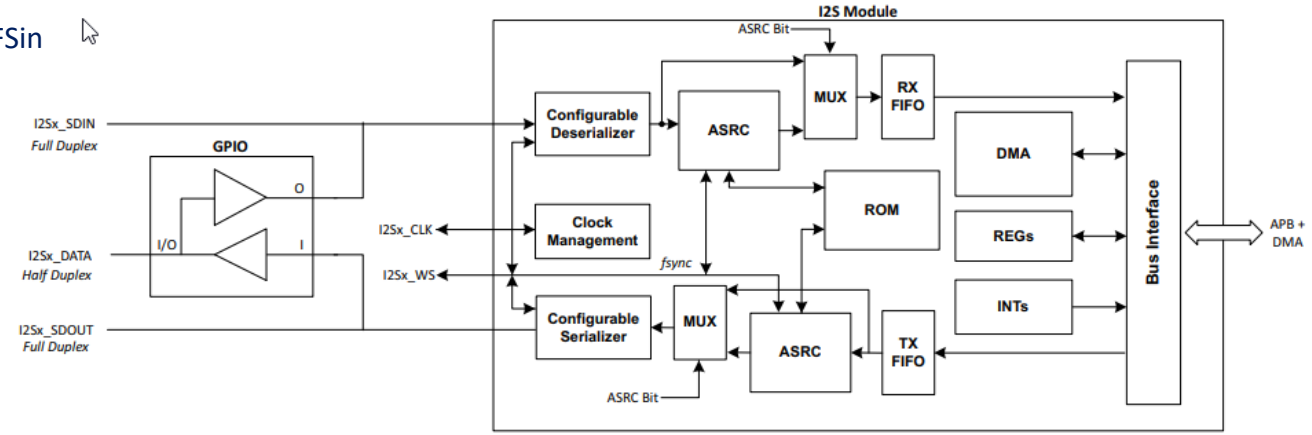
Audio Subsystem

- 1-bit PDM (pulse-density modulated) input for up to 4 pairs of microphone outputs
- 24-bit PCM conversion/output at up to 192 kHz sample rate
- Support Digital Microphone Clock at 512 kHz, 1.024 MHz, 2.048 MHz, 2.45 MHz, 3.072 MHz
- PCM Sampling Rate: 8 kHz, 16 kHz, 32 kHz, 48 kHz, 96 kHz, 192 kHz
- PGA Gain: -12dB - +34.5dB gain with 1.5dB/Step
- High Performance Mode
  - 110dB SNR, BW=20 kHz (A-weighted)
  - 105dB THD+N, BW=20 kHz (A-weighted)
- Mid Performance
  - 107dB SNR, BW=6.7 kHz (A-weighted)
  - 101dB THD+N, BW=6.7 kHz (A-weighted)
- Reduced Performance mode
  - 88dB SNR, BW=6.7 kHz (A-weighted)
  - 83dB THD+N, BW=6.7 kHz (A-weighted)
- Power Down Mode support

OPERATING MODE	F_{PDMA_CKO} (MHz)	F_S (kHz)	OSR	DIV_MCLKQ [1:0]	MCLKDIV [3:0]	SINCRATE [6:0]	SINAD (dB)	DR (dB)
High Performance Mode	6.144	96	64	1	1	32	103	110.8
	3.072	48	64	1	3	32	105.5	108.7
	3.072	24	128	1	3	64	122.8	120.9
	3.072	16	192	1	3	96	116.1	120.4
	1.536	16	96	1	7	48	115.4	120.5
Reduced Performance Mode	3.072	96	32	1	3	16	86	87.9
	1.536	48	32	1	7	16	83.2	88.8
	0.768	16	48	1	15	24	89.7	97.2
Mid Performance Mode	1.536	24	64	1	7	32	101	107.6
	1.024	16	64	1	11	32	100	106.8

Full Duplex I2S with ASRC

- Supported sample rates include 8, 11.025, 16, 22.05, 32, 44.1, 48, 96 and 192 kHz
- Audio sample sizes of 8,16, 24 and 32 bit
- 2 instances (IPB0/IPB1) of full-duplex I2S (stereo TX + stereo RX) using shared CLK & FS
- Master and slave
- Optional Asynchronous Sample Rate Conversion (ASRC) on slave I2S channels
  - 2-channel audio sample rate converter
  - Sample size: 24 bits (for internal processing, but accepts 8, 16, 24 or 32 bits)
  - Lower than -130 dB THD+N for common conversion ratios (when using 24-bit or 32-bit samples)
  - Minimum input clock frequency: FSin
  - Minimum output clock frequency: FSout
  - Extremely fast synchronization time with the input audio stream: 128 / FSin
  - Latency: (FIFO_SIZE / (2FSin)) + (2 / FSout)
  - Automatically adjusts to changes in both input and output sample rates
  - High input jitter tolerance
  - Input sample rate range: 8 kHz to 192 kHz
  - Output sample rate range: 8 kHz to 192 kHz
  - Maximum down conversion of 3.9:1 (silicon rev B)
  - Maximum up conversion of 1:7
  - Fixed FSYNC:SCLK ratio of 64:1 required



