ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΠΜΣ Space Technologies, Applications and seRvices (STAR)

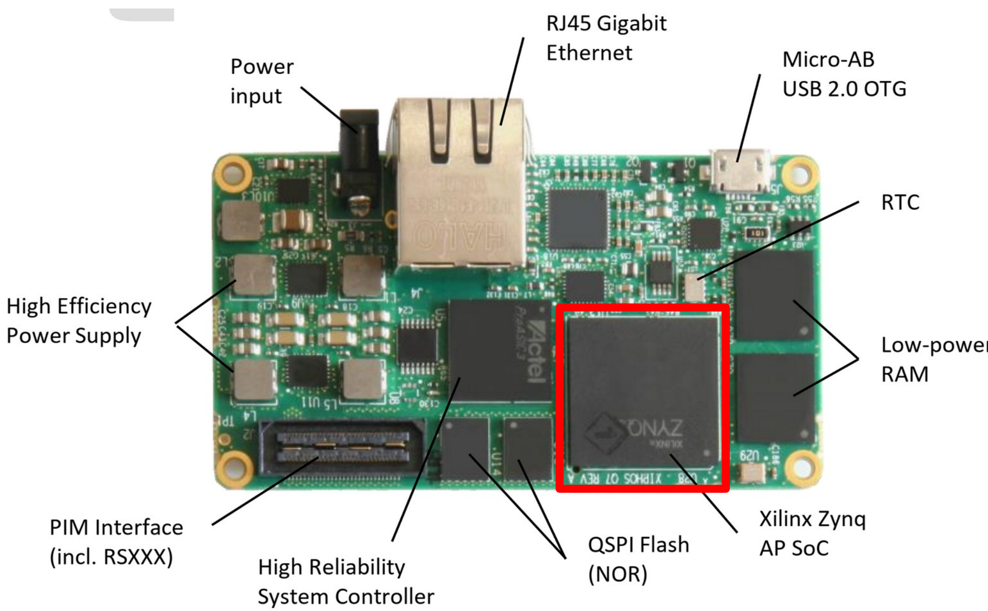
M806 Space Data Systems

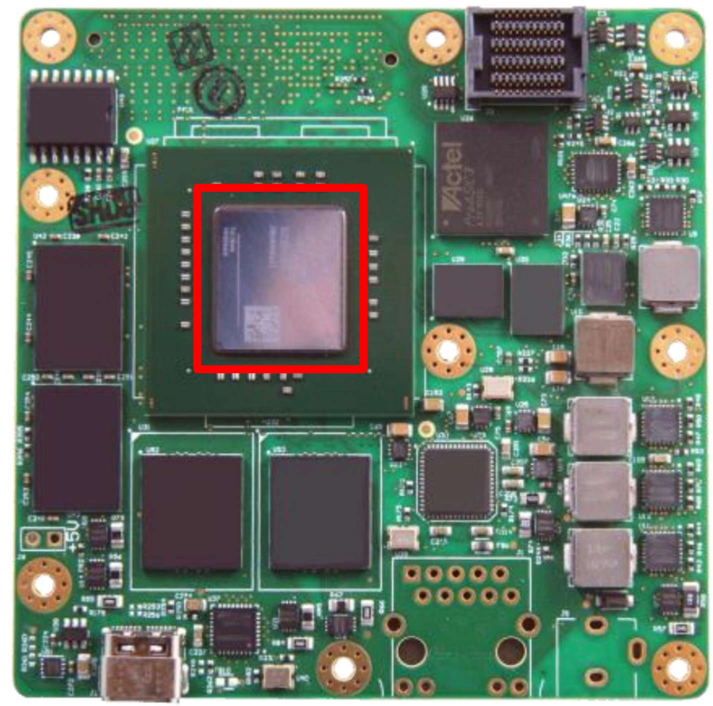
System-On-Chips for Space Data Systems

Ακαδημαϊκό Έτος 2023-2024

Νεκτάριος Κρανίτης

System-on-Chip στο διάστημα: NewSpace/Space 2.0, CubeSats, Nanosatellites





Xiphos Q7 Q-Card (Zynq-7020 FPGA SoC)
ARM dual-core Cortex-A9 MPCore processors each up to 766 MHz
Mass 24g, Power 2W, 78mm X 43mm X 9mm

Xiphos Q8 Q-Card (Zynq UltraScale+ XCZU7EG)
Quad-core ARM Cortex-A53 Application Processing Unit at up to 1.2 GHz
Mass 64g, Power 4W, 85.8mm X 80mm X 11.2mm

System-on-Chip στο διάστημα: NewSpace/Space 2.0, CubeSats, Nanosatellites

Space Inventor Z7000 Payload and Onboard Computing Platform

FEATURES

- On-board computer based on Xilinx Zync 7030 SoC
- Dual ARM® Cortex-A9 Main Processing Units 667 MHz
- Memory: 256KB on-chip memory, 256 MB RAM, 16 GB eMMC mass storage
- FPGA: 125K programmable Logic Cells
- Interfaces
  28V unregulated supply
  CAN bus
  LVDS, SpaceWire
  RS-422
  Ethernet
- Power consumption: <1.5 W idle. Up to 20 W.

Thales Alenia Space multiMIND Multi-mission payload processing system

multiMIND characteristics

PROCESSING

- MPSoC family: Xilinx Zynq Ultrascale+ ZU6EG, ZU9EG or ZU15EG
- Processing System: Quad-core ARM Cortex-A53 up to 1.5 GHz + Dual-core ARM Cortex-R5 up to 600 MHz
- Processing Logic: FF 429k-682k / LUT 215k-341k / DSP 1973-3528

MEMORY

- 4 GByte ECC working memory, 512 KByte MRAM
- 2x16 GByte NAND mission data storage
- 2x128 MByte NOR configuration storage

NKUA ERMIS CubeSat MISSION

- Payload Data Processing Unit (PDPU)
  - Developed by NKUA Onboard Data Processing Laboratory & DSCAL
  - Hosts payload mass memory
  - State-of-the art FPGA acceleration of CCSDS data processing algorithms
    - CCSDS 122
    - CCSDS 123
    - CCSDS 142.1
  - Based on GOMspace Nanomind MK3
    - Hosts a Zynq-7045 SoC

Gomspace Nanomind MK3

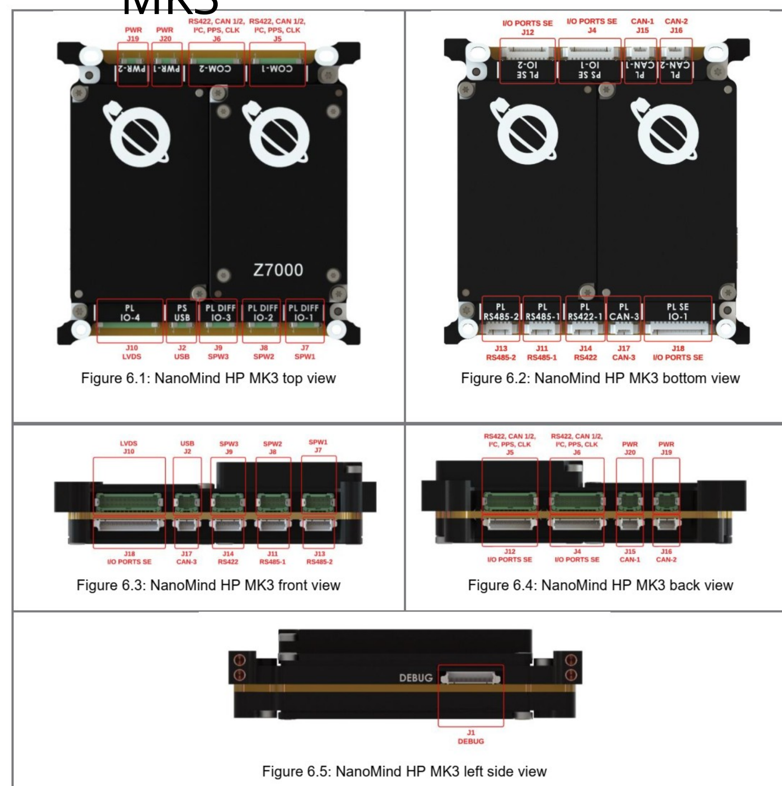


Figure 6.1: NanoMind HP MK3 top view

Figure 6.2: NanoMind HP MK3 bottom view

Figure 6.3: NanoMind HP MK3 front view

Figure 6.4: NanoMind HP MK3 back view

Figure 6.5: NanoMind HP MK3 left side view

GOMspace Nanomind MK3

Highlighted Features

- Xilinx Zynq 7030/7045 Programmable SoC
  - Dual ARM Cortex A9 MPCore, up to 800MHz
  - Powerful FPGA with 125k/350k logic cells
- 1GB DDR3 RAM
  - 512MB with Error Correction Code enabled
- 256GB NOR flash
- 70GB eMMC (pSLC) and 233GB eMMC (MLC)
  - HW Bit-Flip Detection
  - Redundant Firmware Image
- High speed interfaces
  - SpaceWire
  - USB (host/device)

