New SoC platforms for AI and Algorithm Acceleration

(Introduction to Xilinx Versal ACAP)

Gustavo Sutter

gustavo.sutter@uam.es

These talk in context of the “Joint ICTP, IAEA School on FPGA-based SoC and its application to Nuclear and Scientific Applications” at:

The Abdus Salam
International Centre
for Theoretical Physics

Authorized
Training Provider

ATP for Spain – www.electratraining.org

Gustavo Sutter, PhD

Professor at Universidad Autonoma de Madrid

More than 20-year experience in FPGA research, project development, teaching and consultancy

A General Overview

- These talk in context of the “Joint ICTP, IAEA School on FPGA-based SoC and its application to Nuclear and Scientific Applications”

Day 1	14-Nov
Logic design and finite state machines	P. Bazargan
FPGA and System-on-Chip (SoC) technology	C. Sisterna
Digital numbers representation: fixed point vs floating point	P. Bazargan
SoC architecture and design methodology	C. Sisterna
SoC architecture and design methodology (cont.)	C. Sisterna

Day 2	15-Nov
Digital Arithmetic	P. Bazargan
Zynq-7000 SoC: Evaluation and Development Board	C. Sisterna
C for Embedded Systems	C. Sisterna
C for Embedded Systems (cont.)	C. Sisterna
Hardware and Software Interrupts on SoC	C. Sisterna
AXI Bus and Custom IP	C. Sisterna
Connectivity testing for lab activities	L. Garcia

Day 3	16-Nov
Floating point operations: square root and division	P. Bazargan
Programmable hardware acceleration in Communication Networks	L. Valcarenghi
Quantization in Neural Networks: Advantages and Limitations	E. Paolini
Fundamentals of VHDL (Hardware Description Language)	C. Sisterna
VHDL for Synthesis	C. Sisterna
Introduction to Lab Activities	M.L. Crespo, L. Garcia
Lab 1: Hello World	Lab Tutors

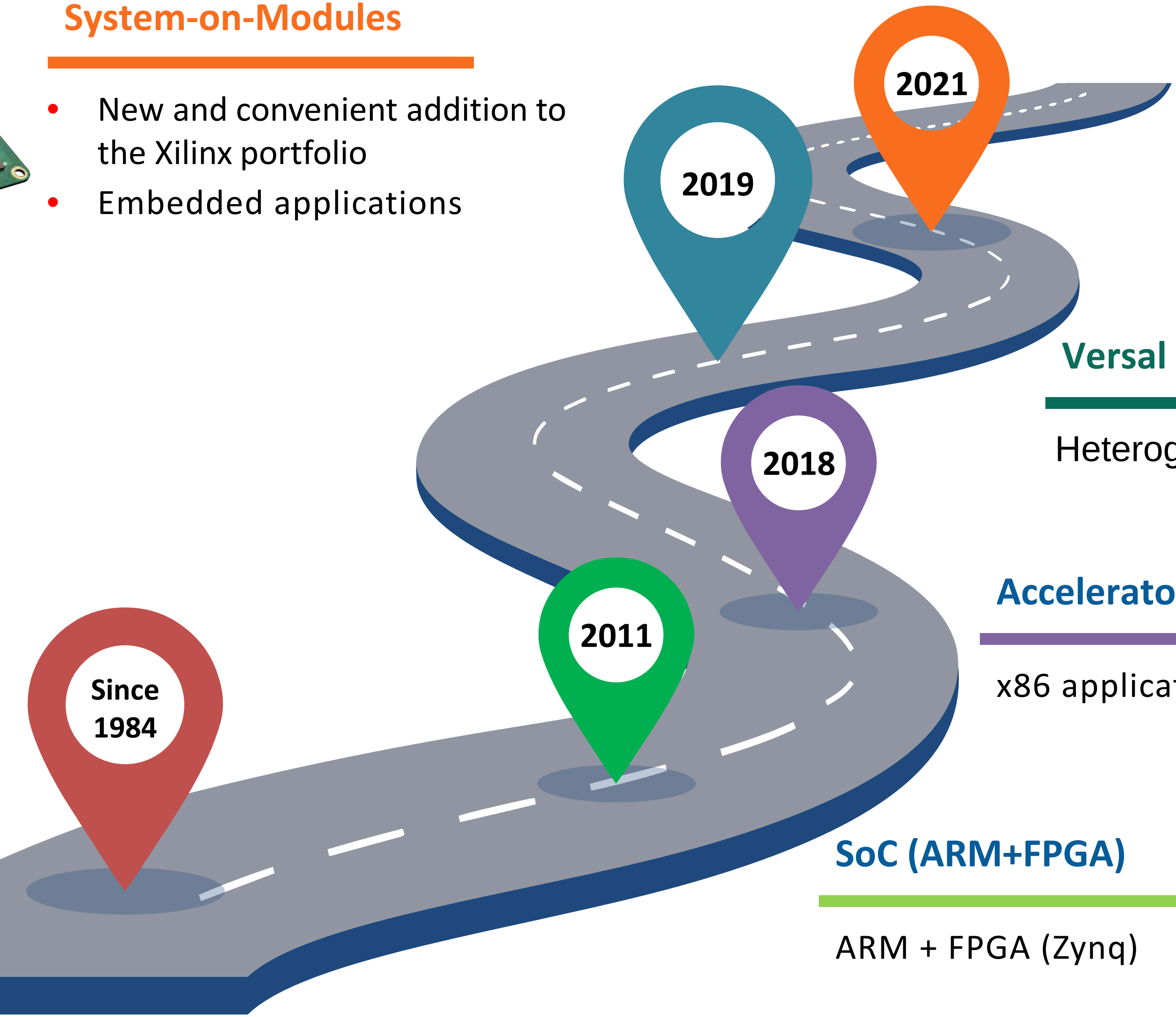
Day 4	17-Nov
The role of programmable hardware acceleration in the edge-cloud continuum	L. Valcarenghi
New SoC platforms for AI and Algorithm Acceleration	G. Sutter
High Level Synthesis (HLS)	F. Rincon
Guide for Lab 2: GPIO In/Out	M. Ballina
Lab 2: GPIO In/Out	Lab Tutors

Day 5	18-Nov
New SoC platforms for AI and Algorithm Acceleration (cont.)	G. Sutter
HLS Tutorial	F. Rincon
VHDL for Simulation (Testbench)	C. Sisterna
Lab 3: Custom IP and VHDL	Lab Tutors

AMD-Xilinx Portfolio Evolution

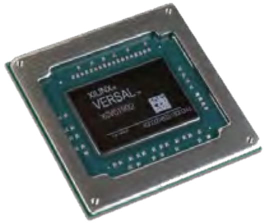
System-on-Modules

- New and convenient addition to the Xilinx portfolio
- Embedded applications



Devices

FPGAs
(+60 families)



Versal ACAP

Heterogeneous computing

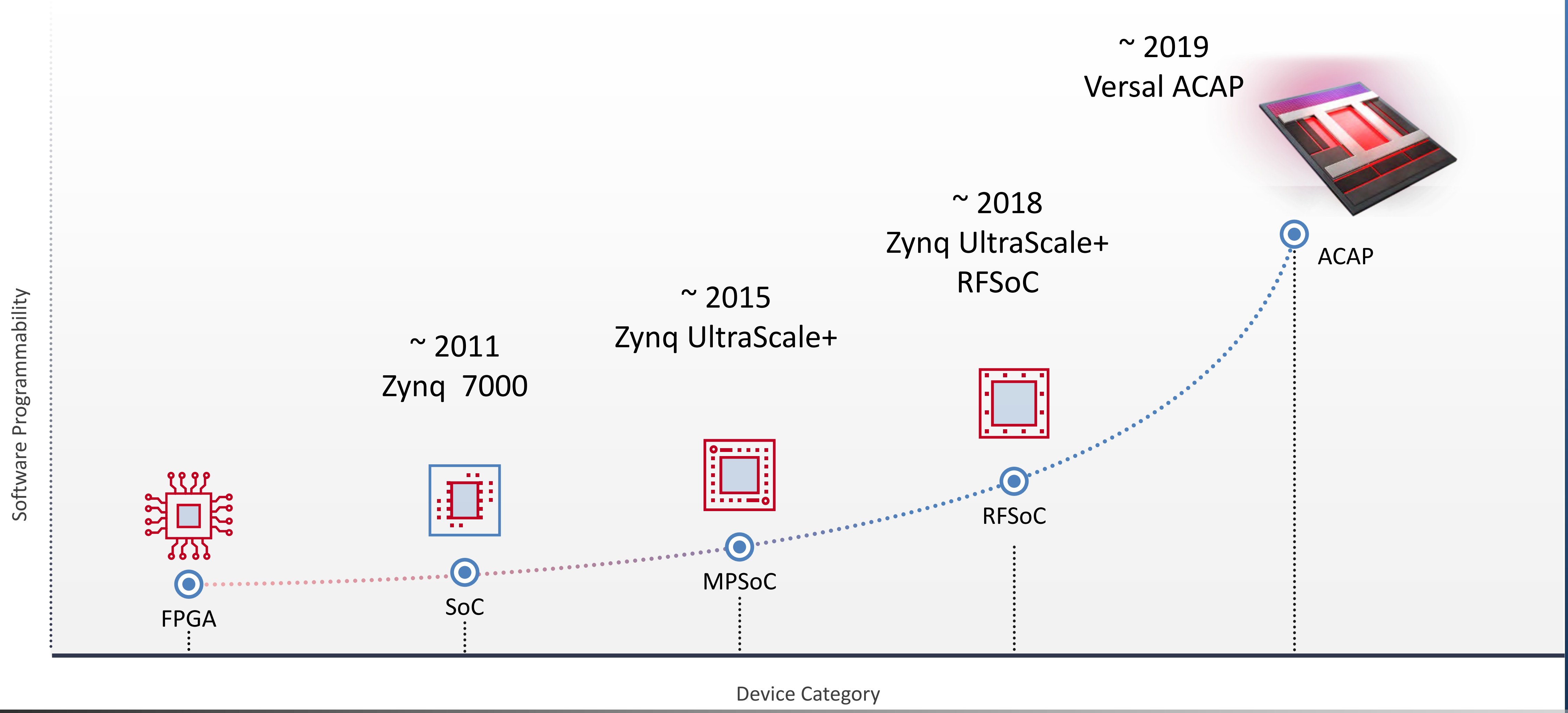
Accelerator Cards

x86 applications

SoC (ARM+FPGA)

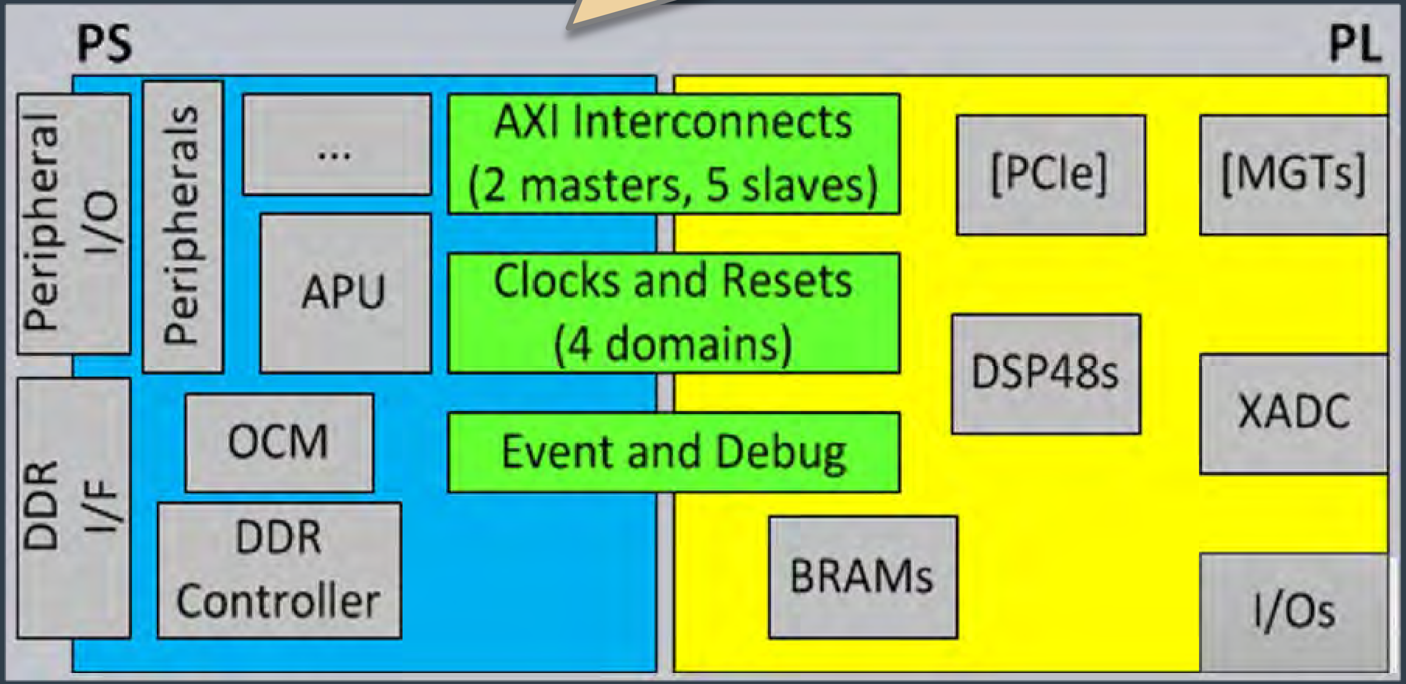
ARM + FPGA (Zynq)

FPGA – SoC – ACAP evolution



Zynq SoC PS and PL

SoC "System-on-Chip"

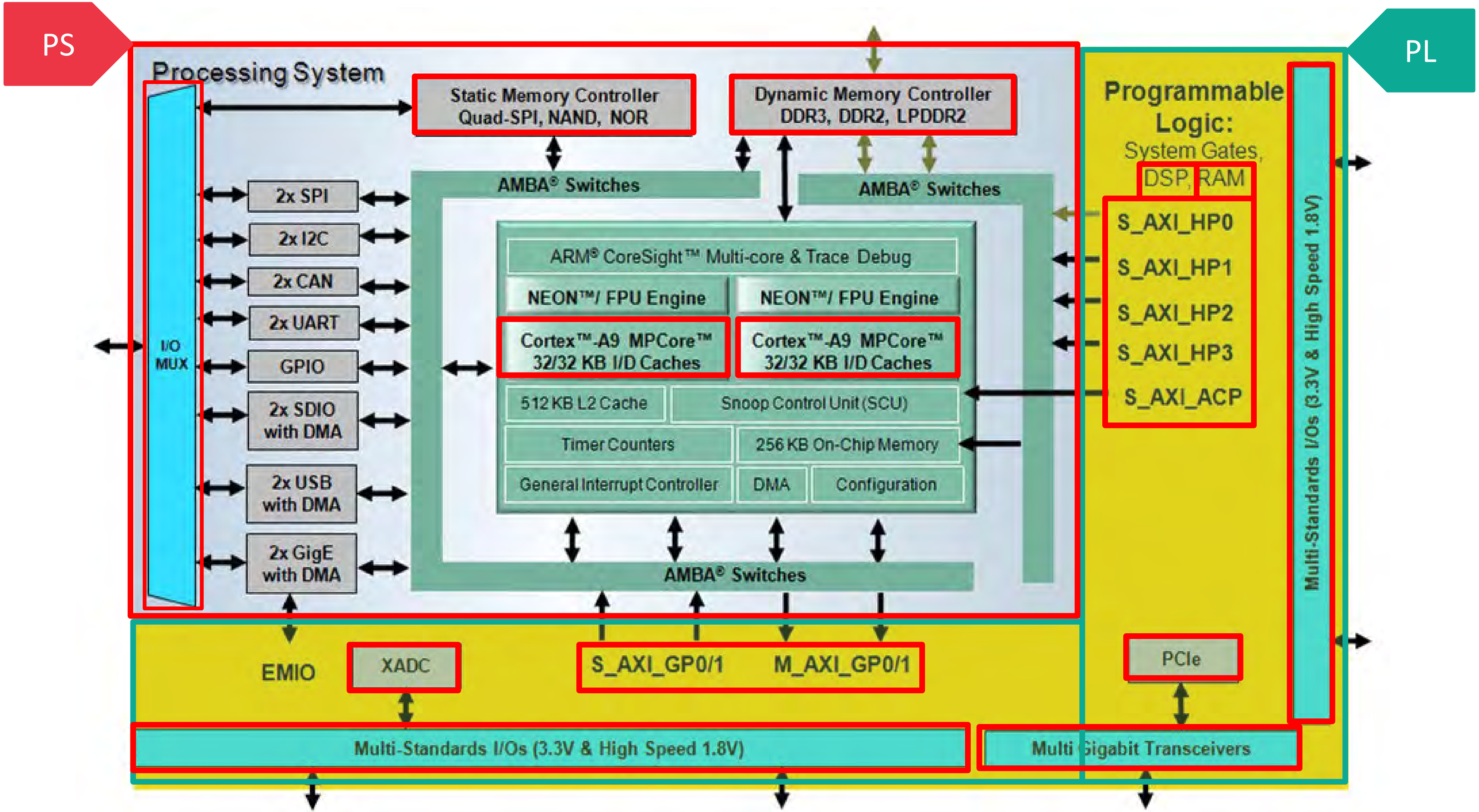


SoCs combine processors, peripherals, and programmable logic

Zynq-7000, MPSoC, RFSoC, Versal® families

Offer ease of software programming and programming techniques to construct hardware

Zynq-7000 SoC Block Diagram



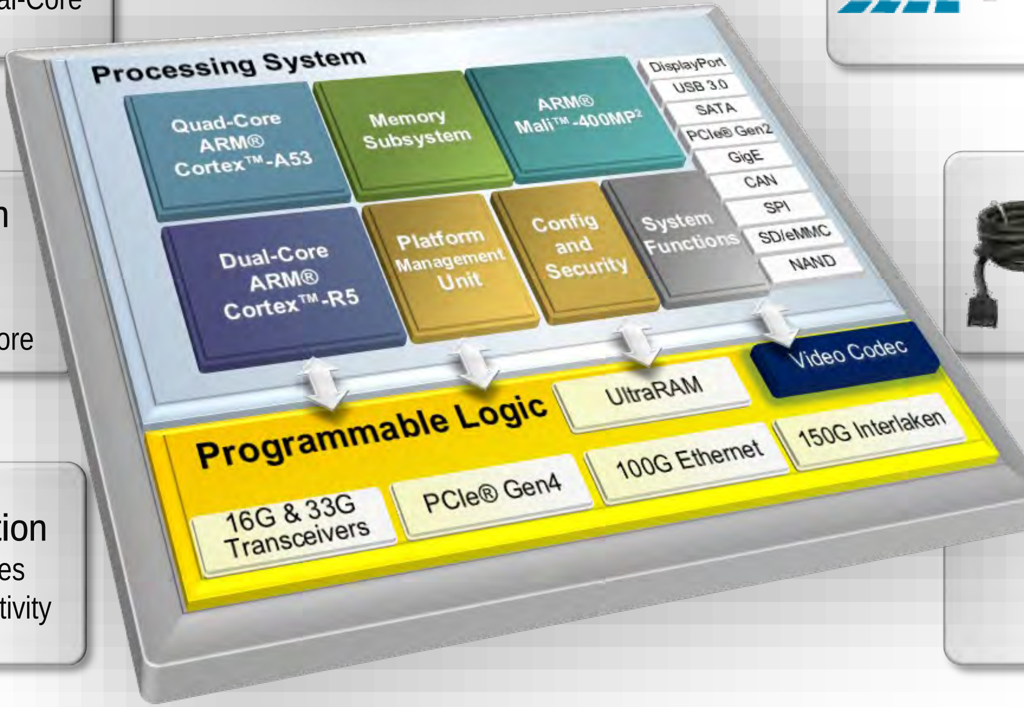
The First Multiprocessing SoC (MPSoC)

Real-Time Processors
32-bit Dual-Core

Memory Subsystem
High Bandwidth
Low Latency

Graphics Processor
ARM Mali-400MP2

Application Processor
64-bit
Dual/Quad-Core



Processing System

Quad-Core ARM® Cortex™-A53

Memory Subsystem

ARM® Mali™-400MP²

DisplayPort
USB 3.0
SATA
PCIe® Gen2
GigE
CAN
SPI
SD/eMMC
NAND

Dual-Core ARM® Cortex™-R5

Platform Management Unit

Config and Security

System Functions

Video Codec

UltraRAM

Programmable Logic

16G & 33G Transceivers

PCIe® Gen4

100G Ethernet

150G Interlaken

High Speed Peripherals
Key Interfaces

Fabric Acceleration
Customizable Engines
High Speed Connectivity

Video Codec
8K4K (15fps)
4K2K (60fps)

Platform & Power Management
Granular Power Control
Functional Safety

Configuration & Security Unit
Anti-Tamper & Trust
Industry Standards

Zynq UltraScale+ MPSoC Block Diagram

Application Processing Unit (APU)

Quad-core or Dual-core ARM Cortex-A53 Processors

ARMv8-A architecture, 64-bit or 32-bit operation (Up to 1.5GHz)

Real-Time Processing Unit (RPU)

Dual ARM Cortex-R5 Processors

ARMv7-R Architecture, 32-bit operation (Up to 600MHz)

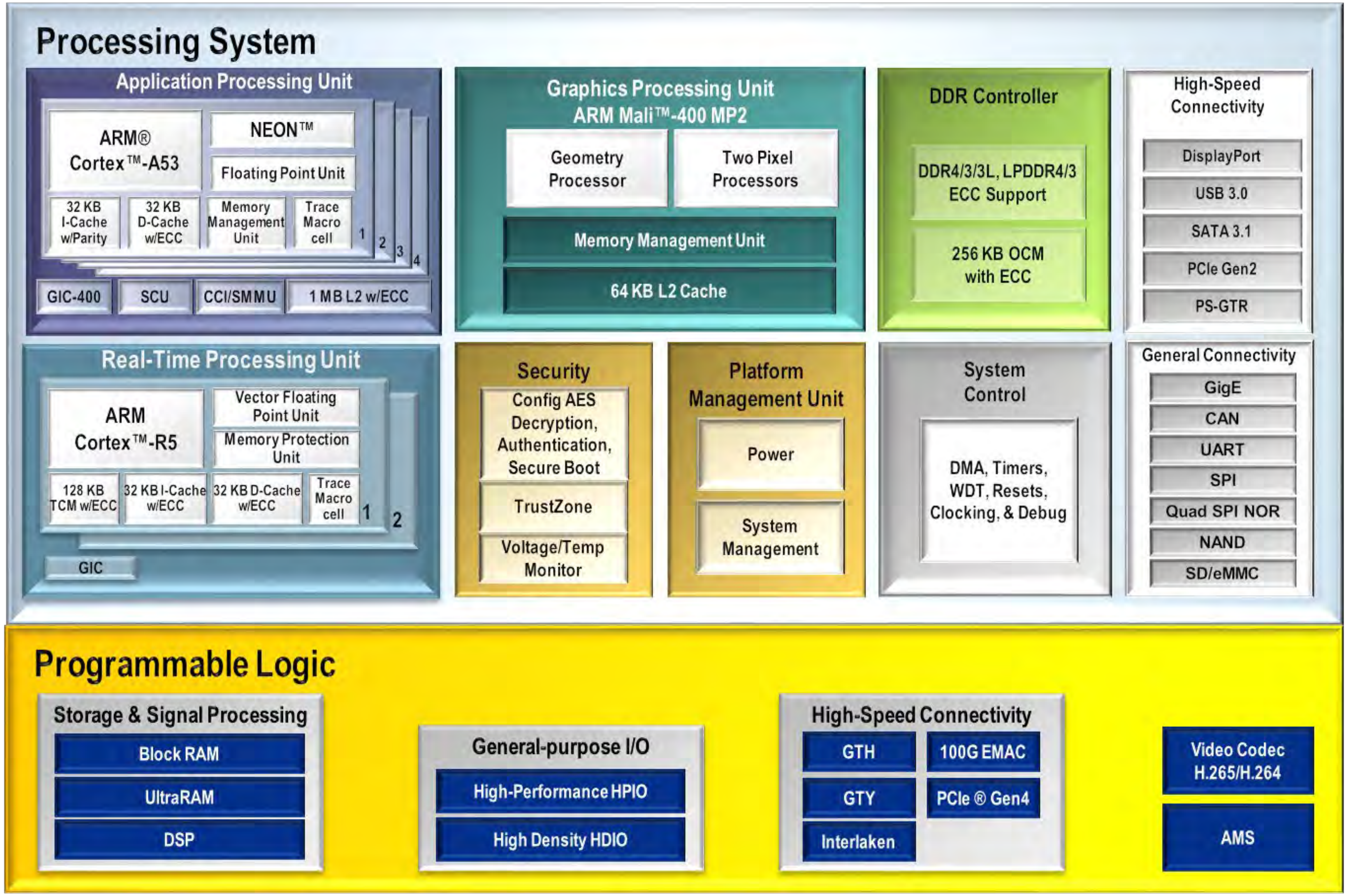
Graphics and Video Processing

Graphic Processing Unit (GPU). ARM Mali-400 MP2

Video Encoder/Decoder (VCU). H.265, H.264

Granular Power Management

Platform Management Unit (PMU)



Processing System

Application Processing Unit

ARM® Cortex™-A53

NEON™

Floating Point Unit

32 KB I-Cache w/Parity

32 KB D-Cache w/ECC

Memory Management Unit

Trace Macro cell

1 2 3 4

GIC-400

SCU

CCI/SMMU

1 MB L2 w/ECC

Graphics Processing Unit ARM Mali™-400 MP2

Geometry Processor

Two Pixel Processors

Memory Management Unit

64 KB L2 Cache

DDR Controller

DDR4/3/3L, LPDDR4/3 ECC Support

256 KB OCM with ECC

High-Speed Connectivity

DisplayPort

USB 3.0

SATA 3.1

PCIe Gen2

PS-GTR

Real-Time Processing Unit

ARM Cortex™-R5

Vector Floating Point Unit

Memory Protection Unit

128 KB TCM w/ECC

32 KB I-Cache w/ECC

32 KB D-Cache w/ECC

Trace Macro cell

1 2

GIC

Security

Config AES Decryption, Authentication, Secure Boot

TrustZone

Voltage/Temp Monitor

Platform Management Unit

Power

System Management

System Control

DMA, Timers, WDT, Resets, Clocking, & Debug

General Connectivity

GigE

CAN

UART

SPI

Quad SPI NOR

NAND

SD/eMMC

Programmable Logic

Storage & Signal Processing

Block RAM

UltraRAM

DSP

General-purpose I/O

High-Performance HPIO

High Density HDIO

High-Speed Connectivity

GTH

100G EMAC

GTY

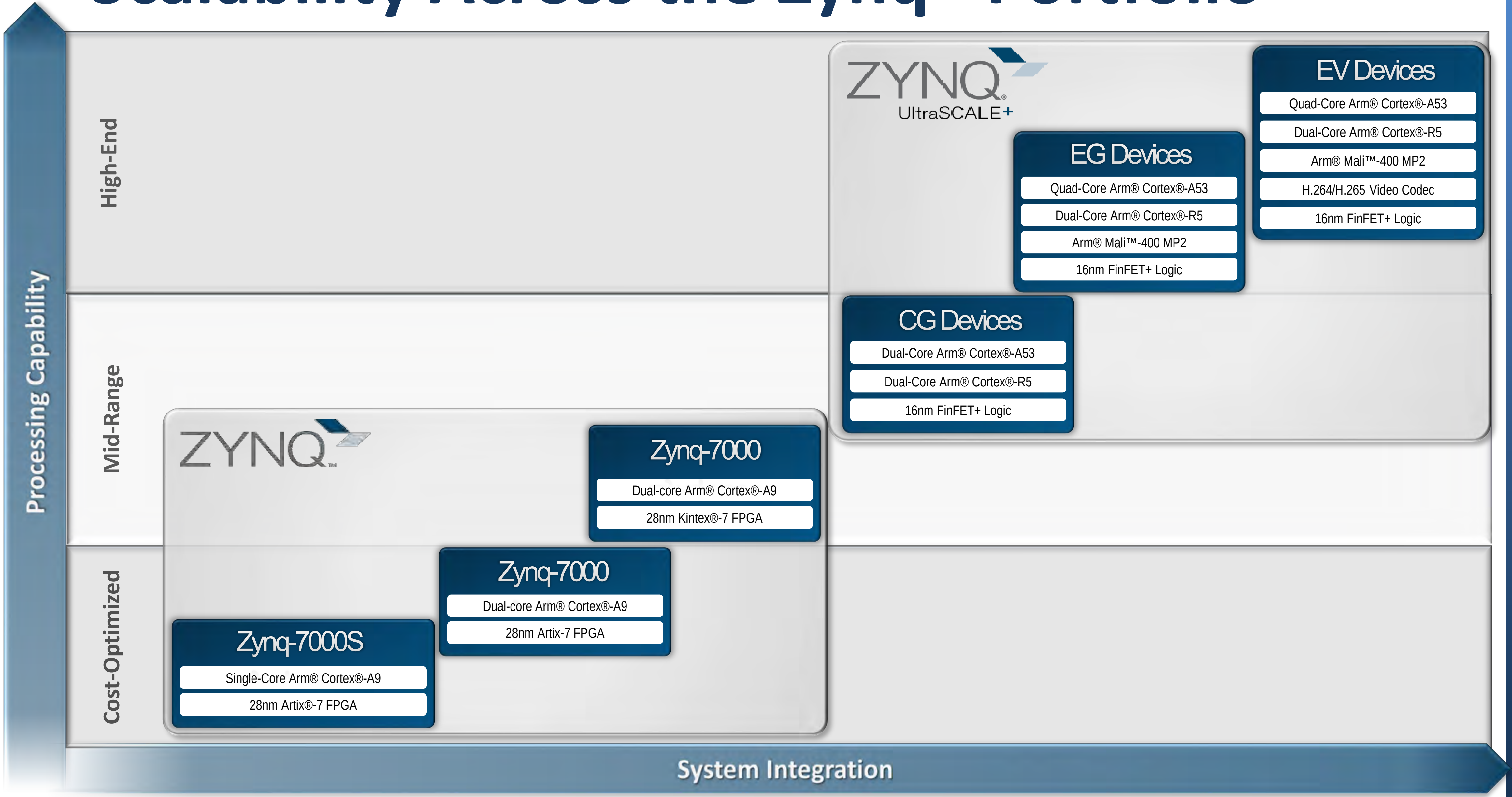
PCIe ® Gen4

Interlaken

Video Codec H.265/H.264

AMS

Scalability Across the Zynq® Portfolio



Processing Capability

System Integration

High-End

Mid-Range

Cost-Optimized

ZYNQ

Zynq-7000S
- Single-Core Arm® Cortex®-A9
- 28nm Artix®-7 FPGA

Zynq-7000
- Dual-core Arm® Cortex®-A9
- 28nm Artix-7 FPGA

Zynq-7000
- Dual-core Arm® Cortex®-A9
- 28nm Kintex®-7 FPGA

ZYNQ UltraSCALE+

CG Devices
- Dual-Core Arm® Cortex®-A53
- Dual-Core Arm® Cortex®-R5
- 16nm FinFET+ Logic

EG Devices
- Quad-Core Arm® Cortex®-A53
- Dual-Core Arm® Cortex®-R5
- Arm® Mali™-400 MP2
- 16nm FinFET+ Logic

EV Devices
- Quad-Core Arm® Cortex®-A53
- Dual-Core Arm® Cortex®-R5
- Arm® Mali™-400 MP2
- H.264/H.265 Video Codec
- 16nm FinFET+ Logic

ZYNQ® RFSoC

Monolithically Integrated RF-Analog on a Production Proven MPSoC Architecture

Processing System

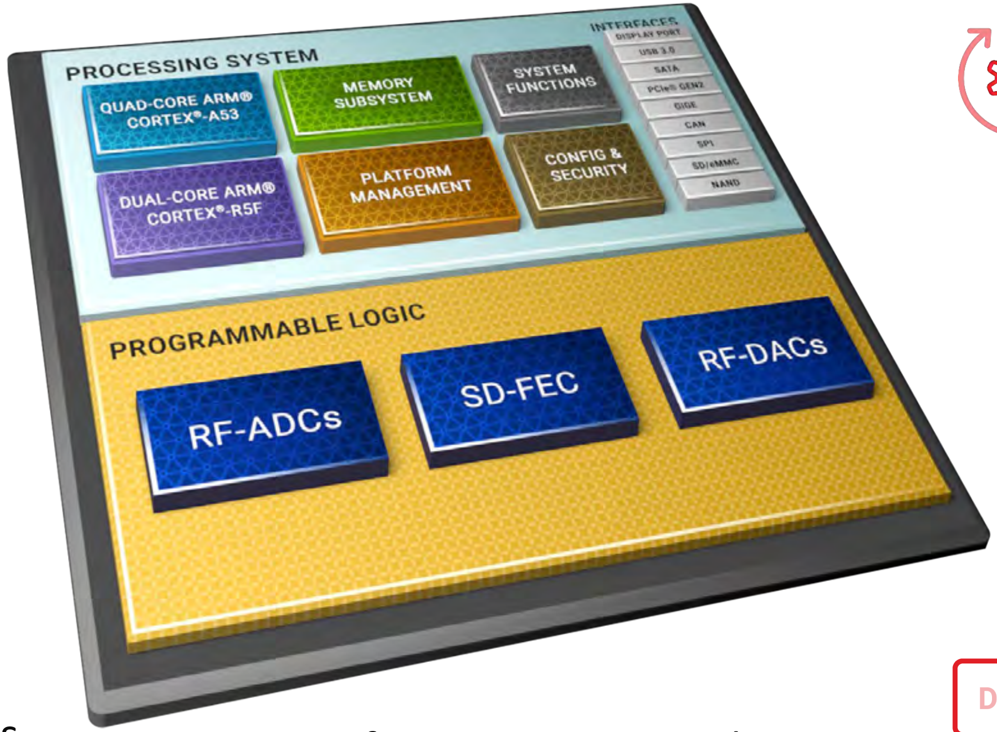
- Quad-Core A53 (64-bit)
- Dual-Core R5F (32-bit)