ambiq

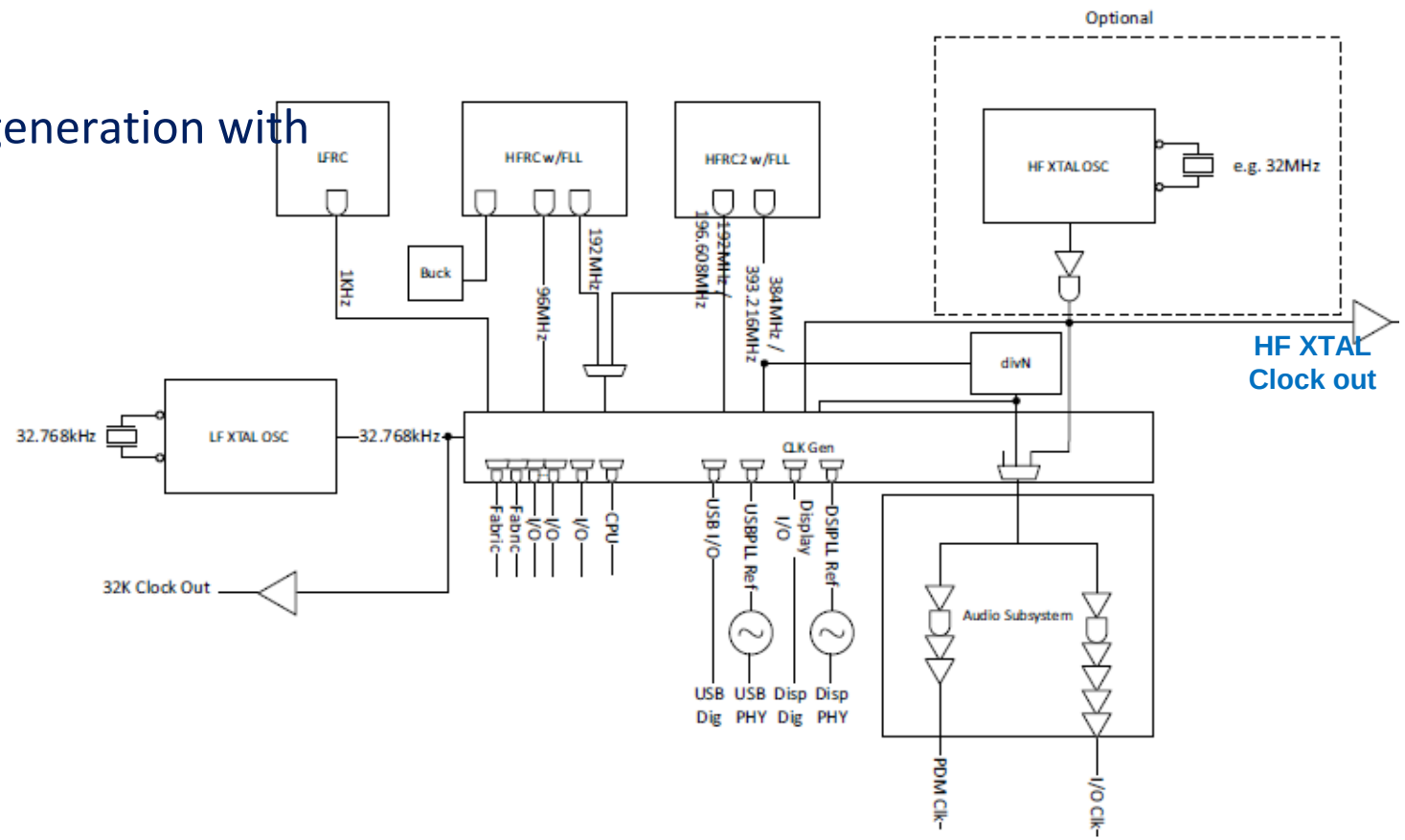
endpoint intelligence

System Management

Clock Generation

System Management

- 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 circuit
  - Low frequency XTAL oscillator circuit
  - 2x High frequency RC oscillator
  - Low frequency RC oscillator
  - High frequency PLL circuits
  - SoC clock generation logic
  - Audio subsystem clock generation logic