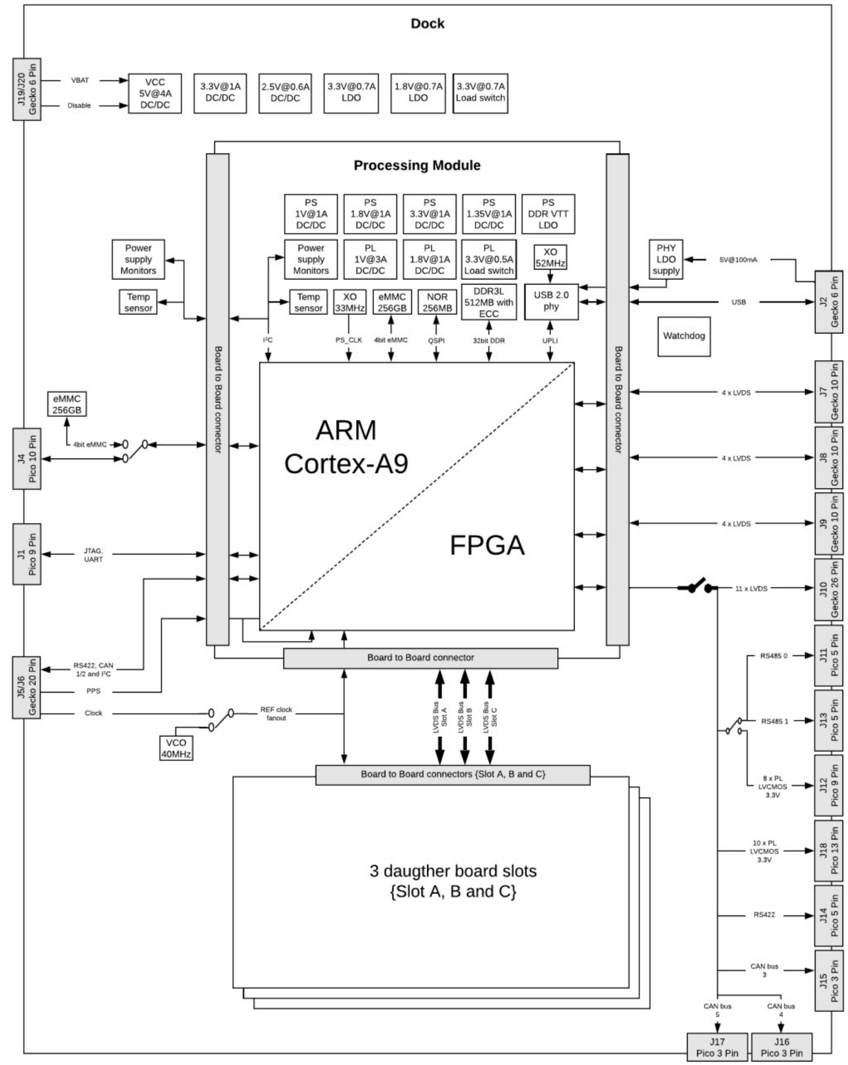


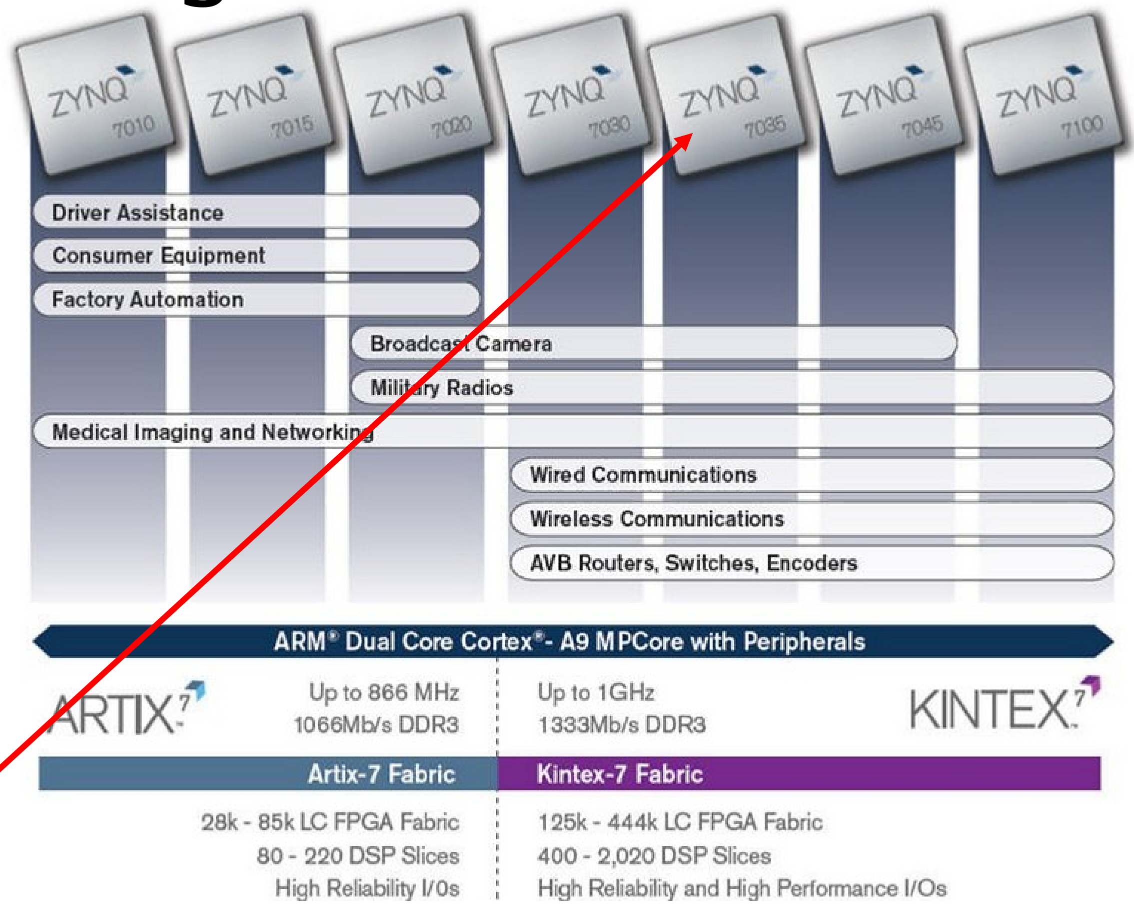
Figure 3.1: NanoMind HP MK3 Block diagram

Embedded System Design with Zynq-7000 SoC

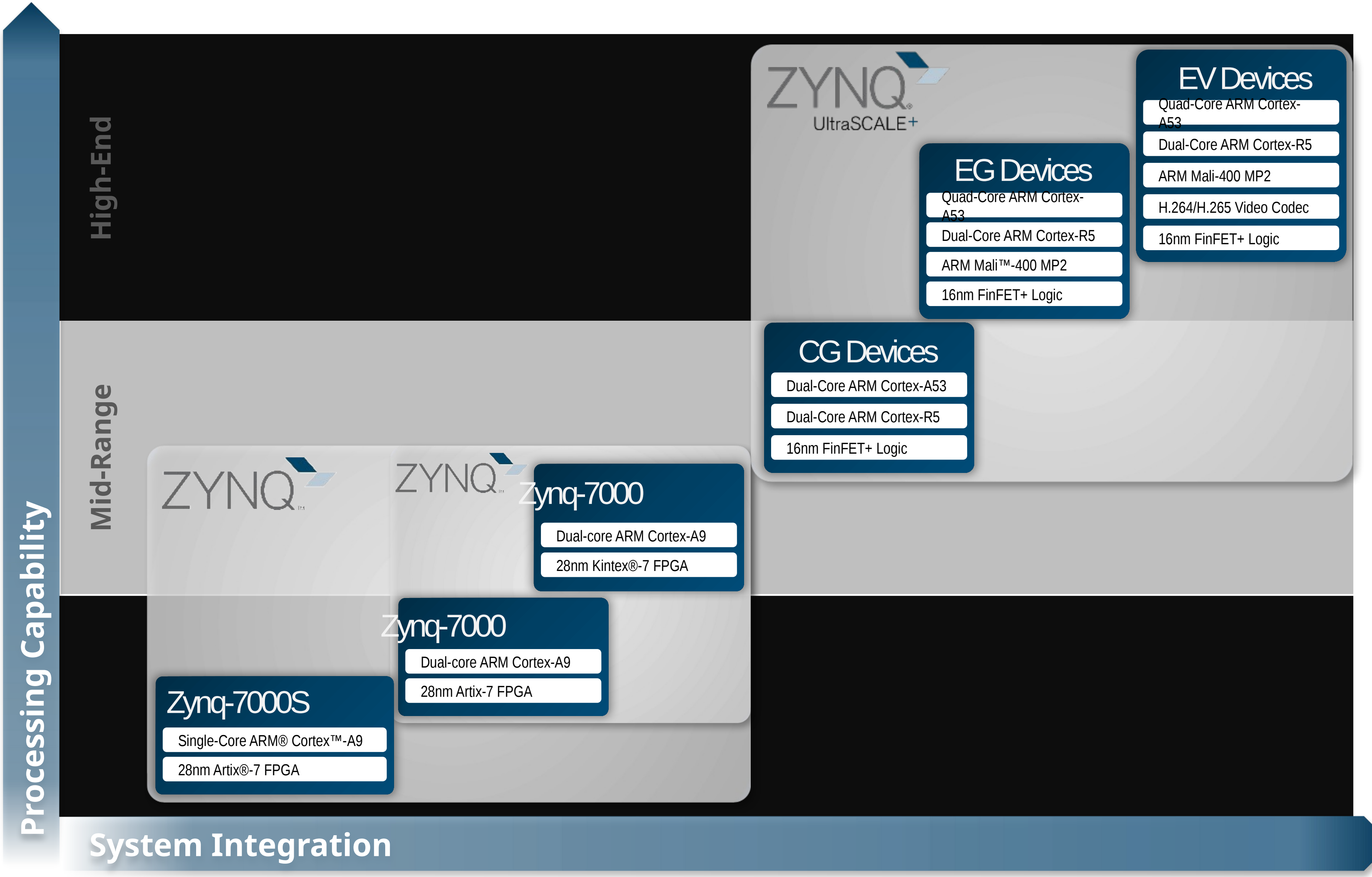
- Embedded design in Zynq-7000 SoC is based on:
  - Processor and peripherals
    - Dual ARM® Cortex™ -A9 processors of Zynq-7000 SoC
    - AXI interconnect
    - AXI component peripherals
    - Reset, clocking, debug ports
  - Software platform for processing system
    - Standalone or other (e.g. Linux) OS
    - C language support
    - Processor services
    - C drivers for hardware
  - User application
    - Interrupt service routines (optional)

