

New SoC platforms for AI and Algorithm Acceleration (Introduction to Xilinx Versal ACAP)

Gustavo Sutter
gustavo.sutter@uam.es



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



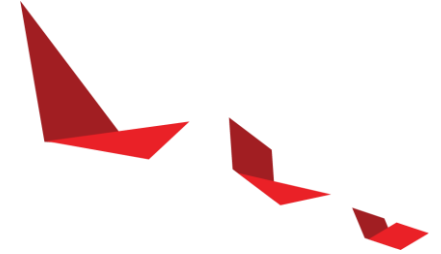


Versal ACAP Introduction

Gustavo Sutter

Presentation for the “Joint ICTP, SAIFR and INESP School on Systems-on-Chip, Embedded Microcontrollers and their Applications in Research and Industry”

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 (Programmable 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

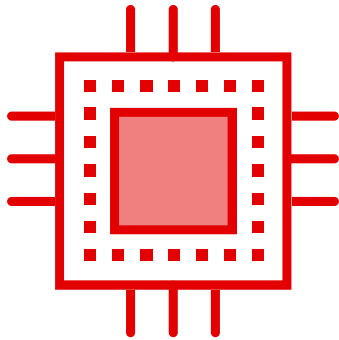
Software programmability

Performance for a diverse
range of applications

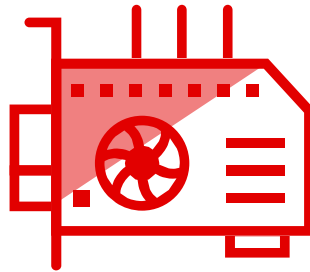
Adaptability to keep pace
with rapid innovation



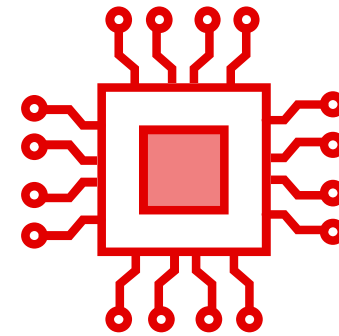
Today's Solutions



CPUs



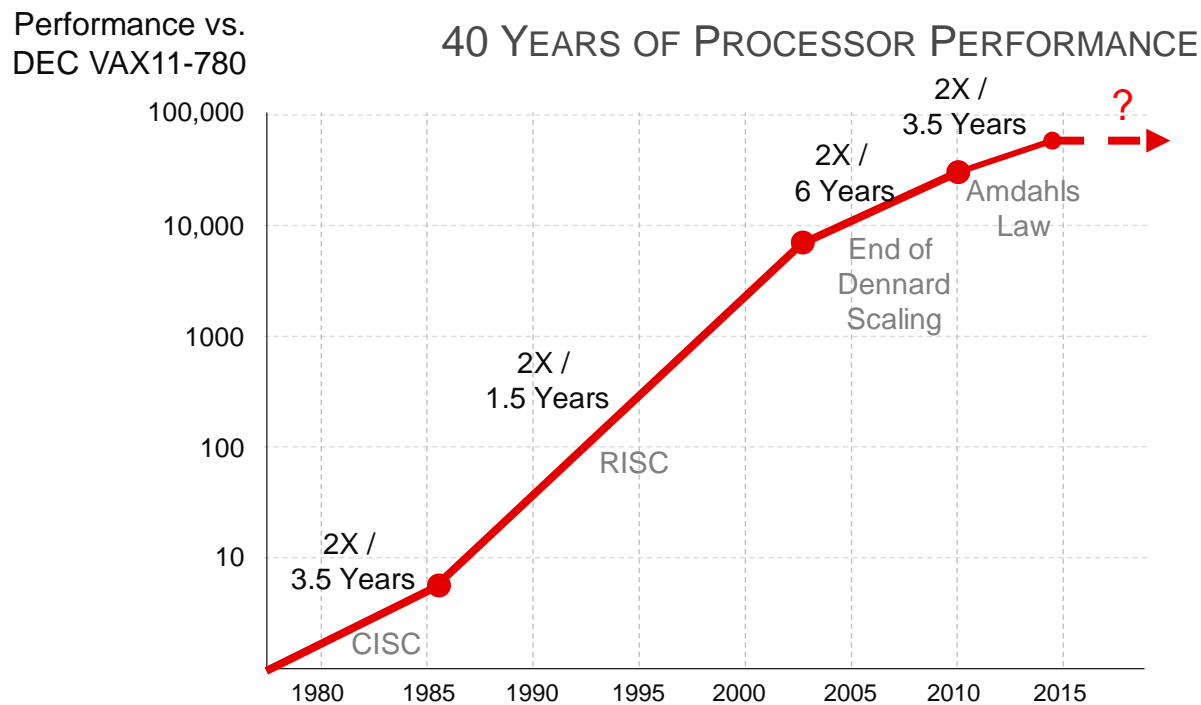
Fixed Function
Accelerators
ASICs/ASSPs/GPUs



FPGAs

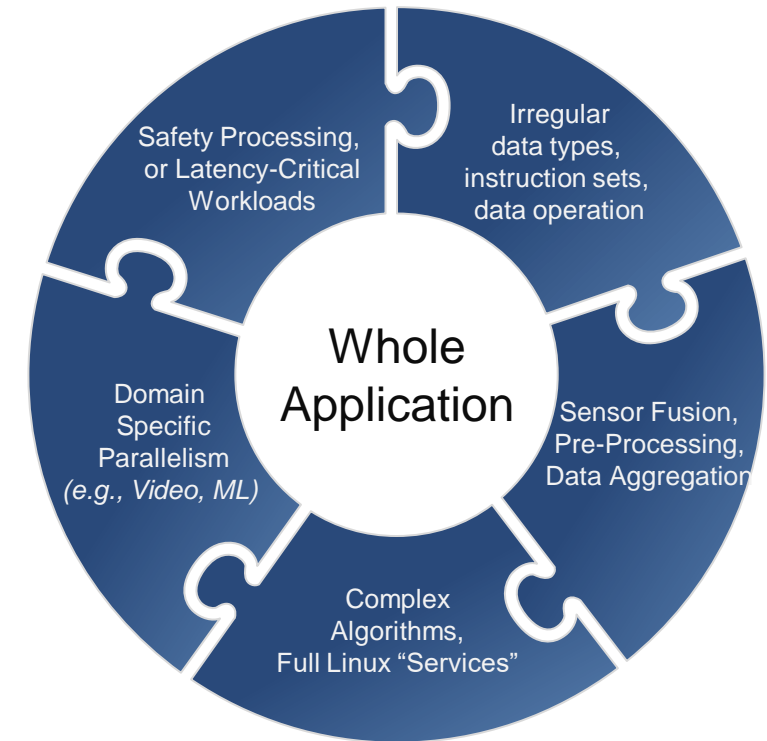
The Technology Conundrum .. And the Need for a New Compute Paradigm

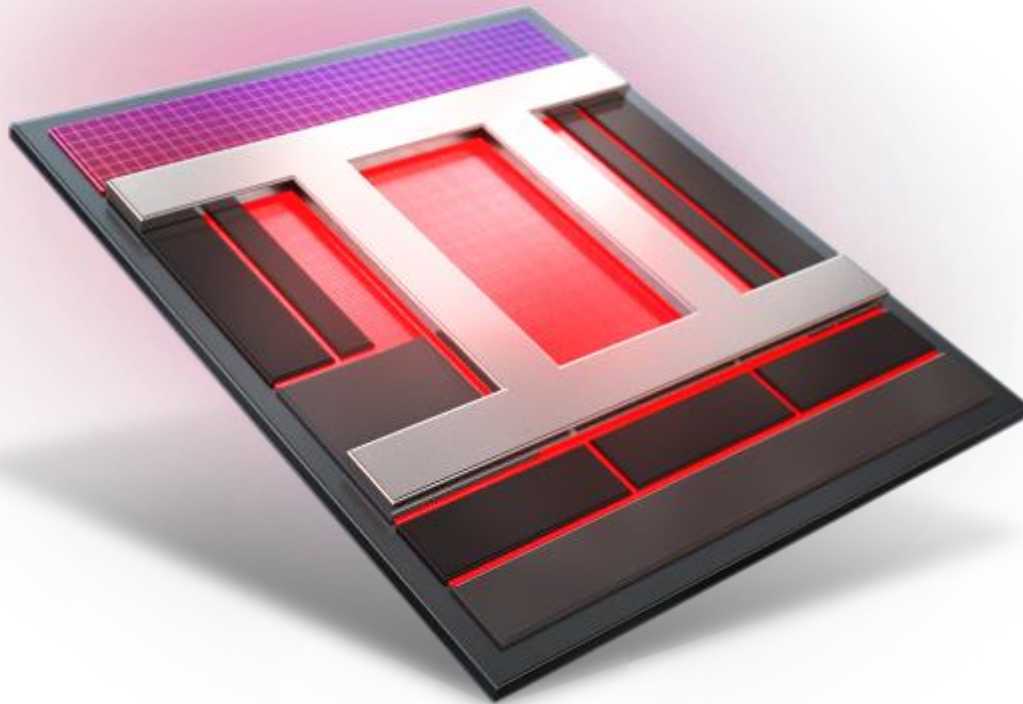
Processing Architectures
are Not Scaling



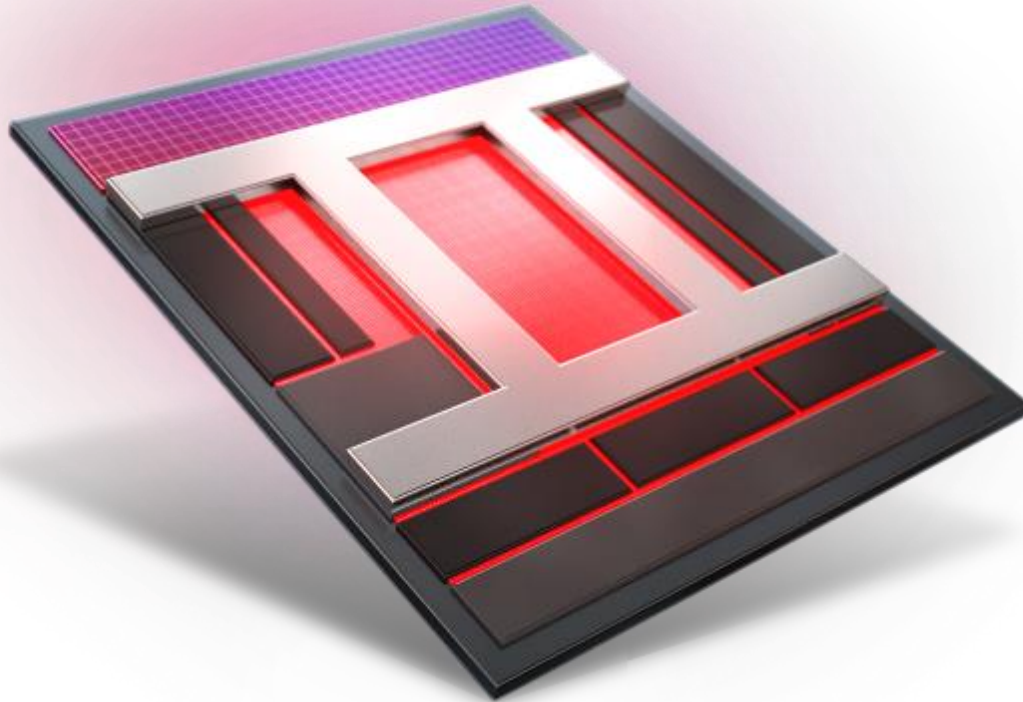
Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e 2018