Hardened Engines

- PCIe® Gen3 & Gen4
- 100G Cores

Programmable Logic

- 16nm FinFET
- UltraScale+™ FPGA Fabric



33G Transceivers

- 33Gb/s
- 28G Backplane Capable

DSP-Intensive

- 4,272 DSP Slices
- 7,612 GMACs

Integrated Direct-RF Analog-to-Digital Converters

Soft Decision Forward Error Correction

- LDPC & Turbo Support

Integrated Direct-RF Digital-to-Analog Converters

Portfolio for Current RFSoC

Performance

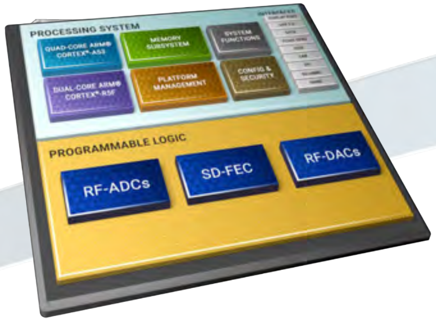
Zynq RFSoC Gen 1

4GHz
of Analog Bandwidth

Zynq RFSoC Gen 2

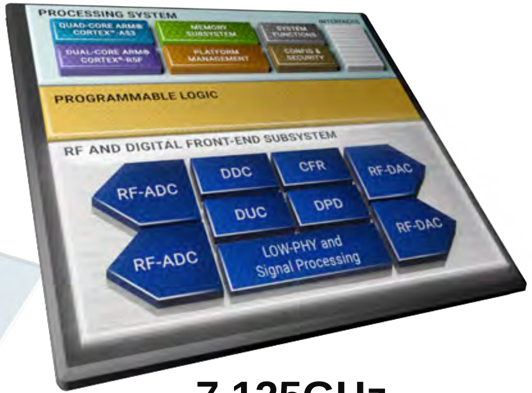
5GHz
of Analog Bandwidth

Zynq RFSoC Gen 3



6GHz
of Analog Bandwidth

Zynq RFSoC DFE



7.125GHz
of Analog Bandwidth
w/Hardened DFE Subsystem

ASIC Performance with Adaptable Logic

- ✓ More processing per channel
- ✓ Greater instantaneous bandwidth
- ✓ More integration
- ✓ ½ the DC power for a given use case
- ✓ Cost-effective for high volume

2018 2019 2020 2021

XILINX

Versal ACAP Introduction

Gustavo Sutter

Presentation for the “Joint ICTP, IAEA School on FPGA-based SoC and its application to Nuclear and Scientific Applications””

Agenda

- Motivation for new SoC/Computing Architectures
- ACAP (Adaptive Compute Acceleration Platform) a first view
- Xilinx Versal Architecture
  - Scalar Engine (Arm Cortex A72 and R5),
  - Adaptable Engines (Programable Logic),
  - Intelligent Engines (DSP and AI Engines),
  - NoC (Network on Chip)
- Design Flow Overview
  - Traditional and platform based flow

Mountains of Unstructured Data

One Architecture Can't Do It Alone

This is the Era of Heterogeneous Compute

Today's Developer Needs

Software programmability

Performance for a diverse range of applications

Adaptability to keep pace with rapid innovation

Today’s Solutions

CPUs

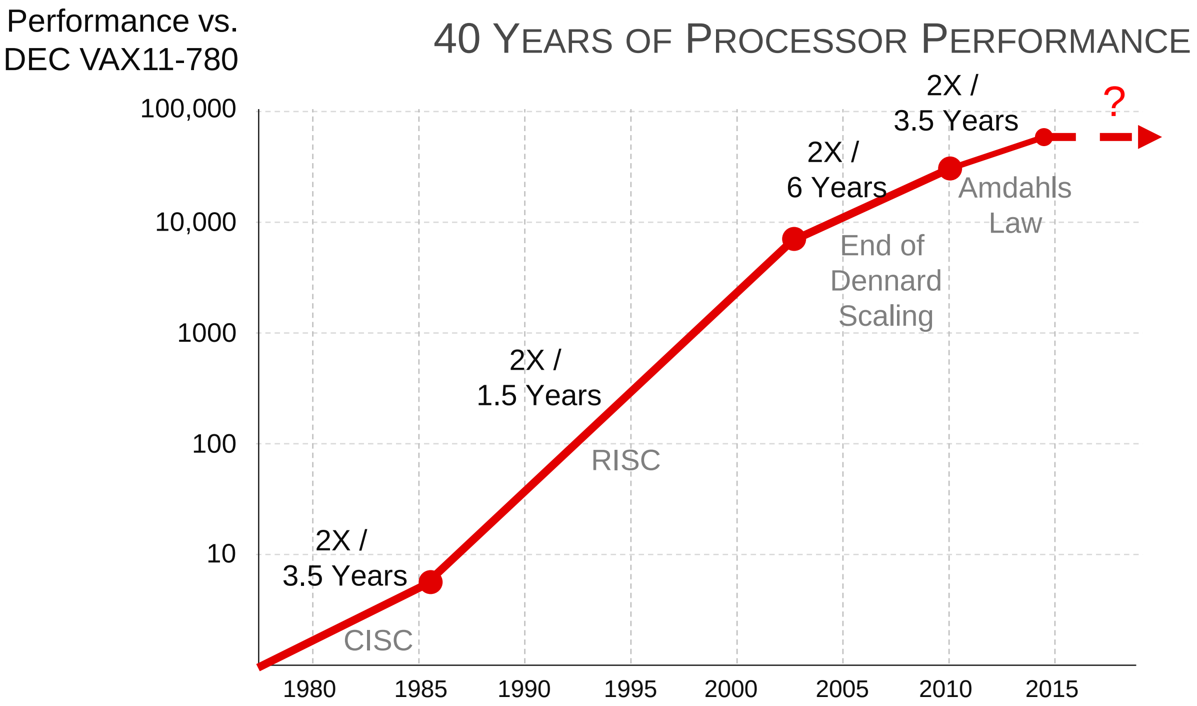
Fixed Function Accelerators

ASICs/ASSPs/GPUs

FPGAs

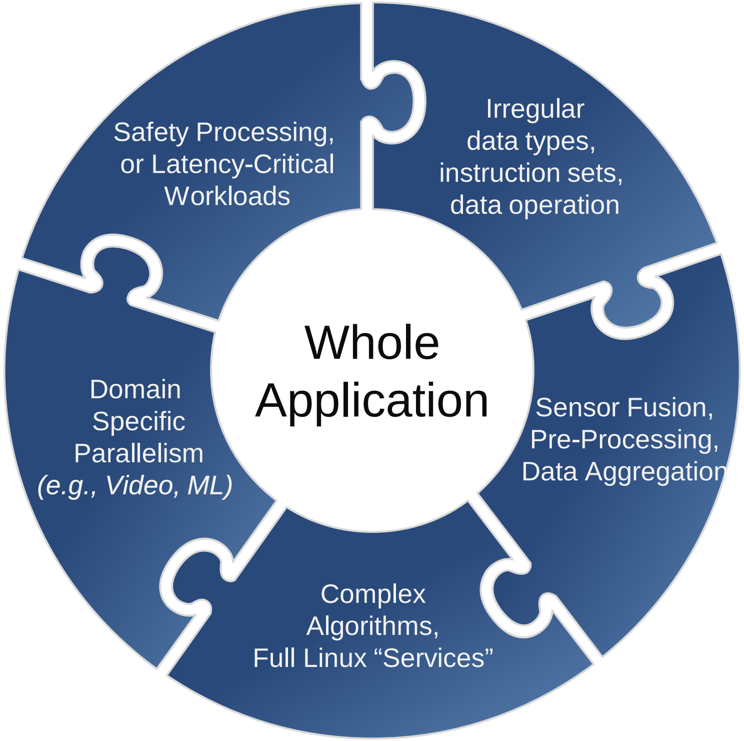
The Technology Conundrum .. And the Need for a New Compute Paradigm

Processing Architectures are Not Scaling

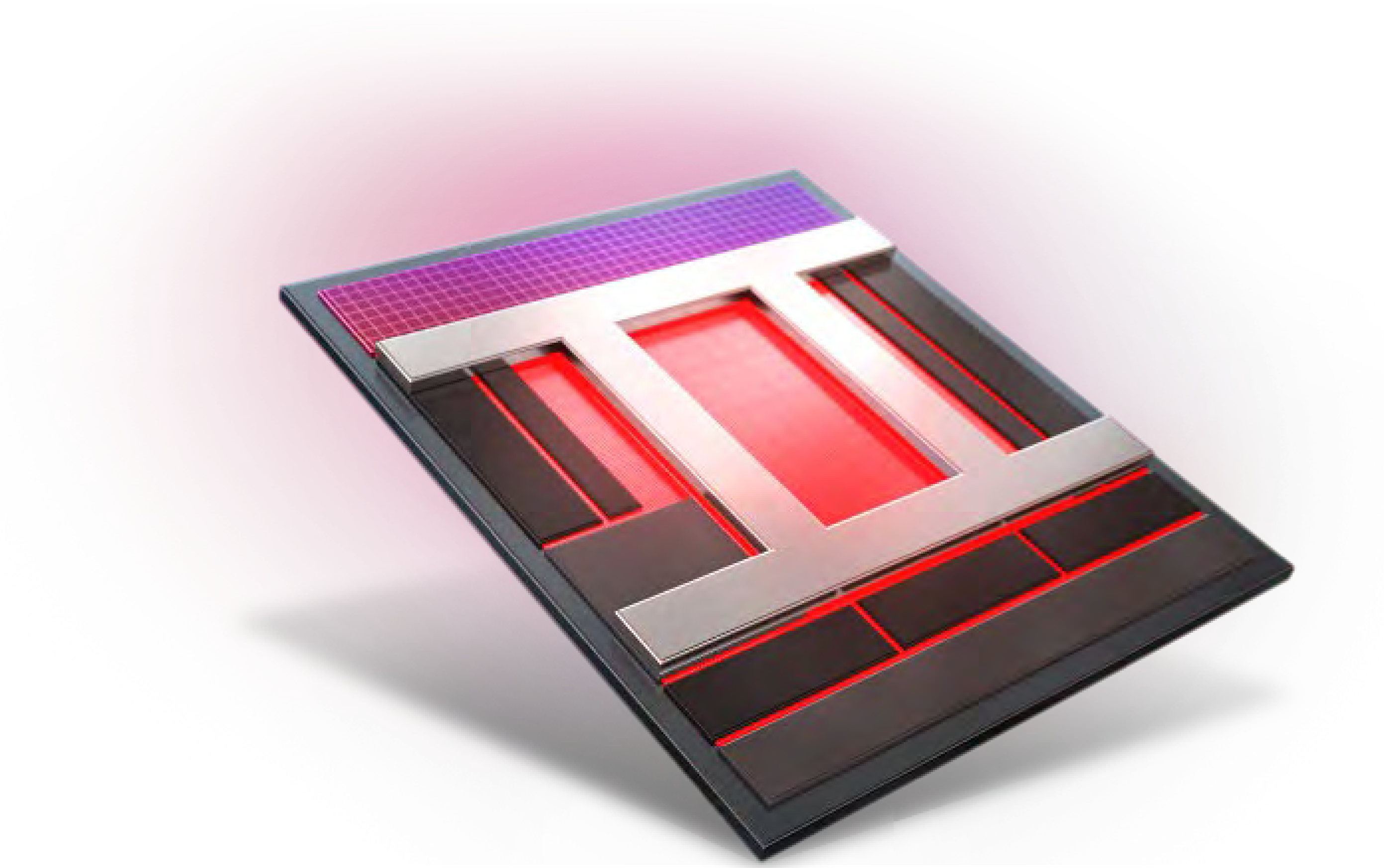


Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e 2018

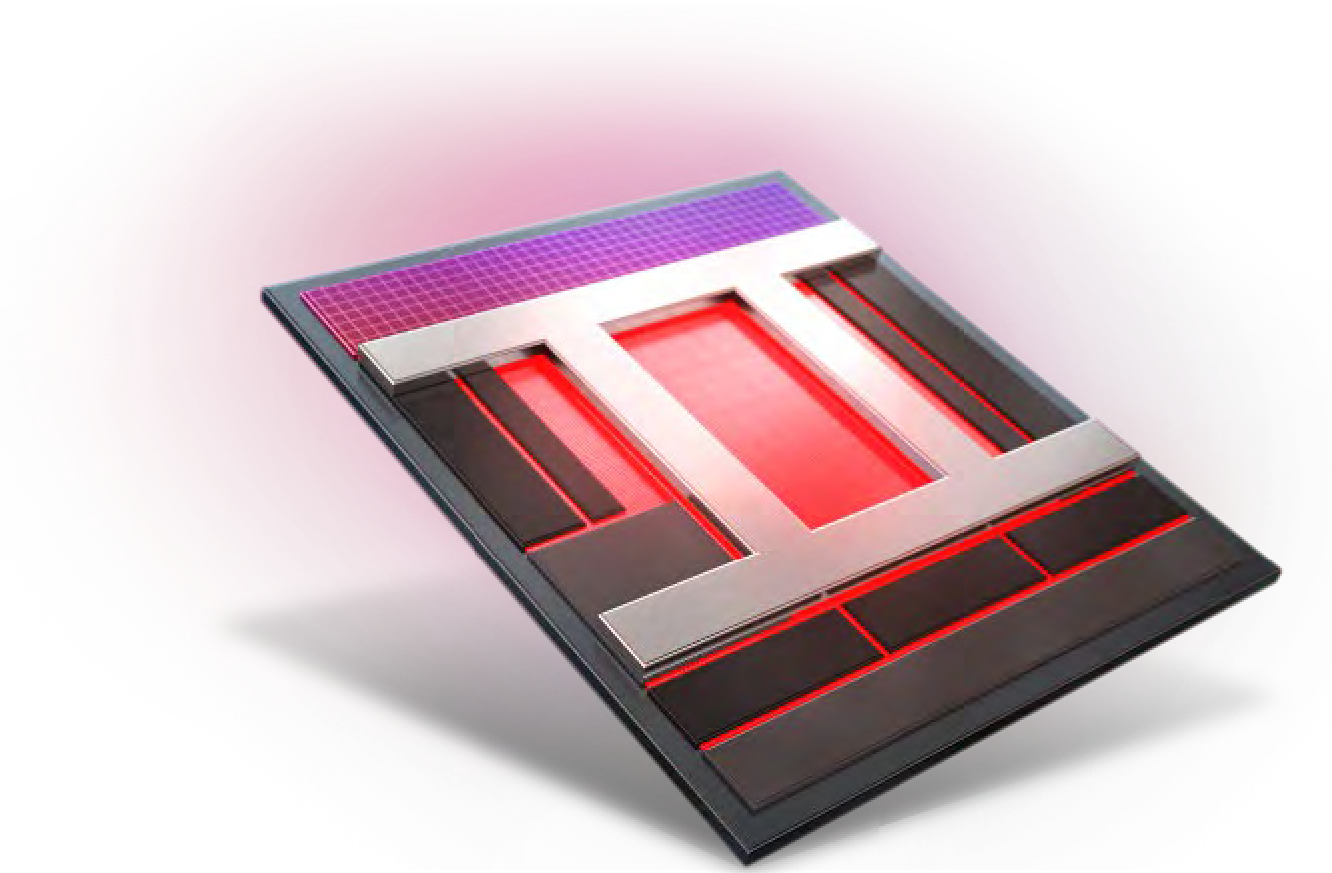
A Single Architecture Can't Do it Alone


Whole Application
Irregular data types, instruction sets, data operation
Sensor Fusion, Pre-Processing, Data Aggregation
Complex Algorithms, Full Linux "Services"
Domain Specific Parallelism (e.g., Video, ML)
Safety Processing, or Latency-Critical Workloads

AMD XILINX

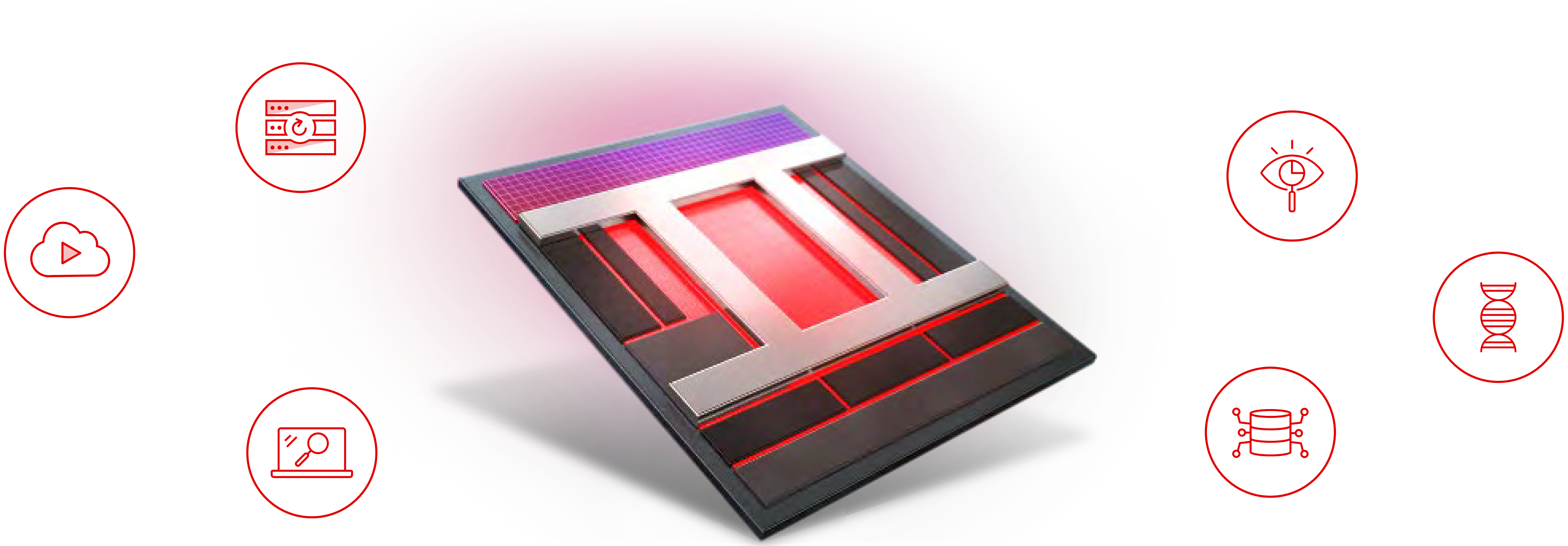


ACAP



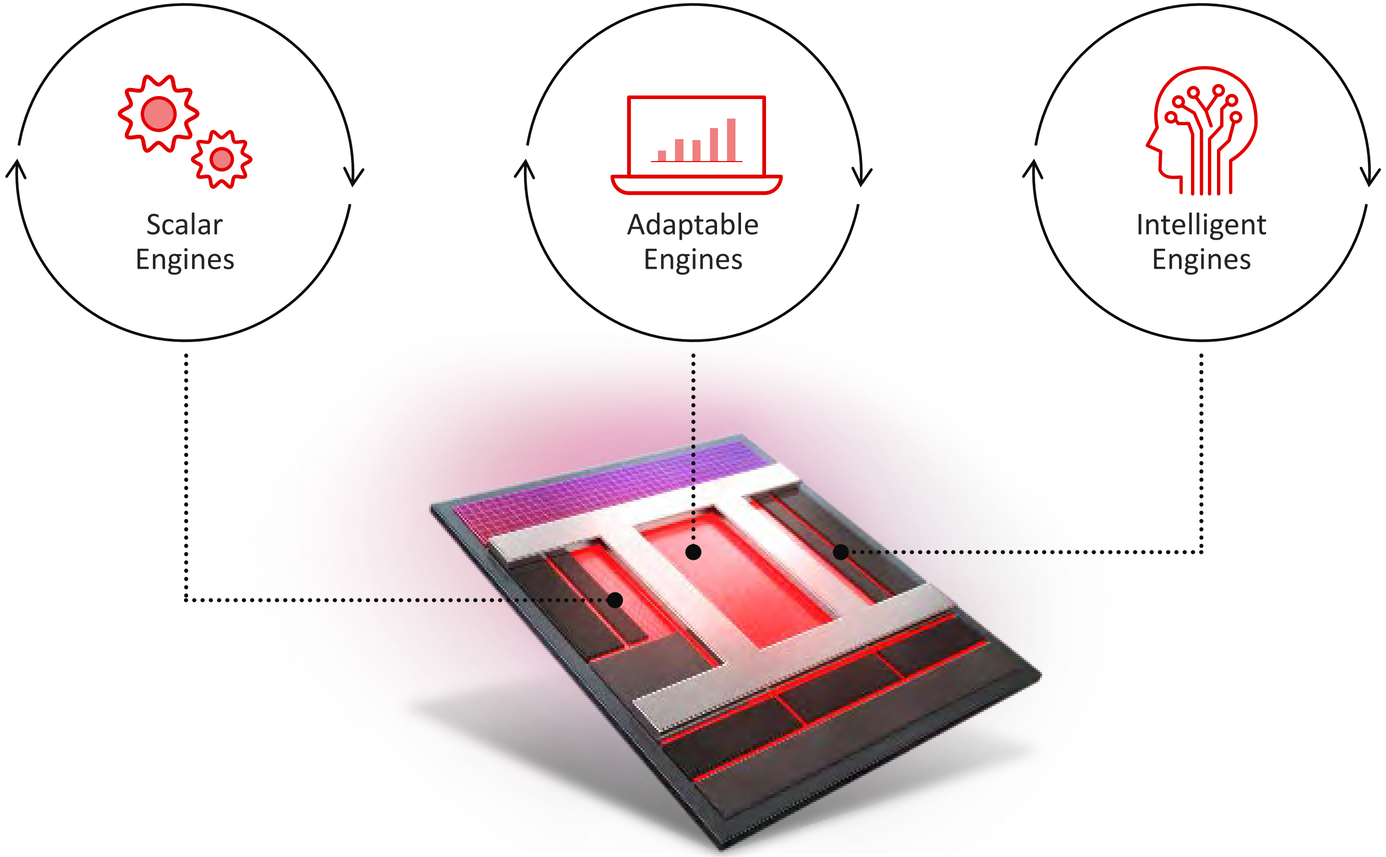
Adaptive
Compute
Acceleration
Platform

Adaptive



Adaptive Hardware for
Domain-Specific Applications

Compute Acceleration



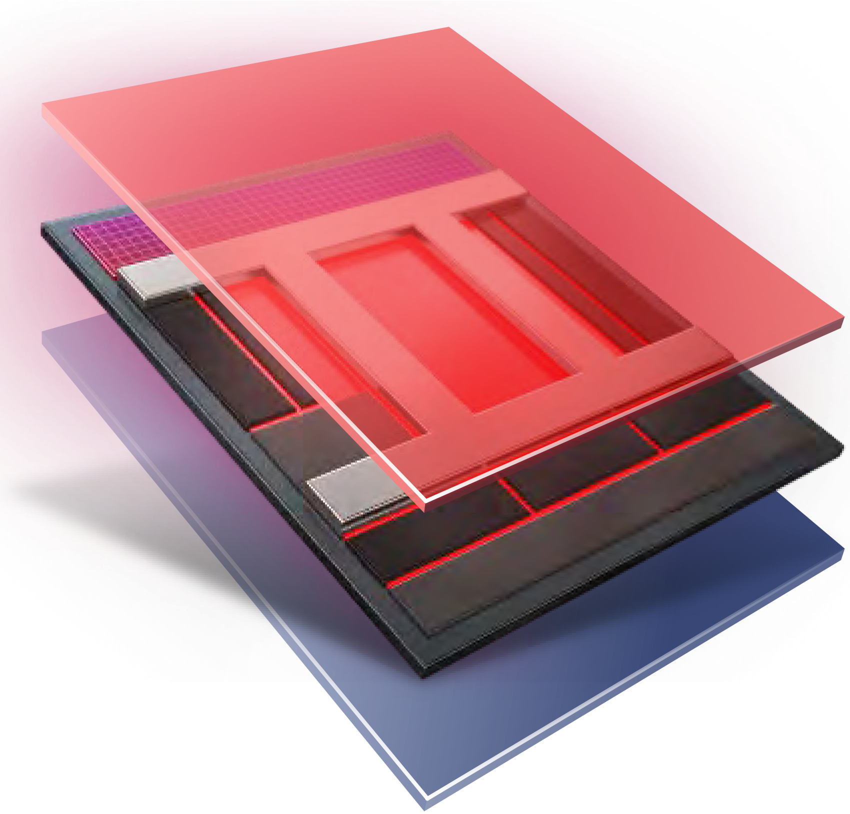
Platform

ENABLING:

Data Scientists

SW App Developers

HW Developers

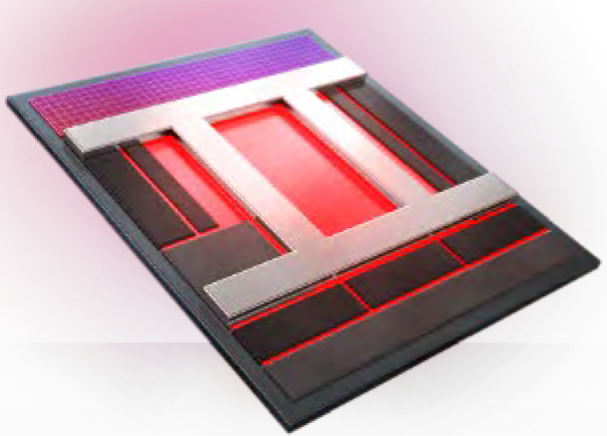


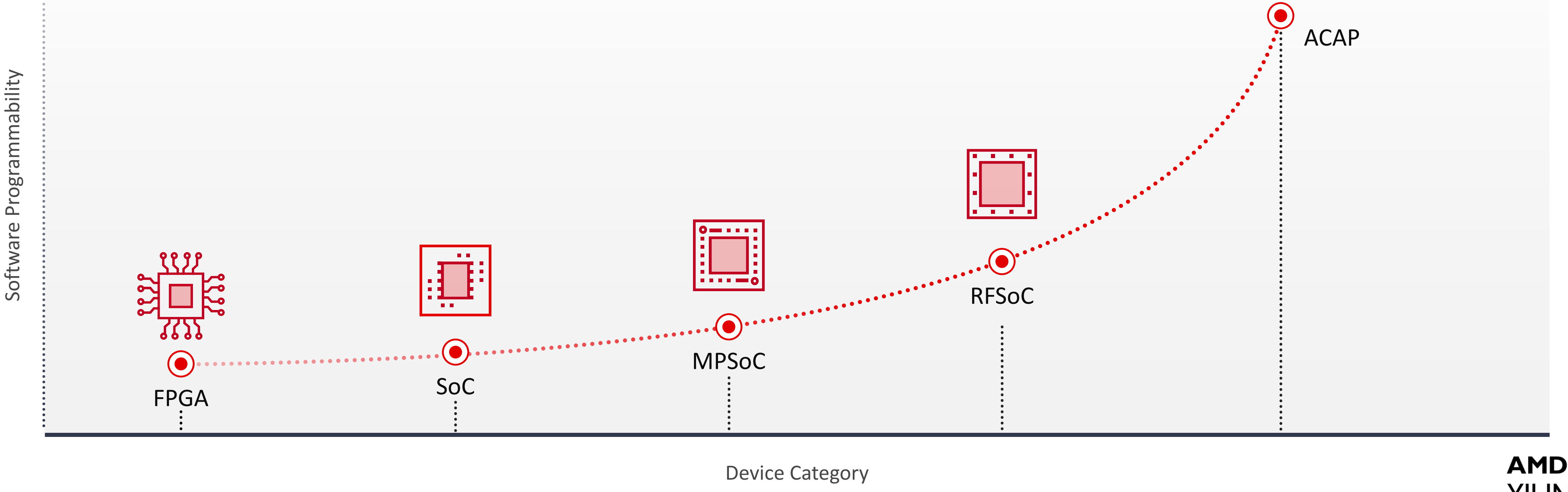
Development Tools
HW/SW Libraries
Run-time Stack

SW Programmable
Silicon Infrastructure

Disruptive Innovation Needed: Enter ACAP

A new class of devices for today's challenges





Introducing the Industry's First ACAP

XILINX® VERSAL™

TILE

XILINX®

VERSAL™

The Industry's First ACAP

Heterogeneous Acceleration

For Any Application

For Any Developer

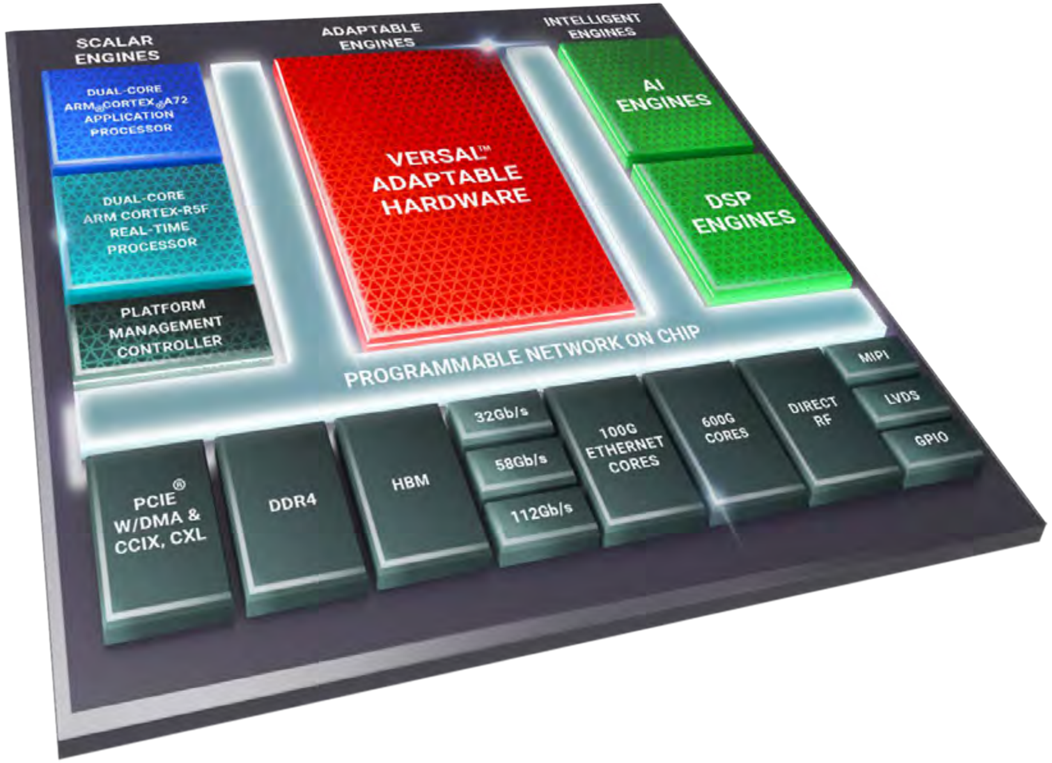
7nm
FinFET

tsmc

XILINX

Versal ACAP

- Scalar Processing Engines
- Adaptable Hardware Engines
- Intelligent Engines
  SW Programmable, HW Adaptable
- Breakout Integration of Advanced Protocol Engines



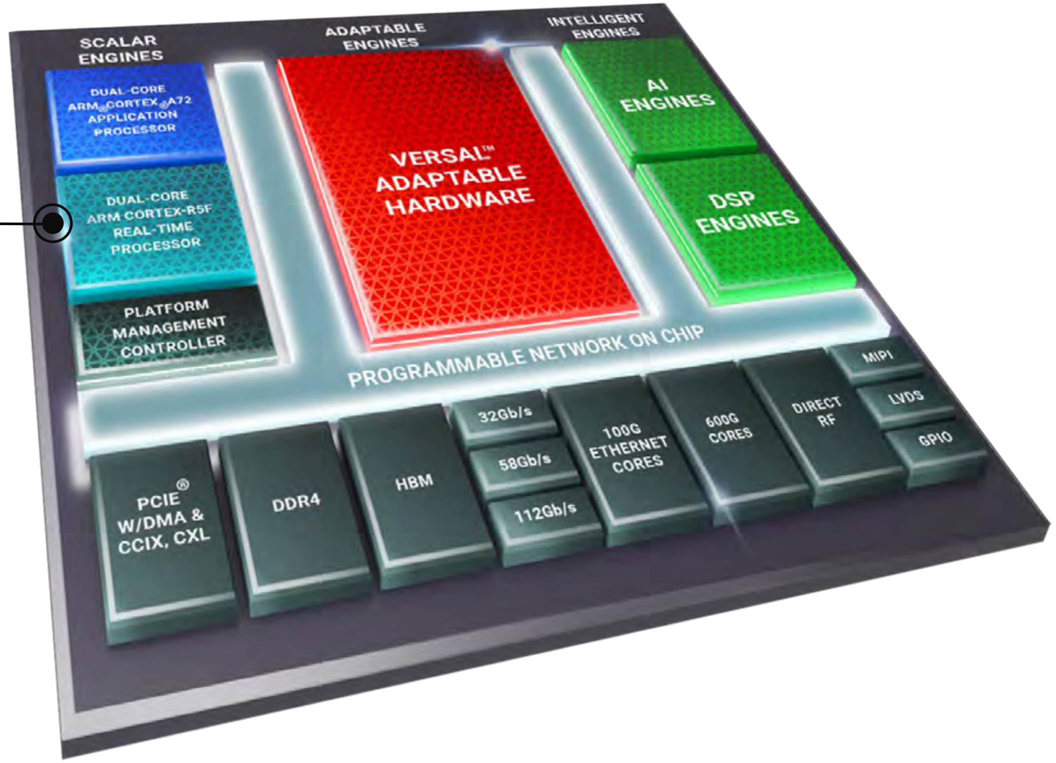
SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL™ ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® W/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G ETHERNET CORES
600G CORES
DIRECT RF
MIPI
LVDS
GPIO