Processing System and Programmable Logic

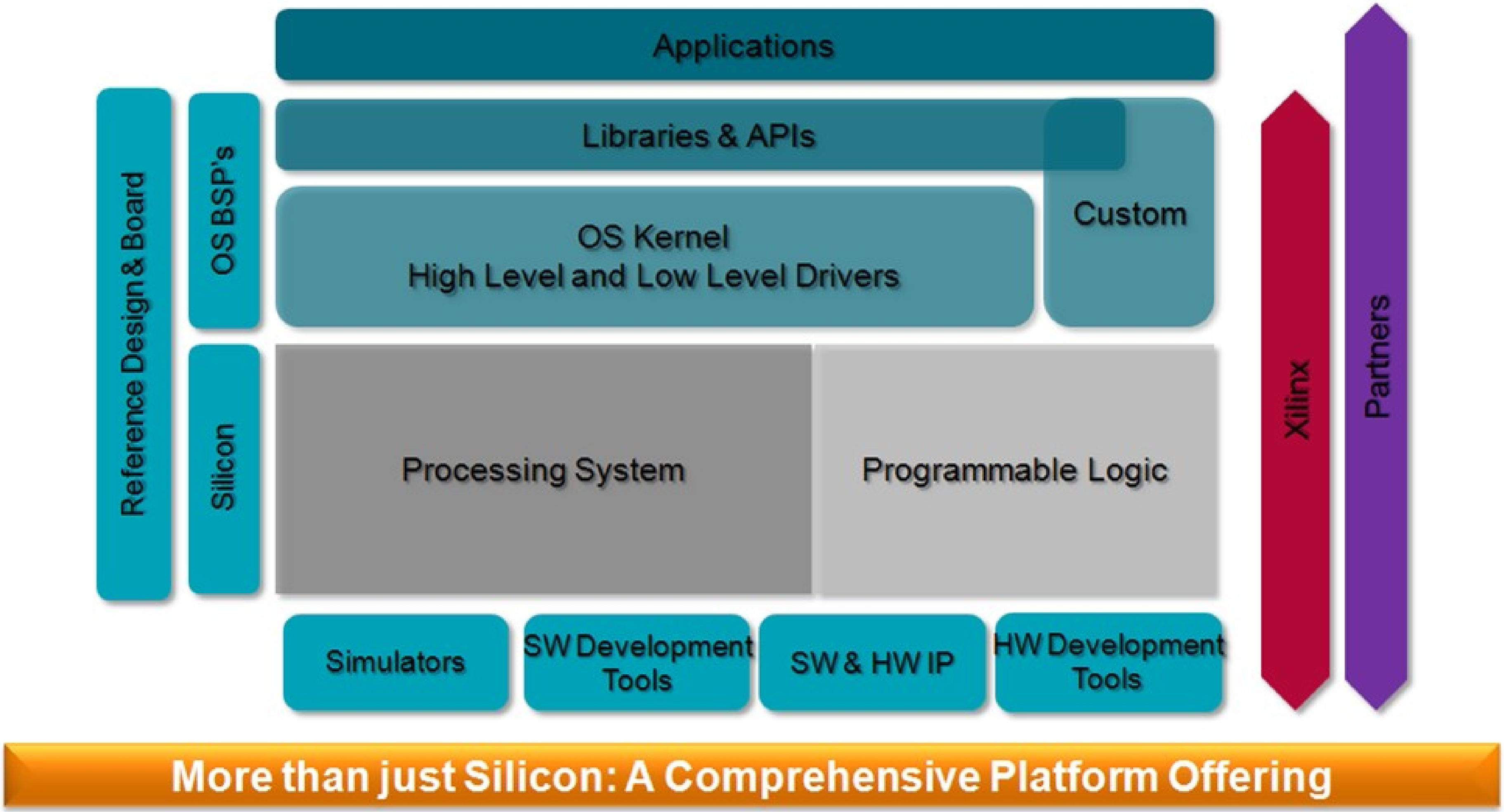
- Zynq-7000 SoC architecture consists of 2 major sections
  - **PS: Processing system**
    - Dual ARM Cortex-A9 CPU
    - Multiple peripherals
    - Hard silicon core
  - **PL: Programmable logic**
    - Uses 7-series prog. logic
    - Artix™-7 devices: Z-7010,
      Z-7015 and Z-7020
    - Kintex™-7 devices: Z-
      d Z-



Extending Scalability Across the Zynq Portfolio



Zynq-7000 SoC



Zynq-7000 Family Highlights

- Complete ARM®-based processing system
  - Application Processor Unit (APU)
    - Dual ARM Cortex™-A9 processors
    - Caches and support blocks
  - Fully integrated memory controllers
  - I/O peripherals
- Tightly integrated programmable logic
  - Used to extend the processing system
  - Scalable density and performance
- Flexible array of I/O
  - Wide range of external multi-standard I/O
  - High-performance integrated serial transceivers
  - Analog-to-digital converter inputs

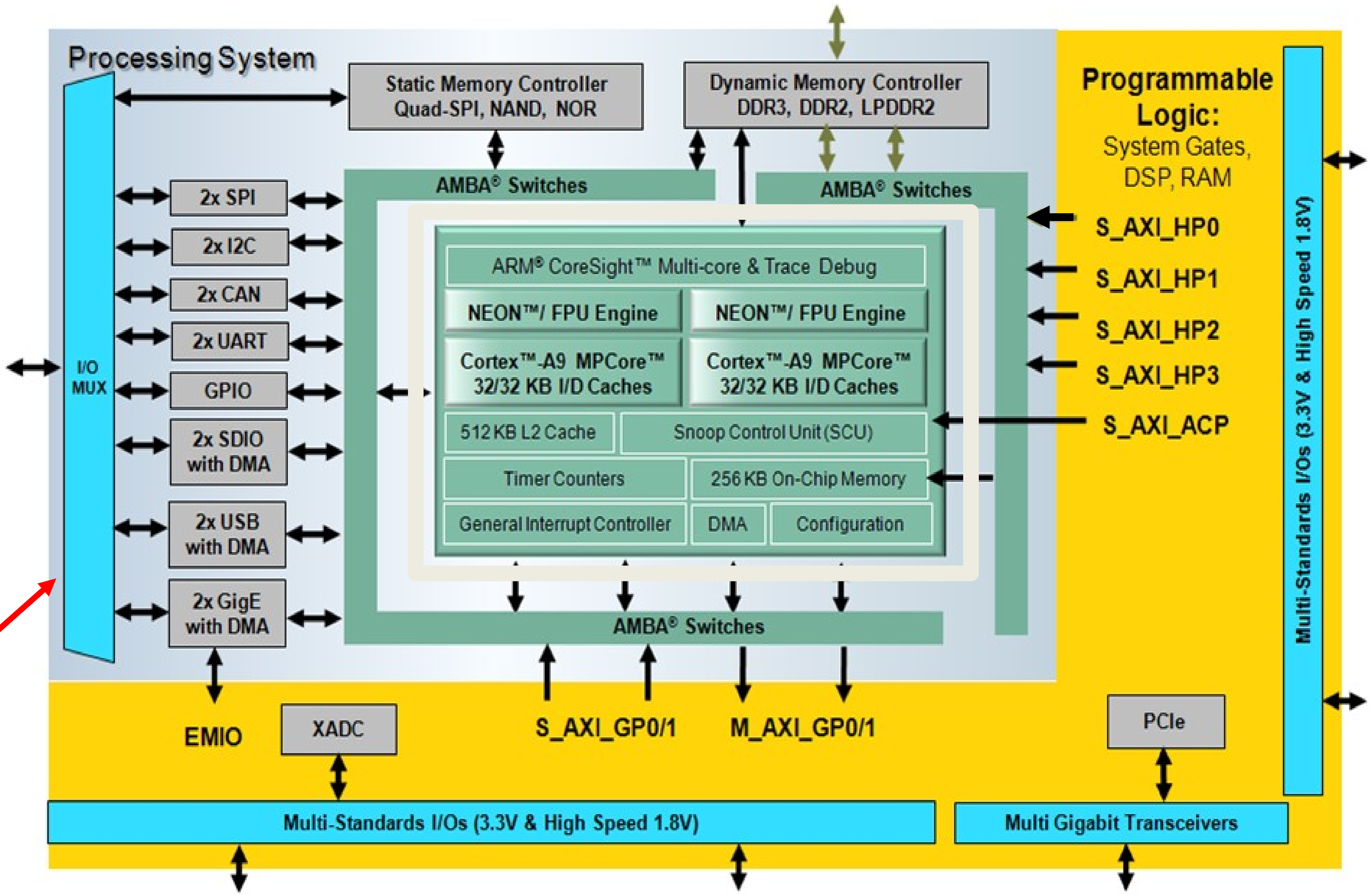
ARM Processor Product Families

- Legacy ARM processors
  - ARM7, ARM9 (not the Cortex-A9 processor), ARM11
- Cortex family of processors
  - Cortex-A#: "**A**" application
    - The products support a memory management unit (MMU)
    - Excellent for operating systems
  - Cortex-R#: "**R**" real time
    - The products support a memory protection Unit (MPU)
    - Better determinism than an MMU
  - Cortex-M#: "**M**" Embedded microcontroller
- There are some products that are implemented differently but use the same ARM Architecture
  - Cortex-A8 and Cortex-A9 processors

ARM Processor Architecture

- ARM Cortex-A9 processor implements ARMv7-A ISA
  - ARMv7 is ARM Instruction Set Architecture (ISA)
    - Thumb instructions: 16 bits; Thumb-2 instructions: 32 bits
    - NEON: ARM's Single Instruction Multiple Data (SIMD) instructions
  - ARMv7-A: Application set, support for Memory Management Unit (MMU)
  - ARMv7-R: Real-time set, support for Memory Protection Unit (MPU)
  - ARMv7-M: Microcontroller set that is the smallest set
- ARM Advanced Microcontroller Bus Architecture (AMBA®) protocol
  - AXI3: Third-generation ARM interface
  - AXI4: Adding to the existing AXI definition (extended bursts, subsets)
- Cortex is the new family of processors
  - ARM family is older generation
  - Cortex is current
  - MMUs in Cortex processors and MPUs in ARM

Zynq-7000 SoC Block Diagram



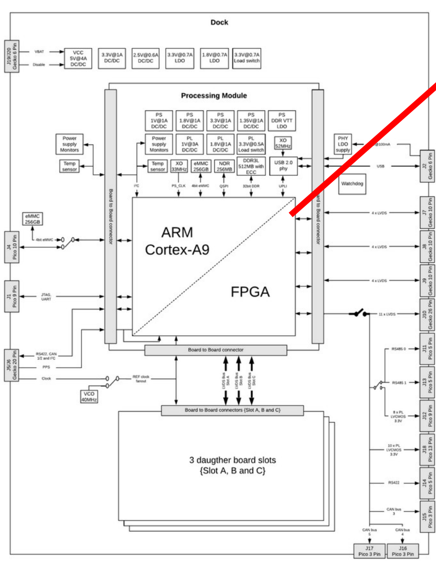


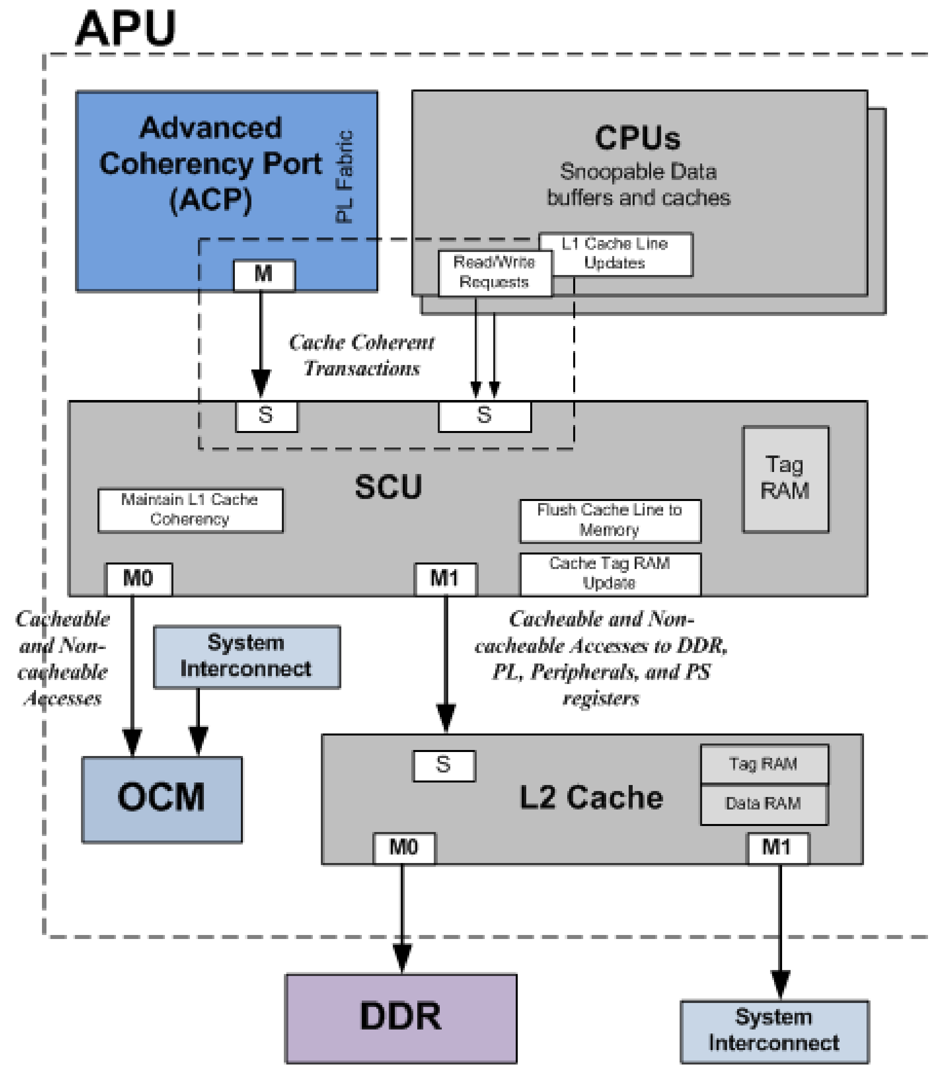
Figure 3.1: NanoMind HP MK3 Block diagram