A Single Architecture
Can't Do it Alone



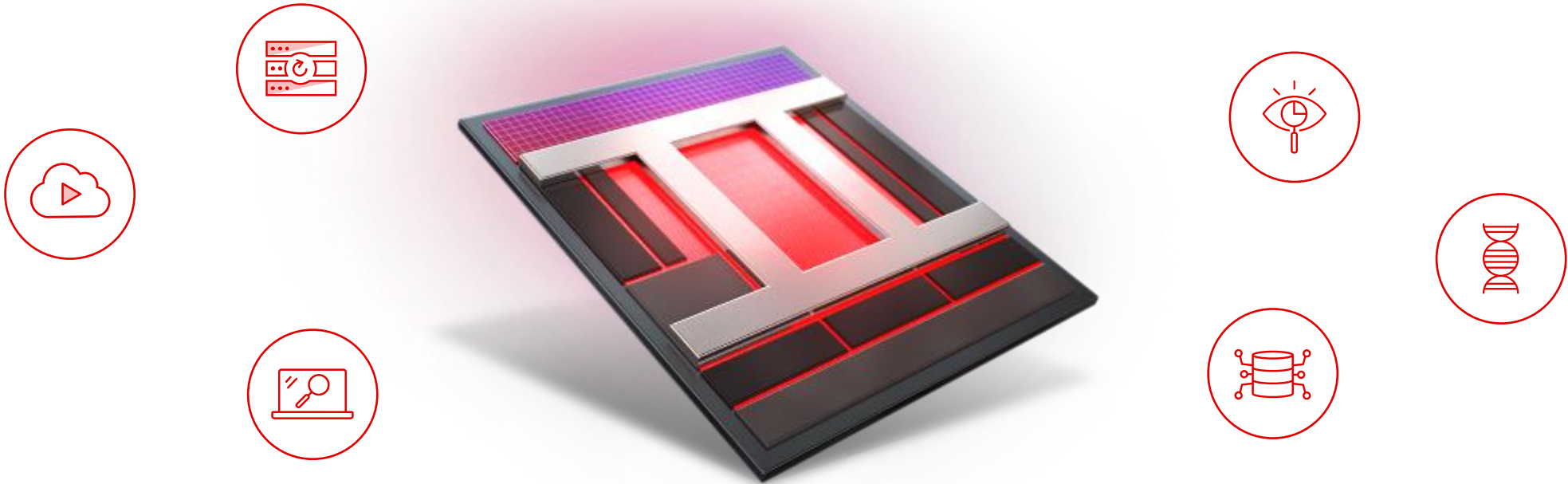


ACAP



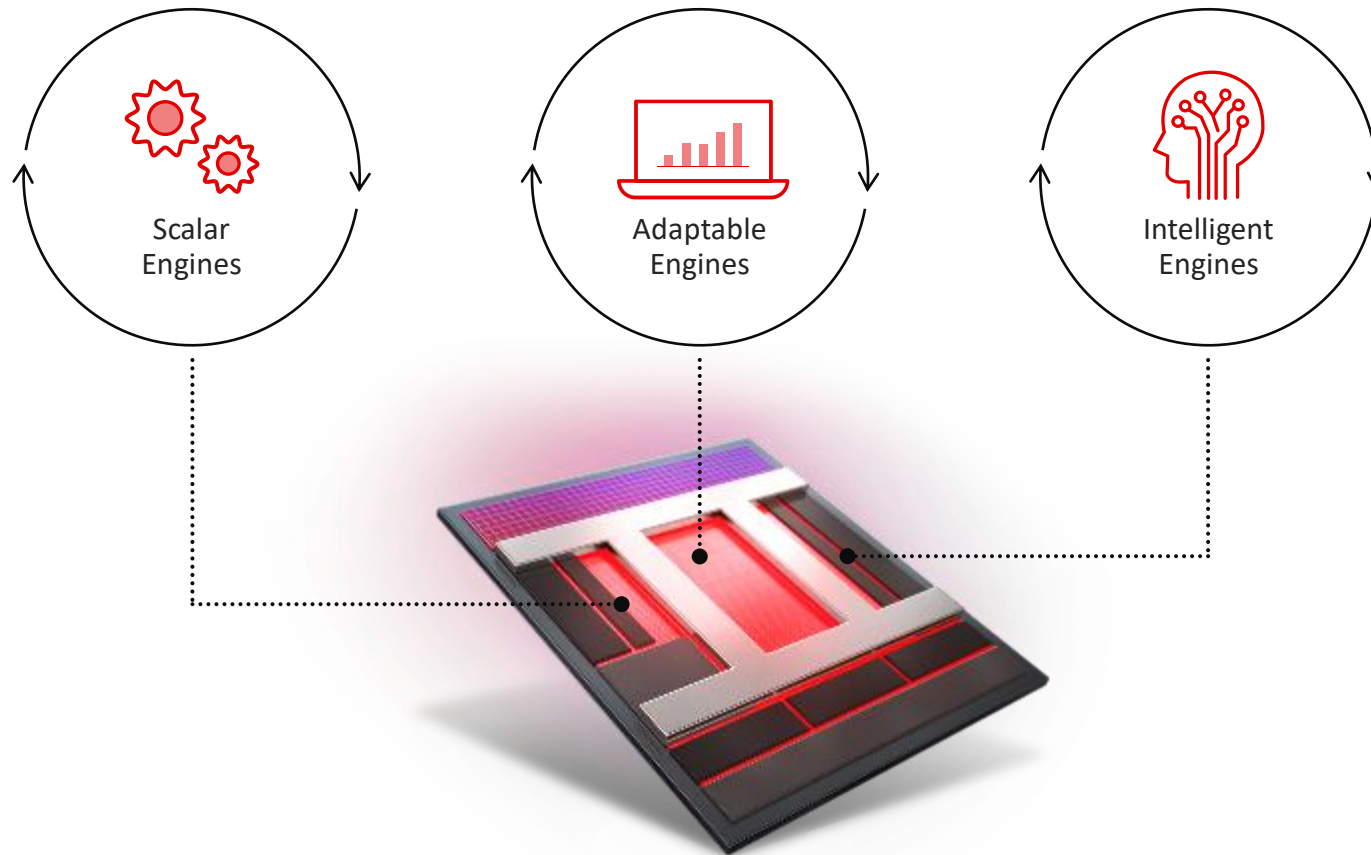
Adaptive Compute Acceleration Platform

Adaptive



Adaptive Hardware for
Domain-Specific Applications

Compute Acceleration



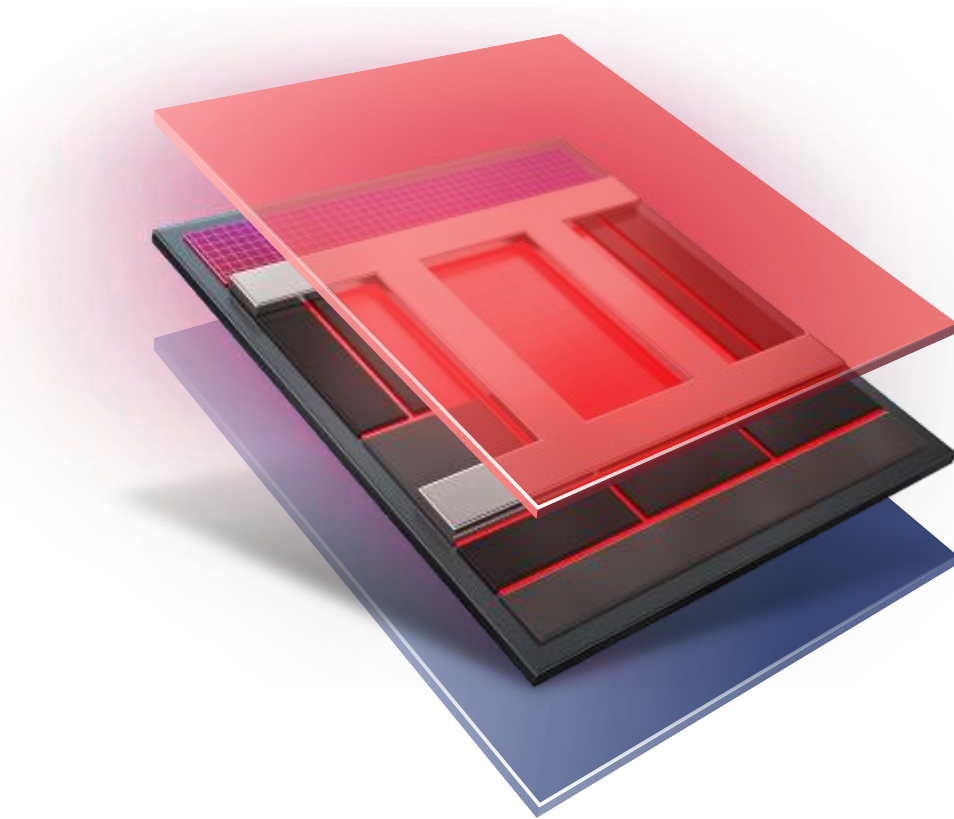
Platform

ENABLING:

Data Scientists

SW App Developers

HW Developers



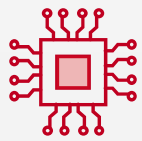
Development Tools
HW/SW Libraries
Run-time Stack

SW Programmable
Silicon Infrastructure

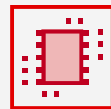
Disruptive Innovation Needed: Enter ACAP