ambiq

Timing: Clock Generation

System Management

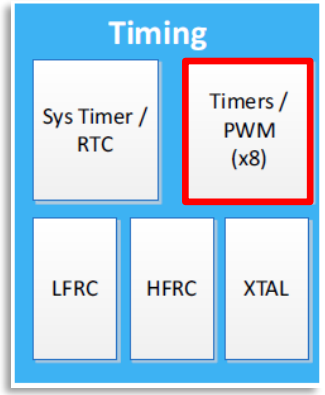
- High Precision XT Oscillator (XT)
  - Tuned to an external 32.768 kHz XTAL
- HFRC (High Freq. RC oscillator)
  - 96 MHz or 192MHz
  - 2% accurate at 25C across voltage (1.8V to 2.2V)
  - Auto adjust function continuously calibrates HFRC to XTAL to ensure < 1% error (<0.1% error at stable temp)
- HFRC2 (High Freq. RC oscillator2)
  - 192MHz (can be adjusted to be harmonic of 44.1KHz or 48KHz audio frequency)
  - Consumes most power
  - Auto adjust function continuously calibrates HFRC2 to high frequency XTAL to ensure < 100ppm
- LFRC (Low Freq. RC oscillator)
  - Nominally 1.024 kHz
  - Low power but inaccurate (25%+ error)

Timing: Clock Generation con't

System Management

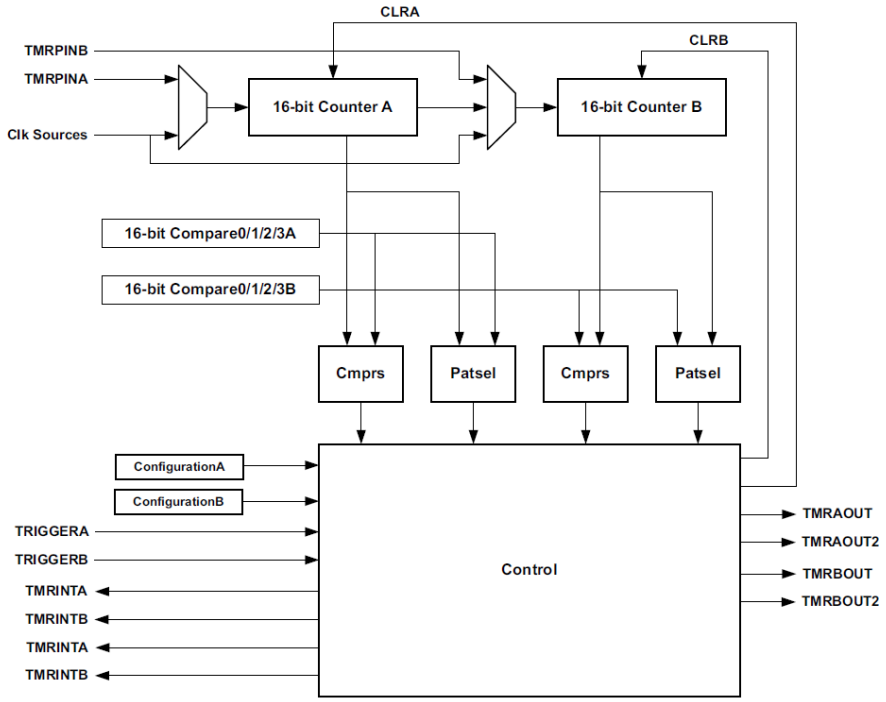
- HFRC Adjust (HFAdj)
  - Auto adjust function used to refine HFRC frequency to account for frequency drift

Timing: Counter/Timers - TIMER

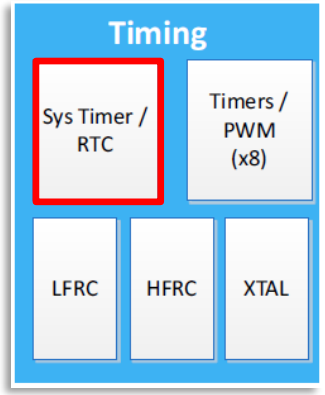


System Management

- Supports multiple clock sources
  - As fast as 24MHz (HFRC/4)
  - As slow as 1Hz (LFRC/1024)
- 16 32-bit timers
- 2 compare modules per TIMER
- Standard Counter functionality:
  - Interrupt after a specified delay
  - Interrupt periodically with a specified period
  - Generate an external pulse of a specified width, after a configured delay
  - Generate an external PWM signal with a specified period and duty cycle