Scalar Processing Engines

Arm Cortex-A72
Application Processor

Arm Cortex-R5
Real-Time Processor

Platform Management Controller

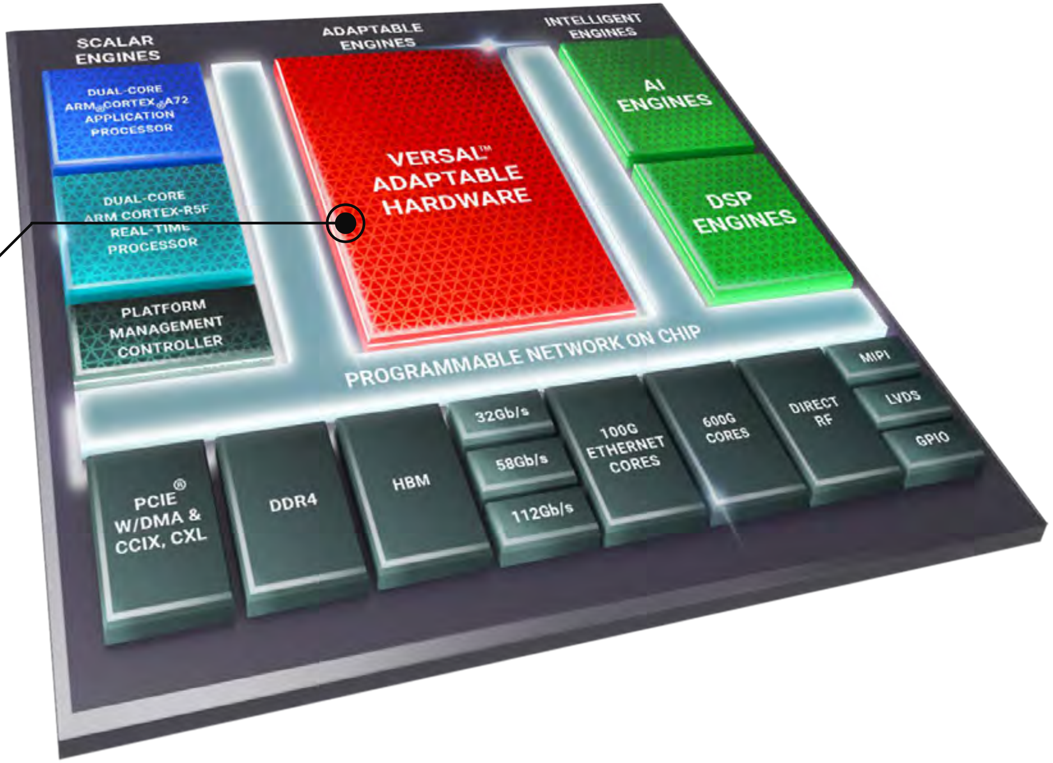

SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL™ ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® W/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G ETHERNET CORES
600G CORES
DIRECT RF
MIPI
LVDS
GPIO

Adaptable Hardware Engines

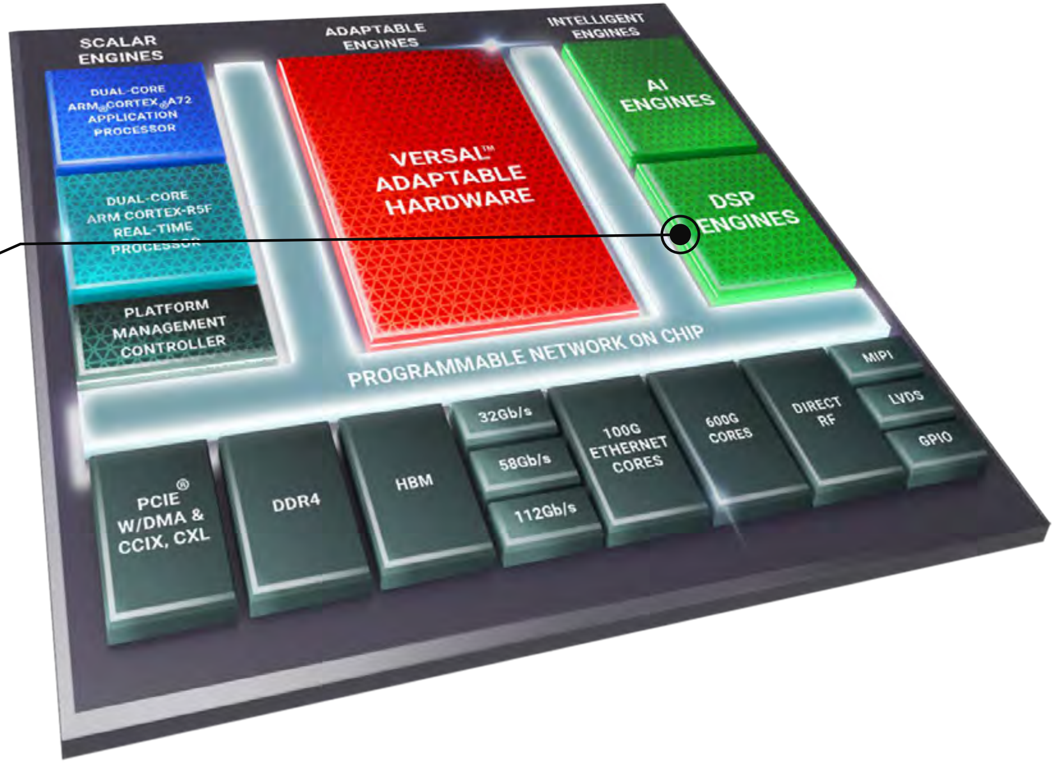
Re-architected foundational HW fabric for greater compute density

Enables custom memory hierarchy

8X Faster Dynamic Reconfiguration ("on-the-fly")


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL™ ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® W/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G ETHERNET CORES
600G CORES
DIRECT RF
MIPI
LVDS
GPIO

Intelligent Engines



SCALAR ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
ADAPTABLE ENGINES
VERSAL™ ADAPTABLE HARDWARE
INTELLIGENT ENGINES
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® W/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G ETHERNET CORES
600G CORES
DIRECT RF
MIPI
LVDS
GPIO

DSP Engines

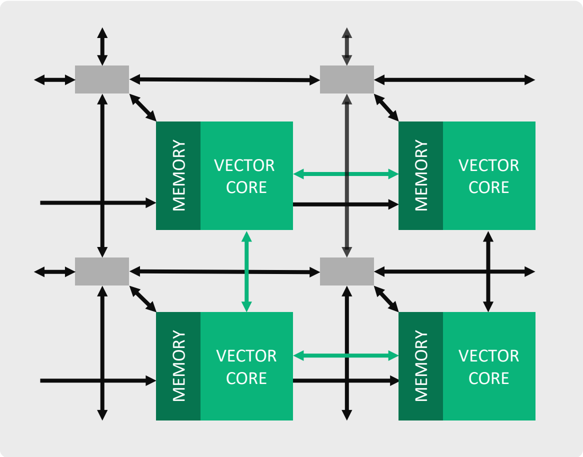
High-precision floating point & low latency

Granular control for customized datapaths

AI Engines

High throughput, low latency, and power efficient

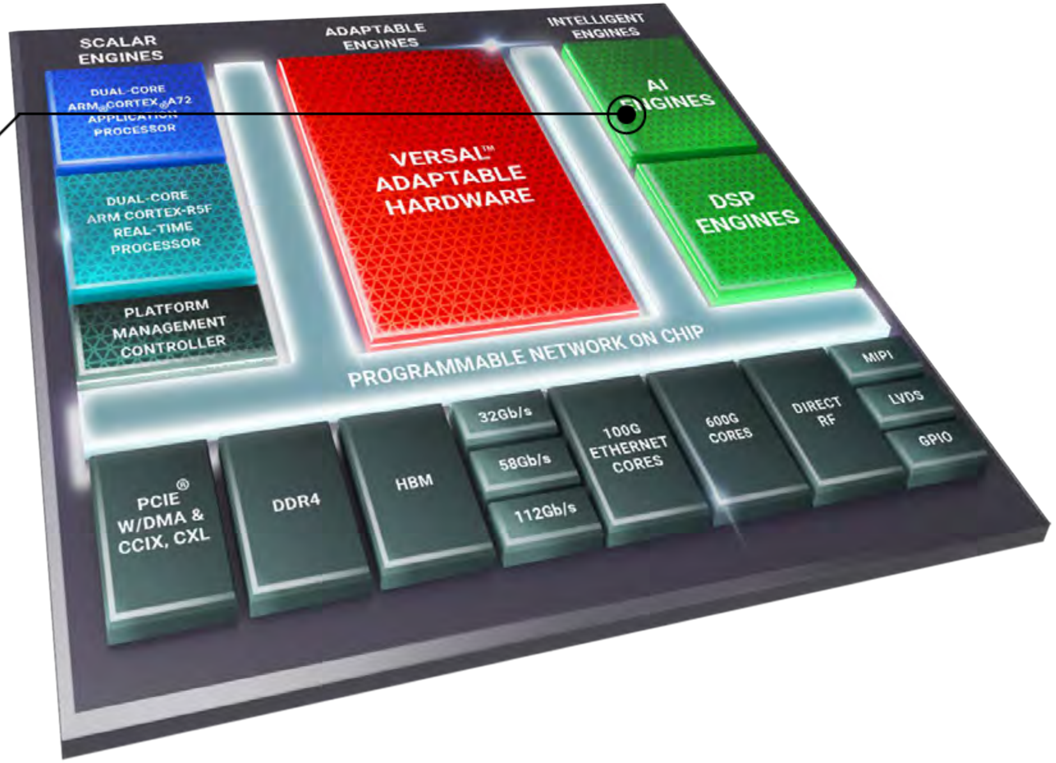
Ideal for AI inference and advanced signal processing



AI Engines

Optimized for AI Inference and Advanced Signal Processing Workloads

- >1GHz VLIW/SIMD vector processor cores
- Massive array of interconnected cores with tightly coupled memory
- Tightly coupled to adaptable hardware engine to enable custom memory hierarchy
- Software programmable, C-, and library-based with hardware adaptability


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL™ ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® W/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G ETHERNET CORES
600G CORES
DIRECT RF
MIPI
LVDS
GPIO

For Any Application

Versal for Multi-Market Applications

CLOUD

NETWORK

EDGE

Data Center

Wired

Wireless

Endpoints

AI ADOPTION ACROSS MARKETS

XILINX® VERSAL™

AI RF Series

HBM Series

AI Core Series

Premium Series

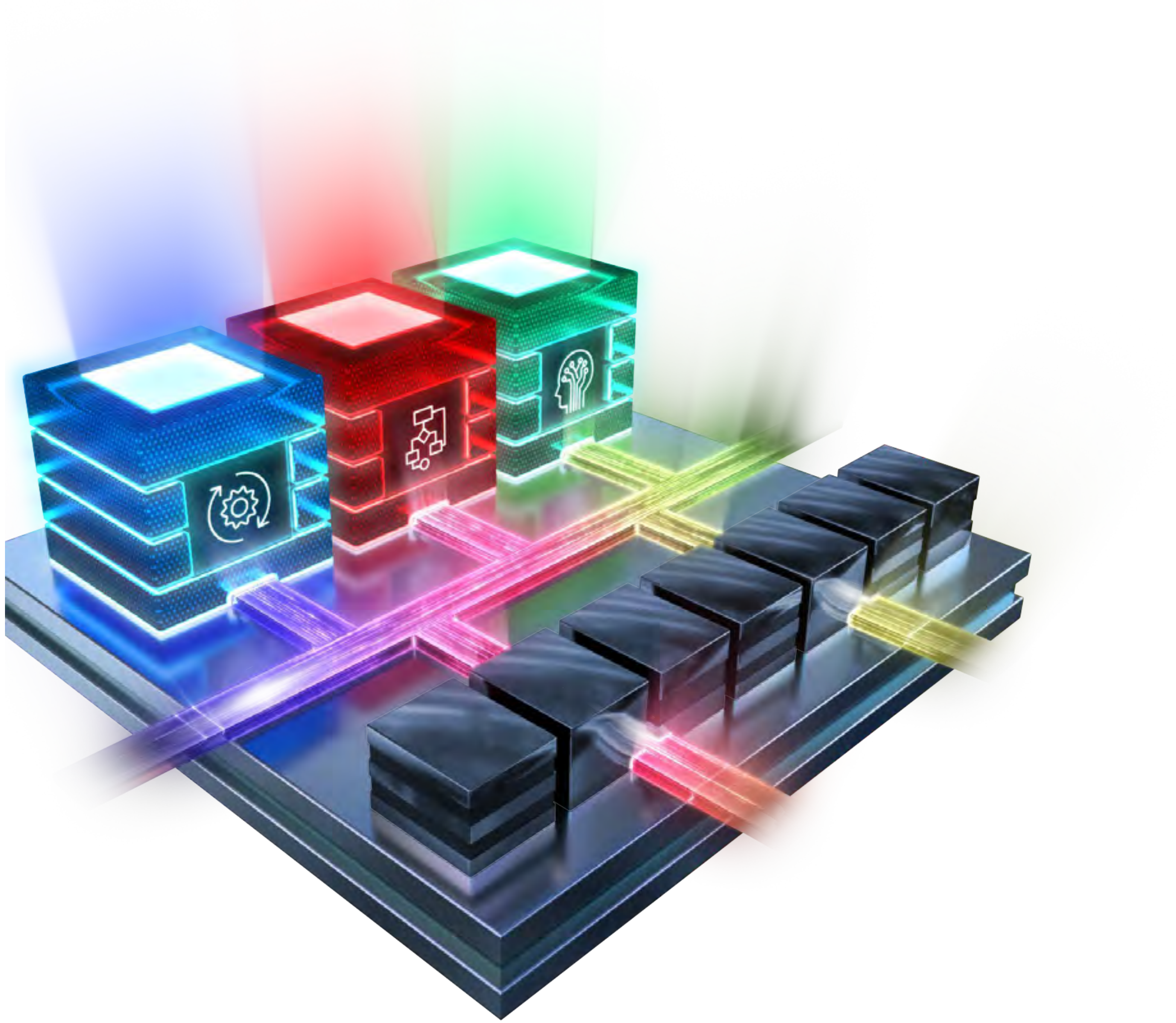
AI Edge Series

Prime Series

Adaptive Compute Acceleration Platform (ACAP)

XILINX VERSAL™

- Heterogeneous acceleration
  - For any application
  - For any developer
- TSMC 7 nm FinFET
  - Software programmability and domain-specific hardware acceleration with adaptability
- Six series of devices
  - Scalability and AI inference capabilities for diverse applications



Versal ACAP: Architecture Overview

Versal Architecture: Overview

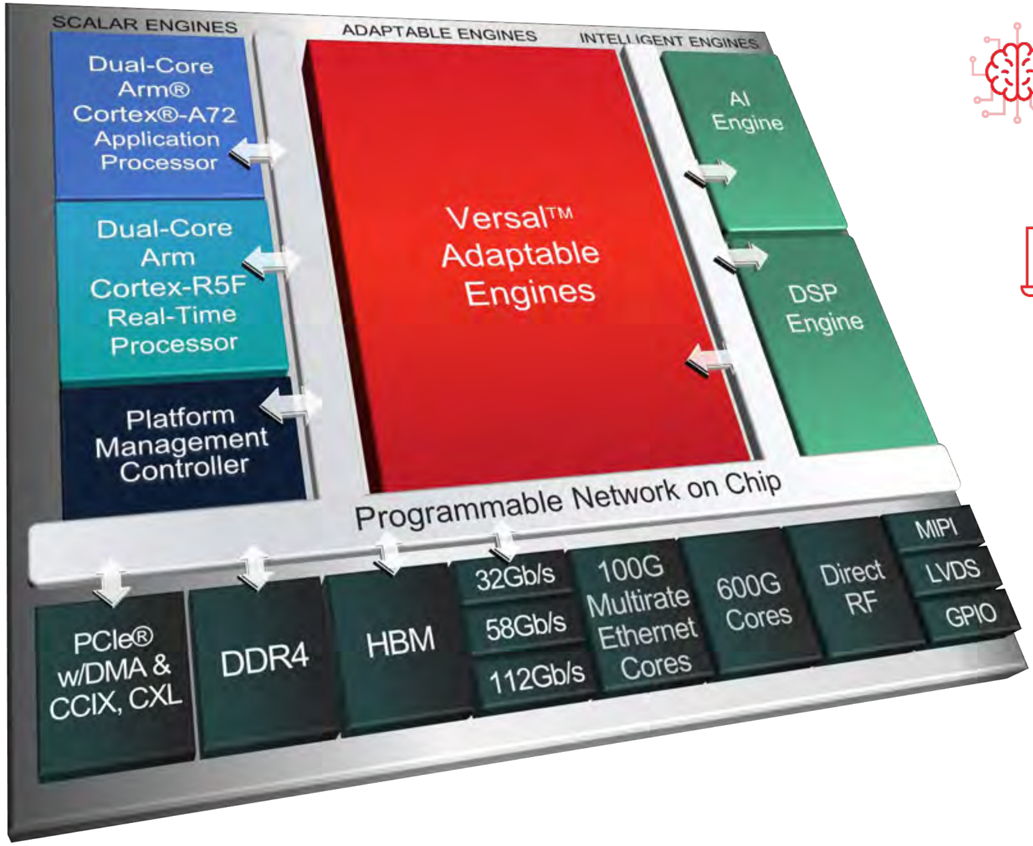
Adaptable Engines
- 2X compute density
- Voltage scaling for perf/watt

Scalar Engines
- Platform control
- Embedded edge compute

PCIe Gen5 & CCIX
- 2X PCIe and DMA bandwidth
- Cache coherency

DDR4 Memory
- 3200-DDR4, 4266-LPDDR4
- 2X bandwidth/pin

Transceiver Leadership
- Broad range, 32G → 112G
- 58G in mainstream devices


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
Dual-Core Arm® Cortex®-A72 Application Processor
Dual-Core Arm Cortex-R5F Real-Time Processor
Platform Management Controller
Versal™ Adaptable Engines
AI Engine
DSP Engine
Programmable Network on Chip
PCIe® w/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G Multirate Ethernet Cores
600G Cores
Direct RF
MIPI
LVDS
GPIO

Intelligent Engines (DSP)
- AI compute
- Diverse DSP workloads

Programmable NoC
- Guaranteed bandwidth
- Efficiently moves data among portions of devices

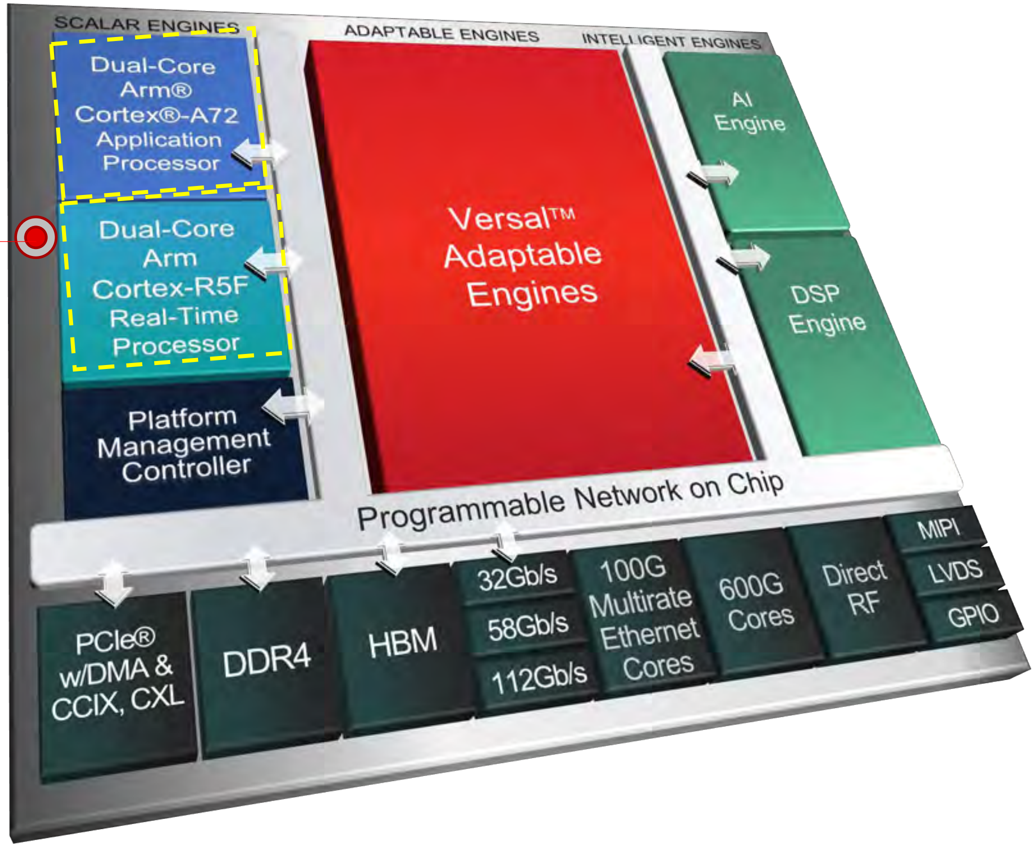
Protocol Engines
- 400G/600G cores
- Power optimized

Programmable I/O
- Any interface or sensor
- Includes 3.2Gb/s MIPI

Scalar Engines

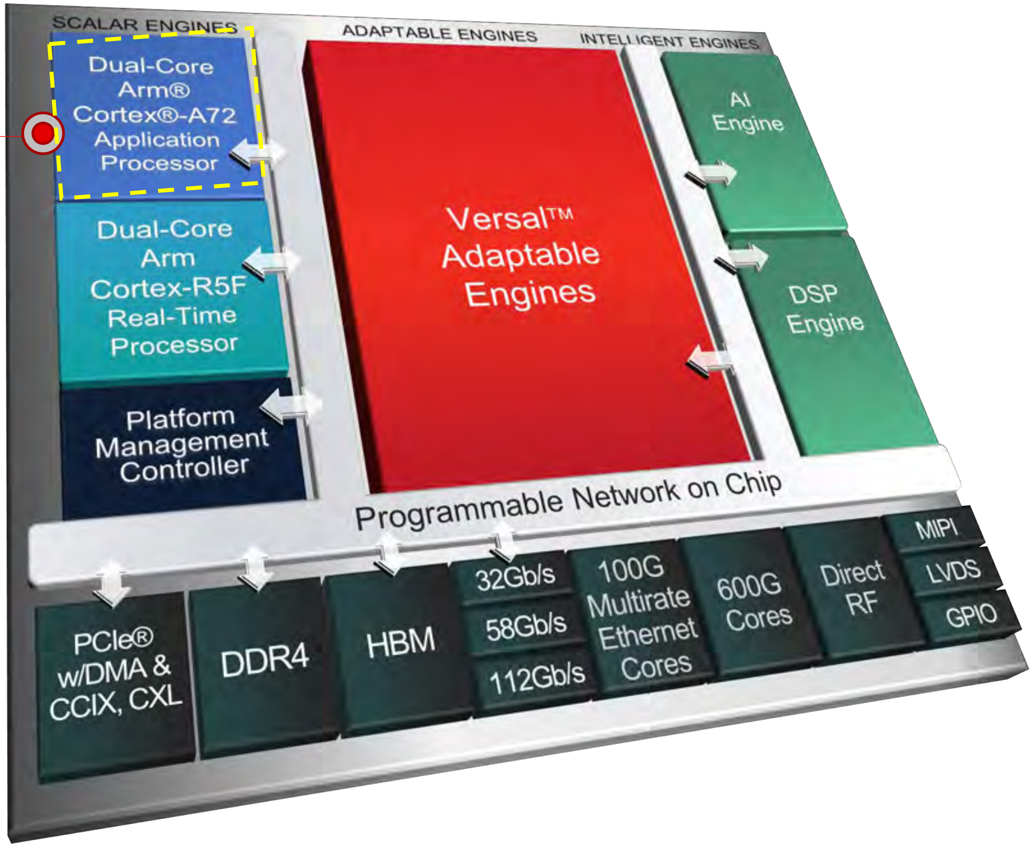
Scalar Engines for Platform Management

- Execute complex algorithms and decision making for autonomous systems
- Provide safety processing and redundancy for mission- and safety-critical applications
- Manage the entire platform
- Load each aspect of the ACAP and monitor status
- Support capability extension
  - PL-instantiated MicroBlaze™ processor


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
Dual-Core Arm® Cortex®-A72 Application Processor
Dual-Core Arm Cortex-R5F Real-Time Processor
Platform Management Controller
Versal™ Adaptable Engines
AI Engine
DSP Engine
Programmable Network on Chip
PCIe® w/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G Multirate Ethernet Cores
600G Cores
Direct RF
MIPI
LVDS
GPIO

Scalar Engines

Application Processing Unit (APU)



SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
Dual-Core Arm® Cortex®-A72 Application Processor
Dual-Core Arm Cortex-R5F Real-Time Processor
Platform Management Controller
Versal™ Adaptable Engines
AI Engine
DSP Engine
Programmable Network on Chip
PCIe® w/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G Multirate Ethernet Cores
600G Cores
Direct RF
MIPI
LVDS
GPIO

Based on Arm Cortex-A72 dual-core processor

- System memory management unit (SMMU)
- Cache coherent interconnect (CCI) unit
- Interface channels
- System peripherals

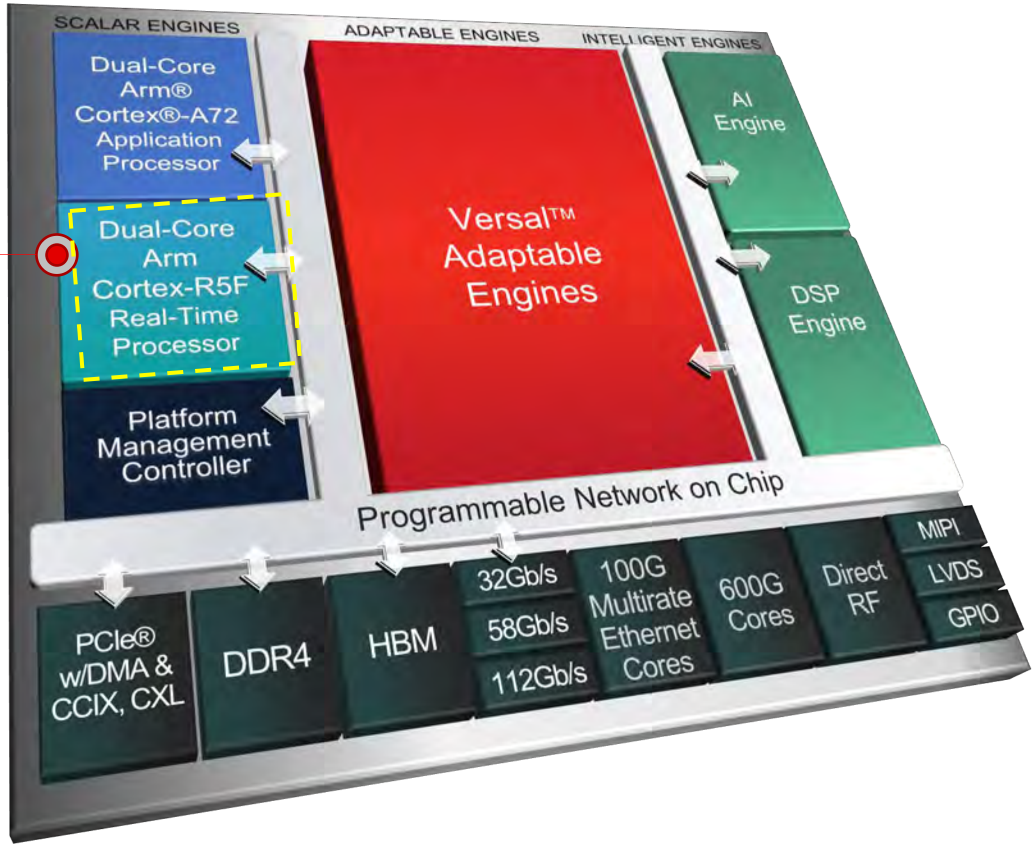
SMMU and CCI provide shared memory environment with PS, PMC, and PL processors

Features:

- Up to 1.7 GHz speed – 2x single-threaded performance
- Armv8 architecture
- Up in seconds
- Supports Linux and bare-metal

Scalar Engines

Real-time Processing Unit (RPU)


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
Dual-Core Arm® Cortex®-A72 Application Processor
Dual-Core Arm Cortex-R5F Real-Time Processor
Platform Management Controller
Versal™ Adaptable Engines
AI Engine
DSP Engine
Programmable Network on Chip
PCIe® w/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G Multirate Ethernet Cores
600G Cores
Direct RF
MIPI
LVDS
GPIO

Based on Arm Cortex-R5F dual-core processor

- L1 caches
- Tightly coupled memories (TCM)
- Configured into dual-processor mode or into lock-step mode for greater performance

Features:

- Functional safety
- Split mode for performance or lock step for safety
- Low latency, determinism, and real-time control for any application
- ASIL/SIL certifiable

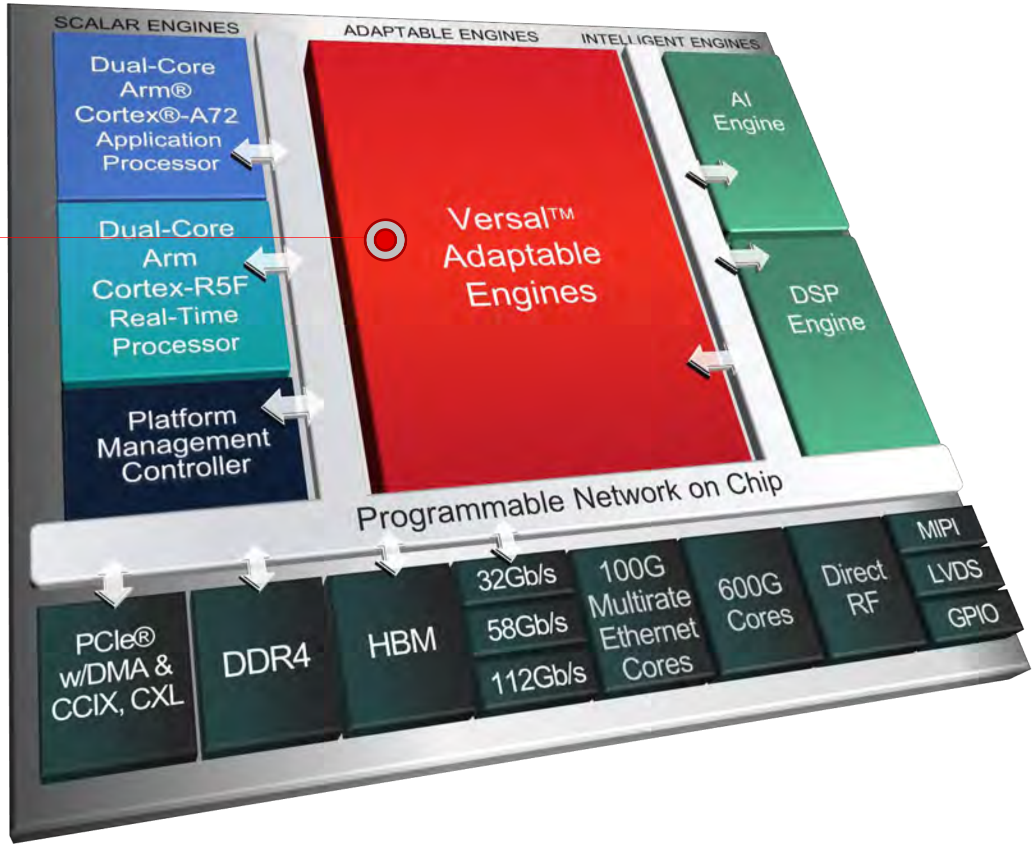
Adaptable Engines

Adaptable Hardware Engines – Programmable Logic

Millions of system logic cells are available for any workload which can be reconfigured on the fly

Parallel, pipelined architectures, and hybrids are all possible with this flexible logic

Wide variety of memory elements providing right compute and memory structures


SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
Dual-Core Arm® Cortex®-A72 Application Processor
Dual-Core Arm Cortex-R5F Real-Time Processor
Platform Management Controller
Versal™ Adaptable Engines
AI Engine
DSP Engine
Programmable Network on Chip
PCIe® w/DMA & CCIX, CXL
DDR4
HBM
32Gb/s
58Gb/s
112Gb/s
100G Multirate Ethernet Cores
600G Cores
Direct RF
MIPI
LVDS
GPIO

Configurable Logic Block

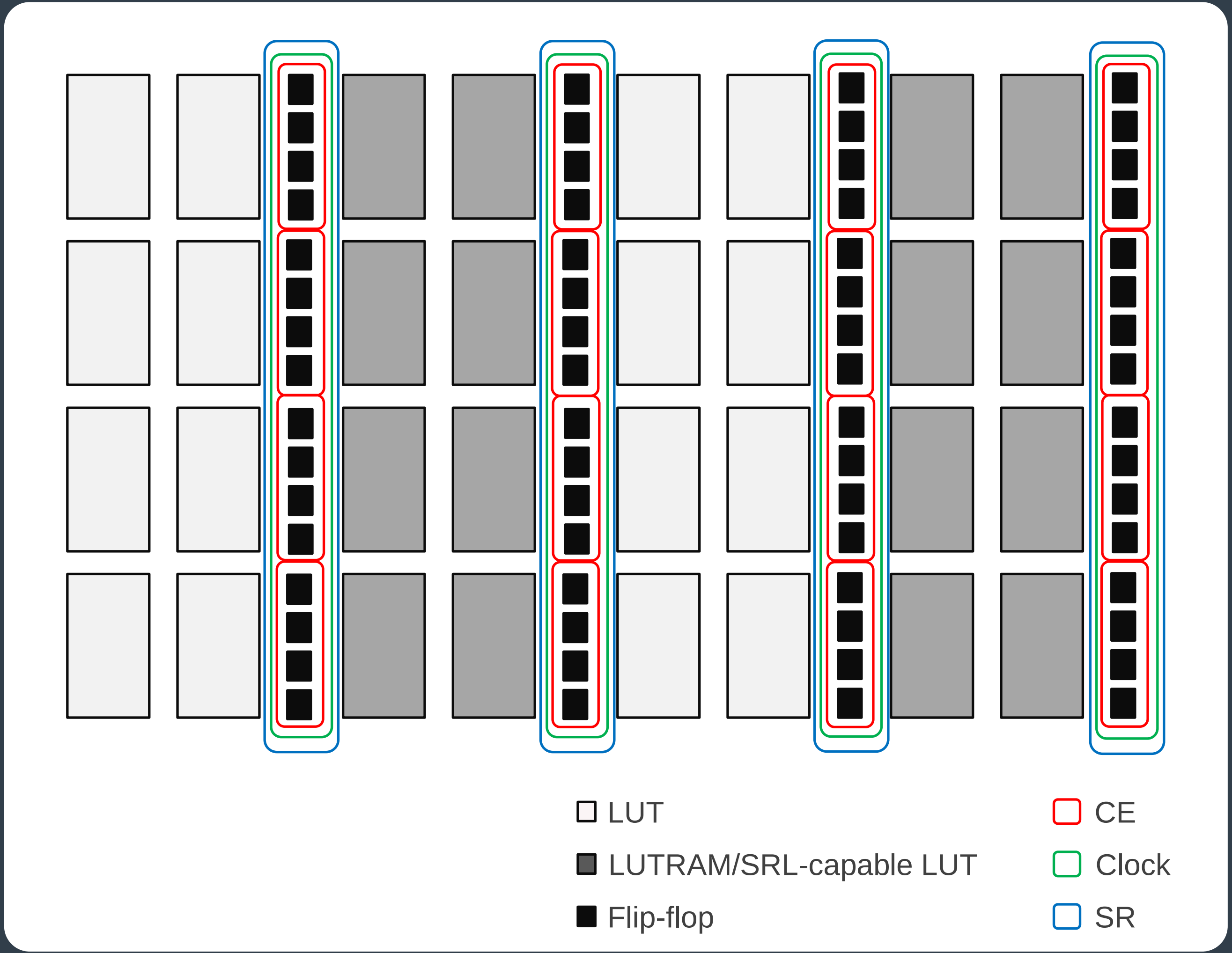
Heart of the programmable logic

CLBs include logic and look-up tables (LUTs)