PS Components

- Application processing unit (APU)
- I/O peripherals (IOP)
  - Multiplexed I/O (MIO), extended multiplexed I/O (EMIO)
- Memory interfaces
- PS interconnect
- DMA
- Timers
  - Public and private
- General interrupt controller (GIC)
- On-chip memory (OCM): RAM
- Debug controller: CoreSight

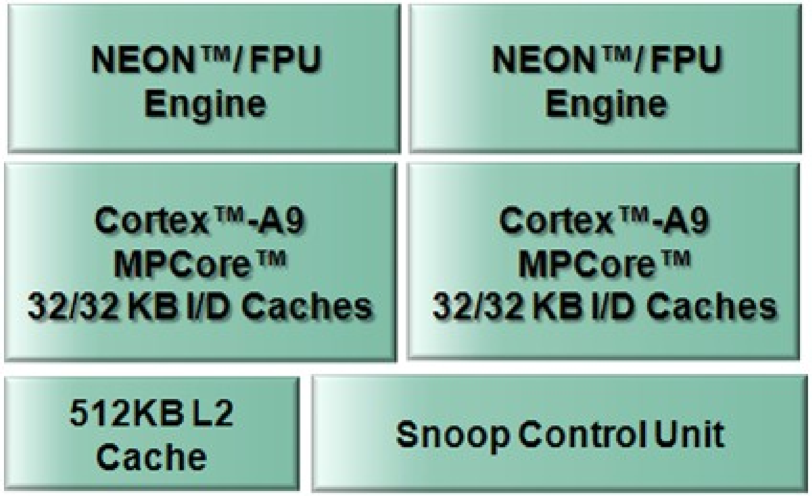
Application Processing Unit (APU)

- Heart of the PS
- Tightly coupled processors and sub-components for maximum performance
- Tied to other PS components and PL via the PS interconnect



Inside the APU

- Dual ARM® Cortex™-A9 MPCore with NEON extensions
  - Up to 1GHz operation
  - 2.5 DMIPS/MHz per core
  - Separate 32KB instruction & data caches
- Snoop Control Unit (SCU)
  - L1 cache snoop control
    - Accelerator coherency port
- Level 2 cache and controller
  - Shared 512 KB cache with parity



APU Sub-Components

- General interrupt controller (GIC)
- On-chip memory (OCM): RAM and boot ROM
- Central DMA (eight channels)
- Device configuration (DEVCFG)
- Private watchdog timer and timer for each CPU
- System watchdog and triple timer counters shared between CPUs
- ARM CoreSight debug technology

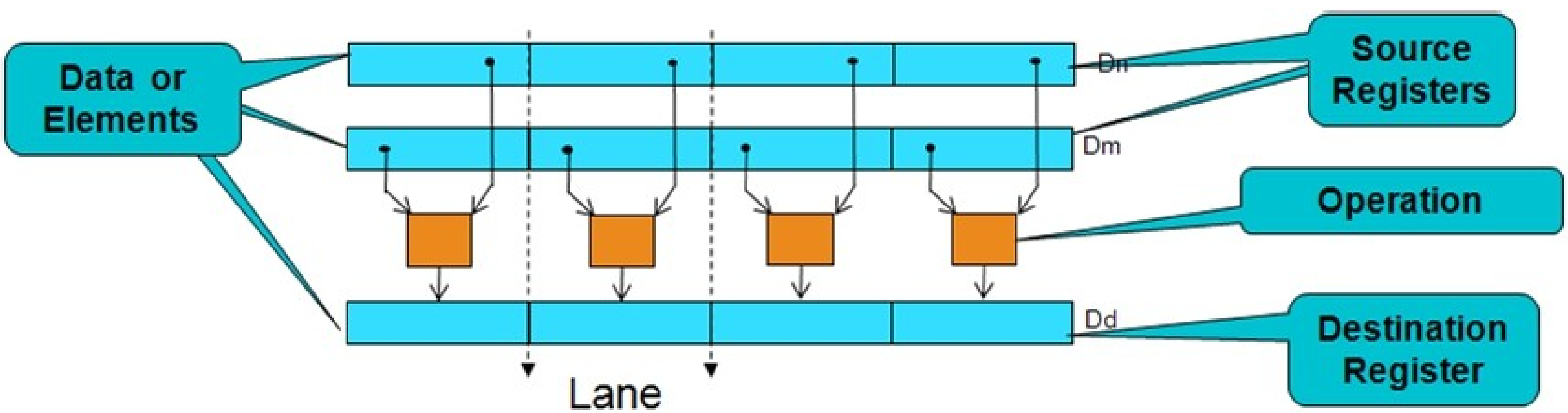
APU Internal Address Map

- All registers for both CPUs are grouped into two contiguous 4KB pages
  - Accessed through a dedicated internal bus
- Fixed at 0xF8F0_0000 with a register block size of 8 KB
  - Each CPU uses an offset into this base address

0x0000-0x00FC	SCU registers
0x0100-0x01FF	Interrupt controller interface
0x0200-0x02FF	Global timer
0x0600-0x06FF	Private timers and watchdog timers
0x1000-0x1FFF	Interrupt distributor

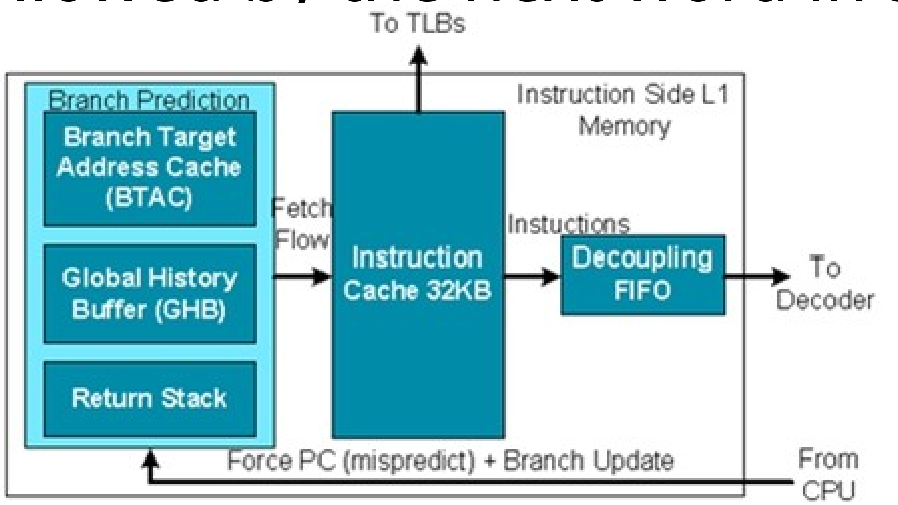
Vector Processing using NEON

- NEON is the ARM codename for the vector processing unit
  - Provides multimedia and signal processing support
- FPU is the floating-point unit extension to NEON
  - Both NEON and FPU share a single set of registers
- NEON technology is a wide Single Instruction, Multiple Data (SIMD) parallel and co-processing architecture
  - 32 registers, 64-bits wide (dual view as 16 registers, 128-bits wide)
  - Data types can be: signed/unsigned 8-bit, 16-bit, 32-bit, 64-bit, or 32-bit float



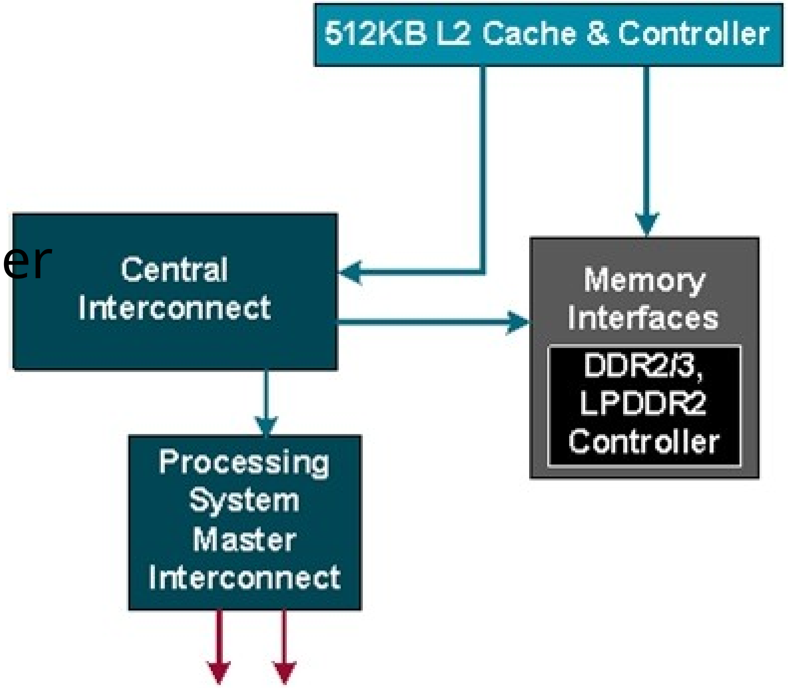
L1 Cache Features

- Separate instruction and data caches for each processor
- Caches are 4-way, set associative and are write-back
- Non-lockable
- Eight words cache length
- On a cache miss, critical word first filling of the cache is performed followed by the next word in sequence