A new class of devices for
today's challenges

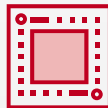
Software Programmability



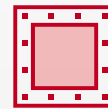
FPGA



SoC



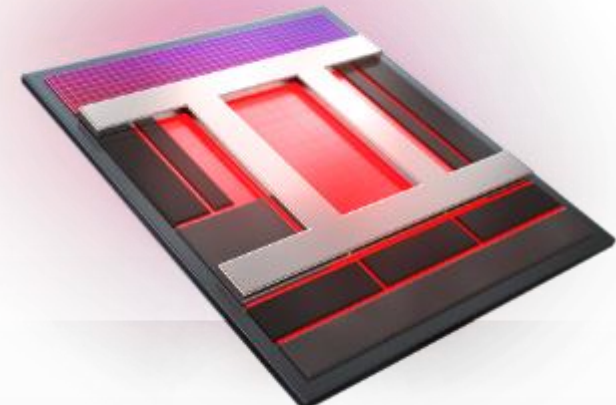
MPSoC



RFSoc



ACAP



Device Category

Introducing the Industry's First ACAP







XILINX®
VERSAL™

The Industry's First ACAP

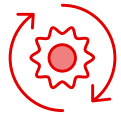
Heterogeneous Acceleration

For Any Application

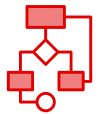
For Any Developer



Versal ACAP



Scalar Processing Engines



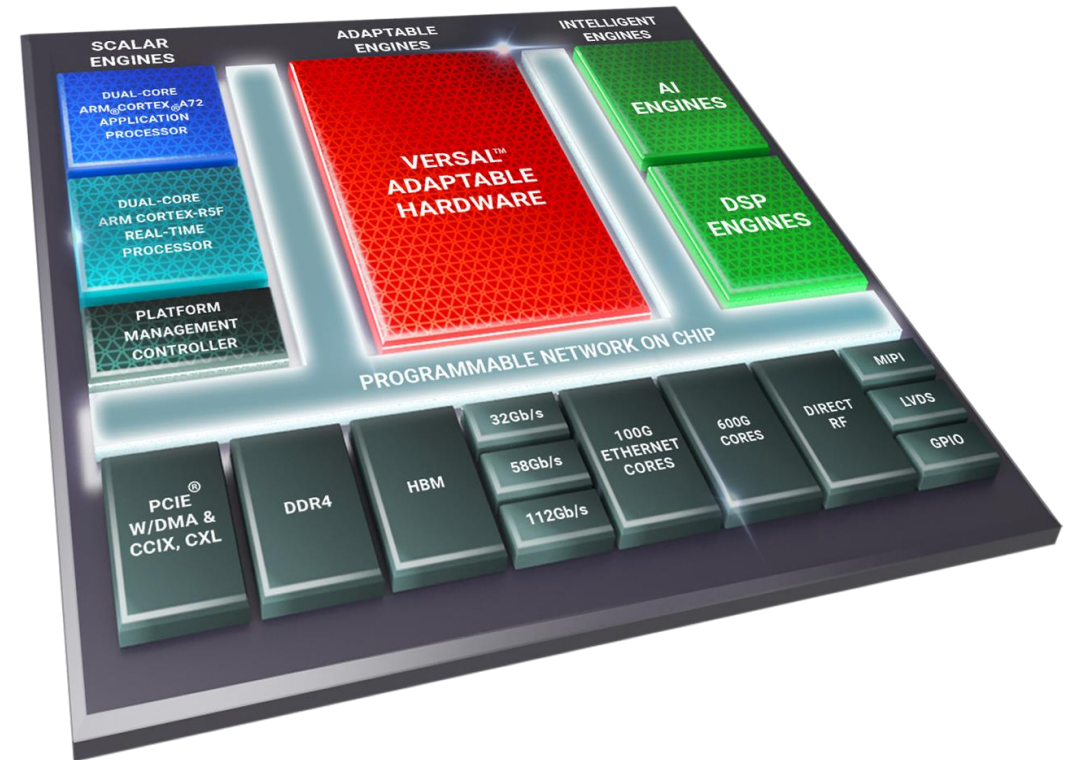
Adaptable Hardware Engines



Intelligent Engines
SW Programmable, HW Adaptable



Breakout Integration of Advanced
Protocol Engines

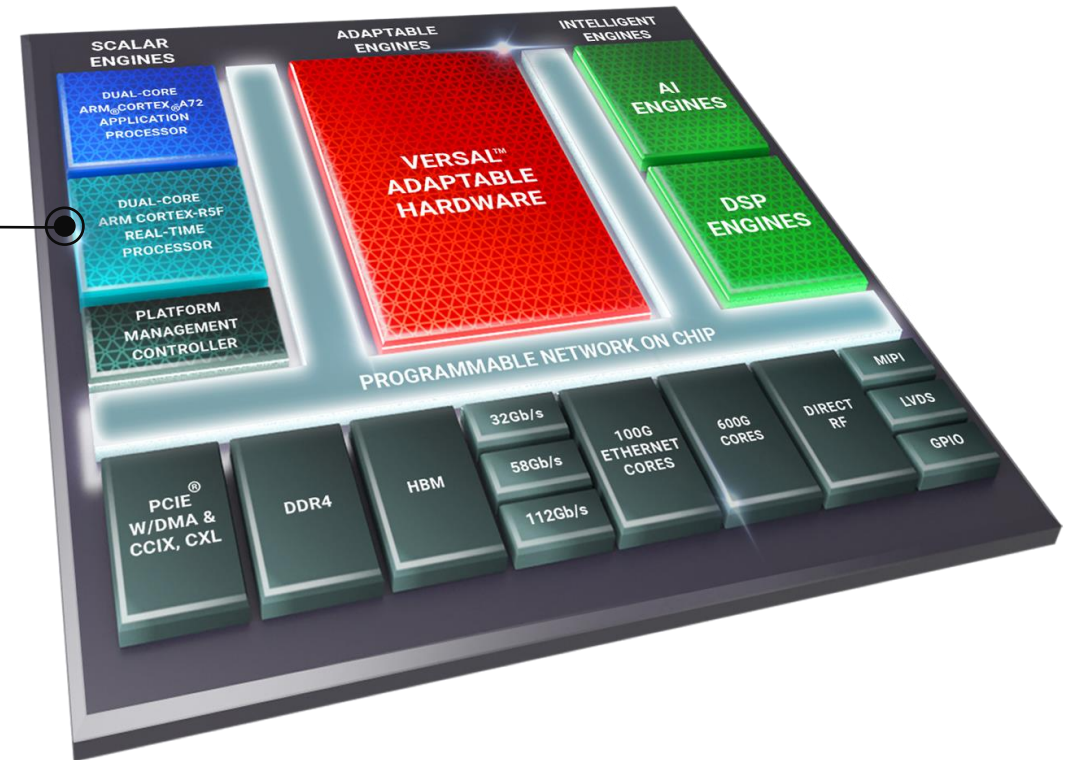


Scalar Processing Engines

Arm Cortex-A72
Application Processor

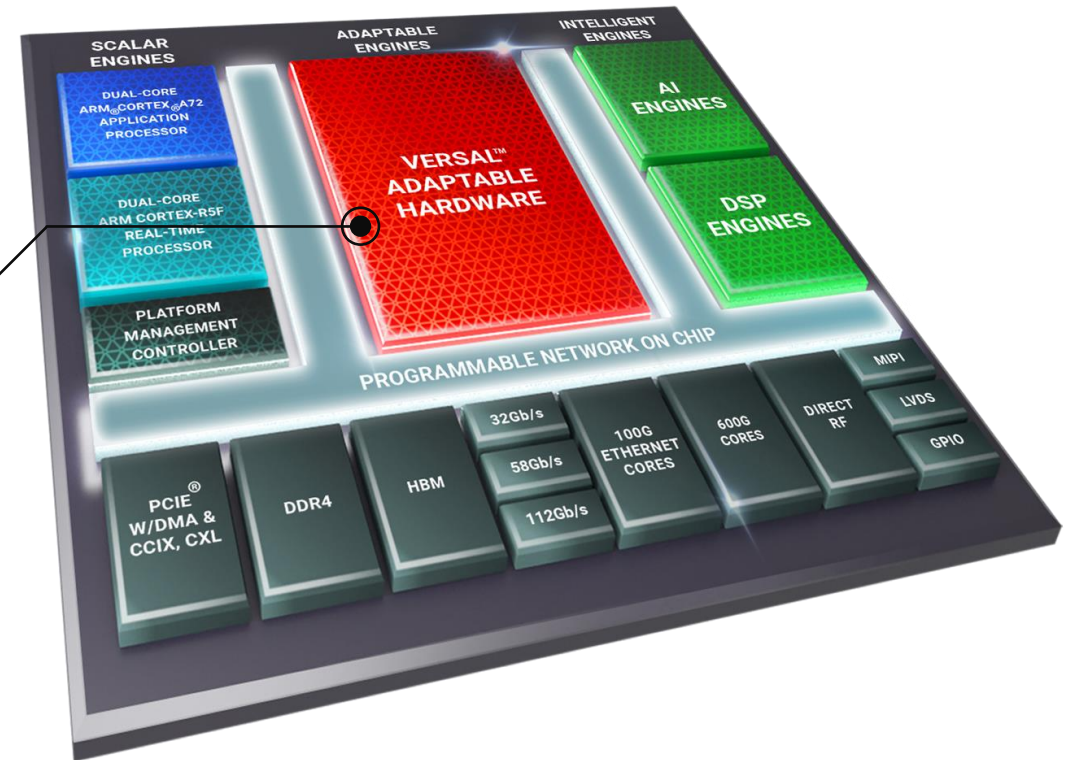
Arm Cortex-R5
Real-Time Processor

Platform Management Controller



Adaptable Hardware Engines

Re-architected foundational HW
fabric for greater compute density
Enables custom memory hierarchy
8X Faster Dynamic Reconfiguration
("on-the-fly")



Intelligent Engines

DSP Engines

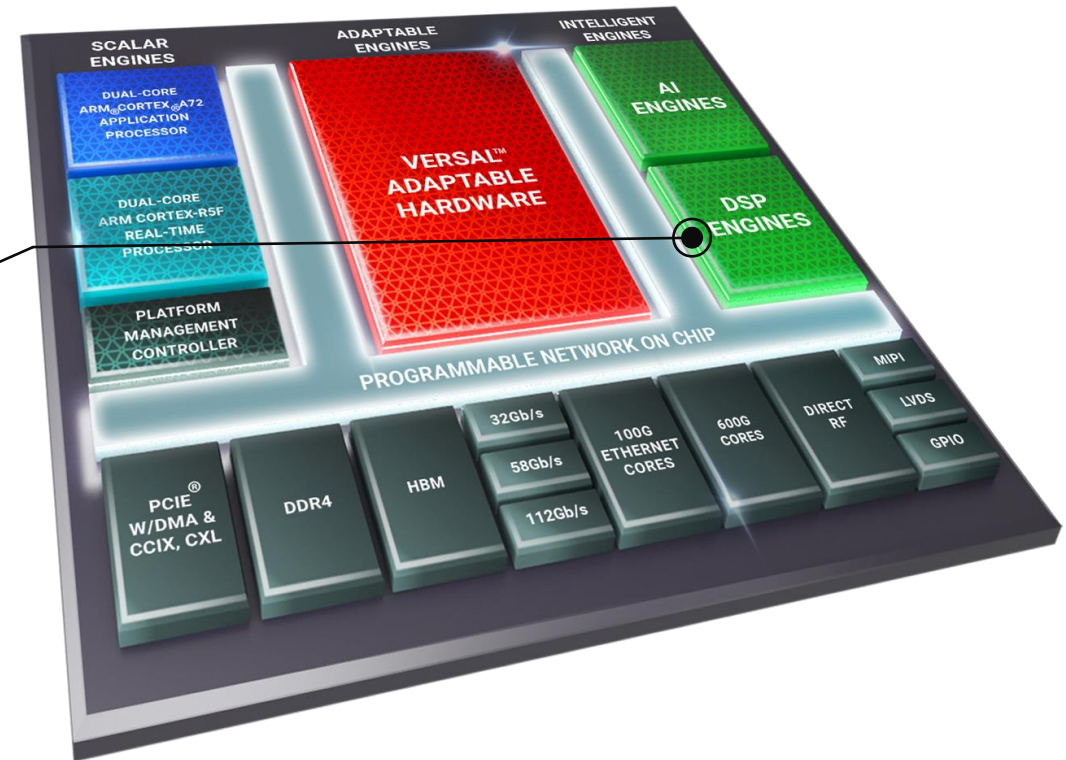
High-precision floating point & low latency

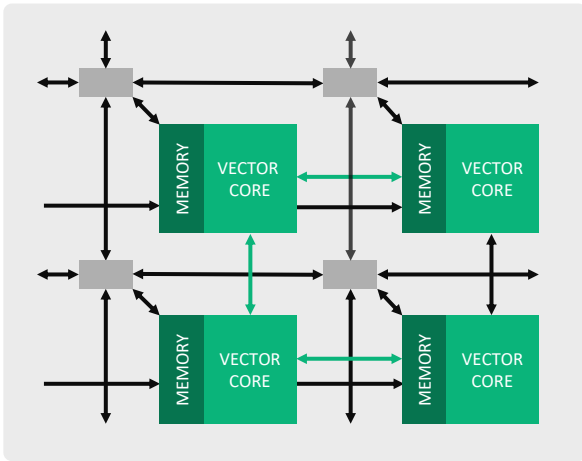
Granular control for customized datapaths