- Can be configured into different combinations
- Can create special-purpose functions, processing units, and other entities

CLB contains:

- 32 LUTs and 64 flip-flops
- Arithmetic carry logic and multiplexers
- Control signals
  - 4 clocks, 4 set/resets, 16 clock enables
- 50% of the LUTs can be combined to form LUTRAM and shift registers
- Dedicated interconnect paths



CLB Features

Four times more logic capacity than before

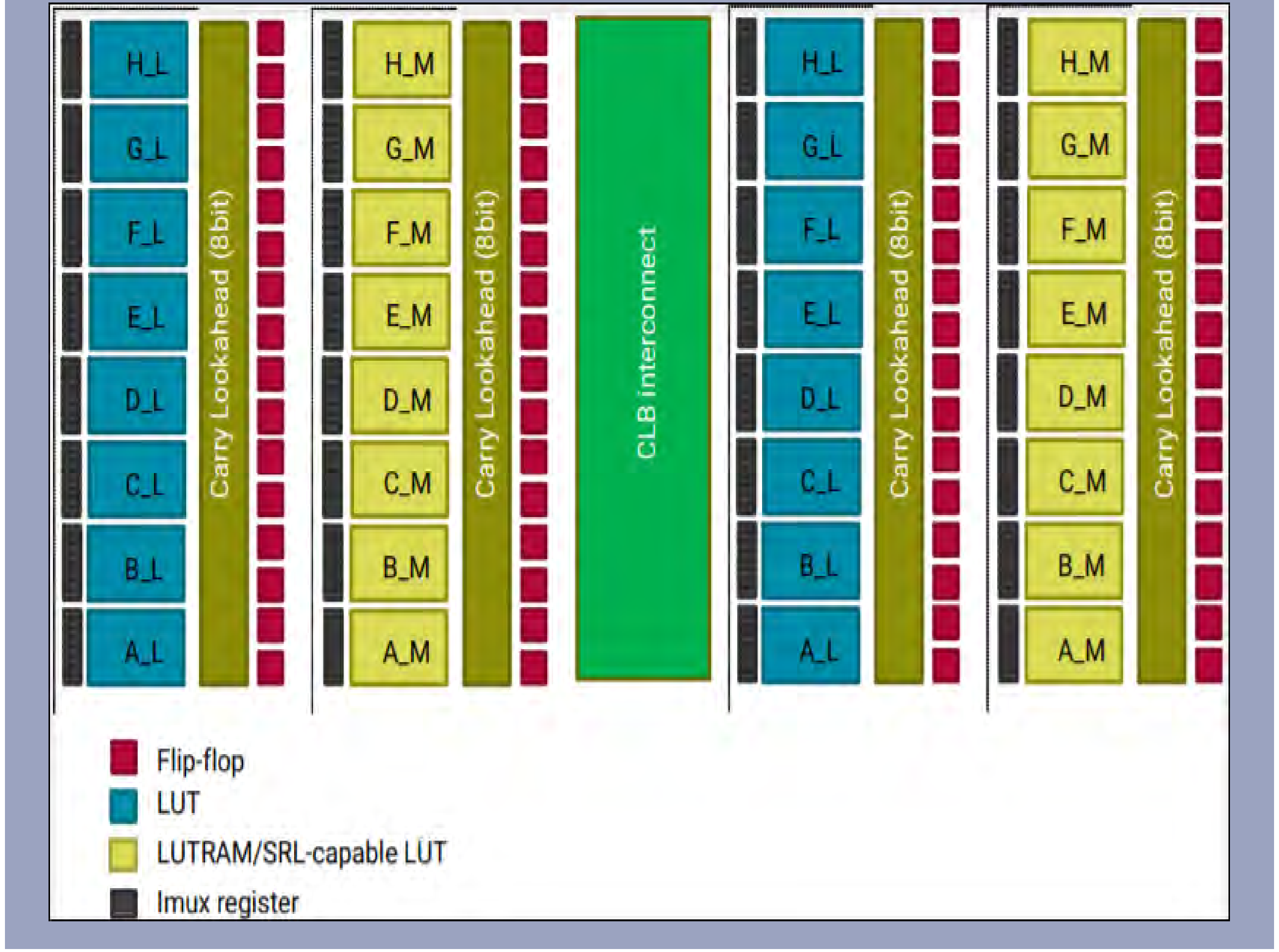
Dedicated LUT-LUT cascade paths exist inside the CLB to reduce delays

Implementation of any arbitrary programmable logic function into functional units (LUTs)

Flip-flops and latches for state retention and pipelining

SLL connections are now part of the CLB

Block Diagram of CLB



Wide function multiplexers are no longer implemented and other LUTs are used instead

New cascade multiplexers enable new carry chains to start at bits 0 and 4

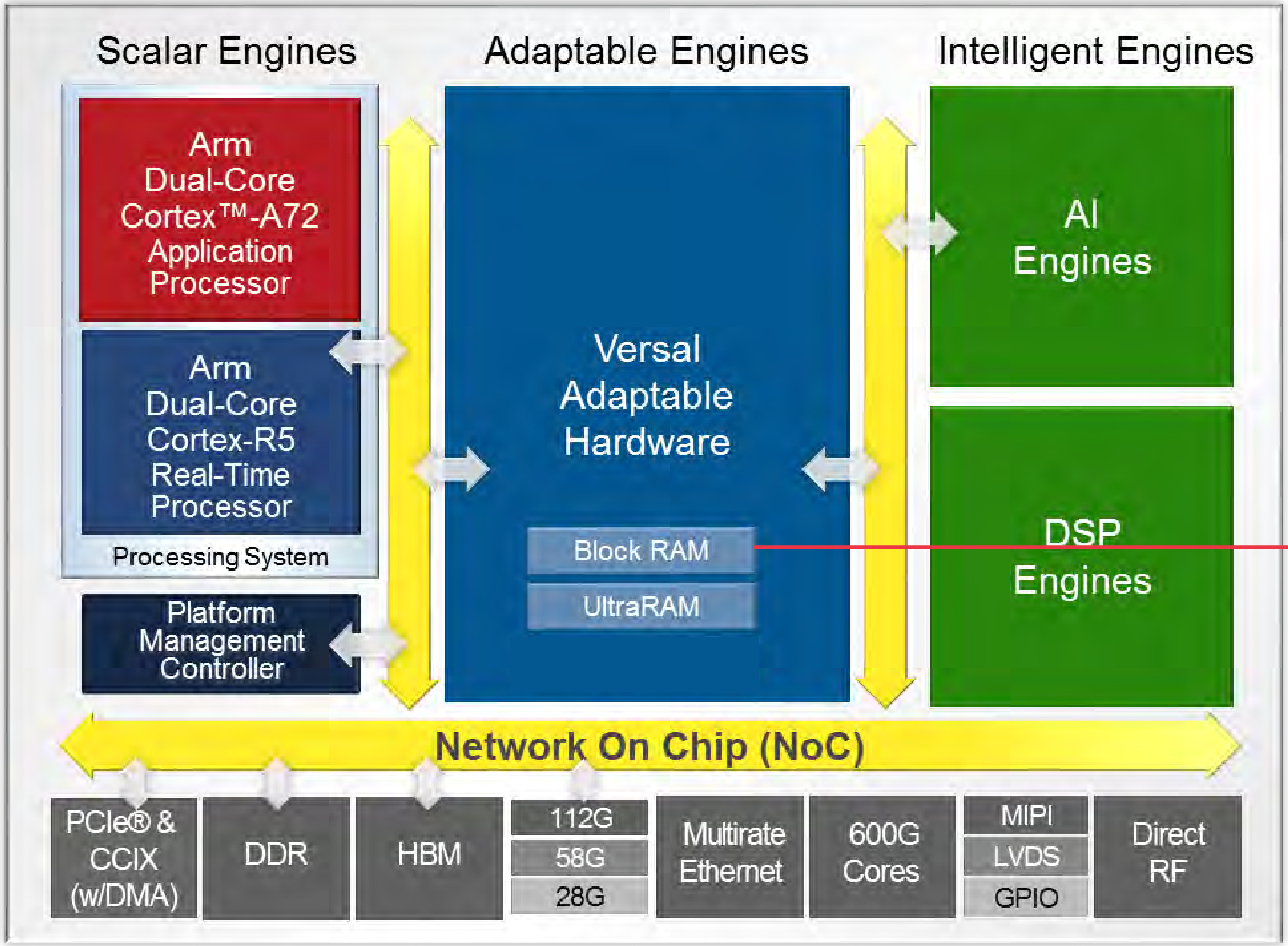
LUTRAMs are simplified, having dedicated hardware to support 32- and 64-bit depths

Output multiplexing in CLBs

Additional registers (Imux registers) are embedded into the CLB and in the local interconnect for all hard blocks connected to programmable logic routing

Adaptable Versal Architecture

Seamless memory-mapped access to the full height and width of the device



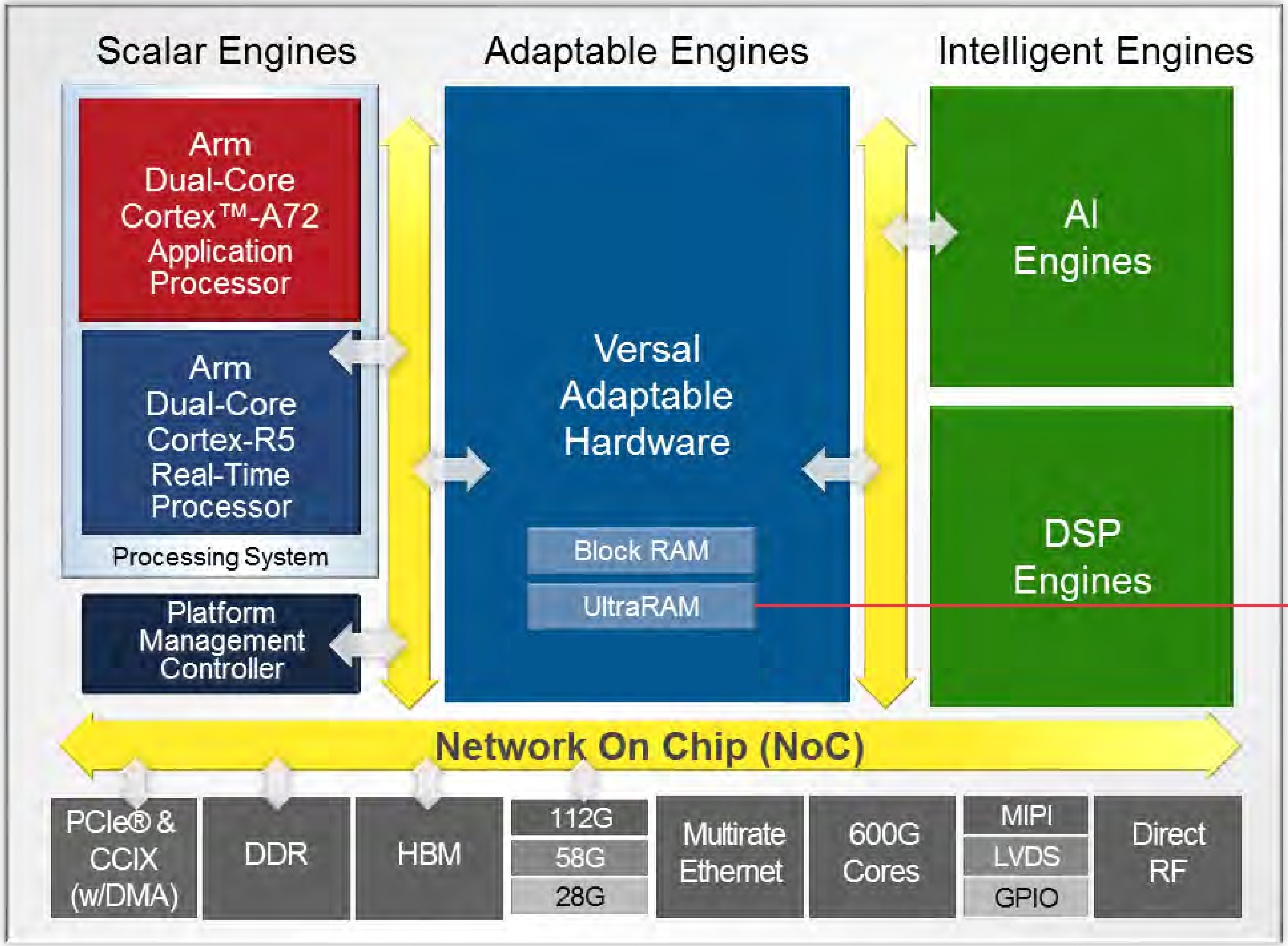
Block RAM

4 KB data with ECC (36 Kb)

- Placed in columns within the CRs and across the device
- Can be cascaded to enable a deeper memory implementation

Adaptable Versal Architecture

Seamless memory-mapped access to the full height and width of the device

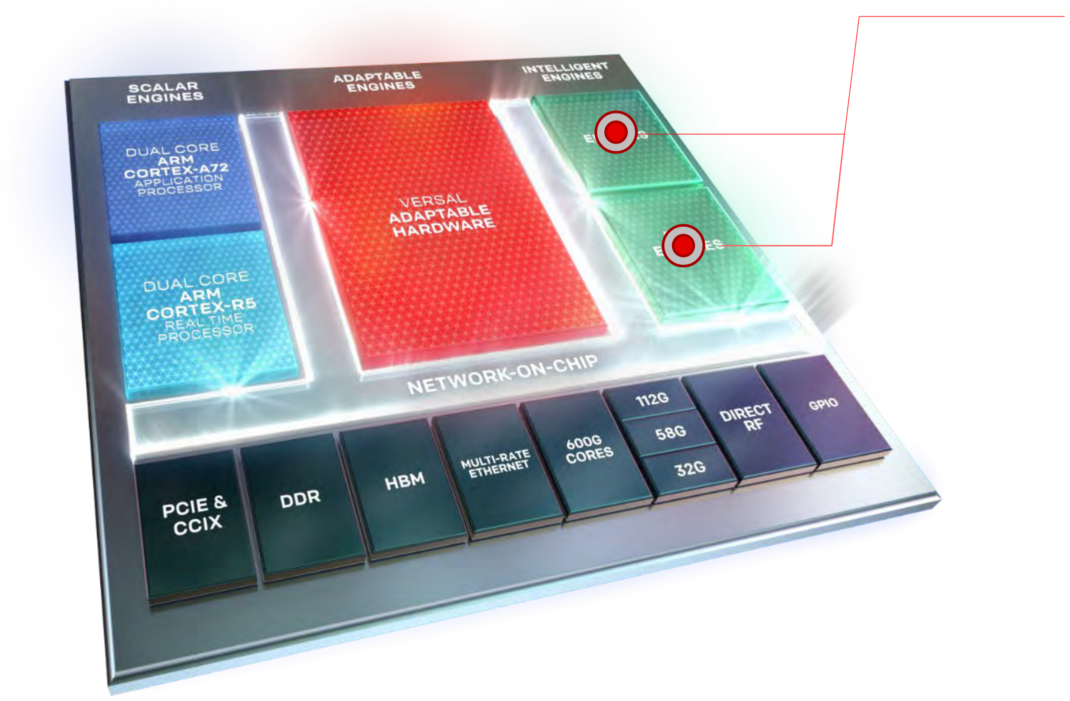


UltraRAM

32 KB data with ECC (288 Kb)

- Single-clocked, two-port, synchronous
- Arranged in the columnar Versal architecture

Intelligent Engines



Wired communications, automotive, and consumer markets

AI Is Everywhere

Intelligent Engines for Diverse Compute

DSP Engines

High-precision, floating-point computation support

Offload additional functions for acceleration

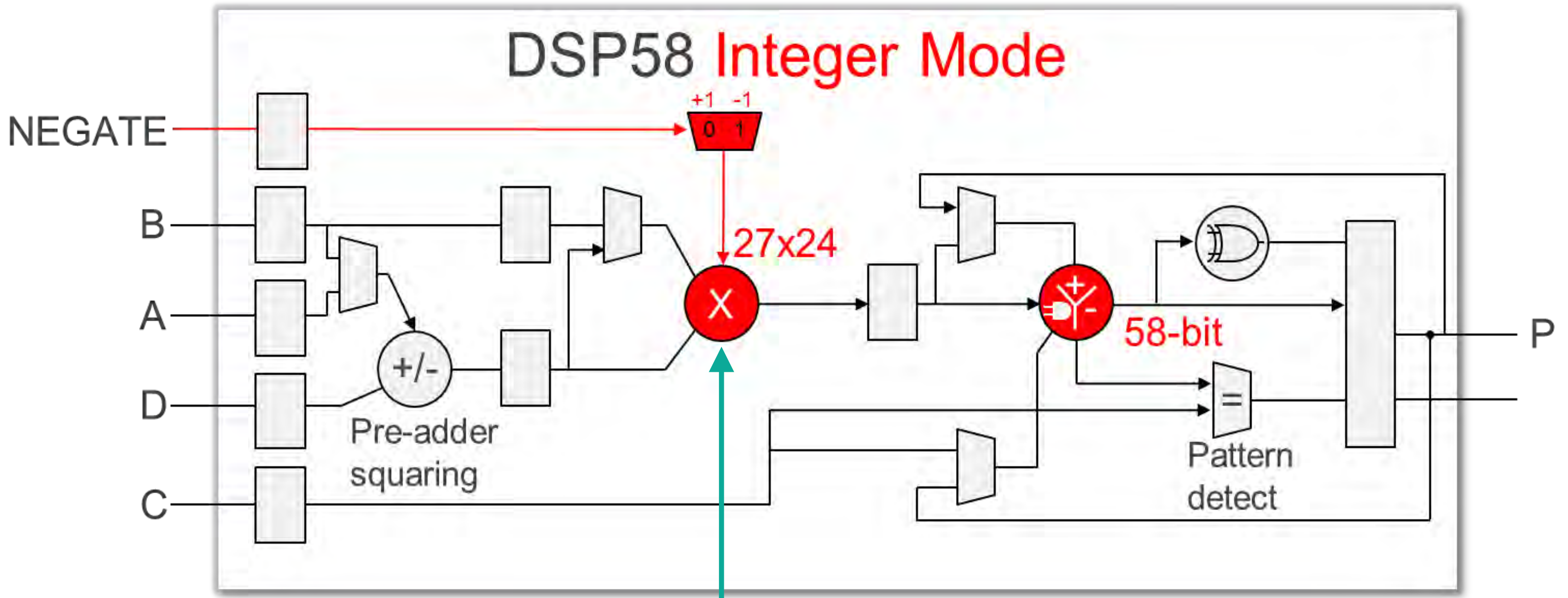
AI Engines

High throughput, low latency, and power efficient

Ideal for AI inference and advanced signal processing

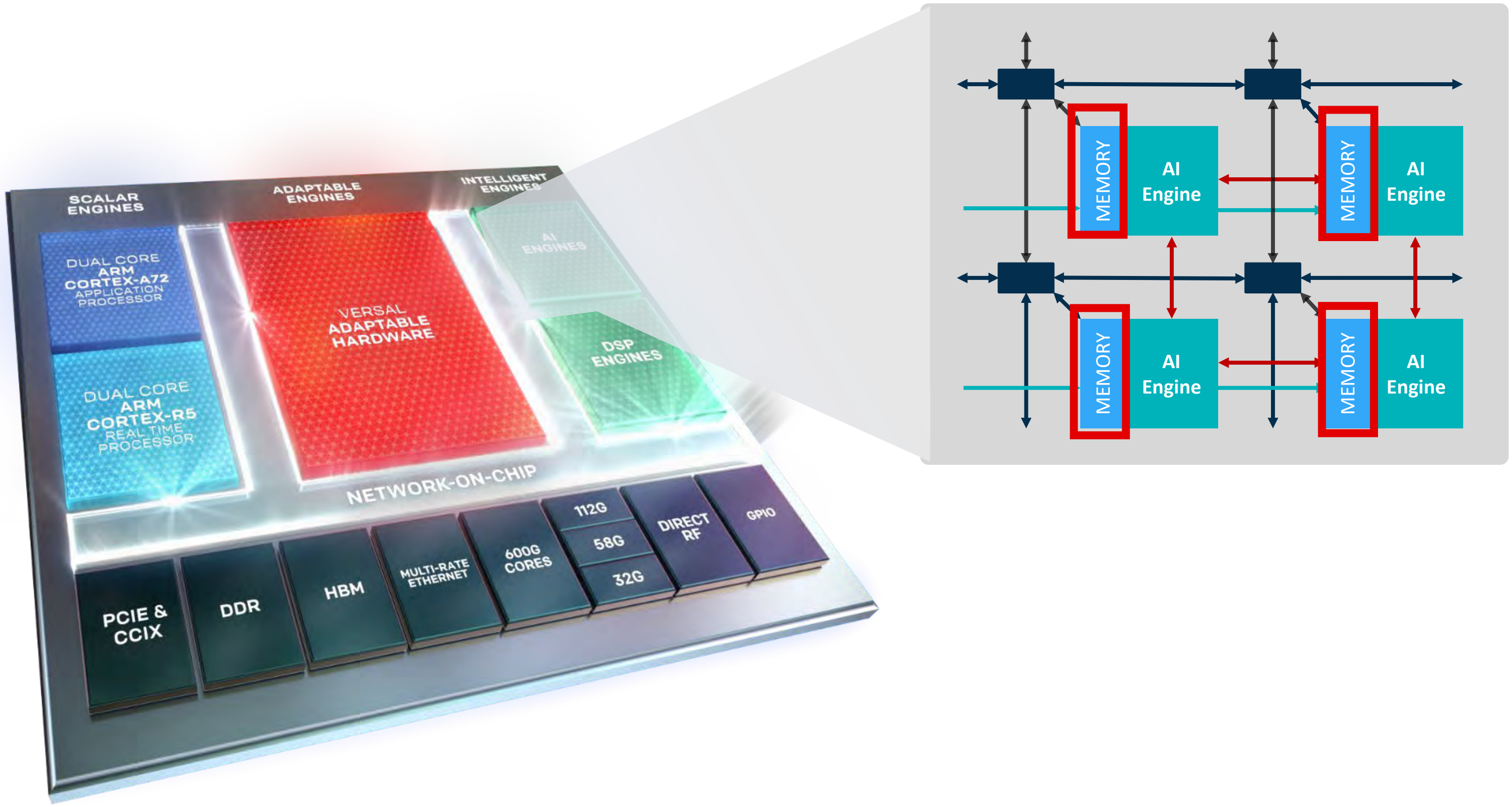
Digital Signal Processing Capability

DSP Engine combines high speed with small size for high performance and system design flexibility



- Multiplier can be dynamically bypassed
- Two 58-bit inputs can feed a SIMD arithmetic unit
- Logic unit can generate a logic function on the two operands

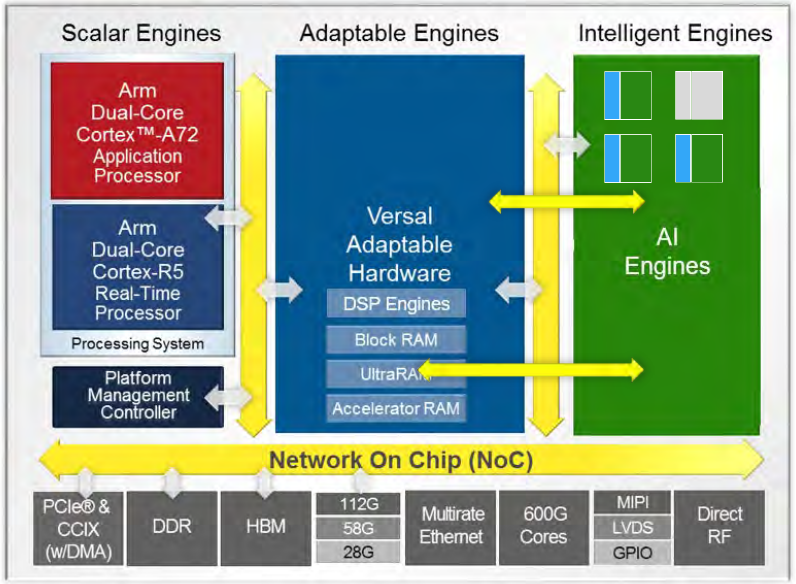
AI Engine: Hardened Compute, Memory, and Interconnect



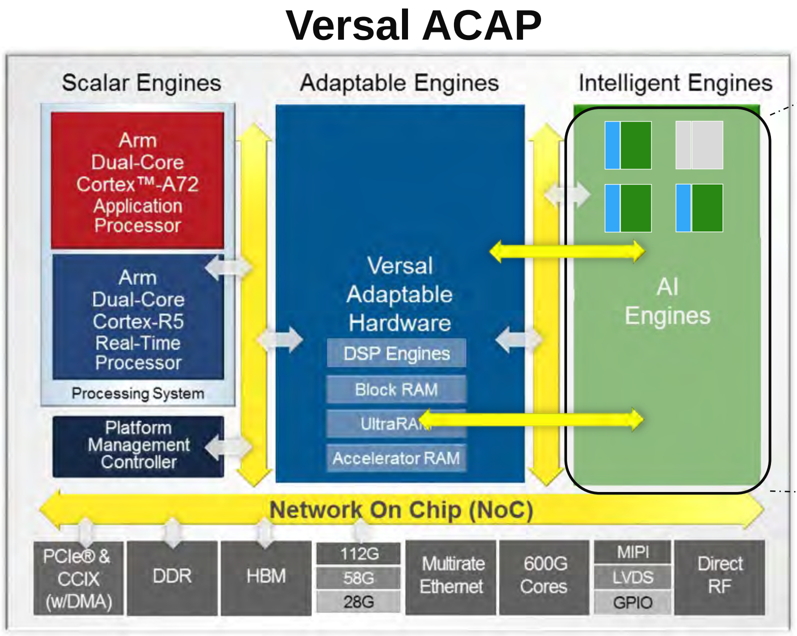
- Immense array of SIMD/VLIW processors with its own instruction and data memory
- Connected on an innovative interconnect built with massive bandwidth
- Form a tightly integrated heterogeneous compute platform
- Provide up to five times higher compute density for vector-based algorithms
- Provide that throughput within the system power and thermal profile by complementing the DSP58s