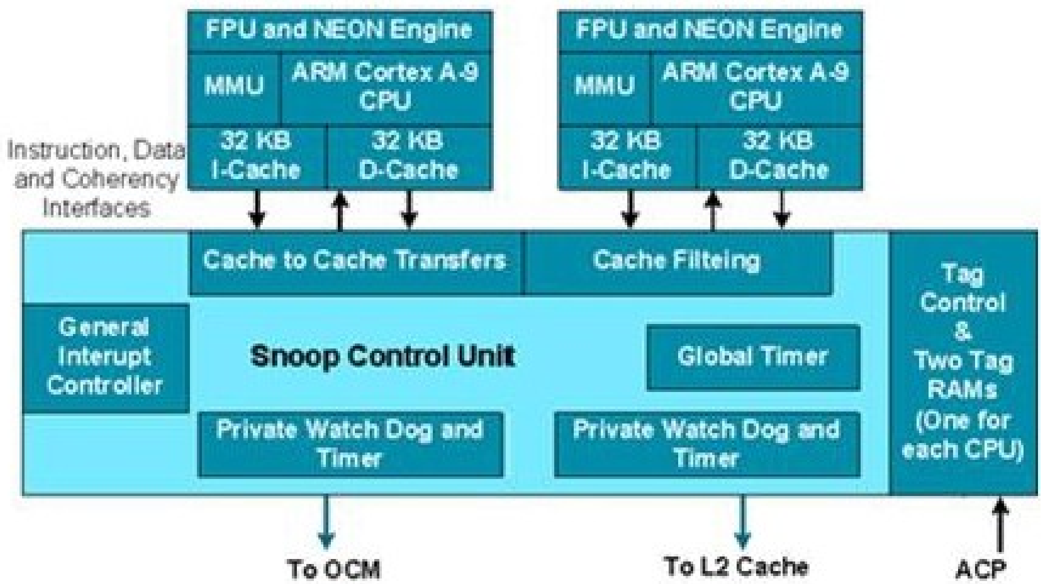
L2 Cache Features

- 512K bytes of RAM built into the SCU
  - Latency of 25 CPU cycles
  - Unified instruction and data cache
- Fixed, 256-bit (32 words) cache line size
- Support for per-master way lockdown between multiple CPU
- Eight-way, set associative
- Two AXI interfaces
  - One to DDR controller
  - One to programmable logic master (to peripherals)



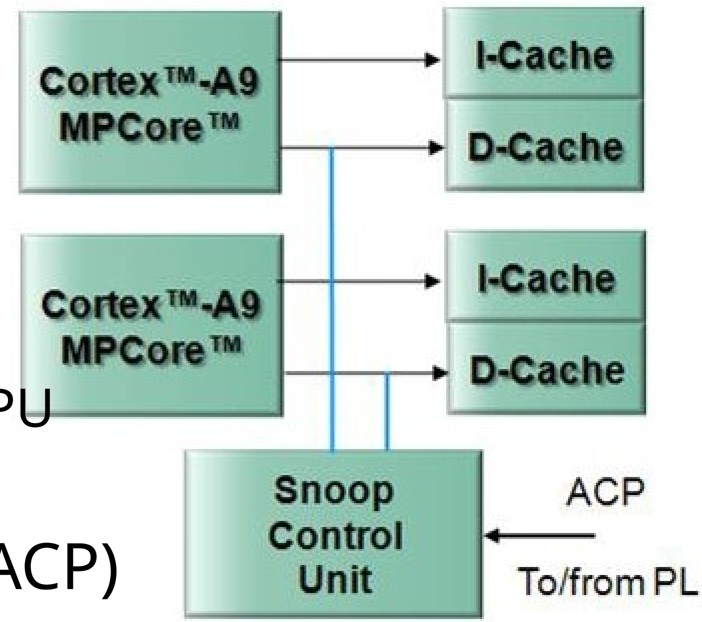
Snoop Control Unit (SCU)

- Shares and arbitrates functions between the two processor cores
  - Data cache coherency between the processors
  - Initiates L2 AXI memory access
  - Arbitrates between the processors requesting L2 accesses
  - Manages ACP accesses
  - A second master port with programmable address filtering between OCM and L2 memory support



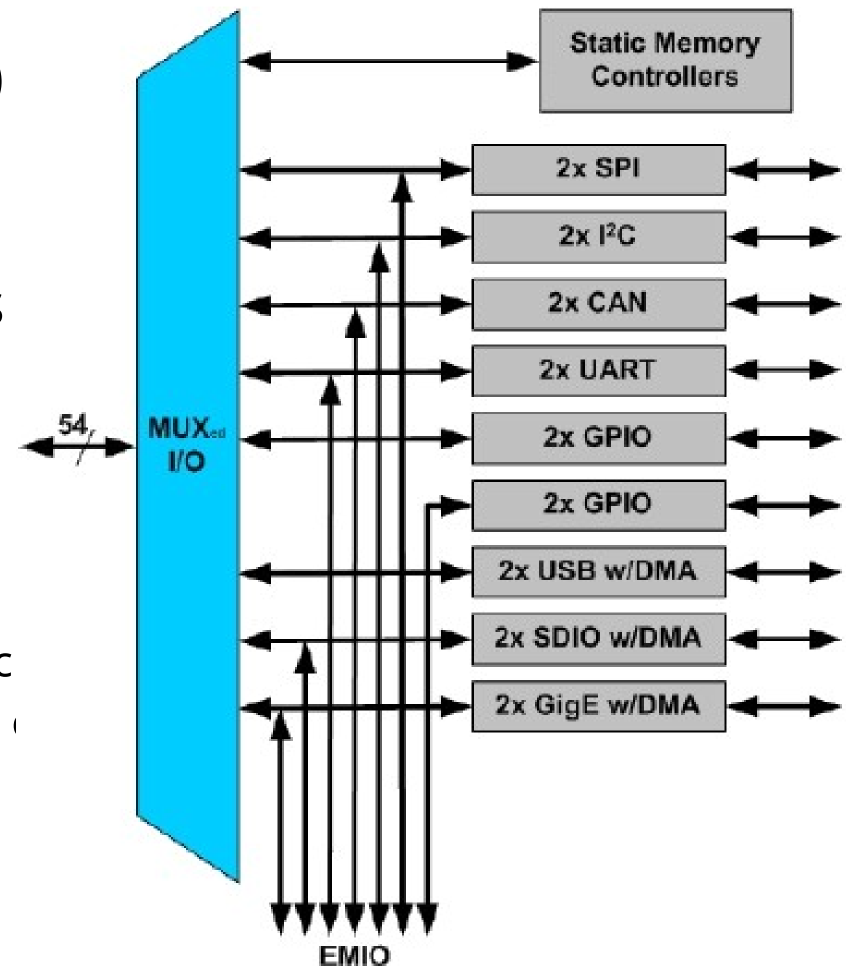
Cache Coherency using SCU

- High-performance, cache-to-cache transfers
- Snoop each CPU and cache each interface independently
- Coherency protocol is MESI
  - M: Cache line has been modified
  - E: Cache line is held exclusively
  - S: Cache line is shared with another CPU
  - I: Cache line is invalidated
- Uses Accelerator Coherence Port (ACP) to allow coherency to be extended to PL



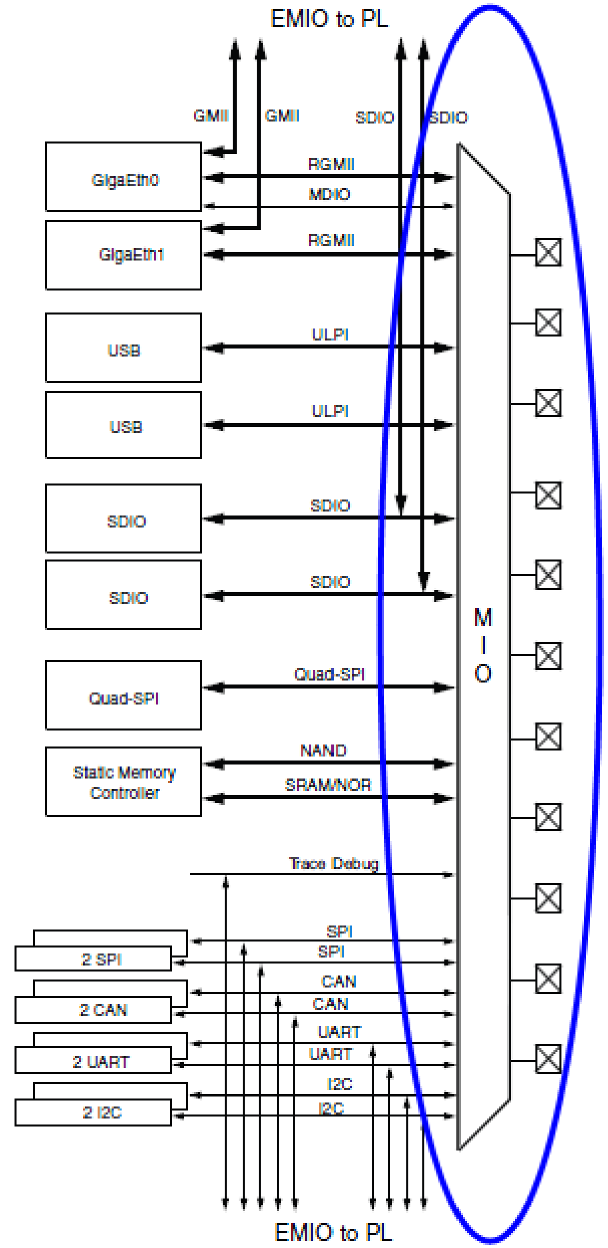
Zynq Architecture Built-in Peripherals

- Two USB 2.0 OTG/Device/Host
- Two Tri- Mode GigE (10/100/1000)
- Two SD/SDIO interfaces
  - Memory, I/O and combo cards
- Two CAN 2.0Bs, SPIs , I2Cs, UARTs
- Four GPIO 32bit Blocks
  - 54 available through MIO
  - other available through EMIO
- Multiplexed Input/Output (MIO)
  - Multiplexed pinout of peripheral and static
  - Two I/O banks; each selectable: 1.8V, 2.5V,
  - Configured using configuration
  - Dedicated pins are used
- Extended MIO
  - Maps PS peripheral ports to the PL
  - Enables use of the SelectIO™ interface with PS peripherals



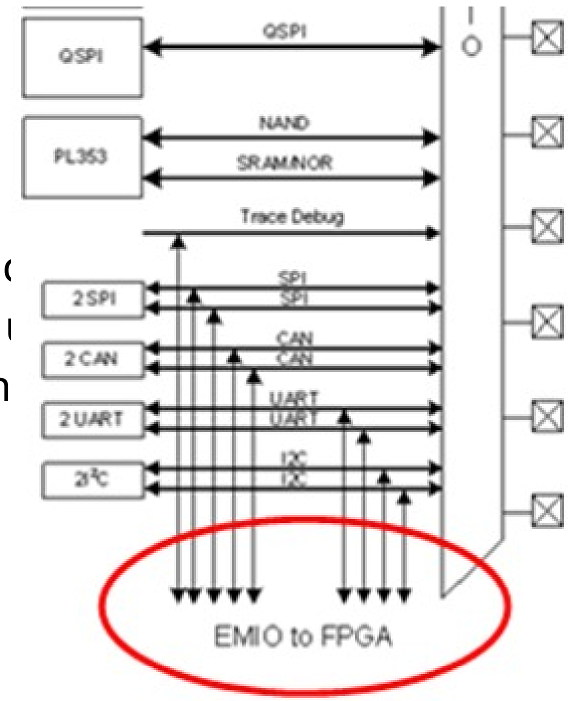
Multiplexed I/O (MIO)