AI Engines

High throughput, low latency, and power efficient

Ideal for AI inference and advanced signal processing

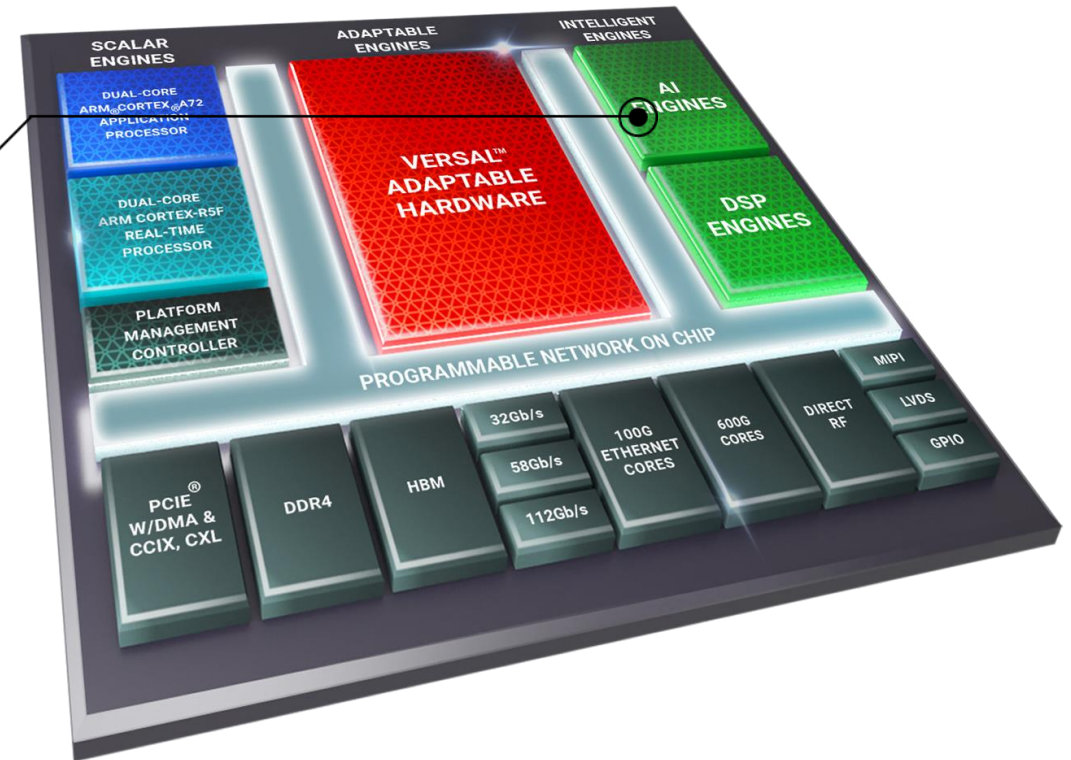




AI Engines

Optimized for AI Inference and Advanced Signal Processing Workloads

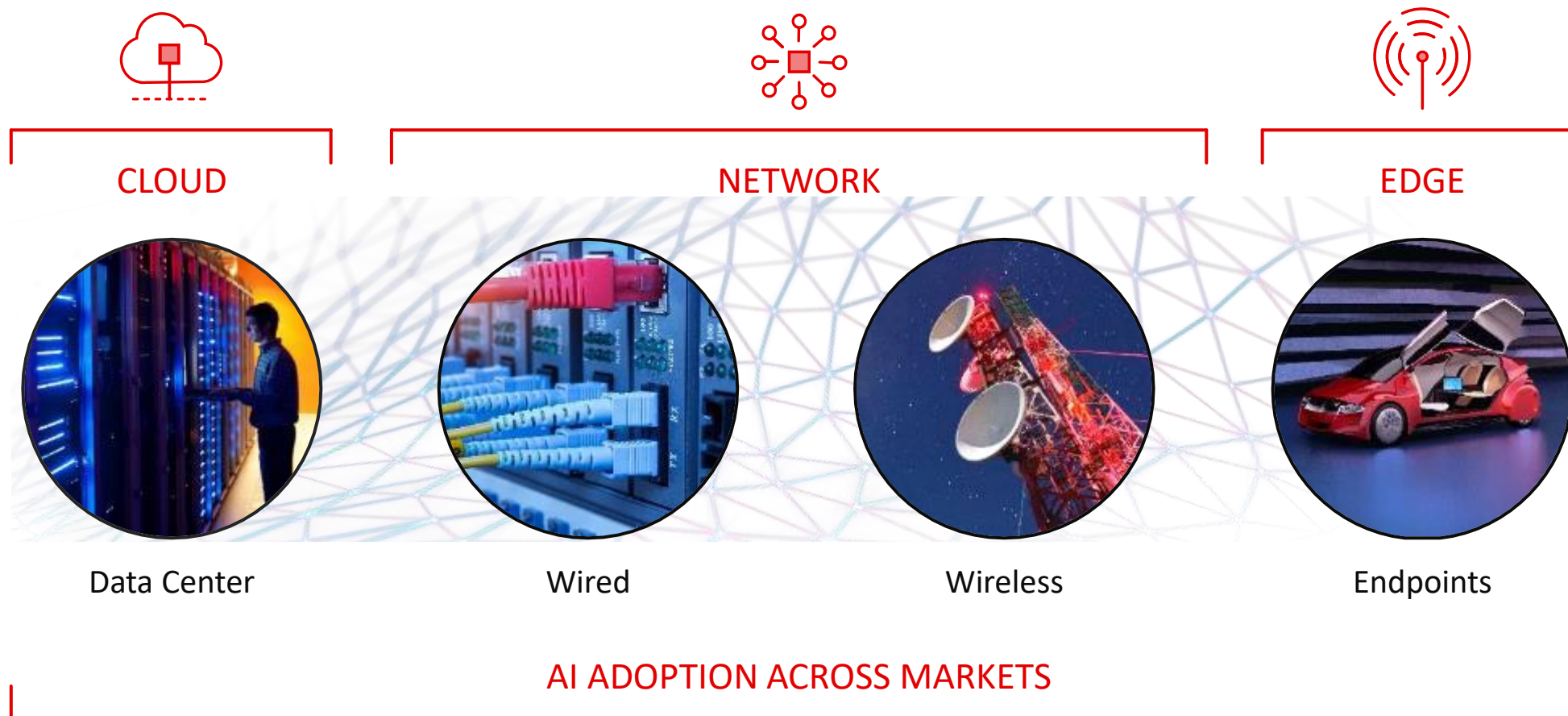
- >1GHz VLIW/SIMD vector processor cores
- Massive array of interconnected cores with tightly coupled memory
- Tightly coupled to adaptable hardware engine to enable custom memory hierarchy
- Software programmable, C-, and library-based with hardware adaptability