AI Engine: Terminology

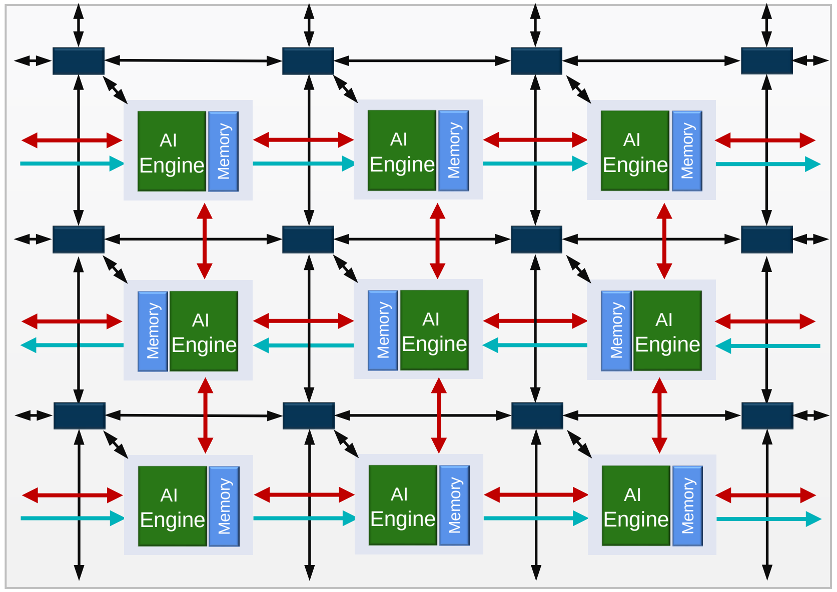
Versal ACAP



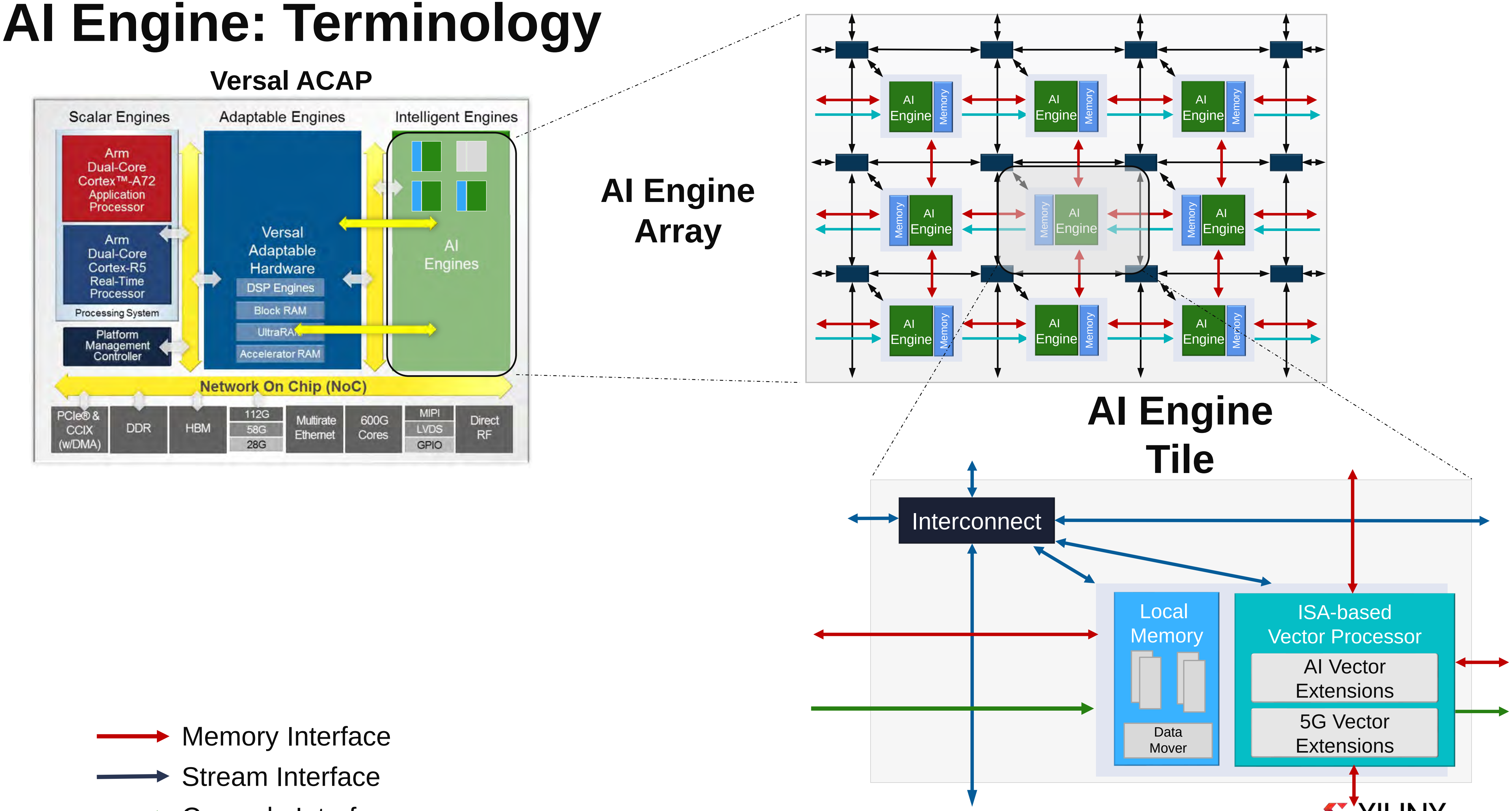
AI Engine: Terminology



AI Engine Array



AI Engine: Terminology



Versal ACAP

AI Engine Array

AI Engine Tile

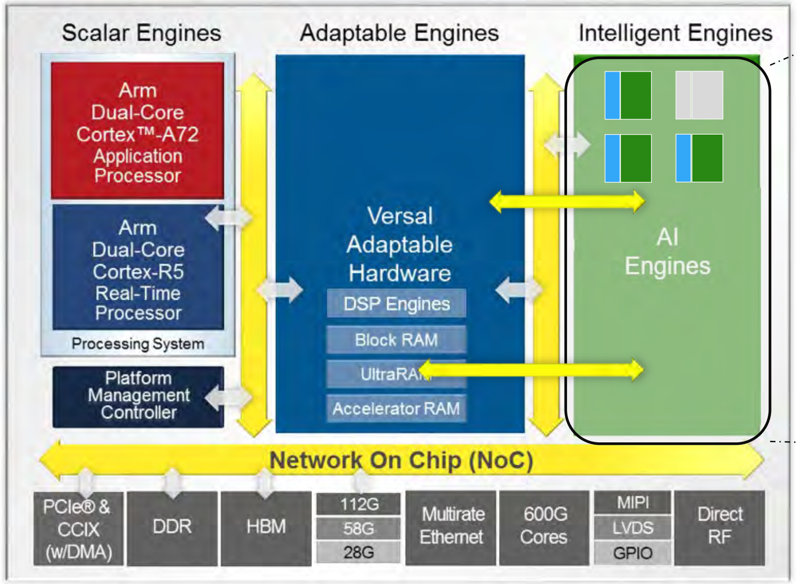
Memory Interface

Stream Interface

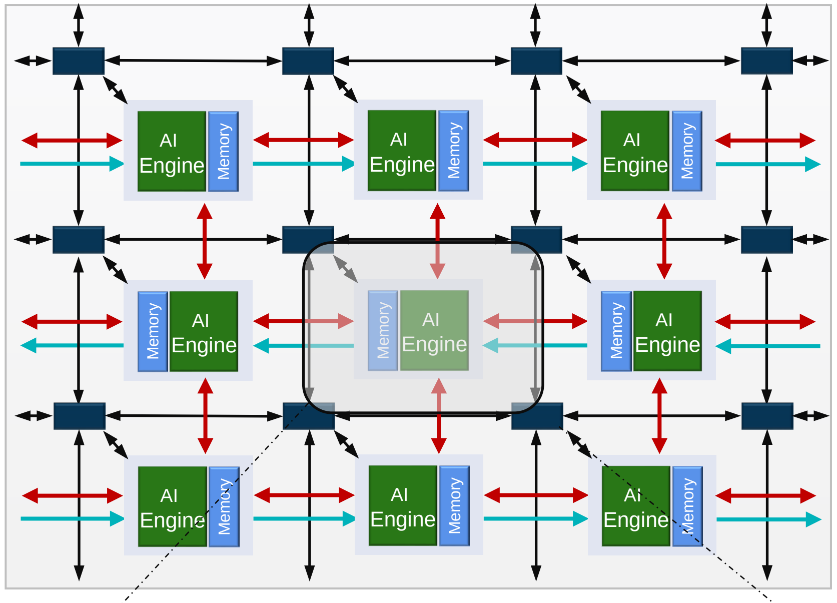
Cascade Interface

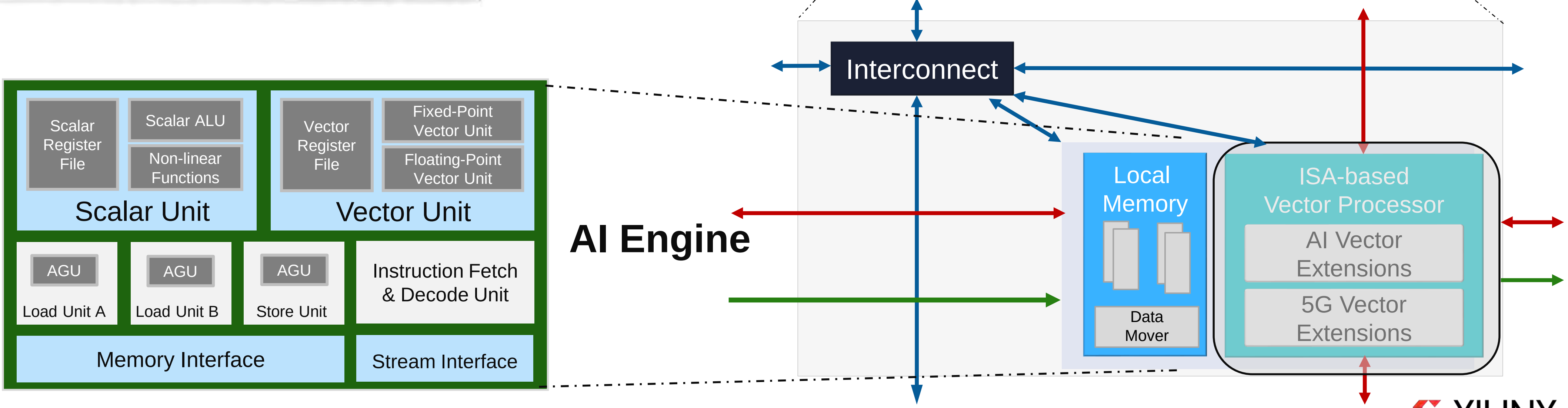
AI Engine: Terminology

Versal ACAP

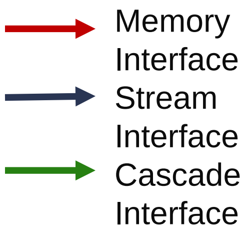


AI Engine Array

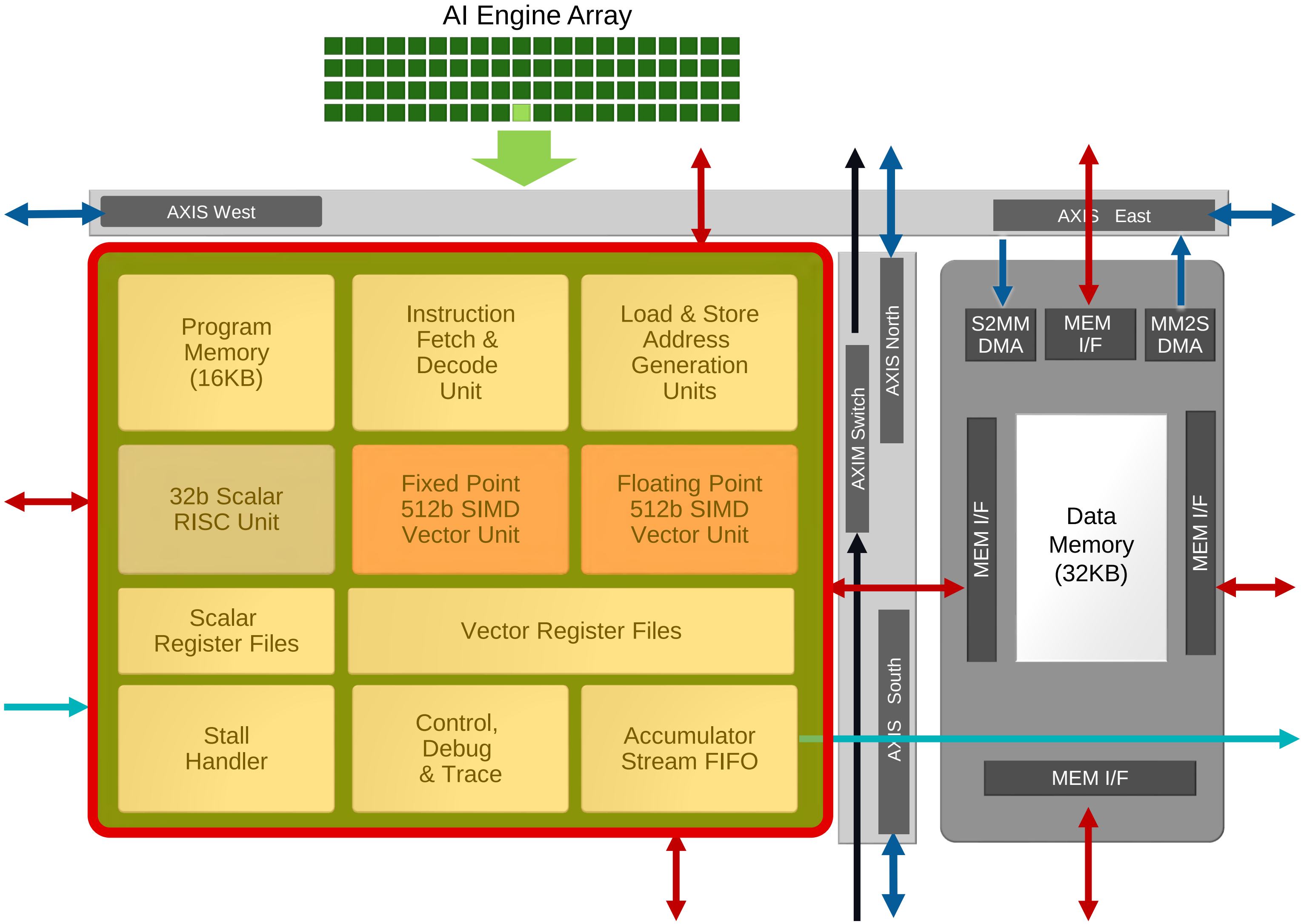




AI Engine



AI Engine: Tile-based Architecture

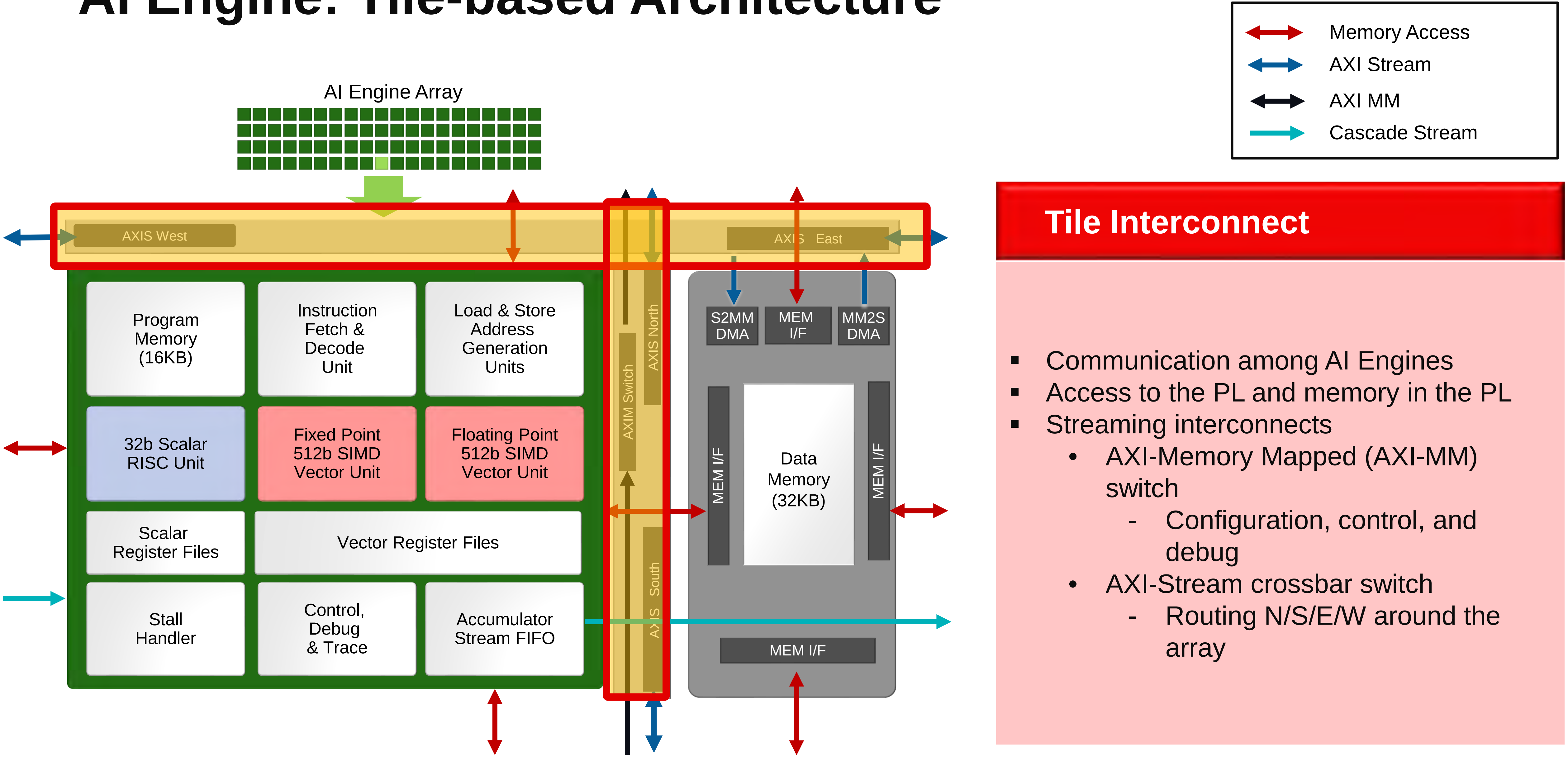


- ⟷ Memory Access
- ⟷ AXI Stream
- ⟷ AXI MM
- → Cascade Stream

AI Engine

- VLIW processor with 512-bit SIMD vector units
  - Application-specific vector extensions
    - Optimized for DSP, 5G, AI, ML, etc.
  - Hardened in 7nm @ 1 GHz
  - Software programmable (e.g., C/C++)
- Debug/trace/profile functionality
  - Debug using memory-mapped AXI4 interface
  - Connect to PMC via JTAG or HSDP

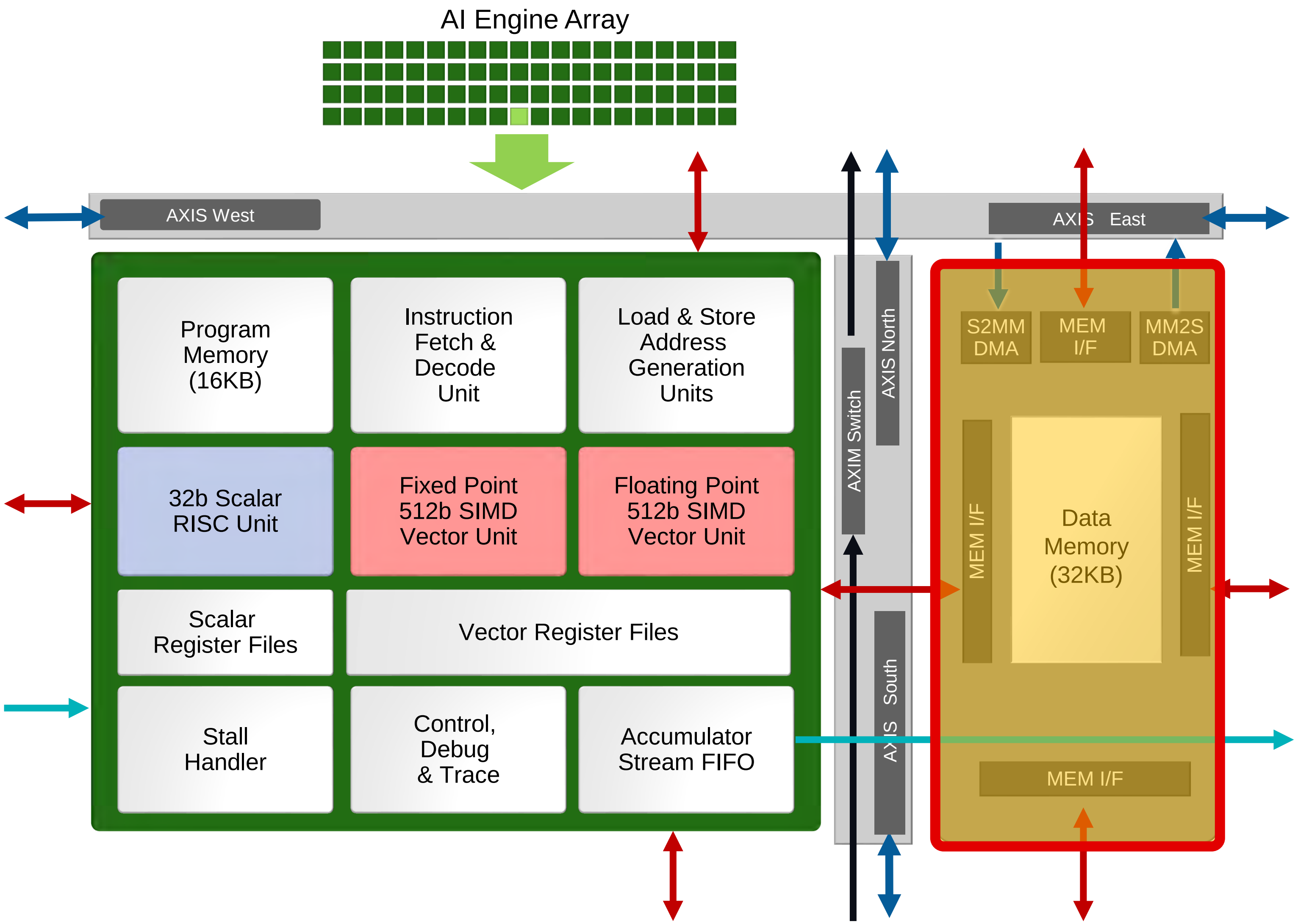
AI Engine: Tile-based Architecture

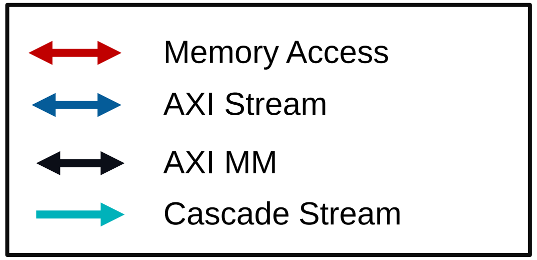


Tile Interconnect

- Communication among AI Engines
- Access to the PL and memory in the PL
- Streaming interconnects
  - AXI-Memory Mapped (AXI-MM) switch
    - Configuration, control, and debug
  - AXI-Stream crossbar switch
    - Routing N/S/E/W around the array

AI Engine: Tile-based Architecture





AI Engine Memory

- **Local memory (32KB)**
  - 8 memory banks, memory interface, DMA (incoming/outgoing), and locks
- 128KB direct core memory access
  - 32KB local
  - 32KB north, 32KB south
  - 32KB east or west (depending on row)
- **Data mover**
  - Non-neighbor data communication
- **Cascade interface**
  - Partial results to next core

XILINX.

ANIMATED

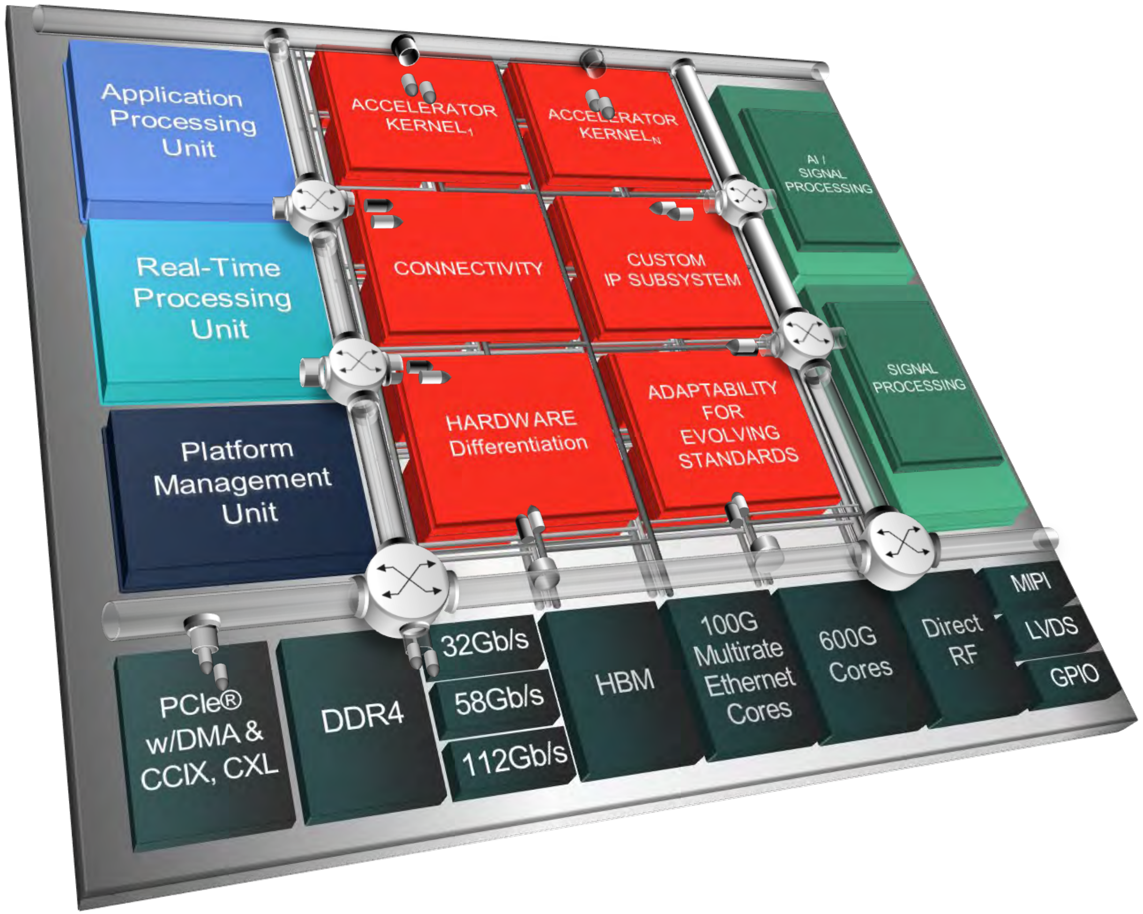
Programmable NoC: Bridging Engines & Hard IP

High-bandwidth, Terabit Programmable NoC

- Massive-bandwidth, programmable network on chip (NoC) that abstracts away the hardware and connects the engines and hard IP together
- Programmed for different data flows between high-speed I/Os and engines

Eases IP and Kernel Placement

- Ability to intuitively compartmentalize the kernels with easy entry and exit points
- Ability to connect any master to any slave
- Global address maps are based on the NoC interconnect
- Configurable NoC is an AXI4-based network to route:
  - High bandwidth
  - Real time
  - Low latency
- Extends in both horizontal and vertical directions to the edges of the device
- Hardened connectivity to the integrated memory controllers and PCIe core, enabling immediate connection to external DDR

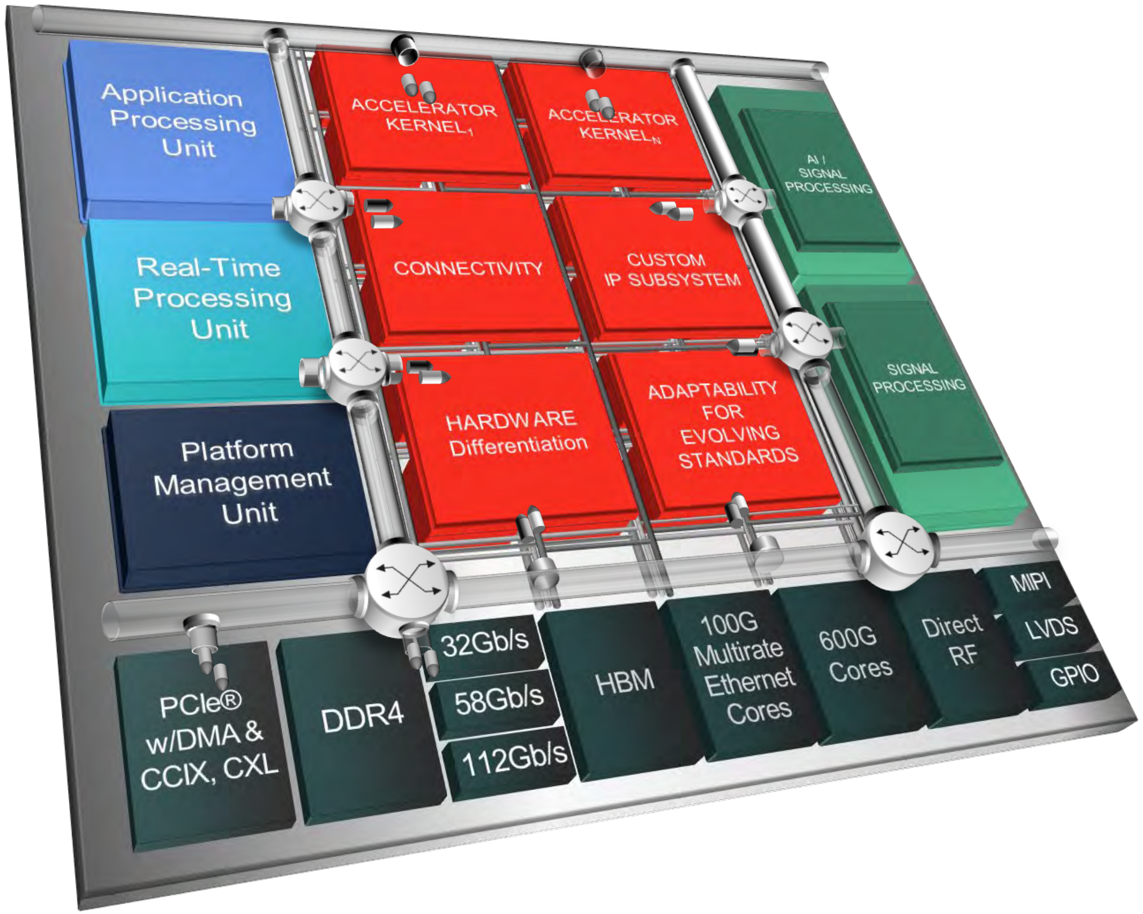


ANIMATED

Programmable NoC: Bridging Engines & Hard IP

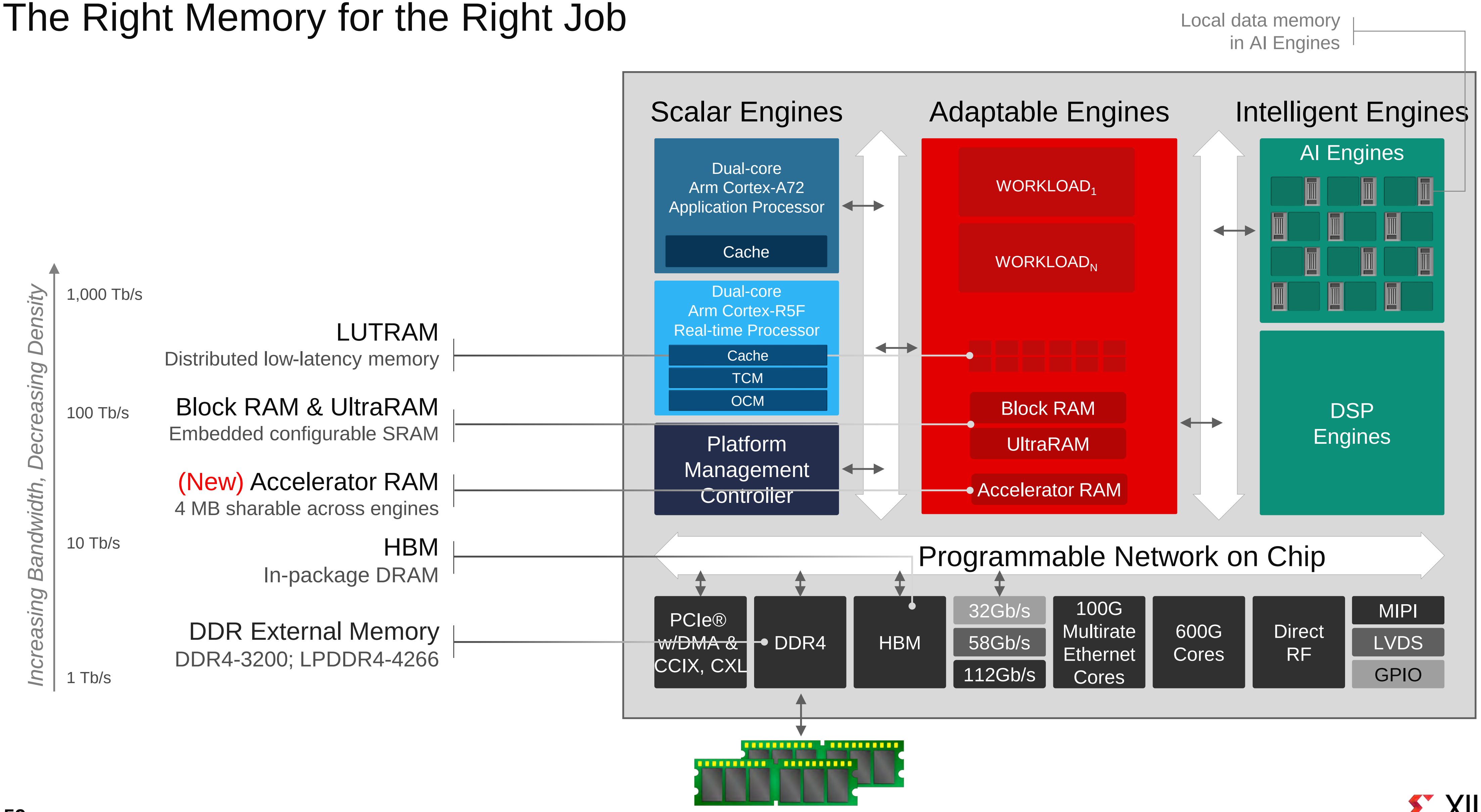
Programming Framework

- NoC is a GUI experience
- Enables an intuitive top-down system design with a straightforward compiler flow



Adaptable Memory Hierarchy

The Right Memory for the Right Job



Versal ACAP Boards & Kits

Prime Series

VMK180
General Purpose Development Kit
(VM1802 Silicon)

Prime
Series

Available Now
https://www.xilinx.com/products/boards-and-kits/vmk180.html

XILINX.

AI Core Series

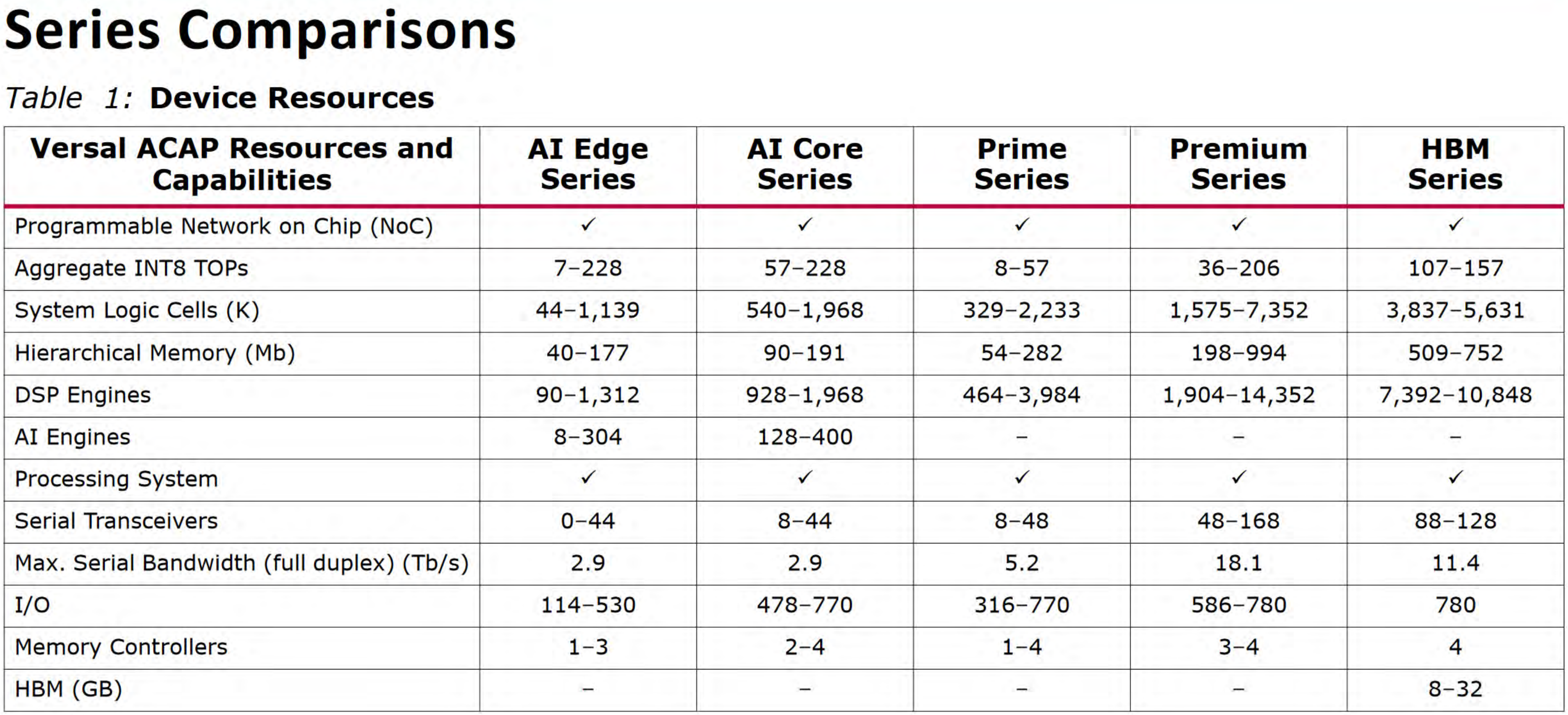
VCK190
AI Engine Development Kit
(VC1902 Silicon)

AI Core
Series

Available Now
https://www.xilinx.com/products/boards-and-kits/vck190.html

XILINX.