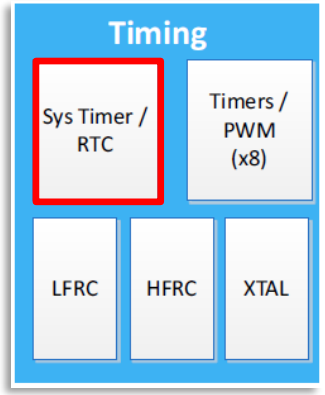
Timing: Real Time Clock (RTC) Module



System Management

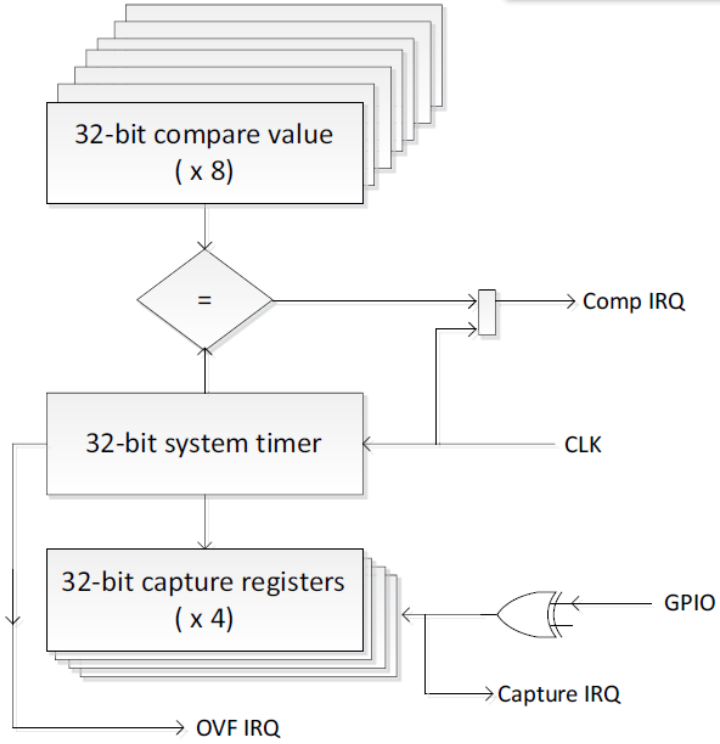
- Same RTC IP as AMx8x5 devices
- Clocked from XTAL or LFRC
- 100th of a second resolution (10ms)
- Automatic leap year calculation
- Programmable alarm interrupts
  - Every 100th sec., 10th sec., sec., min., hour, day, week, month, or year
- Continues to operate through all resets except POA power-on reset which occurs at TBD voltage (~ 1.3V) so time is retained through all other resets including brownout

Timing: System Timer (Stimer) Module

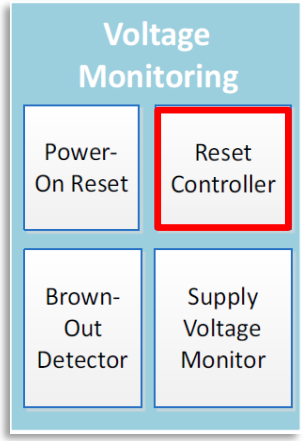


System Management

- 32bit System Timer
- Clocked from XTAL, LFRC, or HFRC/16
- 8x 32bit compare registers with interrupts
- Offsets from “NOW” are written to compare registers and compare register value is determined by HW
- Only reset by POA (Power On Analog - system cold reset). Retains time across all POI and POR (system warm reset) events except full power cycle.
- Contains three 32-bit NVRAM registers that are only reset by POA



Reset Controller

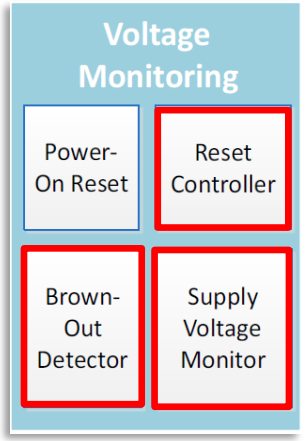


System Management

- Three levels of Reset
  - POR = POR Software Reset = HRESET = SYSRESETn
  - POI = POI Software Reset
  - POA = Power-on Reset
- Most resets trigger POR reset, which is the shallowest Reset
- POI Software reset is slightly deeper reset that triggers INFO space settings to be reloaded from Flash
- POA is the deepest reset and is only triggered by voltage going below POA voltage (~1.3V – See data sheet for exact value)