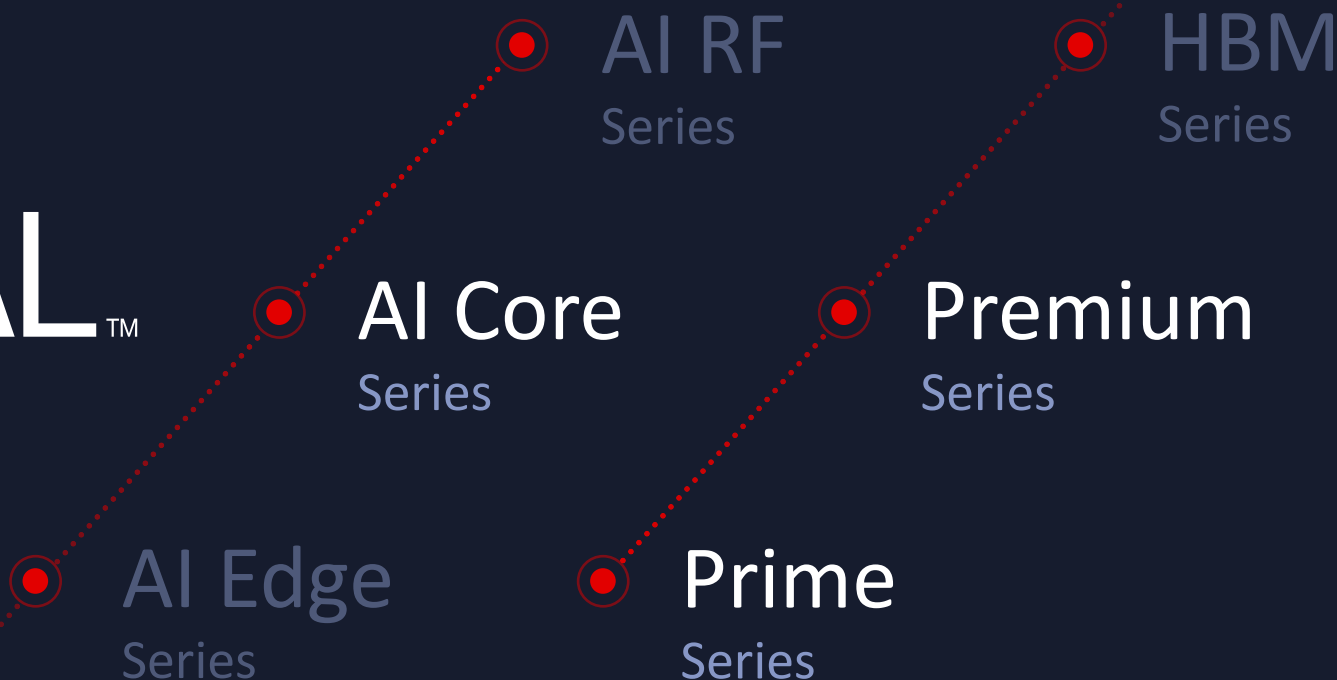
For Any Application

Versal for Multi-Market Applications





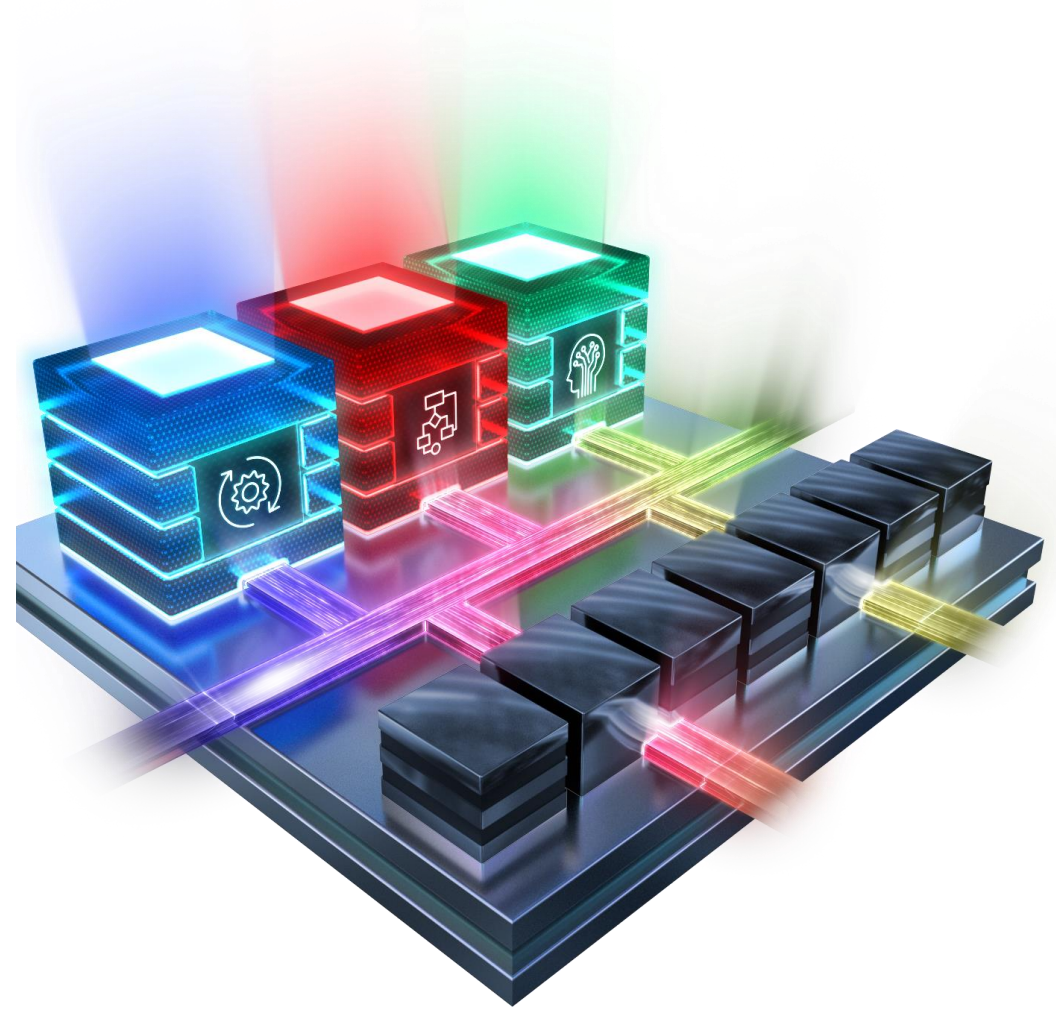
XILINX®
VERSAL™



Adaptive Compute Acceleration Platform (ACAP)



- > 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

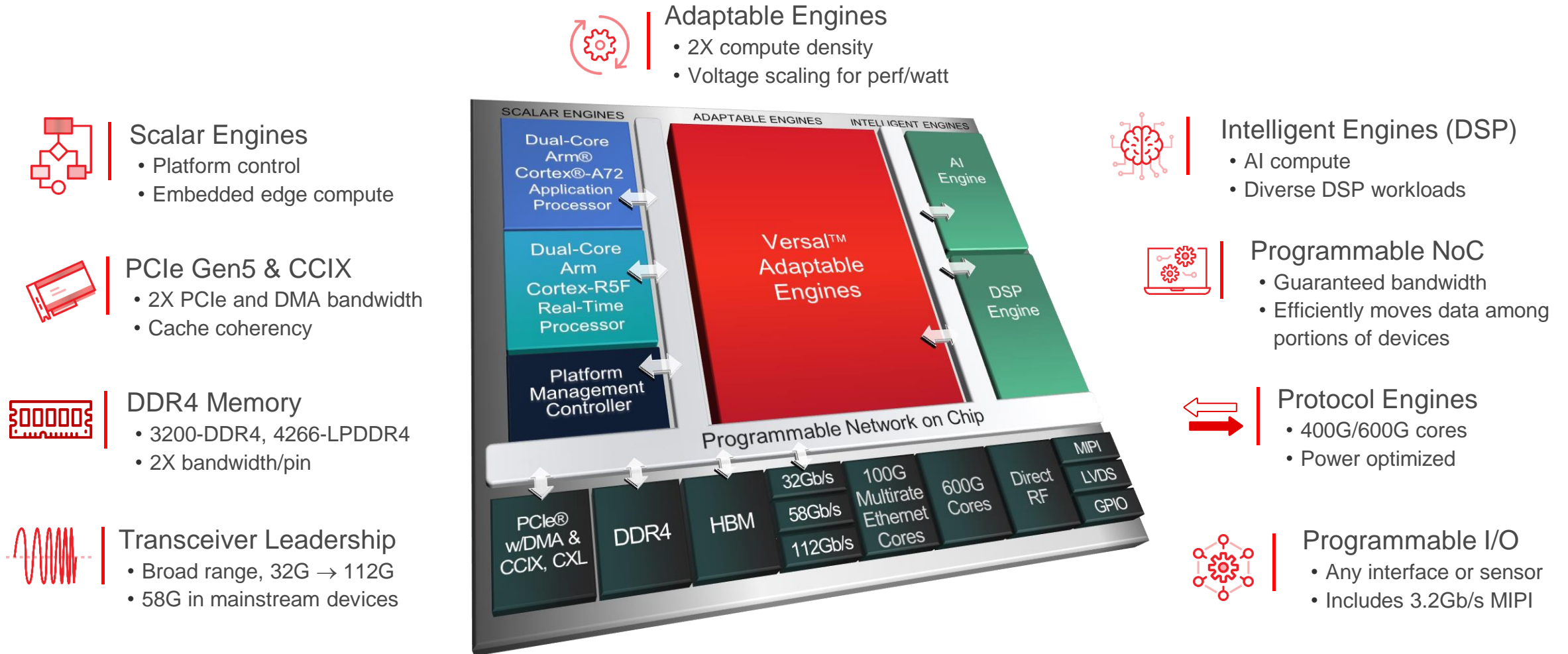




Versal ACAP: Architecture Overview



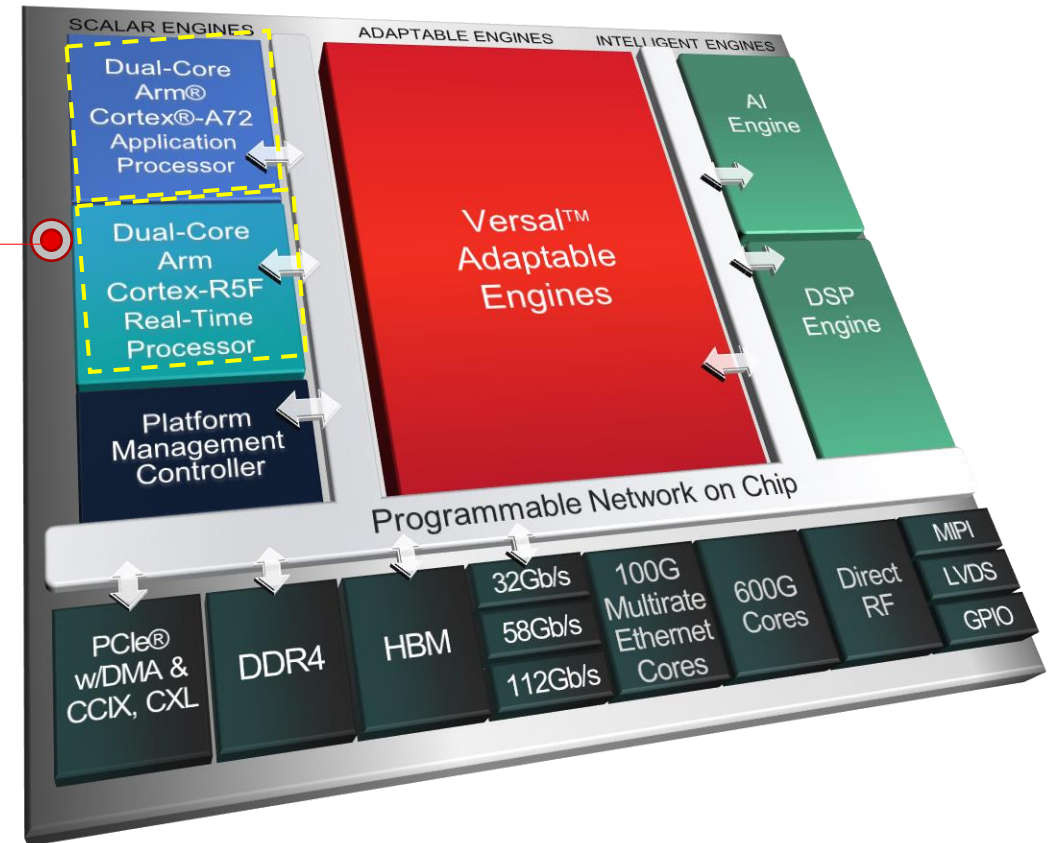
Versal Architecture: Overview



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

Application Processing Unit (APU)

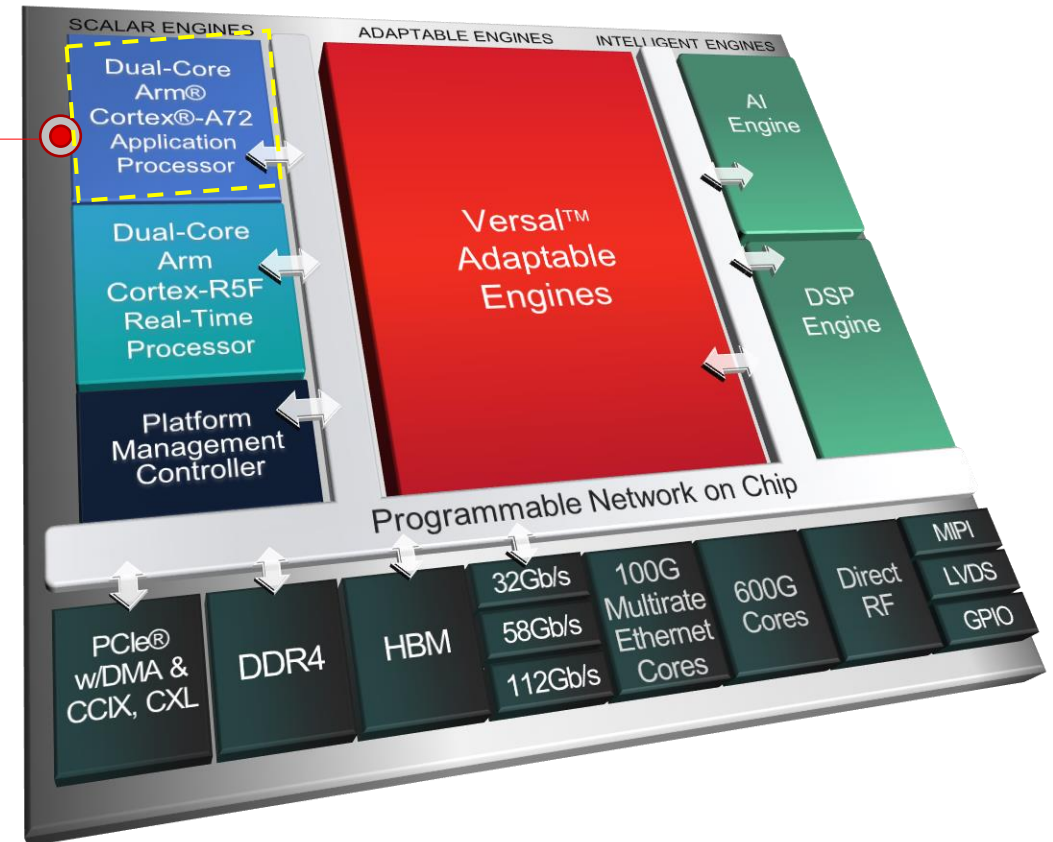
Based on Arm Cortex-A72 dual-core processor