Versal Devices (From DS950: Versal Architecture and Product Data Sheet)



Series Comparisons

Table 1: **Device Resources**

Versal ACAP Resources and Capabilities	AI Edge Series	AI Core Series	Prime Series	Premium Series	HBM Series
Programmable Network on Chip (NoC)	✓	✓	✓	✓	✓
Aggregate INT8 TOPs	7–228	57–228	8–57	36–206	107–157
System Logic Cells (K)	44–1,139	540–1,968	329–2,233	1,575–7,352	3,837–5,631
Hierarchical Memory (Mb)	40–177	90–191	54–282	198–994	509–752
DSP Engines	90–1,312	928–1,968	464–3,984	1,904–14,352	7,392–10,848
AI Engines	8–304	128–400	–	–	–
Processing System	✓	✓	✓	✓	✓
Serial Transceivers	0–44	8–44	8–48	48–168	88–128
Max. Serial Bandwidth (full duplex) (Tb/s)	2.9	2.9	5.2	18.1	11.4
I/O	114–530	478–770	316–770	586–780	780
Memory Controllers	1–3	2–4	1–4	3–4	4
HBM (GB)	–	–	–	–	8–32

Table 4: **Versal AI Core Series**

	VC1352	VC1502	VC1702	VC1802	VC1902	VC2602	VC2802
AI Engines	128	198	304	300	400	0	0
AI Engines-ML	0	0	0	0	0	152	304
AIE/AIE-ML Data Memory (Mb)	32	50	76	75	100	76	152
AIE-ML Shared Memory (Mb)	0	0	0	0	0	304	304
DSP Engines	928	1,032	1,312	1,600	1,968	984	1,312
System Logic Cells	539,840	814,520	981,120	1,585,938	1,968,400	820,313	1,139,040
CLB Flip-Flops	493,568	744,704	897,024	1,450,000	1,799,680	750,000	1,041,408
LUTs	246,784	372,352	448,512	725,000	899,840	375,000	520,704
Distributed RAM (Mb)	7.5	11.3	13.7	22.1	27.5	11.4	15.9
Block RAM Blocks	441	848	954	800	967	476	600
Block RAM (Mb)	15.5	29.8	33.5	28.1	34.0	16.7	21.1
UltraRAM Blocks	209	390	462	325	463	224	264
UltraRAM (Mb)	58.8	109.7	129.9	91.4	130.2	63.0	74.3
Accelerator RAM (Mb)	32	0	0	0	0	0	0
APU	Dual-core Arm Cortex-A72; 48KB/32KB L1 Cache w/ parity and ECC; 1MB L2 Cache w/ ECC						
RPU	Dual-core Arm Cortex-R5F; 32KB/32KB L1 Cache, and TCM w/ECC						
Memory	256KB On-Chip Memory w/ECC						
Connectivity	Ethernet (x2); UART (x2); CAN-FD (x2); USB 2.0 (x1); SPI (x2); I2C (x2)						
NoC Master / Slave Ports	10	21	21	28	28	21	21
DDR Bus Width	128	192	192	256	256	192	192
DDR Memory Controllers	2	3	3	4	4	3	3
PCIe w/DMA & CCIX (CPM4)	–	1 x Gen4x16, CCIX	1 x Gen4x16, CCIX	1 x Gen4x16, CCIX	1 x Gen4x16, CCIX	–	–
PCIe w/DMA & CCIX (CPM5)	–	–	–	–	–	2 x Gen5x8, CCIX	2 x Gen5x8, CCIX
PCIe (PLPCIE4)	1 x Gen4x8	4 x Gen4x8	4 x Gen4x8	4 x Gen4x8	4 x Gen4x8	–	–
PCIe (PLPCIE5)	–	–	–	–	–	4 x Gen5x4	4 x Gen5x4
100G Multirate Ethernet MAC	1	3	4	4	4	2	2
XPIO	378	486	486	648	648	486	486
HDIO	44	22	44	44	44	44	44
GTY Transceivers (32.75Gb/s)	0	32	44	44	44	0	0
GTYP Transceivers (32.75Gb/s)	8	0	0	0	0	32(1)	32(1)
Video Decoder Engines (VDEs)	–	–	–	–	–	2	4

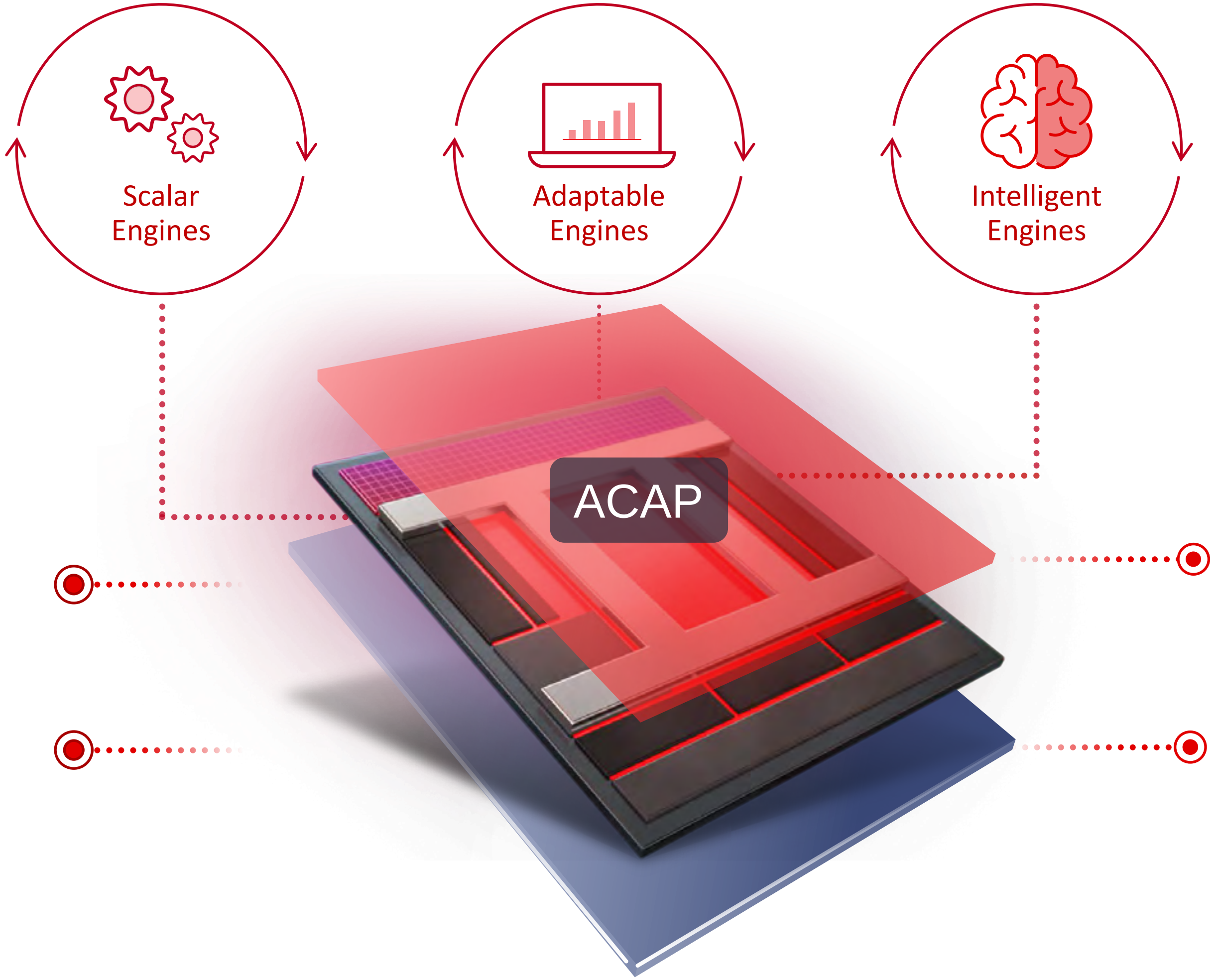
Table 8: **Versal Premium Series**

	VP1102	VP1202	VP1402	VP1502	VP1552	VP1702	VP1802
System Logic Cells	1,574,720	1,969,240	2,233,280	3,763,480	3,836,840	5,557,720	7,351,960
CLB Flip-Flops	1,439,744	1,800,448	2,041,856	3,440,896	3,507,968	5,081,344	6,721,792
LUTs	719,872	900,224	1,020,928	1,720,448	1,753,984	2,540,672	3,360,896
Distributed RAM (Mb)	22	27	31	53	54	78	103
Block RAM Blocks	1,405	1,341	1,981	2,541	2,541	3,741	4,941
Block RAM (Mb)	49	47	70	89	89	132	174
UltraRAM Blocks	453	677	645	1,301	1,301	1,925	2,549
UltraRAM (Mb)	127	190	181	366	366	541	717
DSP Engines	1,904	3,984	2,672	7,440	7,392	10,896	14,352
APU	Dual-core Arm Cortex-A72; 48KB/32KB L1 Cache w/ parity and ECC; 1MB L2 Cache w/ ECC						
RPU	Dual-core Arm Cortex-R5F; 32KB/32KB L1 Cache, and TCM w/ECC						
Memory	256KB On-Chip Memory w/ECC						
Connectivity	Ethernet (x2); UART (x2); CAN-FD (x2); USB 2.0 (x1); SPI (x2); I2C (x2)						
NoC Master / Slave Ports	30	28	42	52	52	76	100
DDR Bus Width	192	256	192	256	256	256	256
DDR Memory Controllers	3	4	3	4	4	4	4
PCIe w/DMA & CCIX (CPM5)	–	2 x Gen5x8, CCIX	–	2 x Gen5x8, CCIX	2 x Gen5x8, CCIX	2 x Gen5x8, CCIX	2 x Gen5x8, CCIX
PCIe w/CXL[2] (PLPCIE5)	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	2 x Gen5x4	8 x Gen5x4	2 x Gen5x4	2 x Gen5x4
Multirate Ethernet MAC	6	2	6	4	4	6	8
600G Ethernet MAC	4	1	8	3	1	5	7
600G Interlaken	2	–	2	1	–	2	3
400G HSC Engine	3	1	5	2	2	3	4
XPIO	486	702	486	702	702	702	648
HDIO	44	–	44	–	–	–	–
GTYP Transceivers 32.75Gb/s	8	28[3]	8	28[3]	68[3]	28[3]	28[3]
GTM Transceivers[1] 58Gb/s (112Gb/s)	64 (32)	20 (10)	96 (48)	60 (30)	20 (10)	100 (50)	140 (70)

Versal ACAP: Design Tool Flow

Adaptive Compute Acceleration Platform (ACAP)

COMPUTE ACCELERATION



Customizable memory hierarchies

NoC to move data

ADAPTIVE

Future Proof for New Algorithms

Support for Common Standards

PLATFORM

Single Heterogeneous Platform for HW/SW Engineers

Frameworks, Libraries, SW, Runtime Stack

Enabling Data Scientists, Software Developers, Hardware Developers

Different Capabilities Means Different Tools

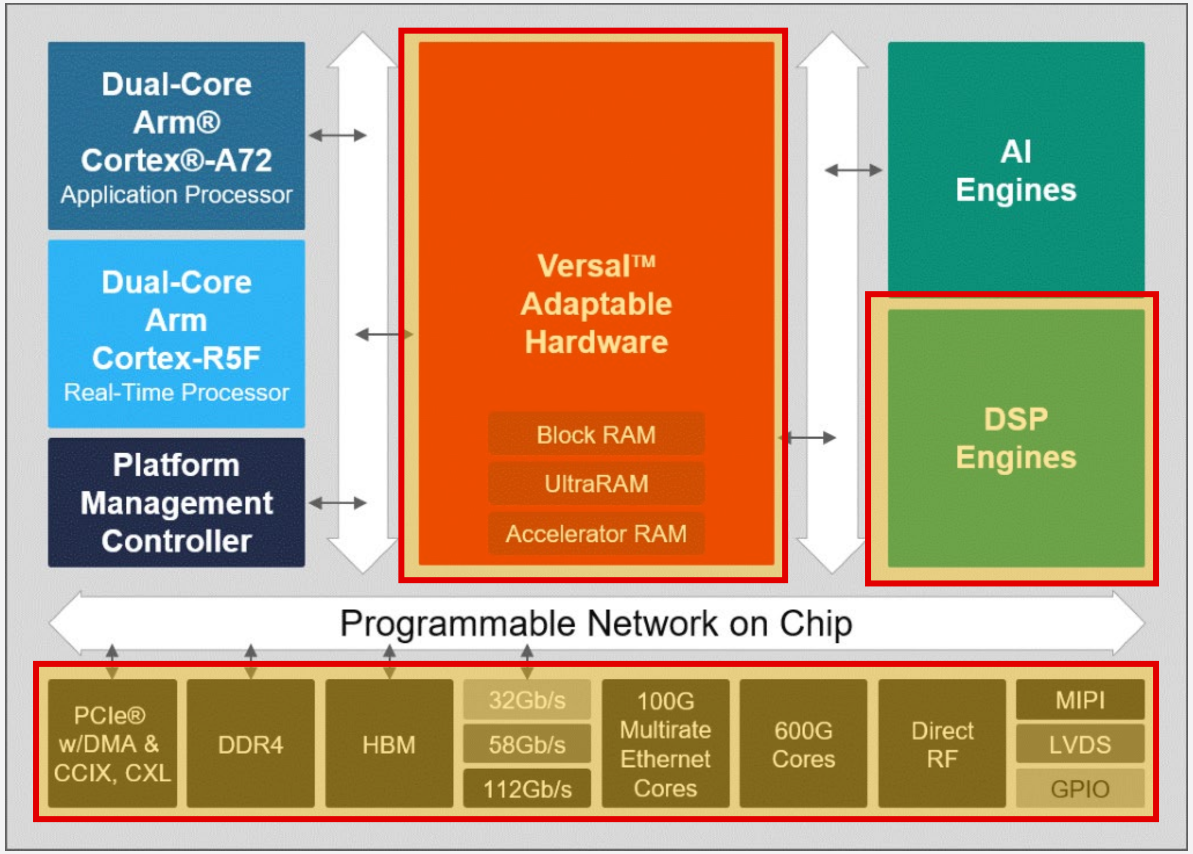
Tools for Development

PL and Its Associated Resources

VIVADO

XILINX VITIS™

- Vivado® Design Suite for manual design
- Vitis™ HLS tools for hardware system creation

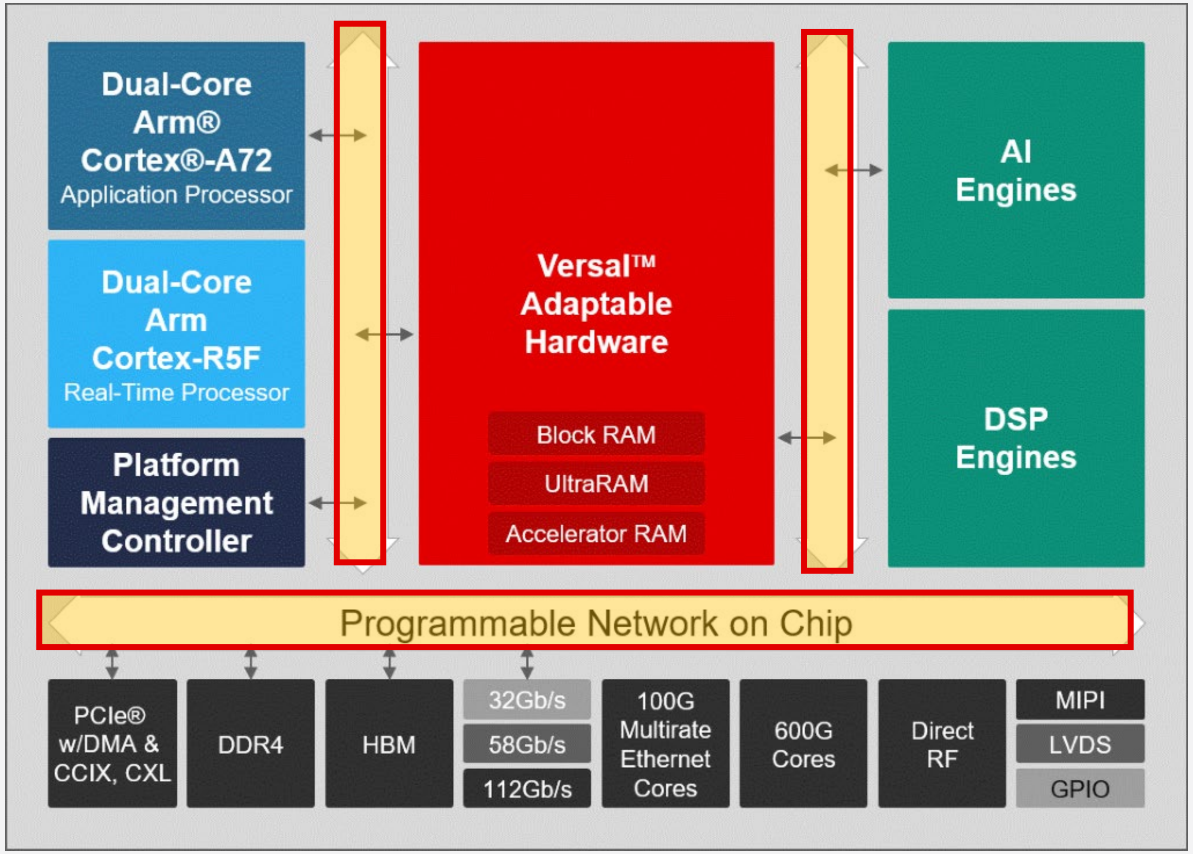


Different Capabilities Means Different Tools

Tools for Development

NoC Resources

- Vivado IPI tool used for connecting NoC to various blocks:
  - Control, Interconnects, and Processing System (CIPS)
  - Memory and memory controllers
  - Communication cores



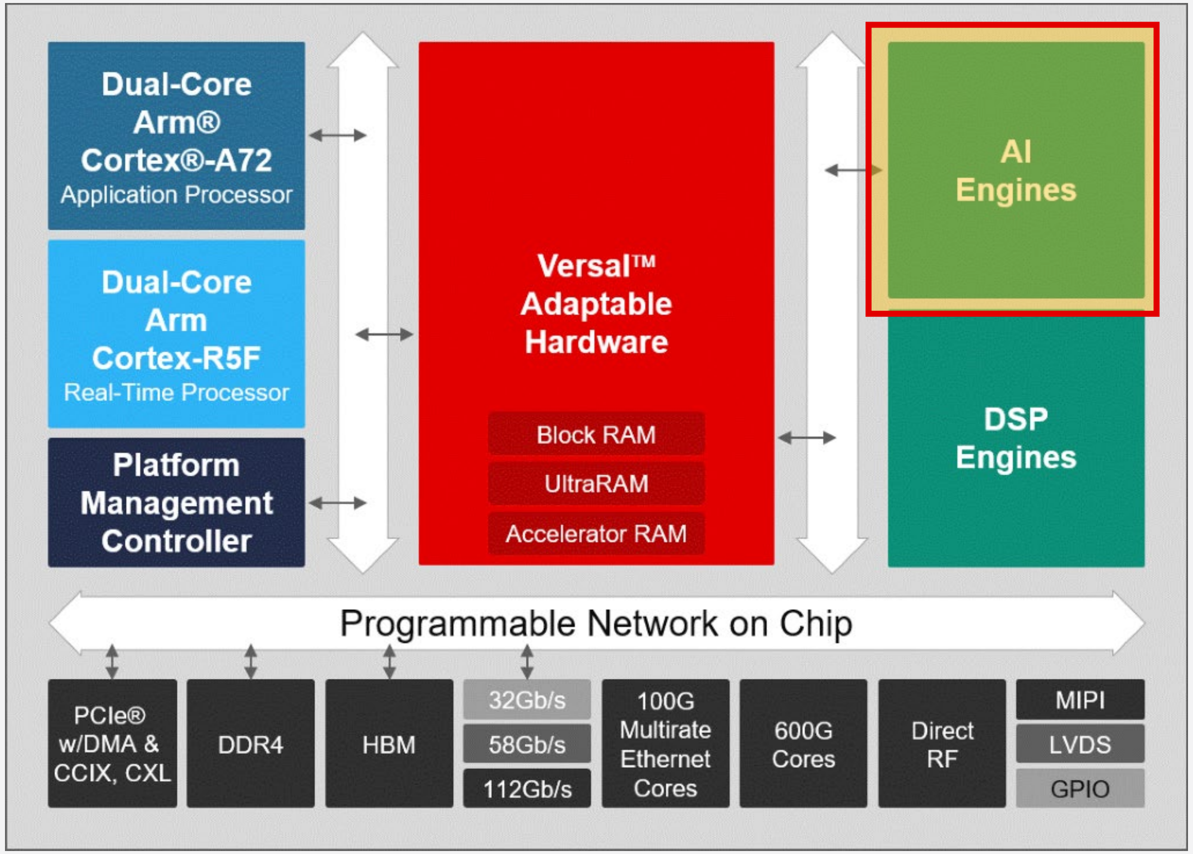
Different Capabilities Means Different Tools

Tools for Development

Adaptable Intelligent Engines

- Array of cores optimized for DSP and ML requires new tools

aiecompiler

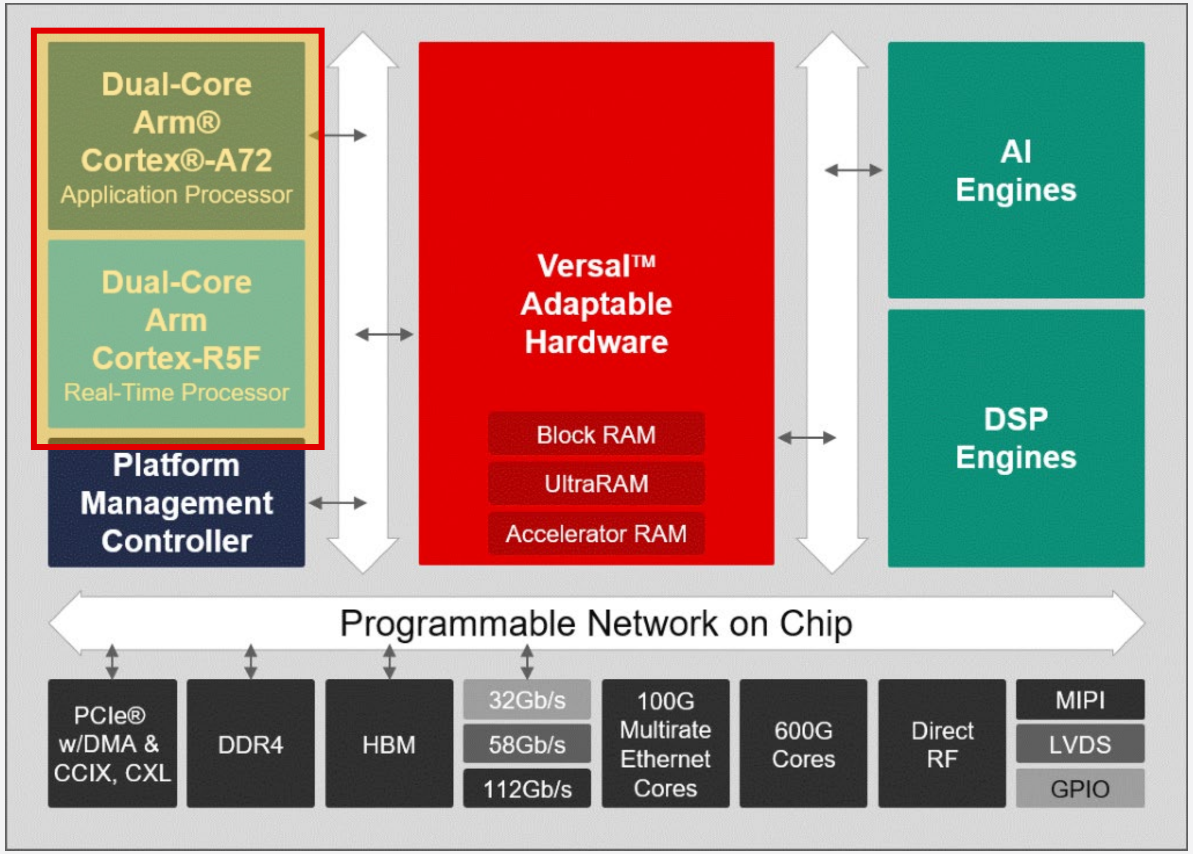


Different Capabilities Means Different Tools

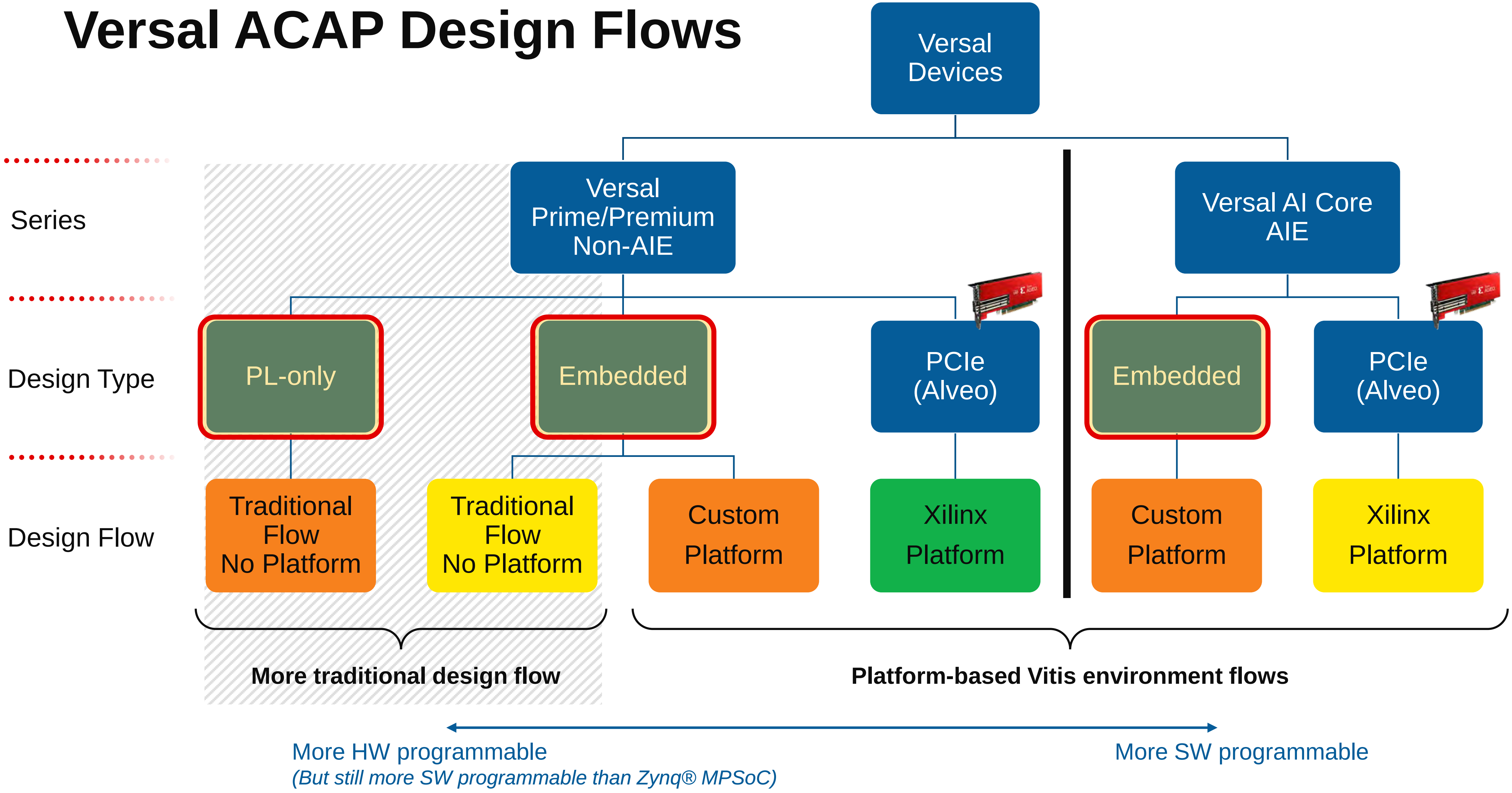
Tools for Development
Scalar Processors

XILINX VITIS™

- Vitis IDE supports Arm® RISC processors
  - Enables bootable image creation



Versal ACAP Design Flows



Versal Devices

Series

Versal Prime/Premium Non-AIE

Versal AI Core AIE

Design Type

PL-only

Embedded

PCIe (Alveo)

Embedded

PCIe (Alveo)

Design Flow

Traditional Flow No Platform

Traditional Flow No Platform

Custom Platform

Xilinx Platform

Custom Platform

Xilinx Platform

More traditional design flow

Platform-based Vitis environment flows

More HW programmable

(But still more SW programmable than Zynq® MPSoC)

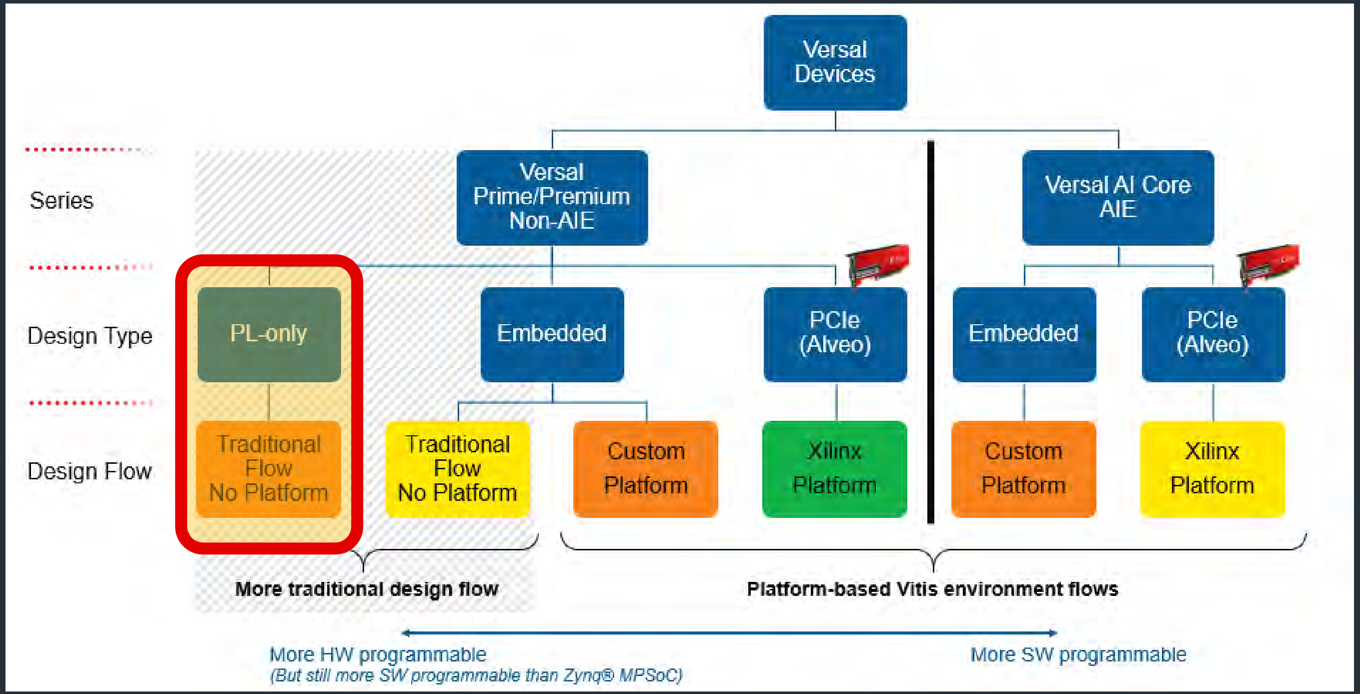
More SW programmable

Versal ACAP Design Flows

Versal

PL-only System

- Hardware-only design
- Created using the traditional design flow using the Vivado Design Suite, which generates a PDI to program the Versal device
- Comprises of a programmable logic (RTL or IP) implementation
- Operates autonomously and the device programming is handled by the Versal ACAP PMC



Versal ACAP Design Flows

Versal

Embedded System

Comprises of:

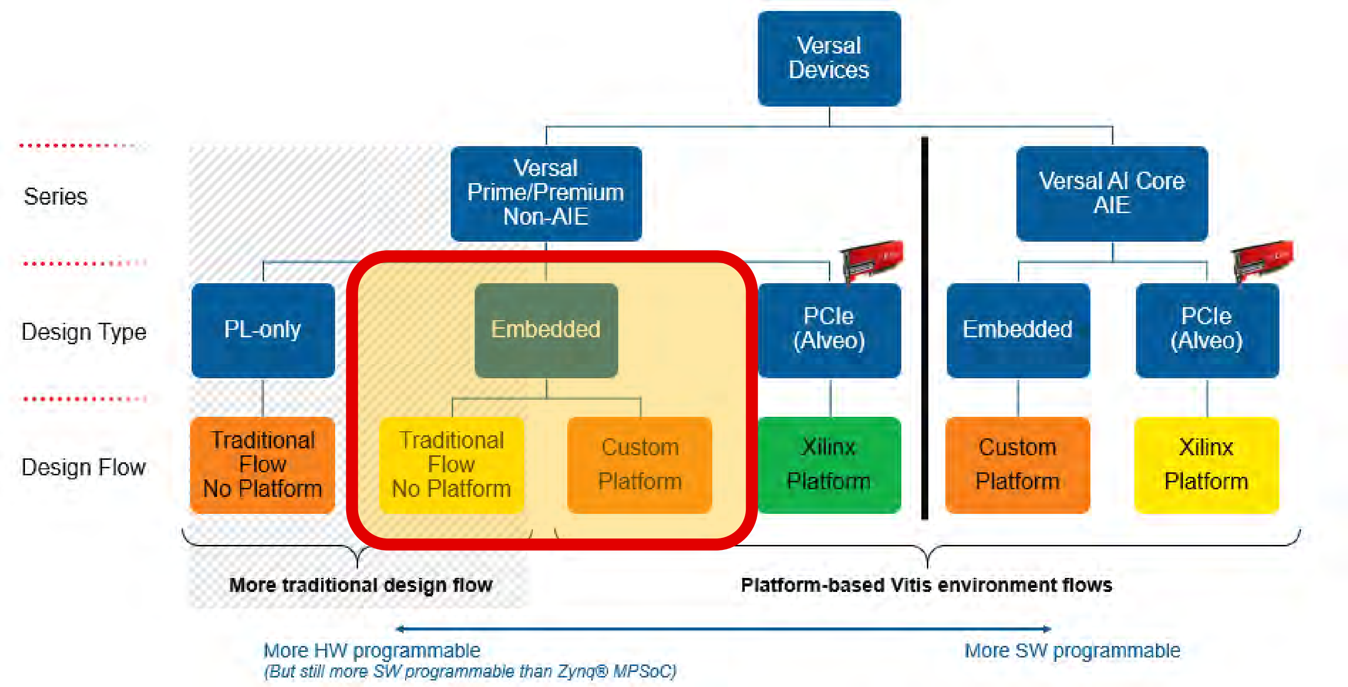
- Embedded processor
- Acceleration logic built into the traditional PL

For the Versal ACAP, the embedded compute system comprises of:

- Arm® Cortex™-A72 and Cortex-R5F processors

Use models can be:

- Sophisticated embedded software stack
- Simple bare-metal stack



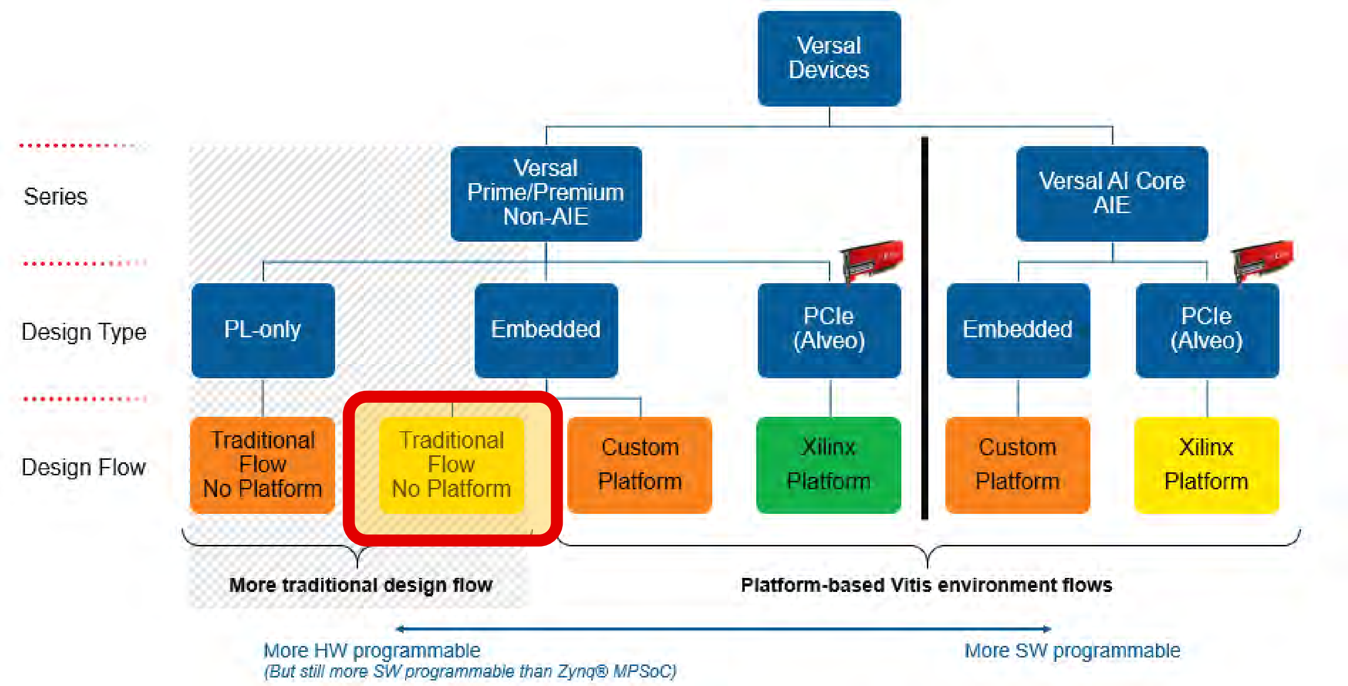
Versal ACAP Design Flows

Versal

Embedded System

Traditional flow:

- No platform
- Hardware design considered a *fixed platform*
  - Vivado tools create the programmable logic on the device
  - Vitis tools cannot modify the PL

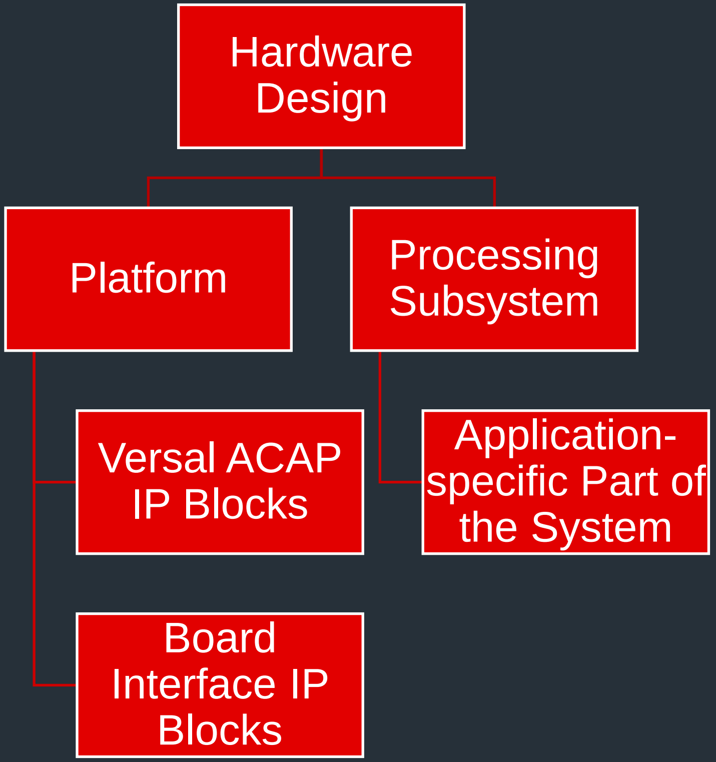


Versal ACAP Design Flows

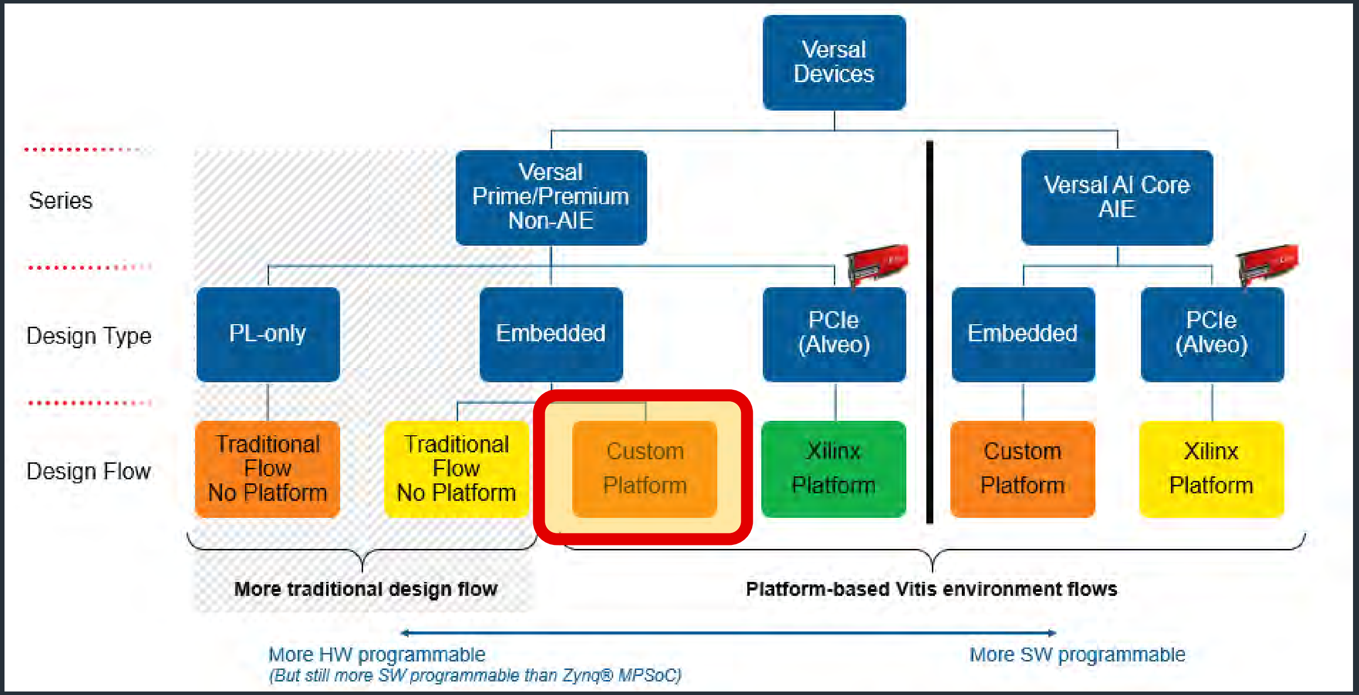
Versal

Embedded System

Platform-based flow:



Hardware Design
Platform
Processing Subsystem
Versal ACAP IP Blocks
Application-specific Part of the System
Board Interface IP Blocks



Versal ACAP Design Flows

Versal

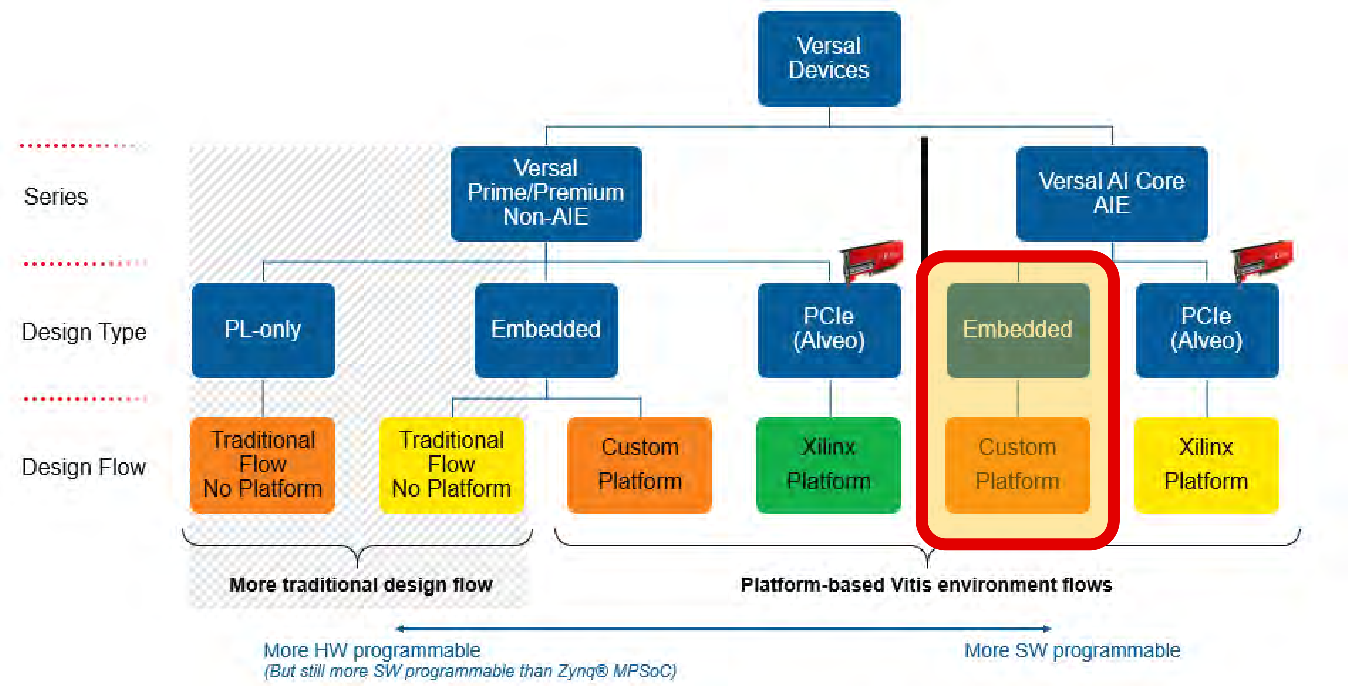
Embedded AI Engine System

Comprises of:

- Embedded processor
- Acceleration logic:
  - Traditional PL
  - AI Engines

For the Versal ACAP:

- Embedded processing system is running on the Arm Cortex-A72 and Cortex-R5F processors
- Hardware content in the PL and algorithmic content in the AI Engines
- Created using platform-based design flow



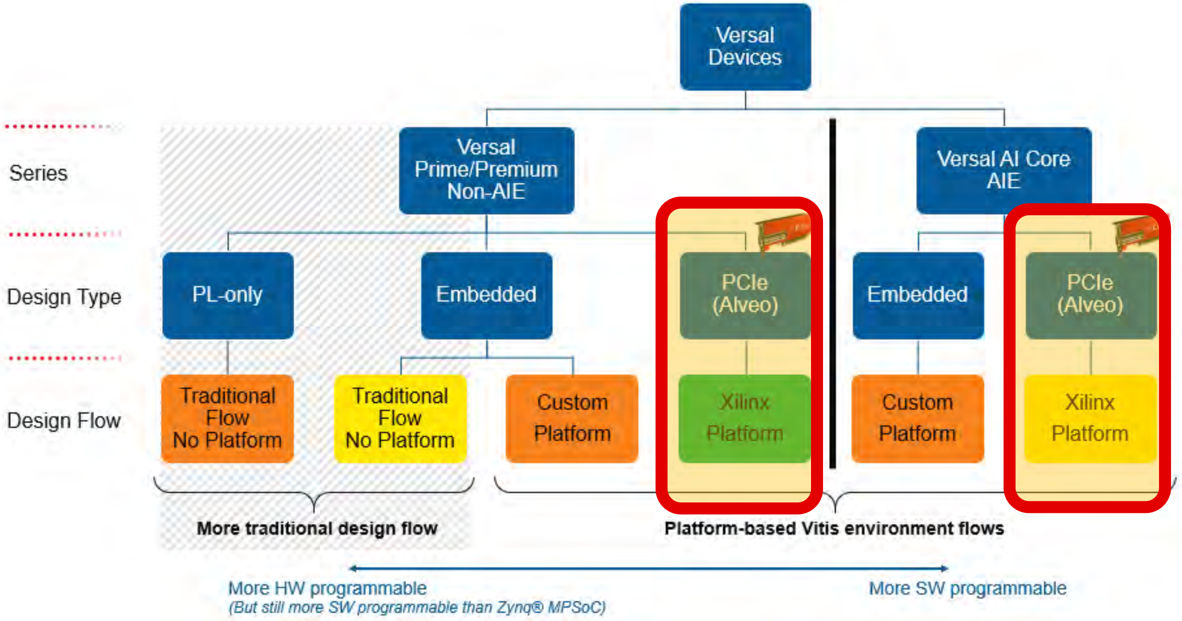
XILINX.

Versal ACAP Design Flows

Xilinx recommends a *parallel development process*

Application team works on the subsystem using a Xilinx pre-built platform

Platform team works independently on bringing up the custom platform

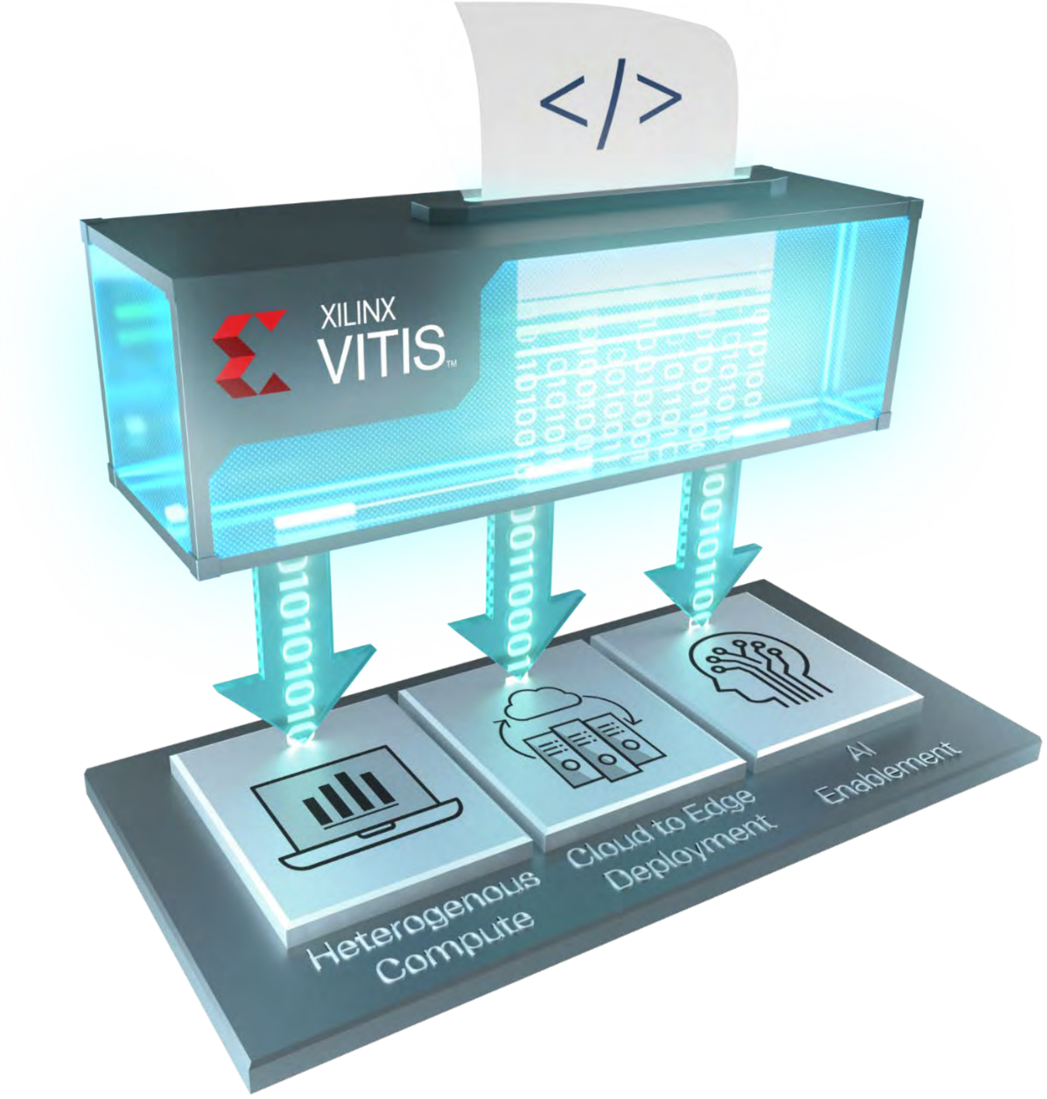


Vitis Unified Software Platform

XILINX VITIS™

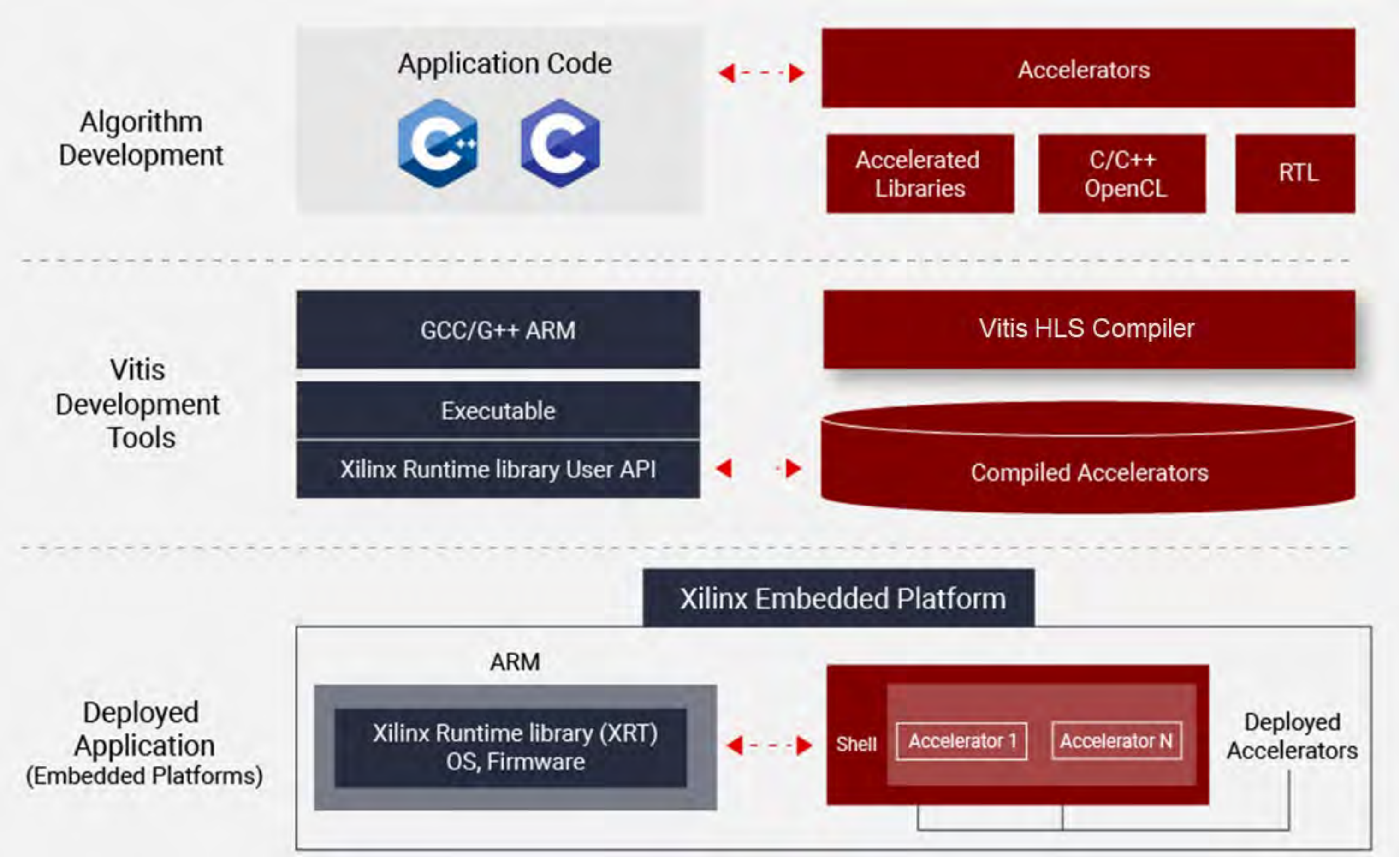
Unified Software Platform

- Adaptive computing
- Easy programmability of Versal ACAP hardware
- Support for open standards
- Software and AI
  - Heterogeneous environment
  - Portability: edge to cloud
  - AI toolkit



Vitis Development Environment for Embedded Systems

- Run applications on an embedded processor platform
- OpenCL APIs and XRT



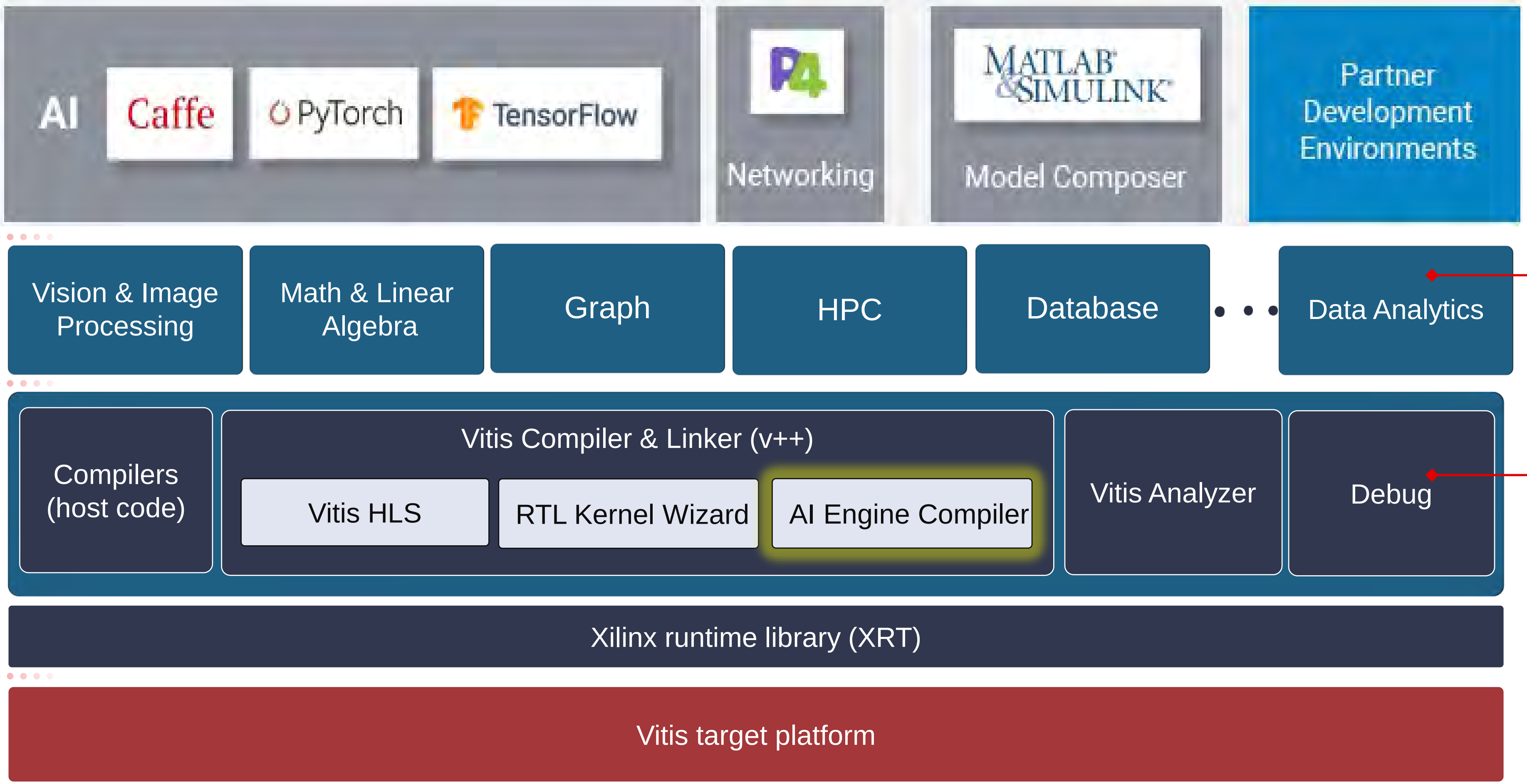
Debug
Profile
Optimize

GUI and Command-line Support
Eclipse Open-source Standard

Vitis Unified Solution Stack

For Heterogenous Compute, Edge to Cloud

Vitis IDE: Environment for development of applications for Xilinx embedded processors



Domain-specific development environments

Vitis accelerated libraries

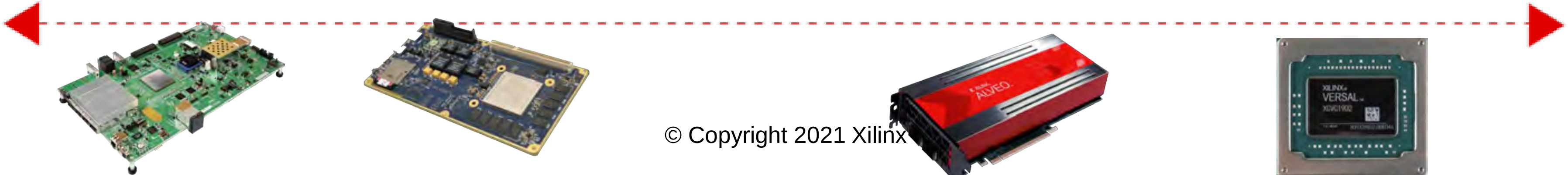
Vitis core development kit

Vitis Target Platform

- Optimized FPGA acceleration
- Libraries for math and domain-specific applications

- Build code
- Fix problems
- Analyze

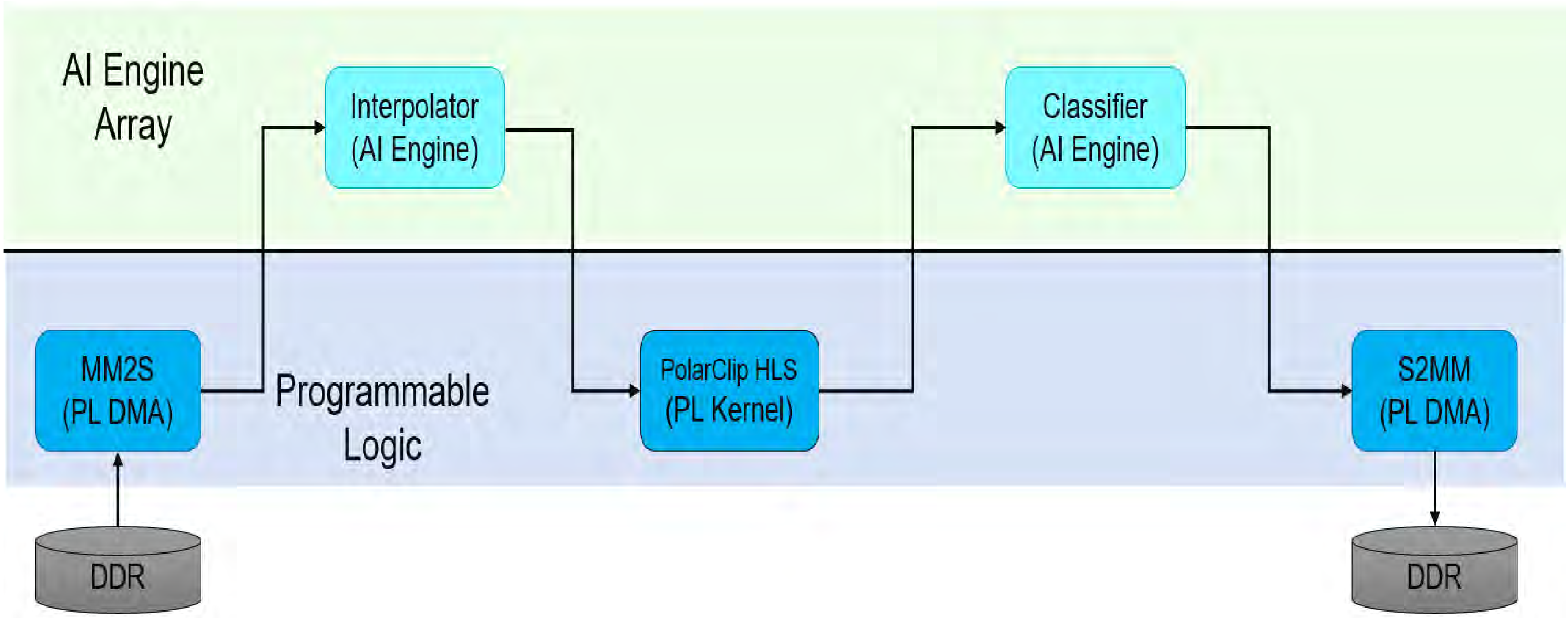
- API and drivers



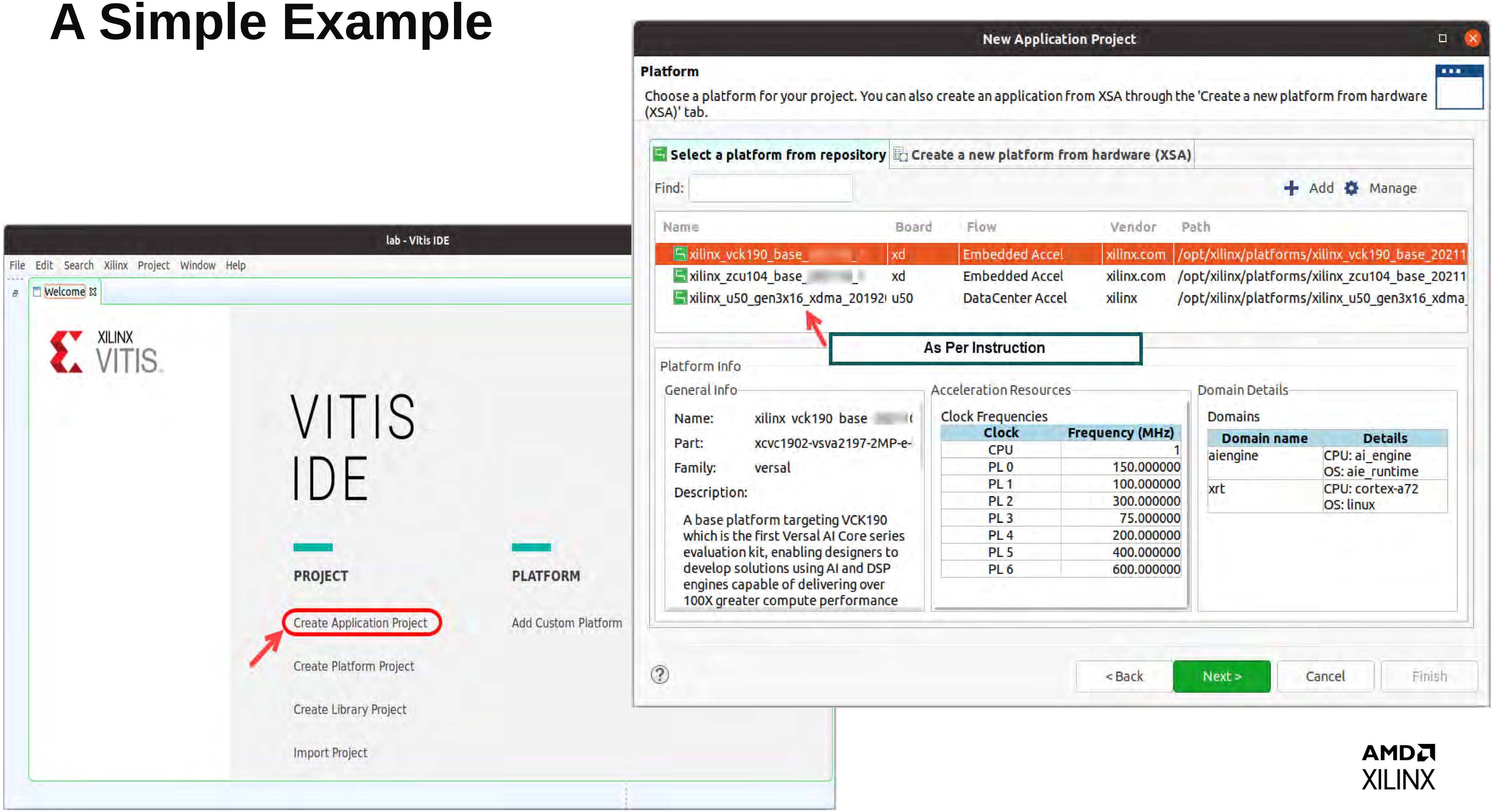
XILINX®

Thank You

A Simple Example



A Simple Example



Platform

Choose a platform for your project. You can also create an application from XSA through the 'Create a new platform from hardware (XSA)' tab.