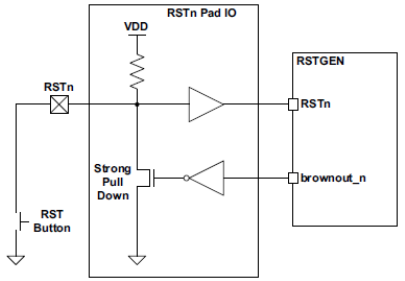
Reset Level	CPU	GPIO	INFO Settings (Trims, Options, and Security)	Peripherals (Except XT, LFRC, RTC, and Stimer)	XT, LFRC, RTC, and Stimer
POR (HRESET/SYSRESETn)	Reset	Reset	Not Reset	Reset	Not Reset
POI	Reset	Reset	Reset	Reset	Not Reset
Power-on Reset (POA)	Reset	Reset	Reset	Reset	Reset

Reset - Voltage Monitoring and nRST pin

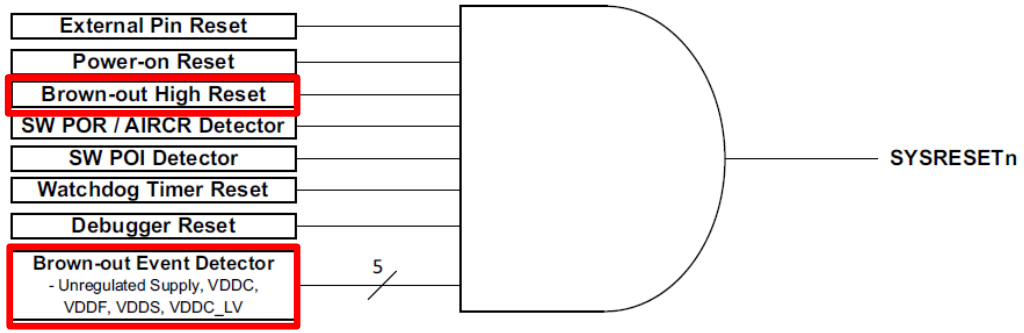


System Management

- BOD Reset max threshold 1.755V
- Optional BODH 2.1V brownout configurable for Reset or Interrupt
- External RSTn pin has built-in weak pull-up resistor
- RSTn pin is driven low when brownout is detected