- 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

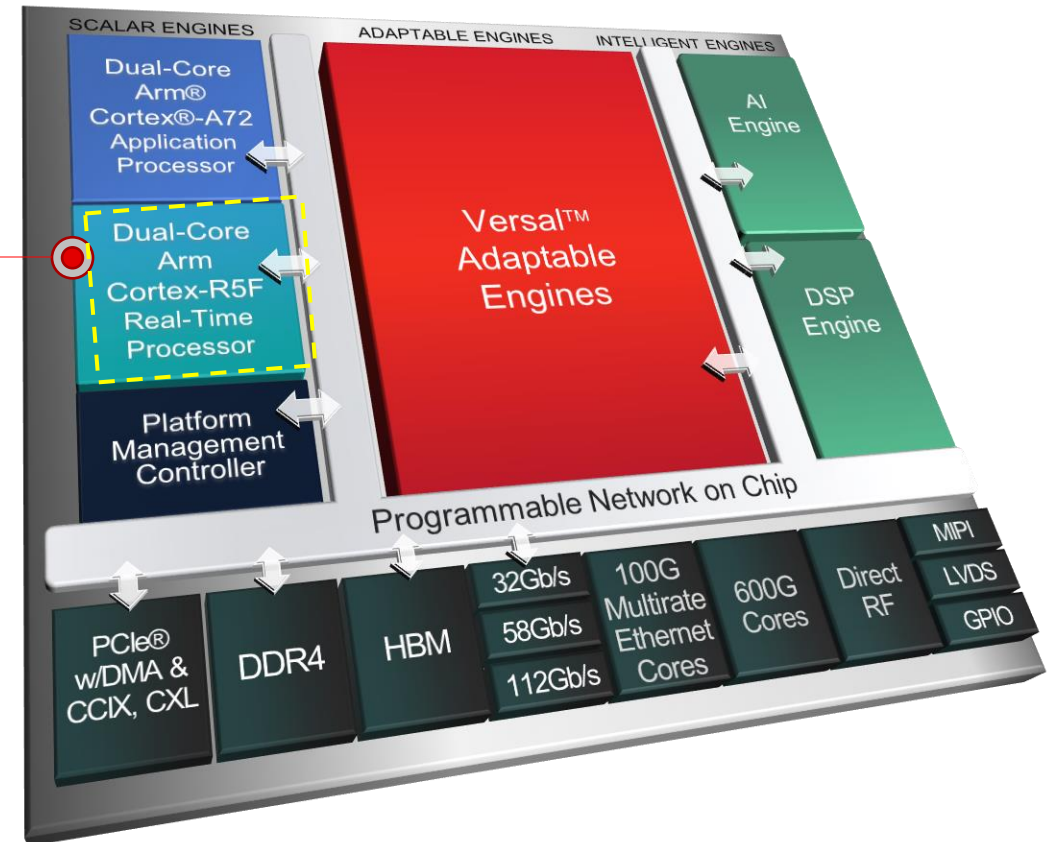
Real-time Processing Unit (RPU)

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



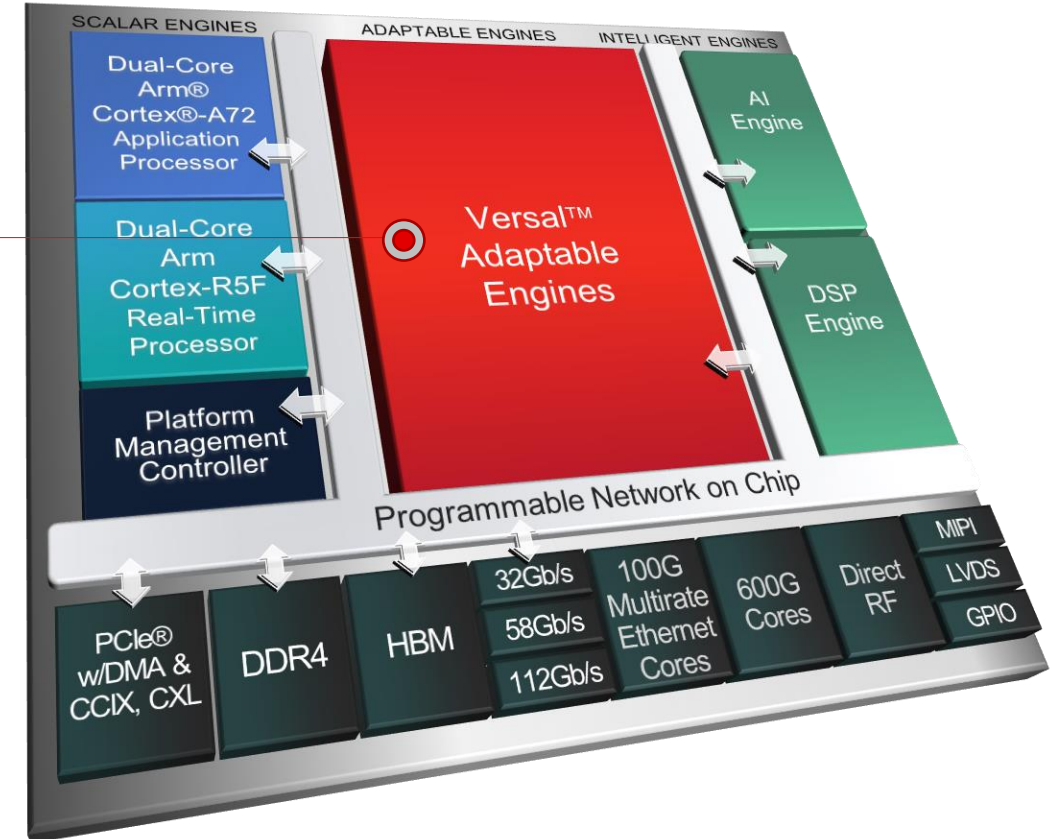
Adaptable Engines

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



Configurable Logic Block

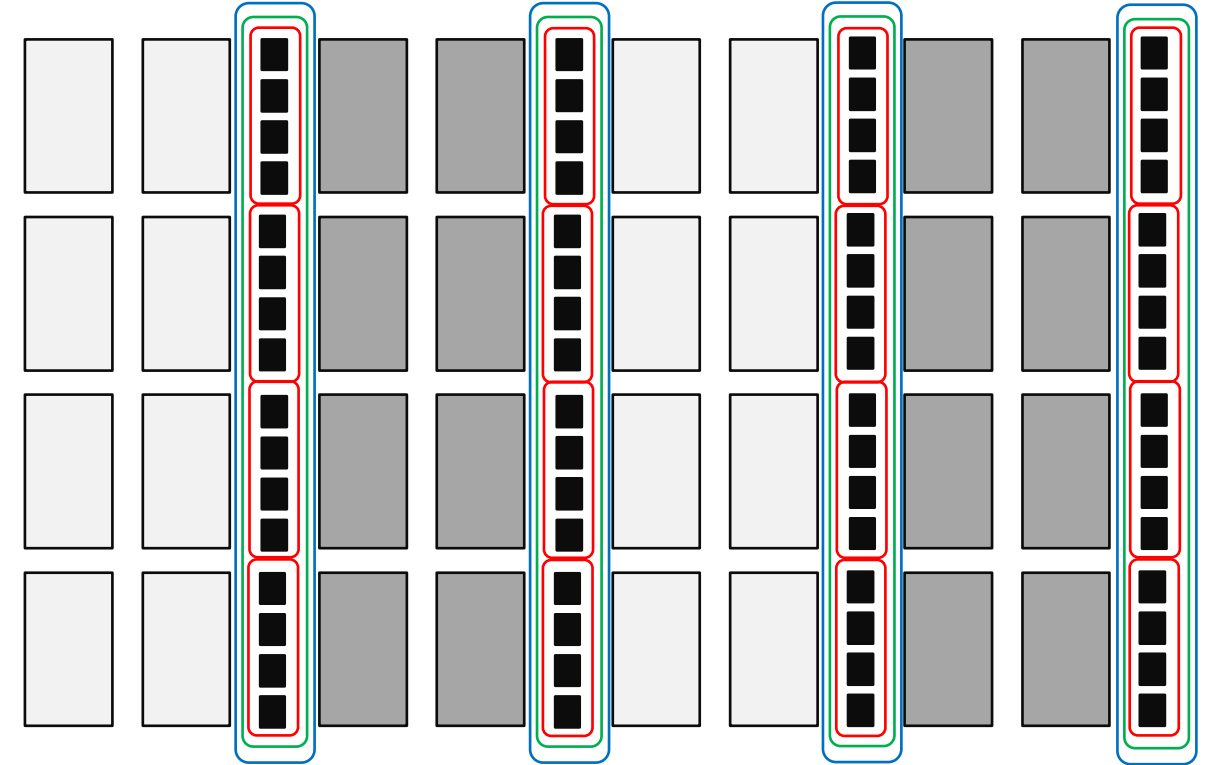
Heart of the programmable logic

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



□ LUT

■ LUTRAM/SRL-capable LUT

■ Flip-flop

□ CE

□ Clock

□ SR

CLB Features

Four times more logic capacity than before

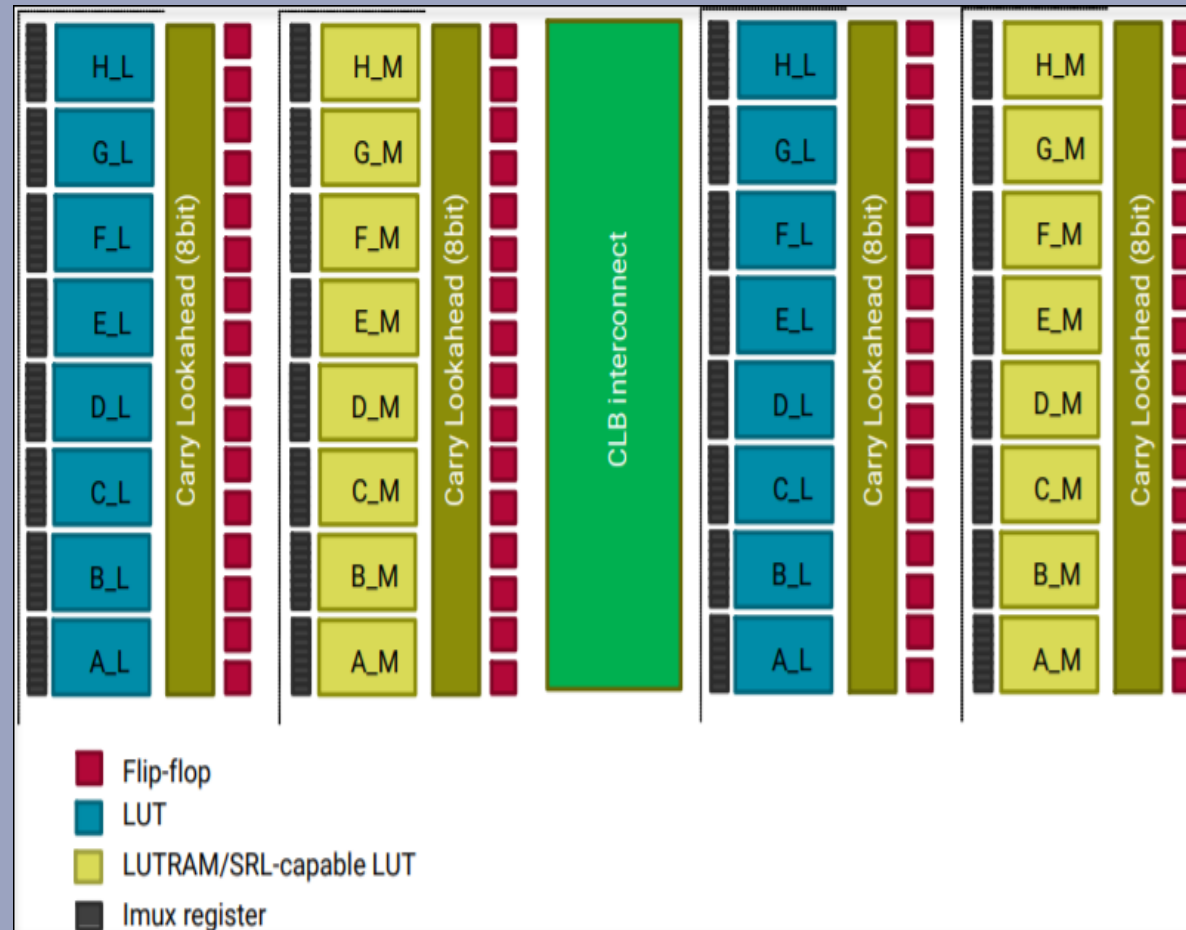
Dedicated LUT-LUT cascade paths exist inside the CLB to reduce delays