Name	Board	Flow	Vendor	Path
xilinx_vck190_base_[redacted]	xd	Embedded Accel	xilinx.com	/opt/xilinx/platforms/xilinx_vck190_base_20211
xilinx_zcu104_base_[redacted]	xd	Embedded Accel	xilinx.com	/opt/xilinx/platforms/xilinx_zcu104_base_20211
xilinx_u50_gen3x16_xdma_20192	u50	DataCenter Accel	xilinx	/opt/xilinx/platforms/xilinx_u50_gen3x16_xdma_

As Per Instruction

Platform Info

General Info

Name: xilinx_vck190_base_[redacted]

Part: xcvc1902-vsva2197-2MP-e-

Family: versal

Description:

A base platform targeting VCK190 which is the first Versal AI Core series evaluation kit, enabling designers to develop solutions using AI and DSP engines capable of delivering over 100X greater compute performance

Acceleration Resources

Clock Frequencies

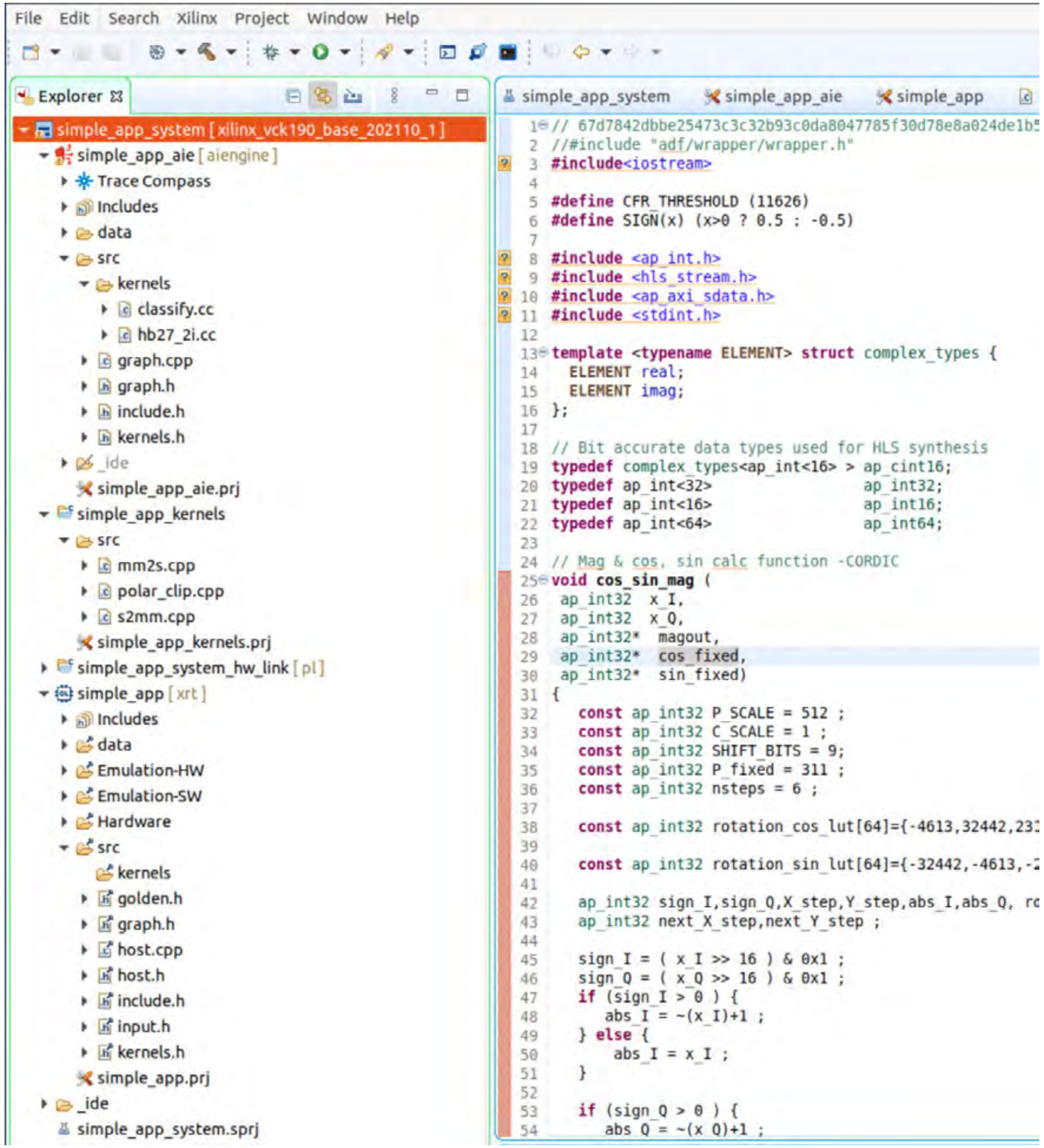
Clock	Frequency (MHz)
CPU	1
PL 0	150.000000
PL 1	100.000000
PL 2	300.000000
PL 3	75.000000
PL 4	200.000000
PL 5	400.000000
PL 6	600.000000

Domain Details

Domains

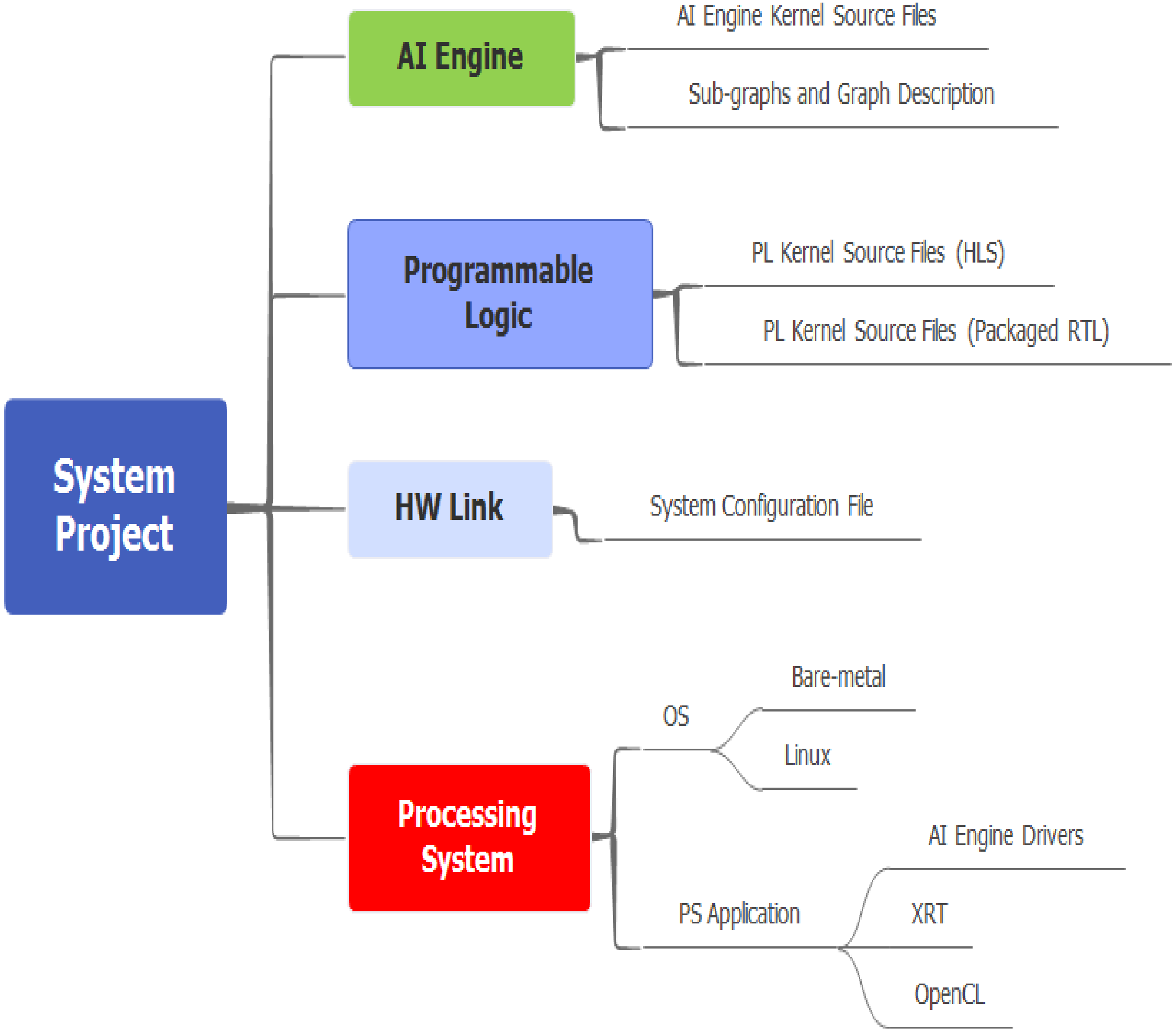
Domain name	Details
aiengine	CPU: ai_engine OS: aie_runtime
xrt	CPU: cortex-a72 OS: linux

A Simple Example: The Project



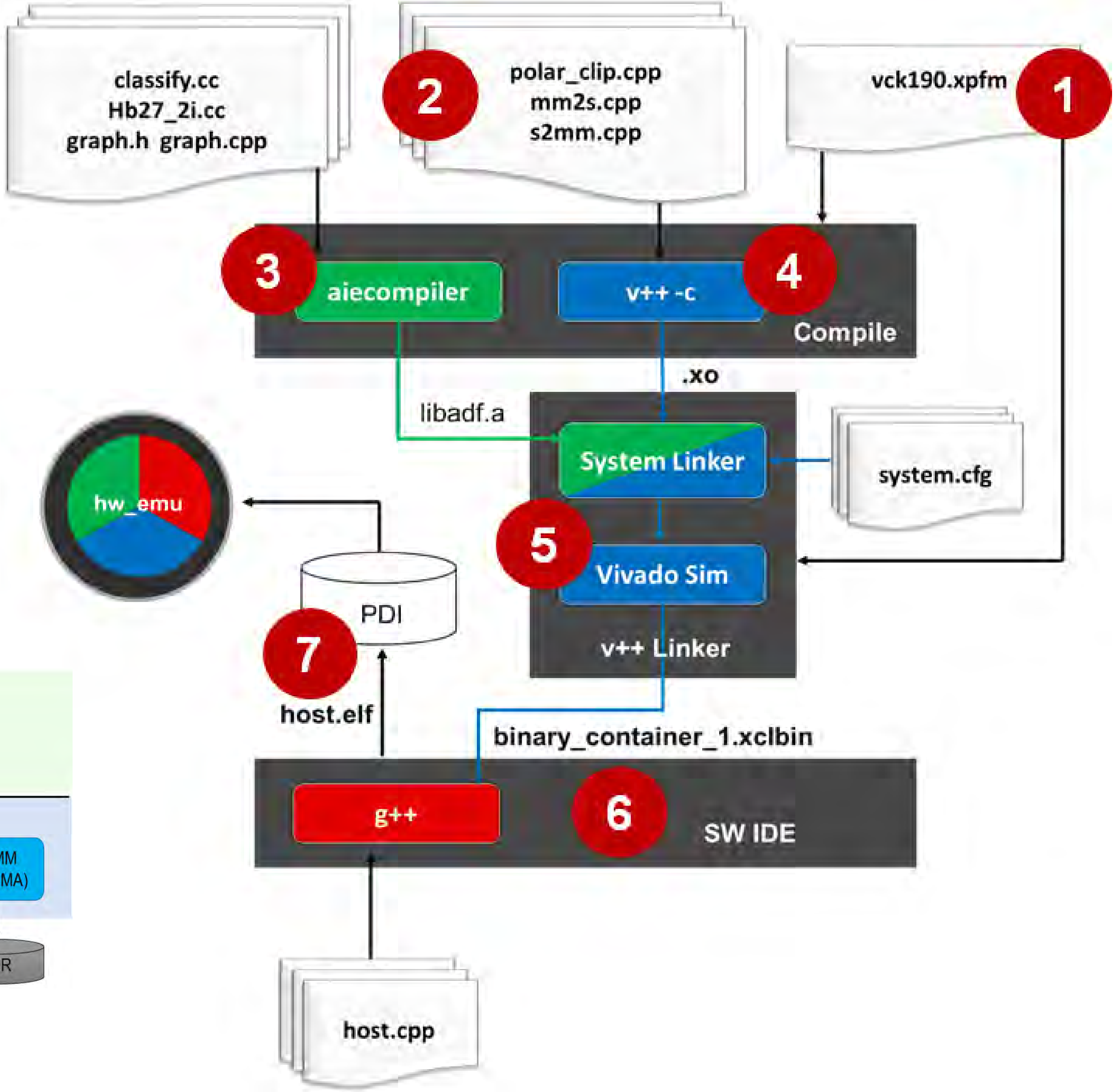
- simple_app_system [xilinx_vck190_base_202110_1]
  - simple_app_aie [aiengine]
    - Trace Compass
    - Includes
    - data
    - src
      - kernels
        - classify.cc
        - hb27_2i.cc
      - graph.cpp
      - graph.h
      - include.h
      - kernels.h
    - _ide
    - simple_app_aie.prj
  - simple_app_kernels
    - src
      - mm2s.cpp
      - polar_clip.cpp
      - s2mm.cpp
    - simple_app_kernels.prj
  - simple_app_system_hw_link [pl]
  - simple_app [xrt]
    - Includes
    - data
    - Emulation-HW
    - Emulation-SW
    - Hardware
    - src
      - kernels
      - golden.h
      - graph.h
      - host.cpp
      - host.h
      - include.h
      - input.h
      - kernels.h
    - simple_app.prj
  - _ide
  - simple_app_system.sprj

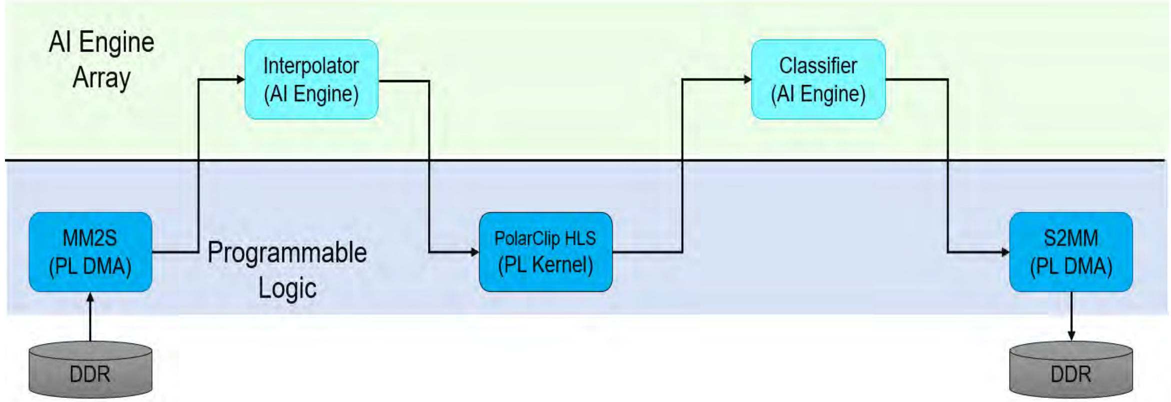
```
// 67d7842dbbe25473c3c32b93c0da8047785f30d78e8a024de1b5
//#include "adf/wrapper/wrapper.h"
#include<iostream>

#define CFR_THRESHOLD (11626)
#define SIGN(x) (x>0 ? 0.5 : -0.5)

#include <ap_int.h>
#include <hls_stream.h>
#include <ap_axi_sdata.h>
#include <stdint.h>

template <typename ELEMENT> struct complex_types {
  ELEMENT real;
  ELEMENT imag;
};

// Bit accurate data types used for HLS synthesis
typedef complex_types<ap_int<16> > ap_cint16;
typedef ap_int<32>                 ap_int32;
typedef ap_int<16>                 ap_int16;
typedef ap_int<64>                 ap_int64;

// Mag & cos, sin calc function -CORDIC
void cos_sin_mag (
 ap_int32  x_I,
 ap_int32  x_Q,
 ap_int32*  magout,
 ap_int32*  cos_fixed,
 ap_int32*  sin_fixed)
{
   const ap_int32 P_SCALE = 512 ;
   const ap_int32 C_SCALE = 1 ;
   const ap_int32 SHIFT_BITS = 9;
   const ap_int32 P_fixed = 311 ;
   const ap_int32 nsteps = 6 ;

   const ap_int32 rotation_cos_lut[64]={-4613,32442,23

   const ap_int32 rotation_sin_lut[64]={-32442,-4613,-

   ap_int32 sign_I,sign_Q,X_step,Y_step,abs_I,abs_Q, rc
   ap_int32 next_X_step,next_Y_step ;

   sign_I = ( x_I >> 16 ) & 0x1 ;
   sign_Q = ( x_Q >> 16 ) & 0x1 ;
   if (sign_I > 0 ) {
      abs_I = ~(x_I)+1 ;
   } else {
       abs_I = x_I ;
   }

   if (sign_Q > 0 ) {
      abs_Q = ~(x_Q)+1 ;
```



System Project
- AI Engine
  - AI Engine Kernel Source Files
  - Sub-graphs and Graph Description
- Programmable Logic
  - PL Kernel Source Files (HLS)
  - PL Kernel Source Files (Packaged RTL)
- HW Link
  - System Configuration File
- Processing System
  - OS
    - Bare-metal
    - Linux
  - PS Application
    - AI Engine Drivers
    - XRT
    - OpenCL

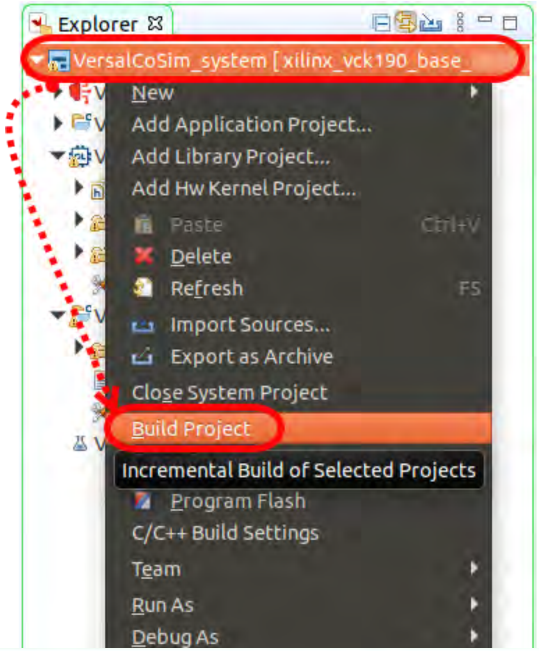
A Simple Example

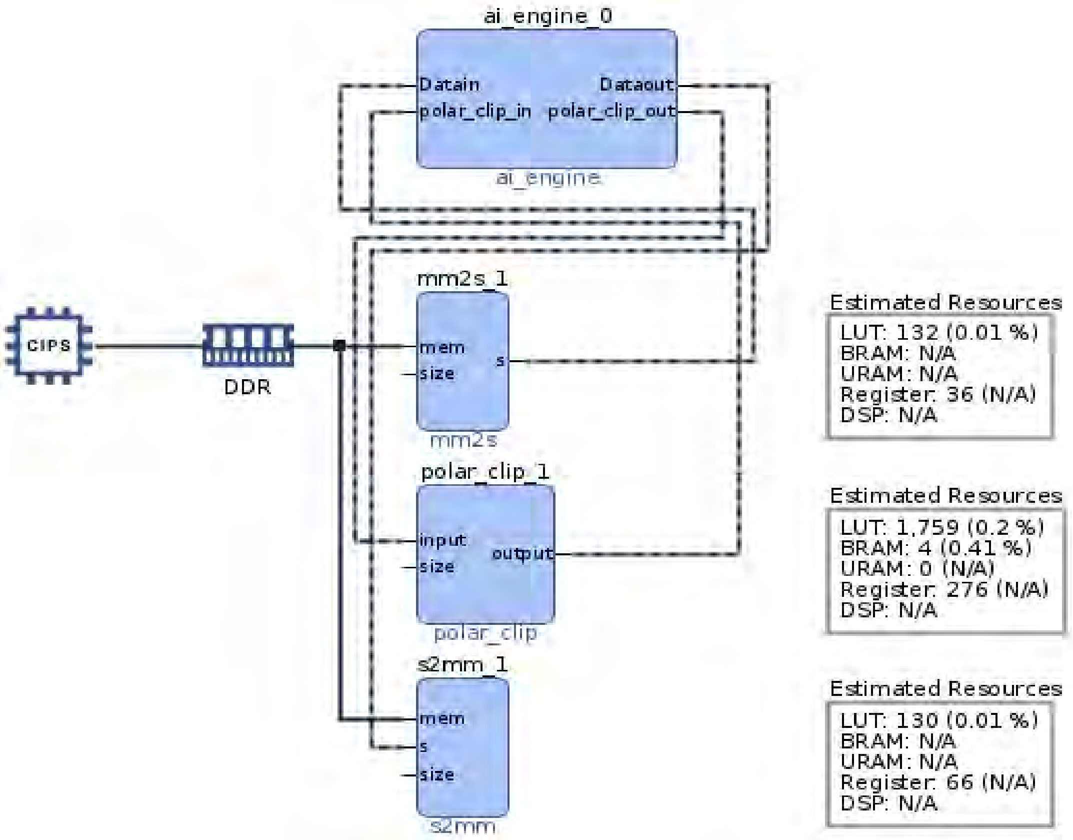


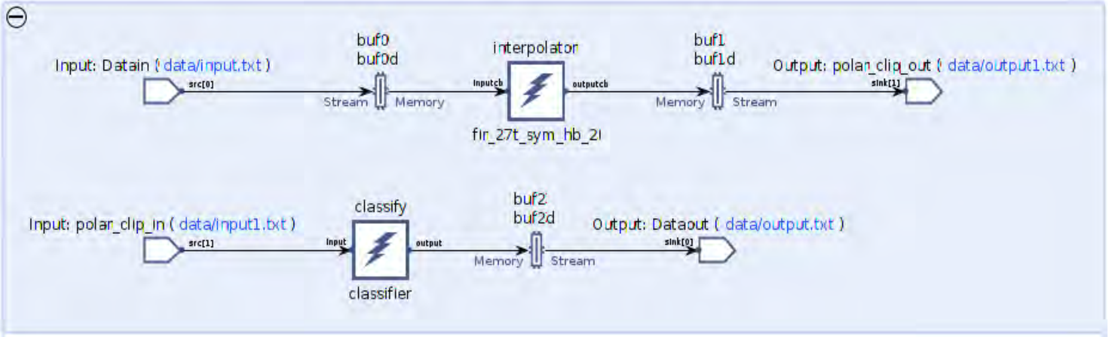


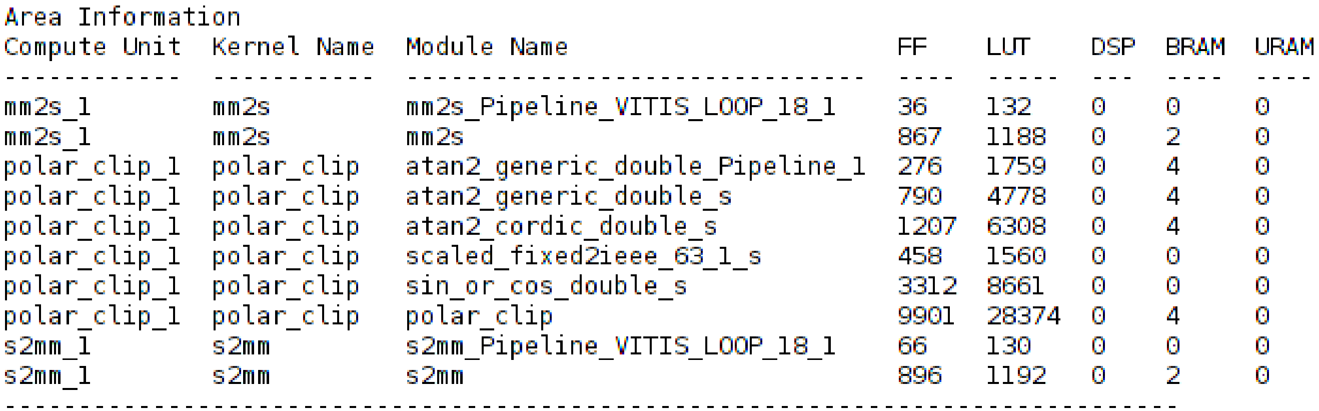
A Simple Example

- Build the system and review results





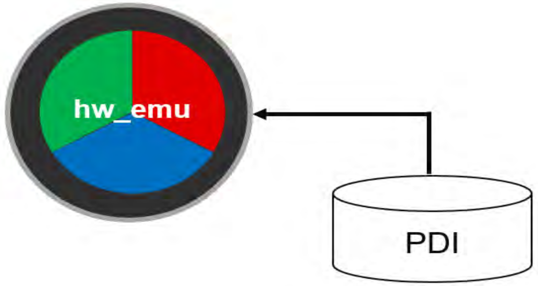


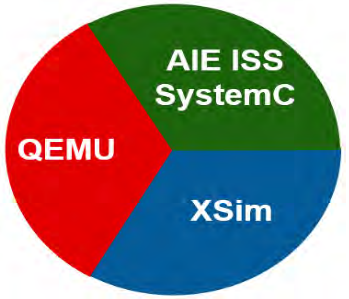


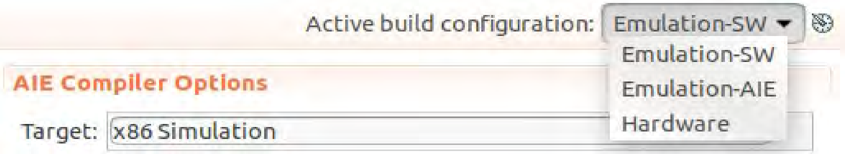
```
Area Information
Compute Unit   Kernel Name   Module Name                       FF    LUT    DSP  BRAM  URAM
------------   -----------   -------------------------------   ----  -----  ---  ----  ----
mm2s_1         mm2s          mm2s_Pipeline_VITIS_LOOP_18_1     36    132    0    0     0
mm2s_1         mm2s          mm2s                              867   1188   0    2     0
polar_clip_1   polar_clip    atan2_generic_double_Pipeline_1   276   1759   0    4     0
polar_clip_1   polar_clip    atan2_generic_double_s            790   4778   0    4     0
polar_clip_1   polar_clip    atan2_cordic_double_s             1207  6308   0    4     0
polar_clip_1   polar_clip    scaled_fixed2ieee_63_1_s          458   1560   0    0     0
polar_clip_1   polar_clip    sin_or_cos_double_s               3312  8661   0    0     0
polar_clip_1   polar_clip    polar_clip                        9901  28374  0    4     0
s2mm_1         s2mm          s2mm_Pipeline_VITIS_LOOP_18_1     66    130    0    0     0
s2mm_1         s2mm          s2mm                              896   1192   0    2     0
-----------------------------------------------------------------------------------------
```

A Simple Example

- You can simulate separately or at system level
  - AIE in SystemC, A72 en QEMU, PL in XSIM

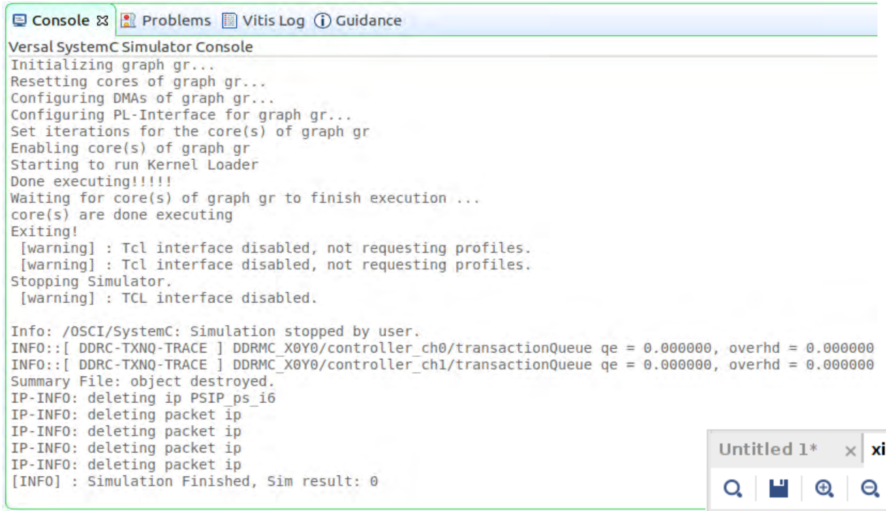





Active build configuration: Emulation-SW

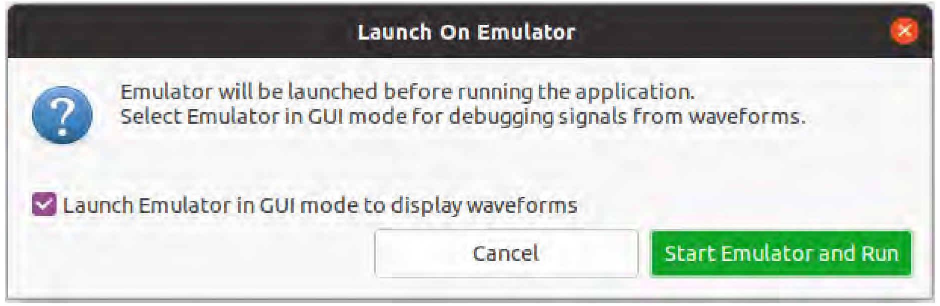
AIE Compiler Options

Target: x86 Simulation



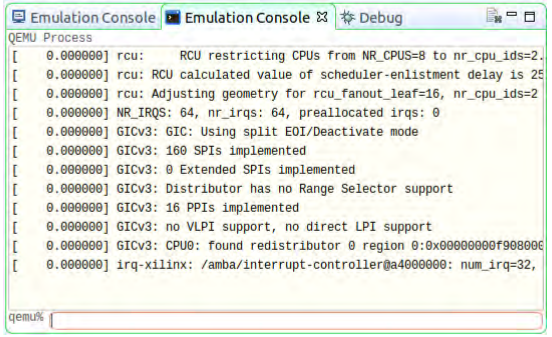
Versal SystemC Simulator Console

```
Initializing graph gr...
Resetting cores of graph gr...
Configuring DMAs of graph gr...
Configuring PL-Interface for graph gr...
Set iterations for the core(s) of graph gr
Enabling core(s) of graph gr
Starting to run Kernel Loader
Done executing!!!!!
Waiting for core(s) of graph gr to finish execution ...
core(s) are done executing
Exiting!
 [warning] : Tcl interface disabled, not requesting profiles.
 [warning] : Tcl interface disabled, not requesting profiles.
Stopping Simulator.
 [warning] : TCL interface disabled.

Info: /OSCI/SystemC: Simulation stopped by user.
INFO::[ DDRC-TXNQ-TRACE ] DDRMC_X0Y0/controller_ch0/transactionQueue qe = 0.000000, overhd = 0.000000
INFO::[ DDRC-TXNQ-TRACE ] DDRMC_X0Y0/controller_ch1/transactionQueue qe = 0.000000, overhd = 0.000000
Summary File: object destroyed.
IP-INFO: deleting ip PSIP_ps_i6
IP-INFO: deleting packet ip
IP-INFO: deleting packet ip
IP-INFO: deleting packet ip
IP-INFO: deleting packet ip
[INFO] : Simulation Finished, Sim result: 0
```


Launch On Emulator

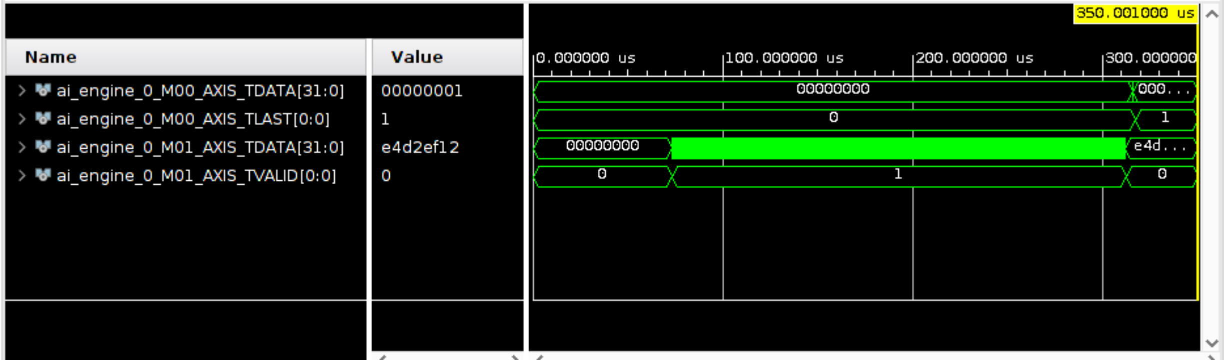
Emulator will be launched before running the application.
Select Emulator in GUI mode for debugging signals from waveforms.

☑ Launch Emulator in GUI mode to display waveforms

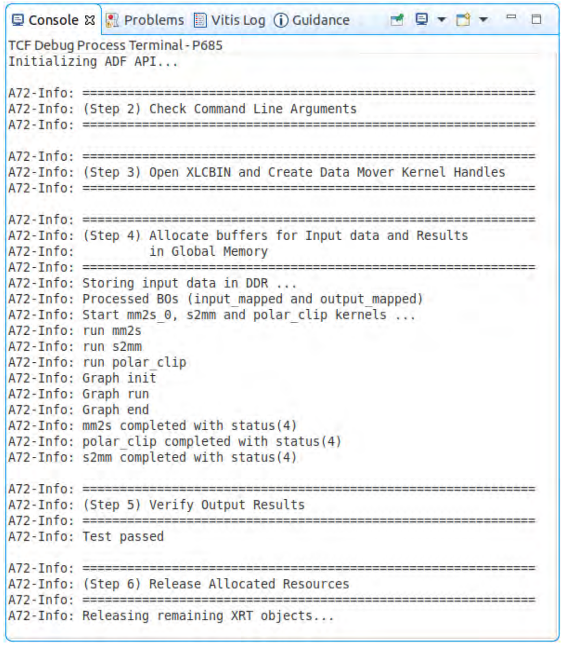


QEMU Process

```
[    0.000000] rcu:     RCU restricting CPUs from NR_CPUS=8 to nr_cpu_ids=2.
[    0.000000] rcu: RCU calculated value of scheduler-enlistment delay is 25
[    0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
[    0.000000] NR_IRQS: 64, nr_irqs: 64, preallocated irqs: 0
[    0.000000] GICv3: GIC: Using split EOI/Deactivate mode
[    0.000000] GICv3: 160 SPIs implemented
[    0.000000] GICv3: 0 Extended SPIs implemented
[    0.000000] GICv3: Distributor has no Range Selector support
[    0.000000] GICv3: 16 PPIs implemented
[    0.000000] GICv3: no VLPI support, no direct LPI support
[    0.000000] GICv3: CPU0: found redistributor 0 region 0:0x00000000f908000
[    0.000000] irq-xilinx: /amba/interrupt-controller@a4000000: num_irq=32,

qemu%
```



Name	Value
ai_engine_0_M00_AXIS_TDATA[31:0]	00000001
ai_engine_0_M00_AXIS_TLAST[0:0]	1
ai_engine_0_M01_AXIS_TDATA[31:0]	e4d2ef12
ai_engine_0_M01_AXIS_TVALID[0:0]	0



TCF Debug Process Terminal - P685

```
Initializing ADF API...

A72-Info: ============================================================
A72-Info: (Step 2) Check Command Line Arguments
A72-Info: ============================================================

A72-Info: ============================================================
A72-Info: (Step 3) Open XLCBIN and Create Data Mover Kernel Handles
A72-Info: ============================================================

A72-Info: ============================================================
A72-Info: (Step 4) Allocate buffers for Input data and Results
A72-Info:          in Global Memory
A72-Info: ============================================================
A72-Info: Storing input data in DDR ...
A72-Info: Processed BOs (input_mapped and output_mapped)
A72-Info: Start mm2s_0, s2mm and polar_clip kernels ...
A72-Info: run mm2s
A72-Info: run s2mm
A72-Info: run polar_clip
A72-Info: Graph init
A72-Info: Graph run
A72-Info: Graph end
A72-Info: mm2s completed with status(4)
A72-Info: polar_clip completed with status(4)
A72-Info: s2mm completed with status(4)

A72-Info: ============================================================
A72-Info: (Step 5) Verify Output Results
A72-Info: ============================================================
A72-Info: Test passed

A72-Info: ============================================================
A72-Info: (Step 6) Release Allocated Resources
A72-Info: ============================================================
A72-Info: Releasing remaining XRT objects...
```