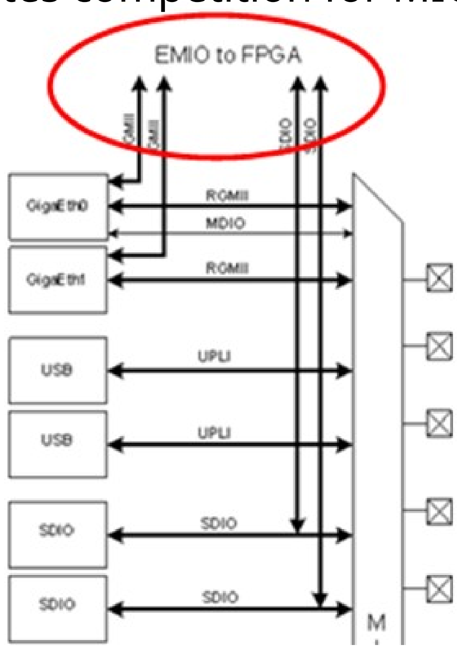
- External interface to PS I/O periphe ports
  - 54 dedicated package pins available
  - Software configurable
    - Automatically added to bootloader by tools
  - Not available for all peripheral ports
    - Some ports can only use EMIO



Extended Multiplexed I/O (EMIO)

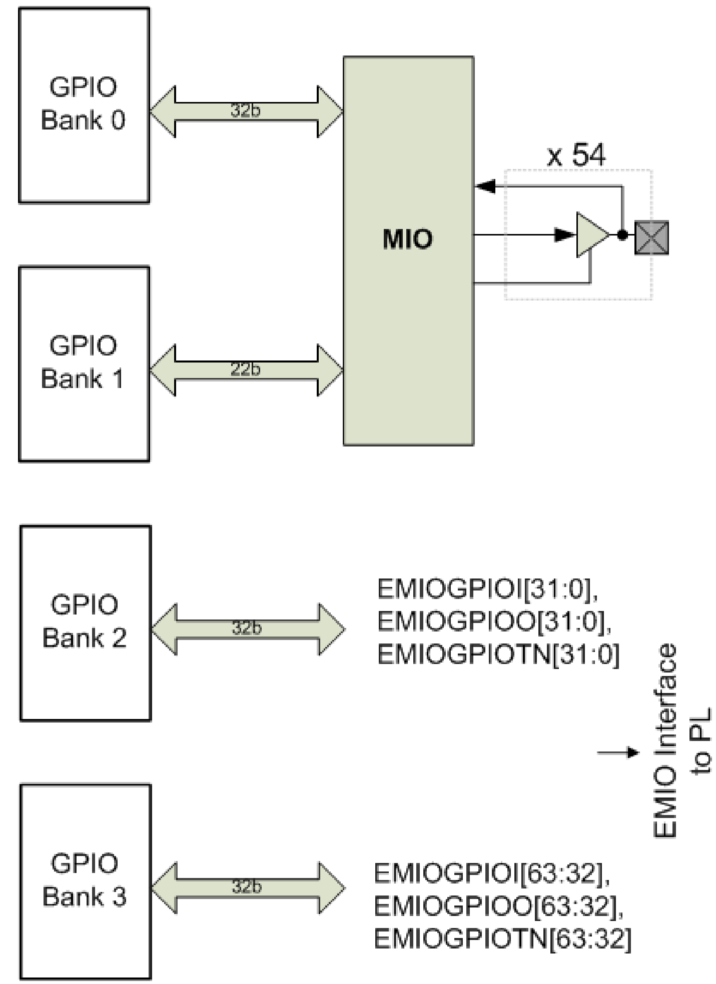
- Extended interface to PS I/O peripheral ports
  - EMIO: Peripheral port to PL
  - Alternative to using MIO
  - Mandatory for some peripheral ports
  - Facilitates
    - Connection to peripheral in programmable logi
    - Use of general I/O pins to supplement MIO pin
    - Allows additional signals for many of the periph
    - Alleviates competition for MIO pin usage



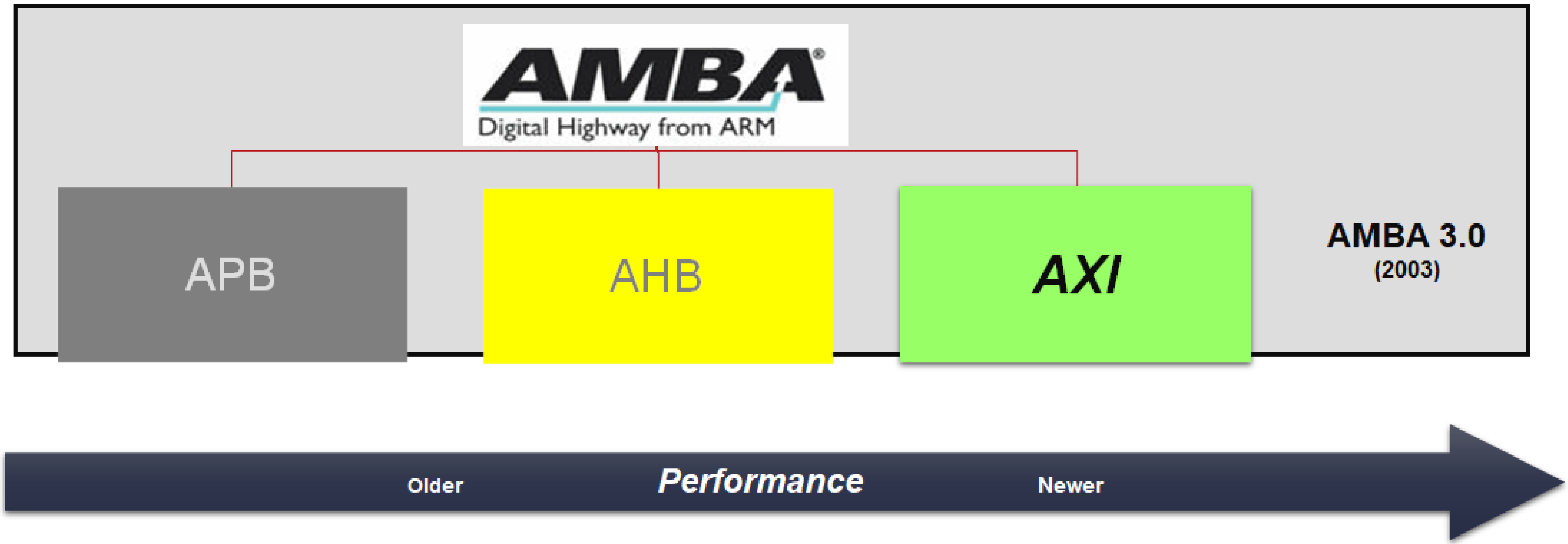


General-Purpose I/O

- GPIO blocks
  - Four separate banks of 32 GPIO bits each
    - Two banks connect to the 54 MIO pins
      - 32 bits and 22 bits, respectively
    - Two banks connect to EMIO (64 bits)
  - Each GPIO bit can be dynamically programmed as input or output
  - Reset values independently configurable for each bit
  - Programmable interrupt generation for each bit
    - One interrupt generated per GPIO bank

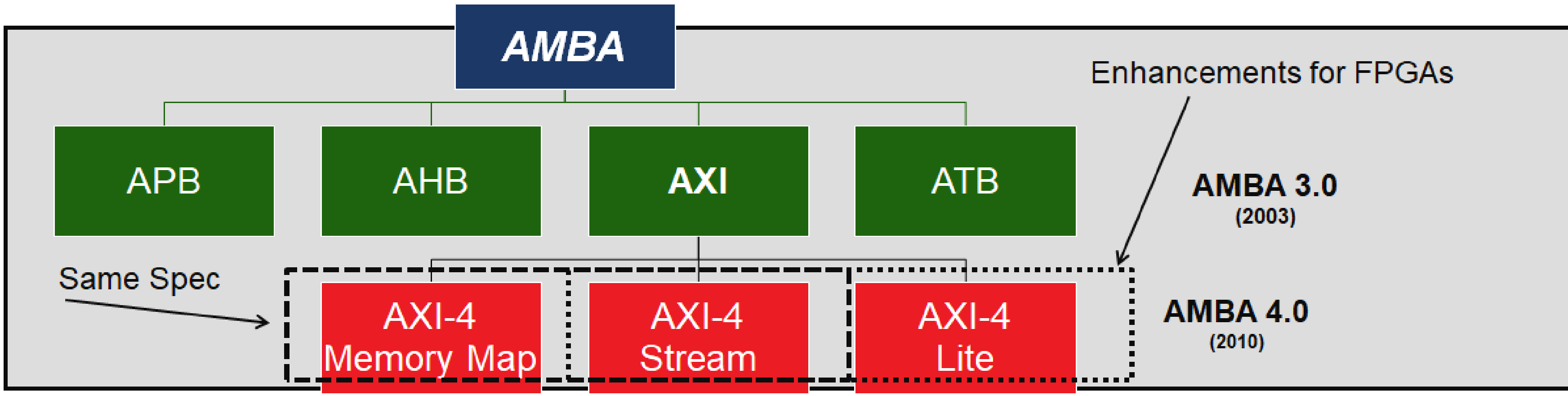


AXI interfaces



AMBA: Advanced Microcontroller Bus Architecture
AXI: Advanced Extensible Interface