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

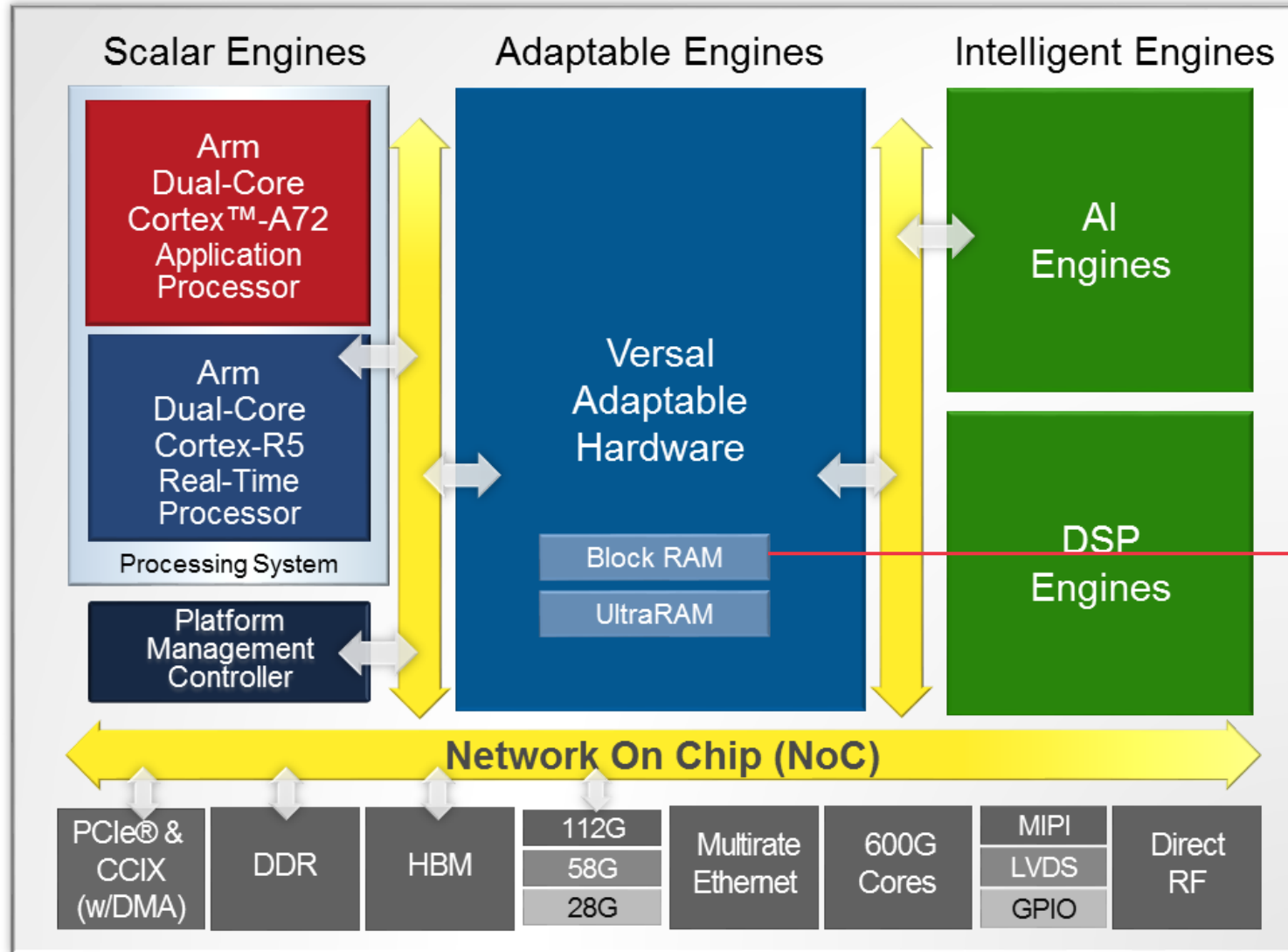
LUTRAMs are simplified, having dedicated hardware to support 32- and 64-bit depths