RSTn Pin

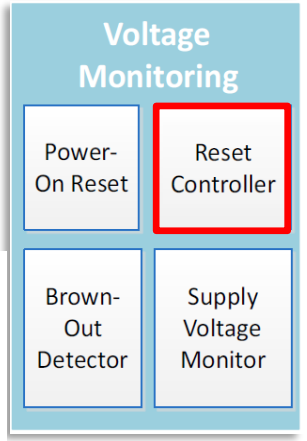


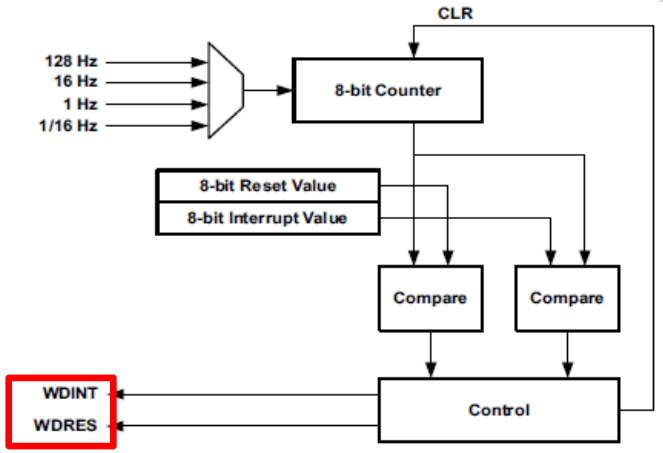
SYSRESET Sources

Reset – Watchdog Timer

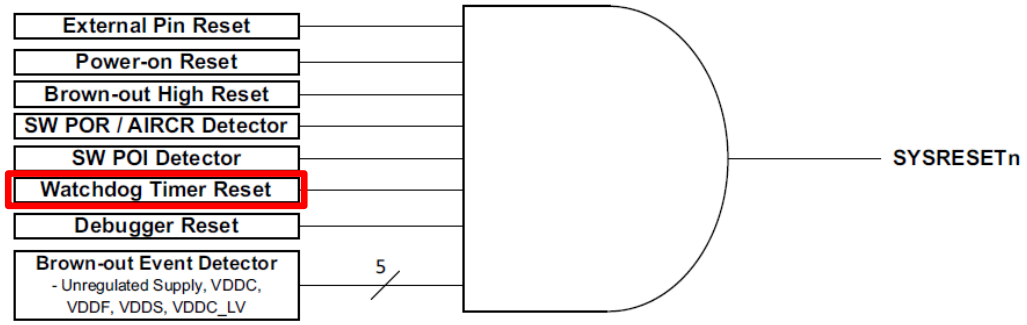
System Management

- WDT (watchdog timer)
  - Clocked from LFRC with divider options from 1/16Hz to 128Hz
  - Two configurable WDT 8-bit register compare values:
    - One compare for Interrupt
    - One compare for Reset



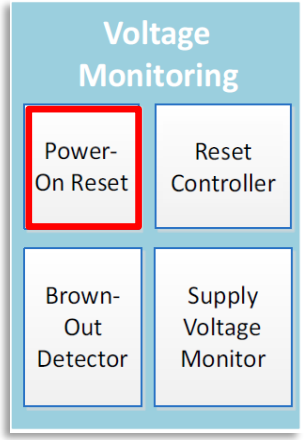


Watchdog Timer



SYSRESET Sources

Reset – Power on Reset (POA)



System Management

- Some registers are only reset by System Power-Down (POA) which occurs ~ 1.3V (See data sheet for exact voltage)
- During all other resets, including RSTn, BOD, WDT, and POR and POI Software resets, the following are retained:
  - All RTC registers retain state
  - RTC and STIMER counters continue operation from 32kHz XTAL or from LFRC (if below BODL). If clock sources stop oscillating at very low voltage, the RTC and STIMER will continue to maintain state.
  - Clock configuration registers retain state