AXI is Part of AMBA



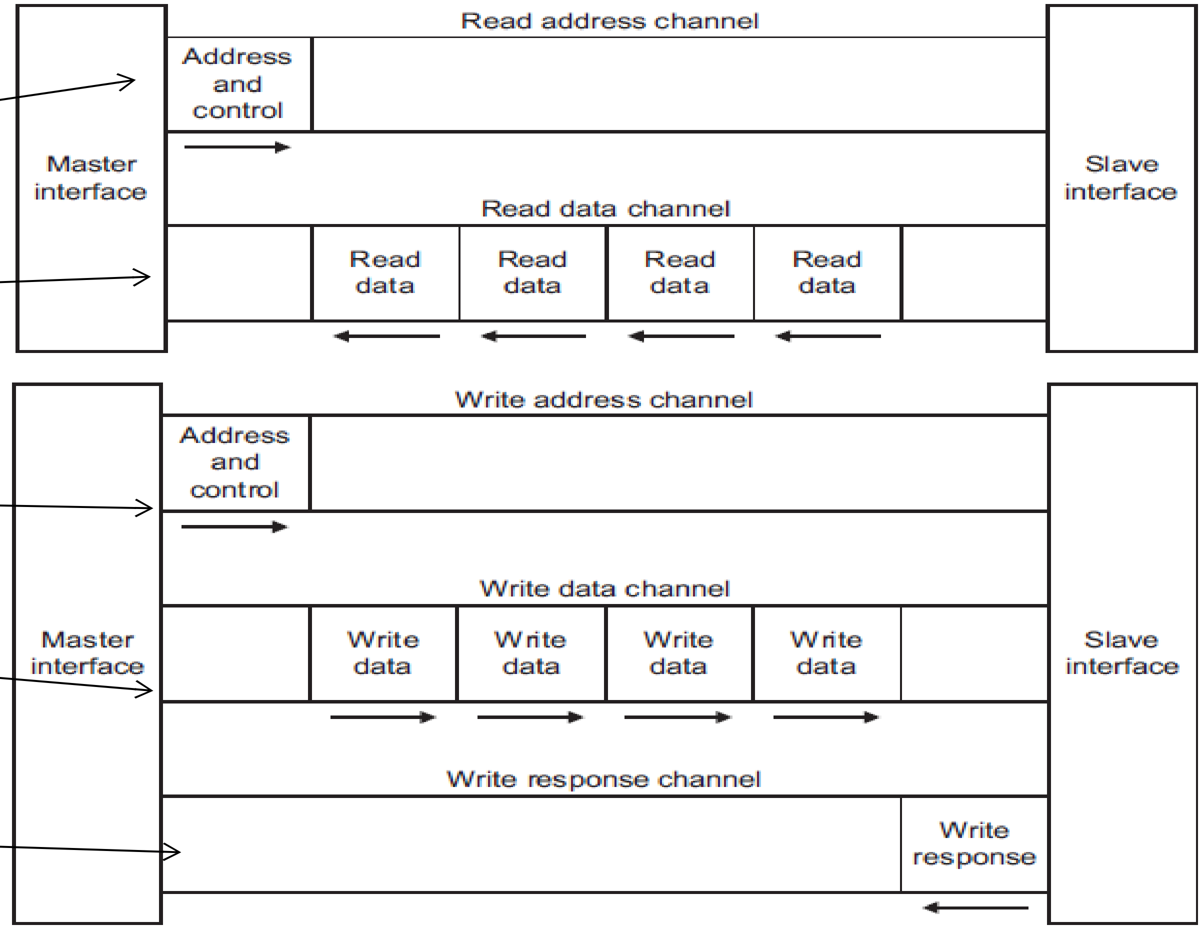
Interface	Features
Memory Map / Full (AXI4)	Traditional Address/Data Burst (single address, multiple data)
Streaming (AXI4-Stream)	Data-Only, Burst
Lite (AXI4-Lite)	Traditional Address/Data—No Burst (single address, single data)

AXI Interconnect

- AXI: interconnect system to tie processors to peripherals
  - **AXI Full** memory map: Full performance bursting interconnect
  - **AXI Lite**: Lower performance non bursting interconnect
    (saves programmable logic resources)
  - **AXI Streaming**: Non-addressed packet based or raw interface

Basic AXI Signaling – 5 Channels

1. Read Address Channe
2. Read Data Channel
3. Write Address Chann
4. Write Data Channel
5. Write Response Char



All AXI Channels Use A Basic "VALID/READY" Handshake

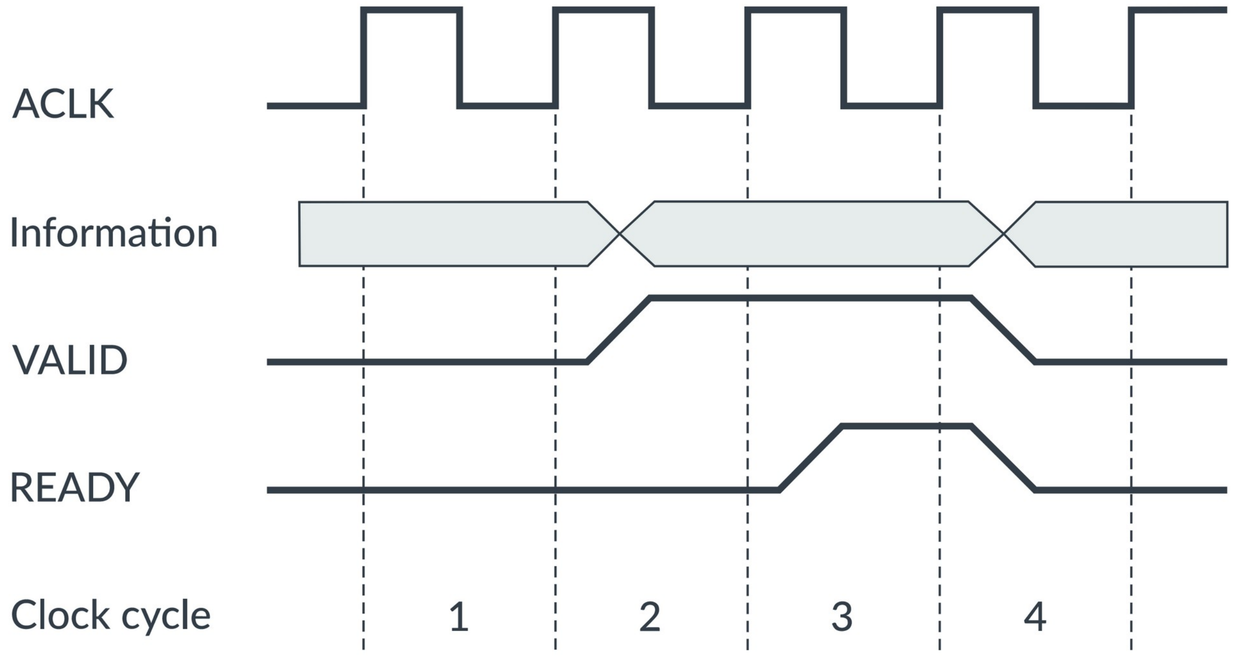
- SOURCE asserts and holds VALID when data is available
  - VALID must remain asserted until destination accepts the DATA
- DESTINATION asserts READY if able to accept data
- Data transferred when VALID and READY = 1
- SOURCE sends next data or deasserts VALID
- DESTINATION deasserts READY if no longer able to accept data
- Mechanism is not an asynchronous handshake, and requires the rising edge of the clock for the handshake to complete
- Read and write handshakes must adhere to the following rules:
  - SOURCE cannot wait for READY to be asserted before asserting VALID. (this is to prevent deadlock because READY *can* depend on VALID for assertion)
  - DESTINATION can wait for VALID to be asserted before asserting READY
  - These rules mean that READY can be asserted before or after VALID, or even at the same time

Differences between transfers and transactions

- Using standard terminology makes understanding the interactions between connected components easier
- AXI makes a distinction between transfers & transactions:
  - **Transfer:** single exchange of information, with one VALID and READY handshake
  - **Transaction:** entire burst of transfers, containing an address transfer, one or more data transfers, and, for write sequences, a response transfer

Channel Transfer Examples (1)

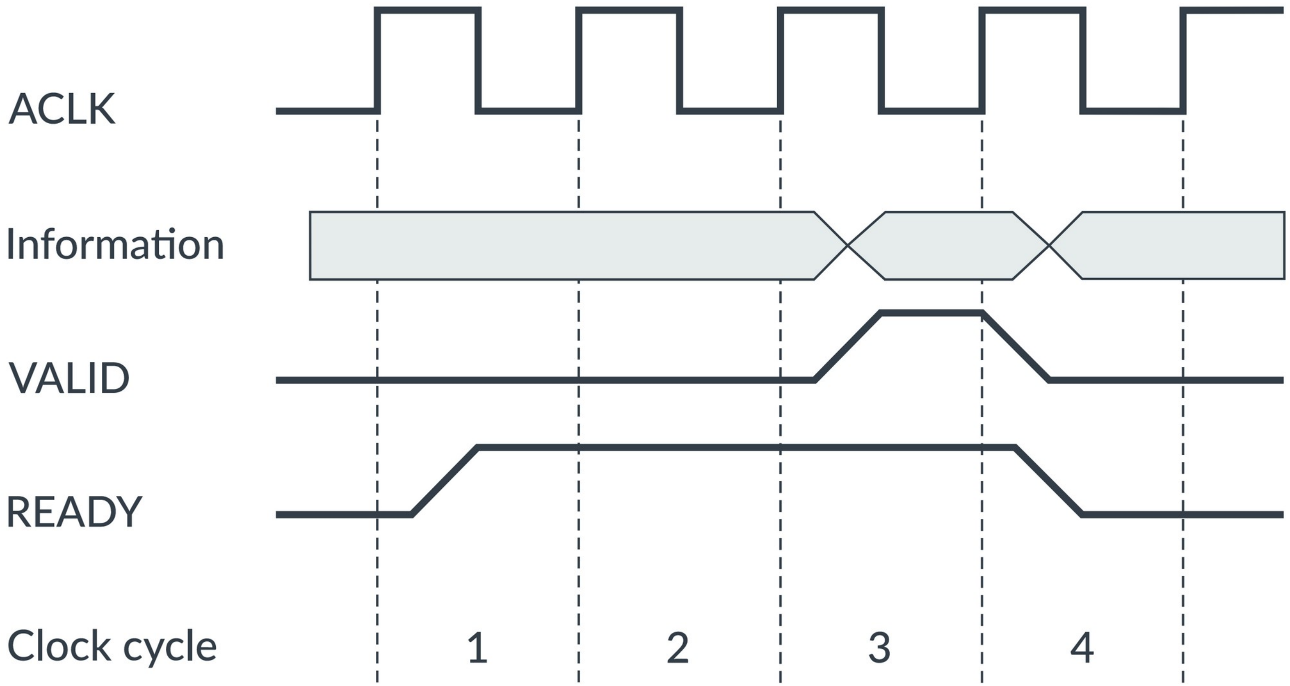
VALID before READY handshake



- In CC 2, VALID signal is asserted, indicating that the data on the information channel is valid
- In CC 3, the following clock cycle, READY signal is asserted
- Handshake completes on rising edge of CC 4, because both READY and VALID are asserted

Channel Transfer Examples (2)