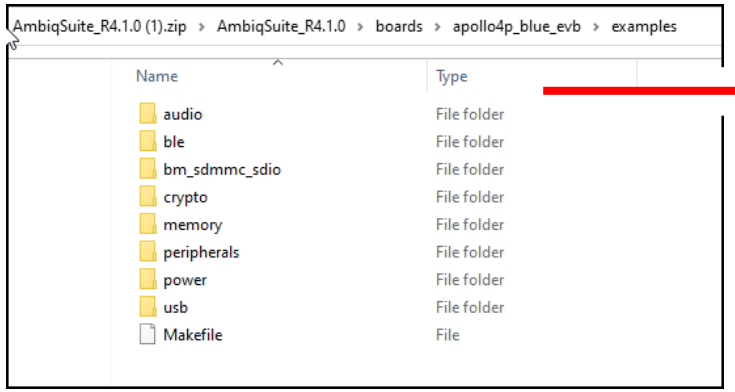
ambiq

endpoint intelligence

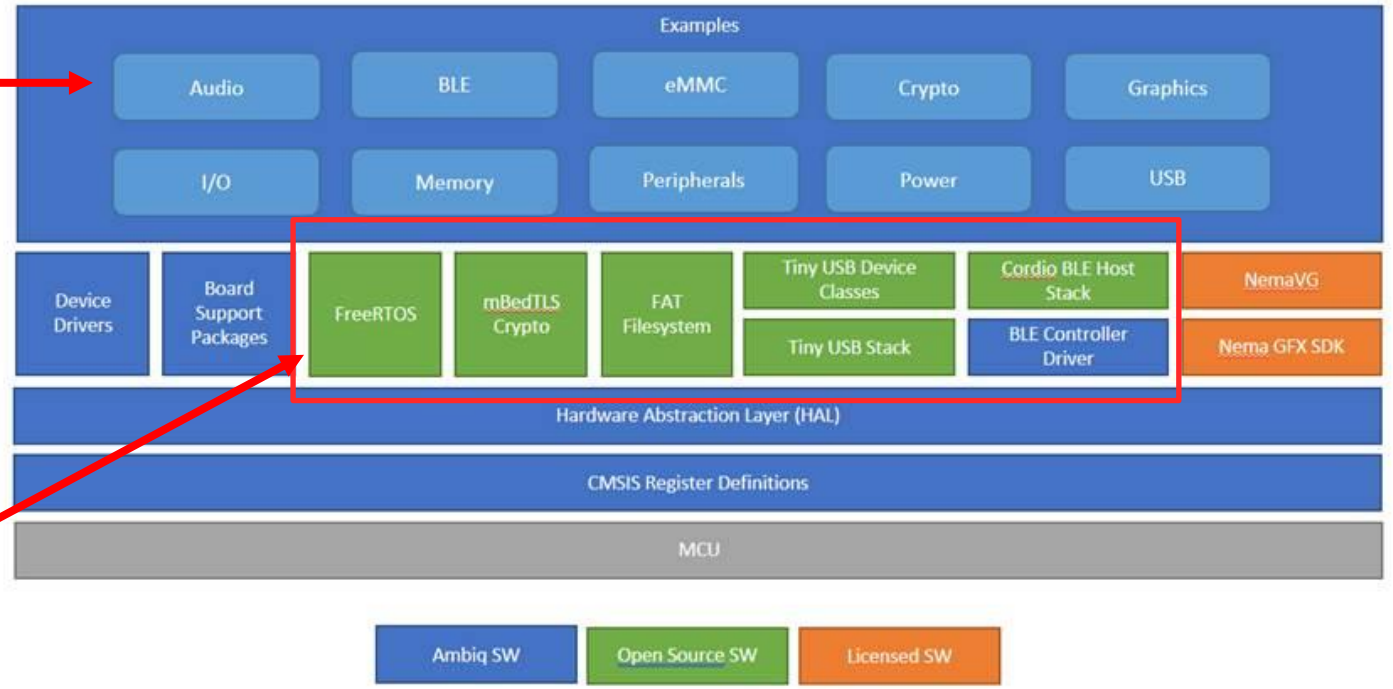
SDK Overview

AmbiqSuite SDK Rev 4.x

SDK



Apollo4 Family Software Stack



No customer license fees required

Open source license agreements included in SDK zip file

AmbiqSuite SDK Rev 4.x

SDK

- Register files (CMSIS compatible)
- HAL source code
- Cordio BLE Stack
- BSP source code for each EVB
- API documentation

- Extensive software examples
  - Source code
  - Keil, IAR, and GCC project files
- Application Notes
- SDK Getting Started Guide

Links to development tools that support Apollo4 family devices[1]:

- SEGGER J-Link Software: https://www.segger.com/downloads/jlink
- KEIL uVision 5[2]: https://www.keil.com/demo/eval/arm.htm
  - Latest Keil Pack: http://www.keil.com/dd2/pack/#/third-party-download-dialog
- IAR Version[3]: https://www.iar.com/iar-embedded-workbench/tools-for-arm/arm-cortex-m-edition/
- GCC: https://gcc.gnu.org

Ambiq Suite: HAL

SDK

- HAL (hardware abstraction layer) makes using Apollo4 easier
- Abstraction provides simple functions and macros for enabling and using peripherals
- HAL also provides macros that make constants easier to read and use
  - AM_HAL_ADC_CLOCK_12MHZ instead of 0x01000000
- Several functions based on C macros that minimize number of executed instructions

Ambiq Suite: BSPs

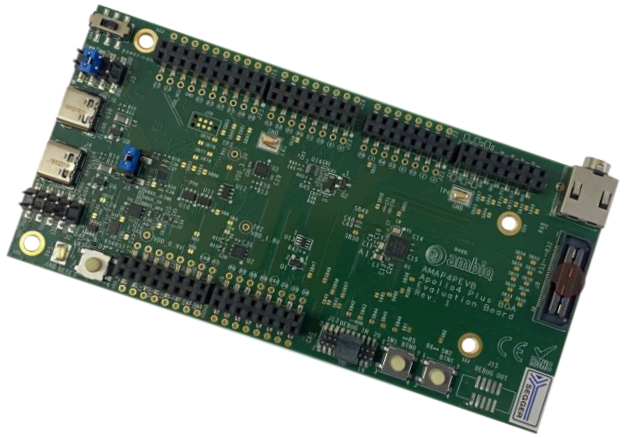
SDK

- A BSP (board support package) contains additional functions and macros for use with specific board hardware
- Each EVK board has its own BSP
- BSPs make prototyping with Ambiq evaluation materials easier
- All Ambiq Suite code examples include use of BSP functions

Debug Tools

SDK

- Ambiq MCUs are supported by industry-standard tools
- Standard Cortex SWD debug interface.
  - Support for Apollo MCU family included in Segger J-Link software
  - Apollo family EVBs have an onboard J-Link interface. Just connect USB.
- All example code provided in Ambiq SDK come with Keil and IAR projects and GCC Makefile
- Keil and IAR downloads with Apollo family support built-in provided by their respective manufacturers