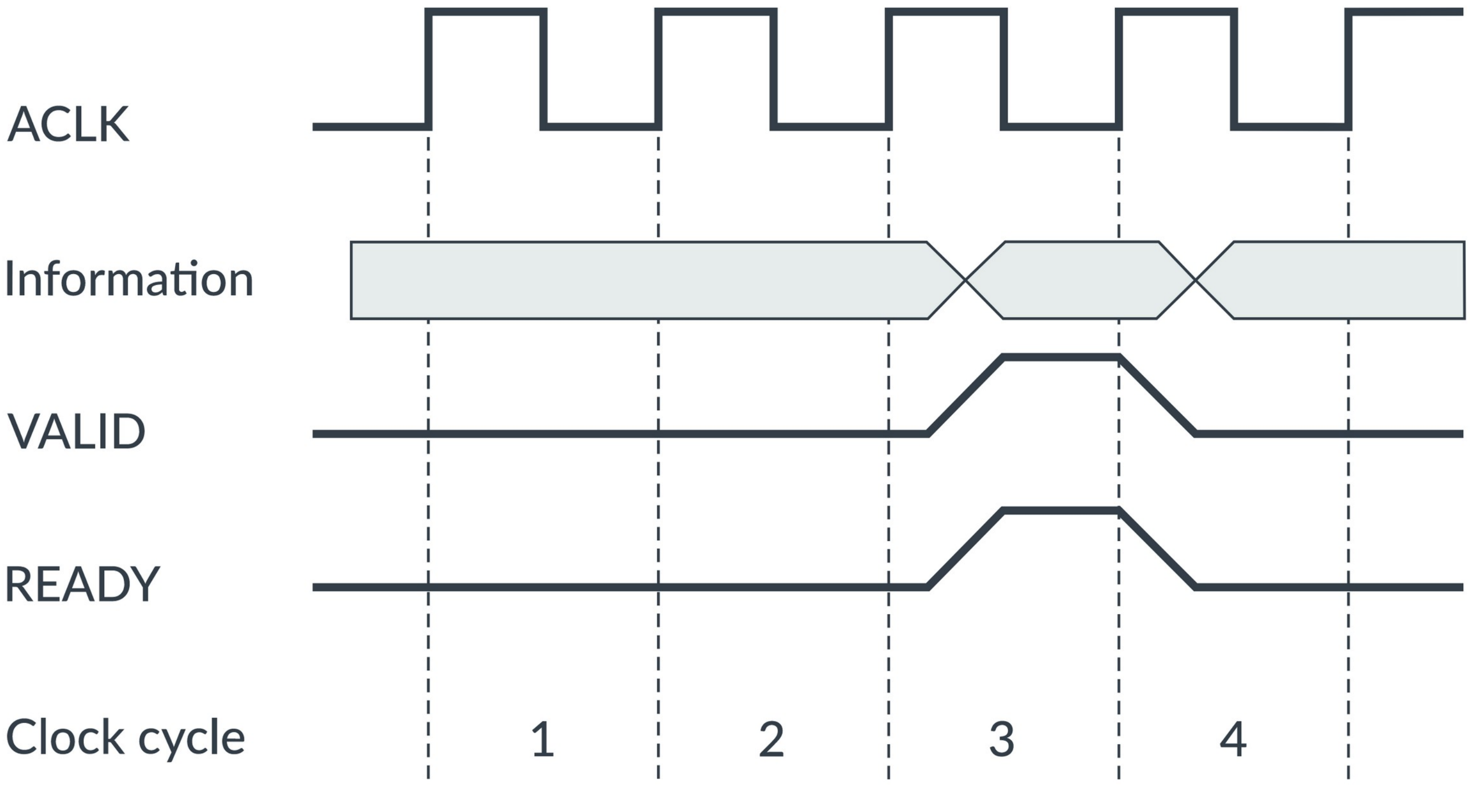
READY before VALID handshake



- In CC 1, READY signal is asserted
- VALID signal is not asserted until clock cycle 3
- Handshake completes on rising edge of CC 4, when both VALID and READY are asserted

Channel Transfer Examples (3)

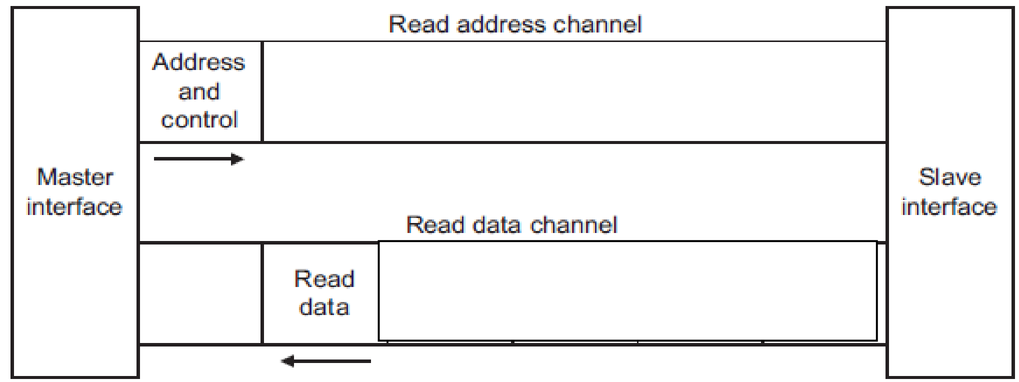
VALID with READY handshake



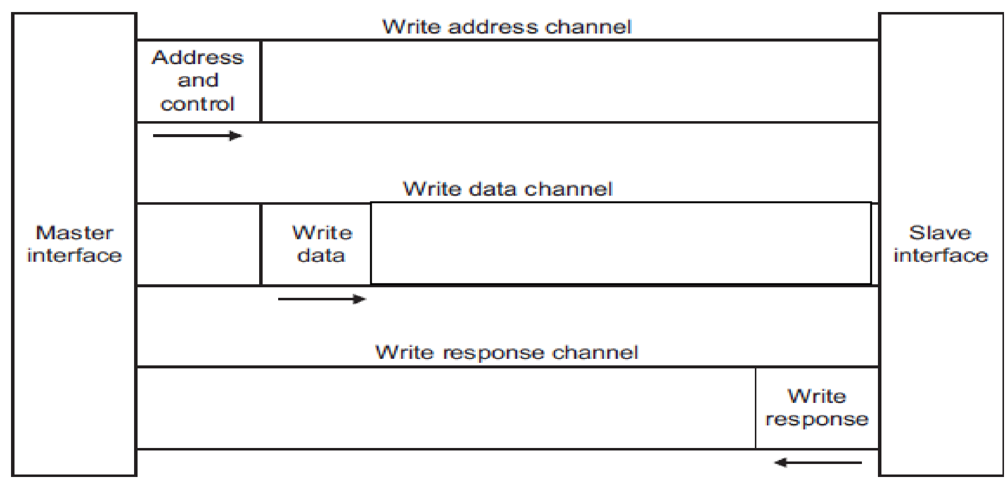
- Both VALID and READY signals being asserted during CC3
- Handshake completes on rising edge of CC4, when both VALID and READY are asserted

The AXI Interface—AXI4Lite

- No burst
- Data width 32 or 64 only
  - Xilinx IP only supports 32-bits
- Very small footprint
- Bridging to AXI4 handled automatically by AXI_Interconnect (if needed)



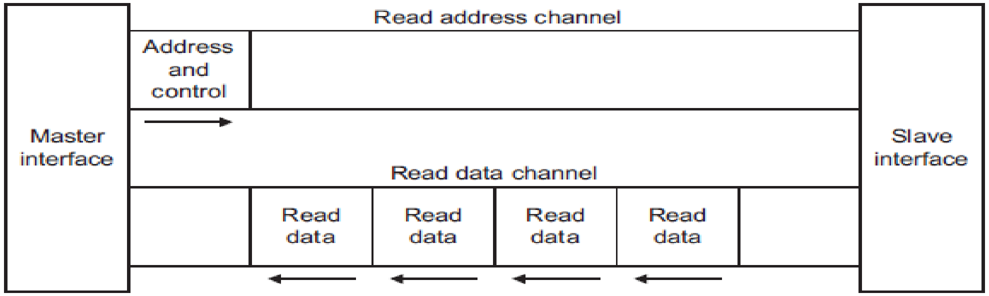
AXI4-Lite Read



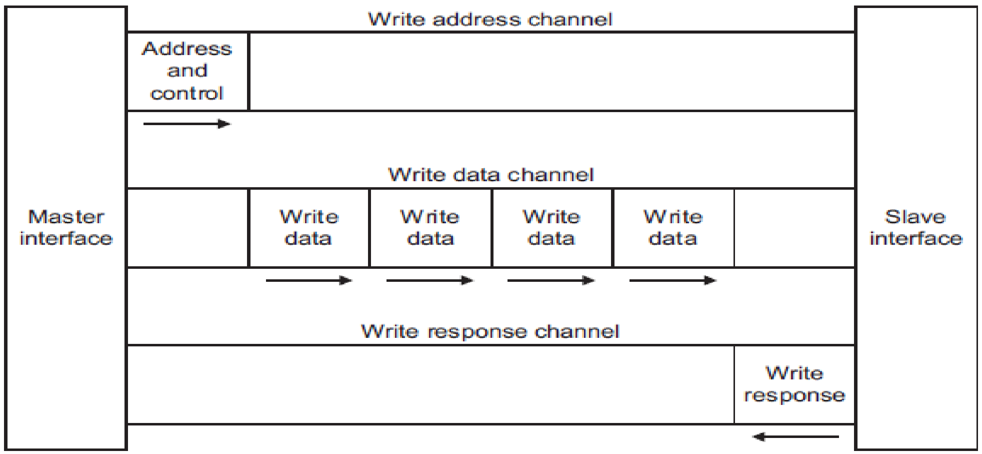
AXI4-Lite Write

The AXI Interface—AXI4

- Sometimes called "Full AXI" or "AXI Memory Mapped"
  - Not ARM-sanctioned names
- Single address multiple data
  - Burst up to 256 data beats
- Data Width parameterizable
  - 1024 bits



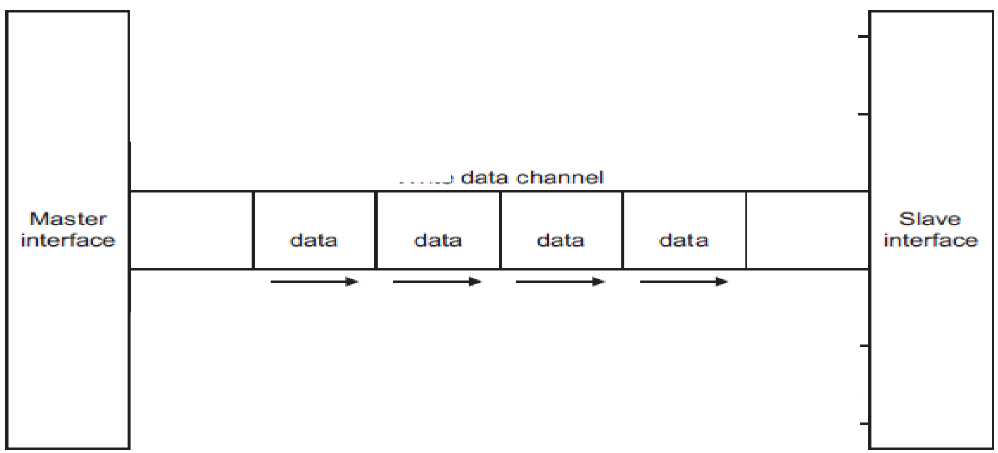
AXI4 Read



AXI4 Write

The AXI Interface—AXI4Stream

- No address channel, no read and write, always just master to slave
  - Effectively an AXI4 “write data” channel
- Unlimited burst length
  - AXI4 max 256
  - AXI4-Lite does not burst
- Virtually same signaling as AXI Data Channels
  - Protocol allows merging, packing, width conversion
  - Supports sparse, continuous, aligned, unaligned streams



AXI4-Stream Transfer

Zynq Device Processing System Configuration