ARM KEIL Microcontroller Tools IAR SYSTEMS GCC

Apollo4 Plus SW and Documentation

- Silicon
  - Apollo4 Plus SoC Datasheet
  - Apollo4 Blue Plus SoC Datasheet
  - Apollo4 Plus Errata List
- AmbiqSuite R4.x SDK
  - Apollo4 Family Programmers Guide
  - Apollo4 Family Getting Started Guide
- App Notes
  - Apollo4 and Apollo4 Blue Secure Update Users Guide
  - Knowledge Base Articles
- Security
  - Apollo4 OEM Provisioning Update and Tools Users Guide
  - Apollo SoC Security Users Guide
  - Apollo4 Blue BLE Controller XTAL32MHz CLK Request
  - Apollo4 Plus EVB intro and HW design files
- Marketing Collateral
  - Product Brief for each SoC
  - Apollo SoC Selector Guide
  - Apollo SoC Family Brochure
  - Apollo4 Plus Display Kit Product Brief

https://ambiq.com/apollo4-plus/

Apollo4 Plus Display Kit

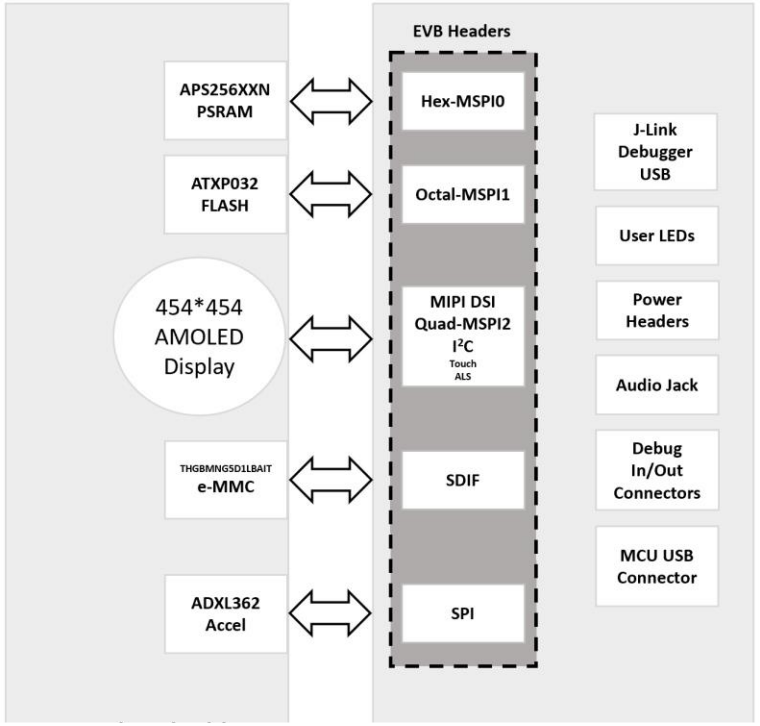
Highlights

- Kit includes Apollo4 EVB and a compatible Apollo4 Display Shield
- Features a 454x454 AMOLED Display with Touch capability
- Shield contains PSRAM, Flash, an eMMC module, and accelerometer
- Develop with AmbiqSuite SDK and NemaGFX API
- Supports NemaGFX GUI Builder and Embedded Wizard GUI Builder

Graphics Capability

- Fully programmable engine with VLIW instruction set
- 4:1, 4:1 with alpha and 6:1 Compression
- Color formats: 24/16/8-bit with/out alpha, grayscale, RGB
- 2D drawing: pixel/line drawing, filled rectangles, triangles (Gouraud shaded)
- Blit support for rotation, mirroring, stretch, source and/or destination color keying, and format conversions
- Full alpha blending with programmable blending modes
- Anti-aliasing, vector graphics, and dithering
- Image transformation: 3D perspective correct projections and texture mapping

Apollo4 Plus Display Kit (AMAP4PDISP)



Additional Resources

- Ambiq Documents and Hardware
  - Ambiq Website
  - Ambiq Content Portal
  - Knowledge Base Articles
  - Request an EVB
- Supported GUI Development Tools
  - Embedded Wizard
  - Think Silicon NEMA® | GUI-Builder
  - Koru

Part	Part Number	EVB
Apollo4 **Plus**	AMAP42KP-KBR	AMAP4PEVB AMAP4PDISP
Apollo4 Blue **Plus (KXR)**	AMAB42KP-KXR	AMA4BP**X**EVB
Apollo4 Blue **Plus** (KBR)	AMAB42KP-KBR	AMA4BPEVB

ambiq

endpoint intelligence

Thank **You**!