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



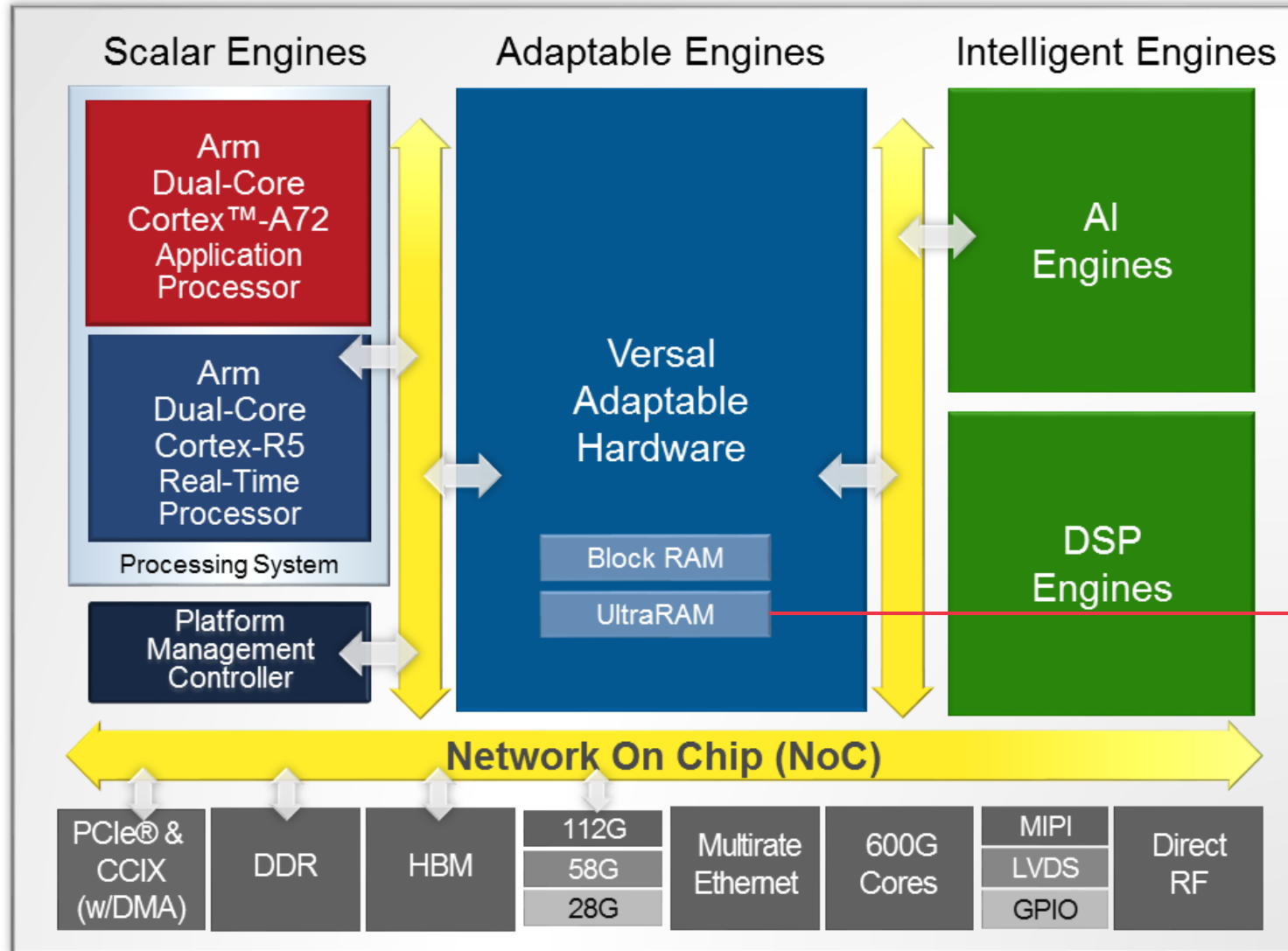
Block RAM

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

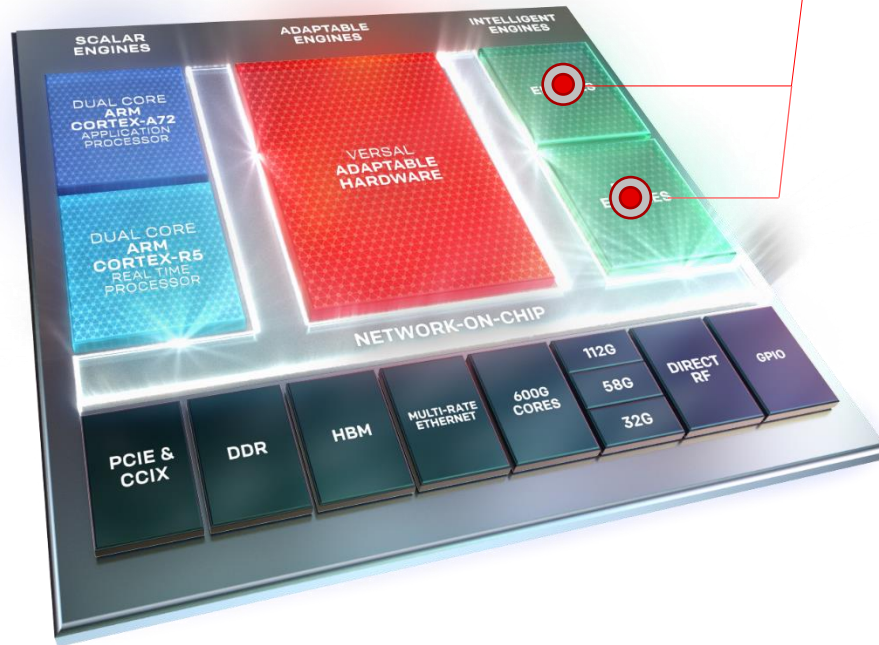


UltraRAM

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

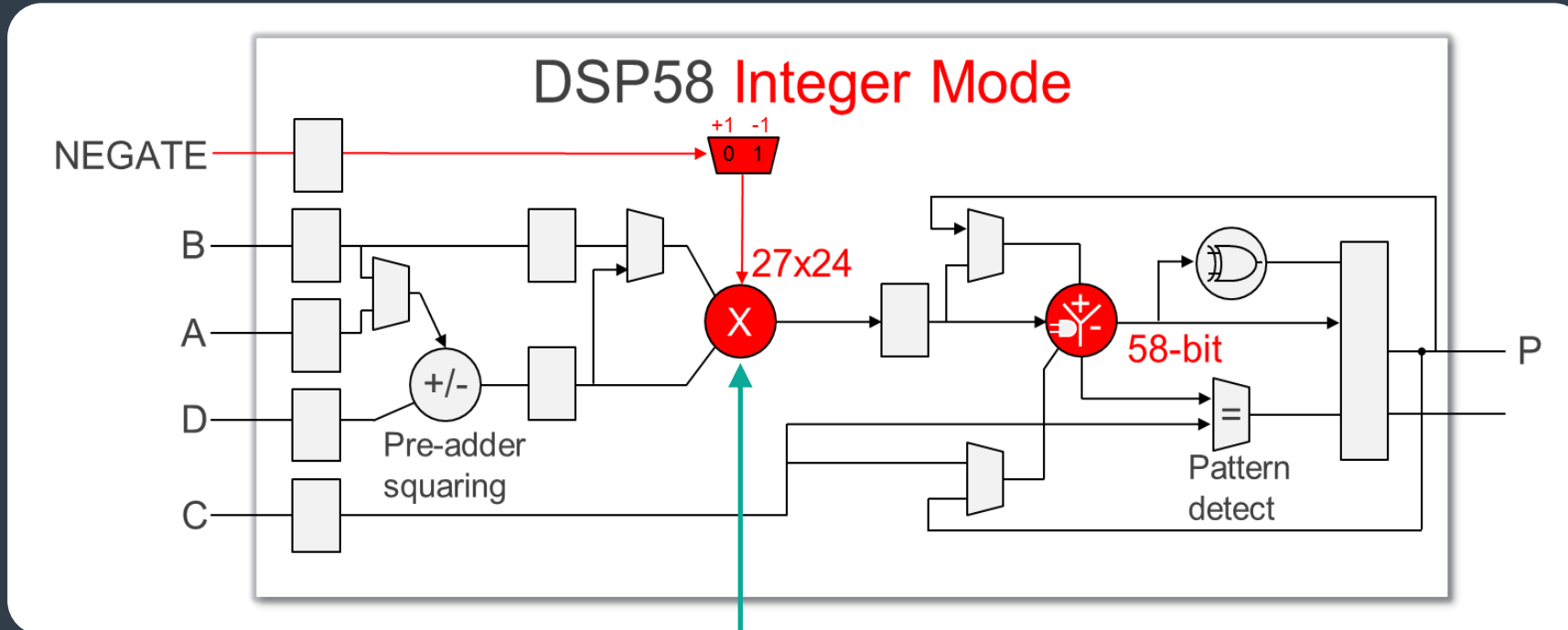
AI Engines

High throughput, low latency, and power efficient

Ideal for AI inference and advanced signal processing

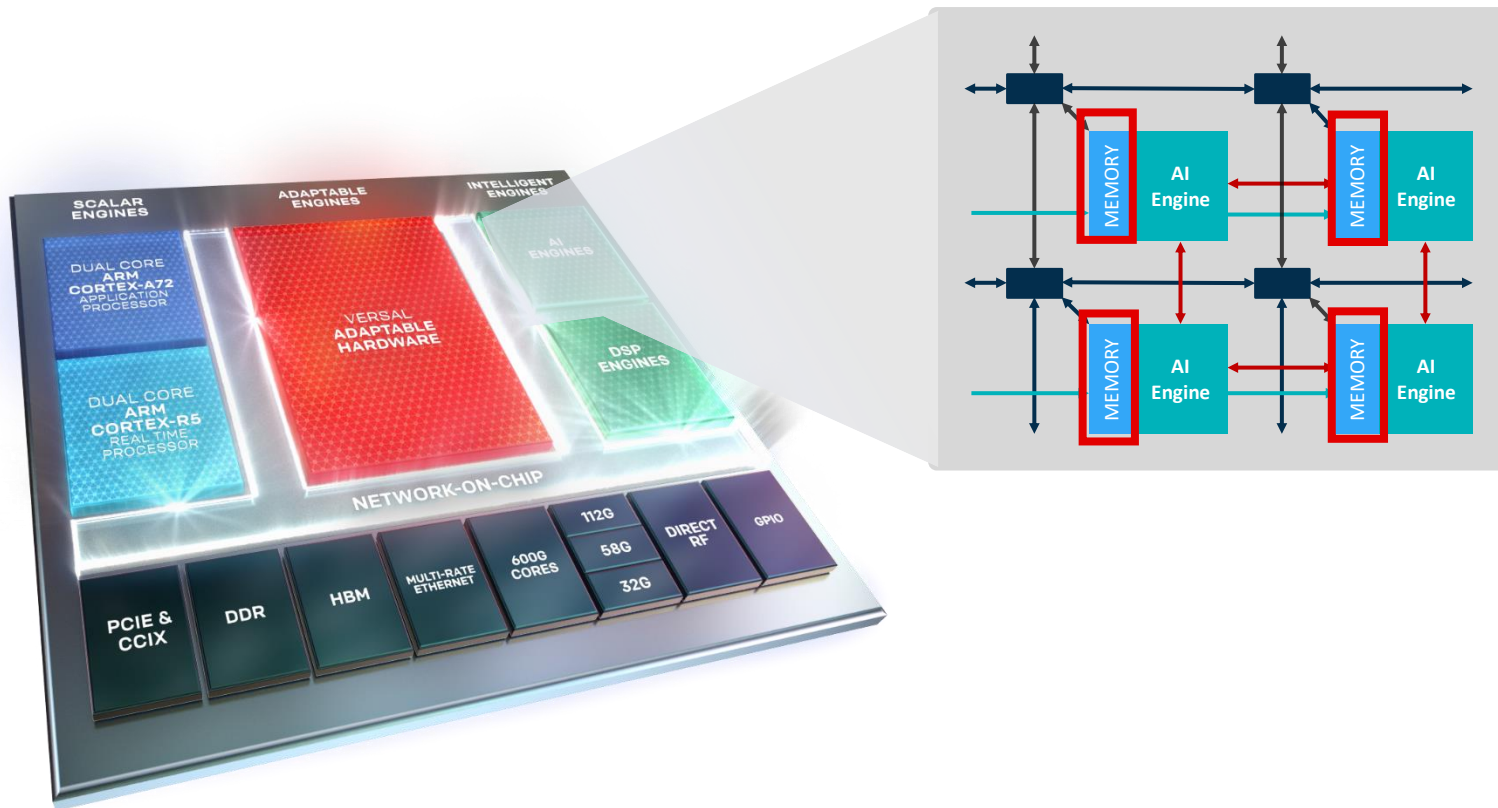
Digital Signal Processing Capability

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

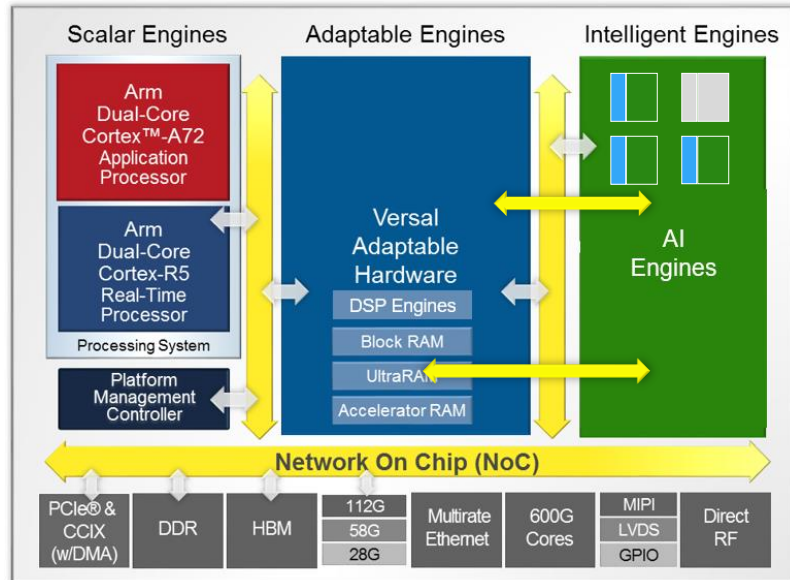
AI Engine: Hardened Compute, Memory, and Interconnect



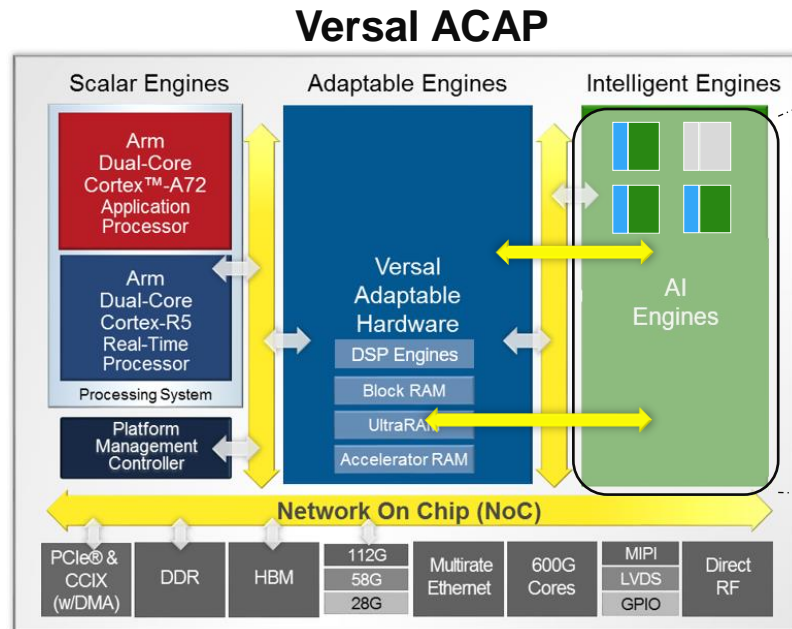
- Immense array of SIMD/VLIW processors with its own instruction and data memory
- Connected on an innovative interconnect built with massive bandwidth
- Form a tightly integrated heterogeneous compute platform
- Provide up to five times higher compute density for vector-based algorithms
- Provide that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

AI Engine: Terminology

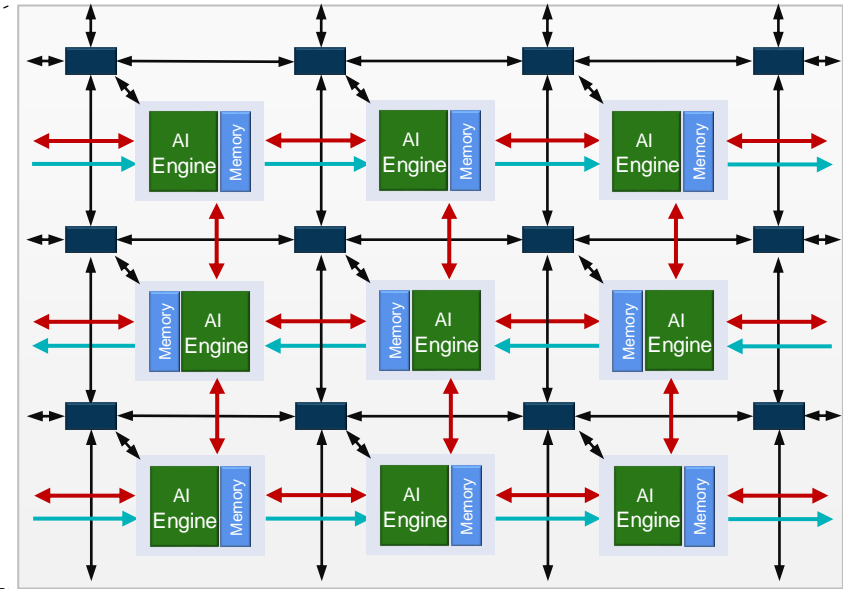
Versal ACAP



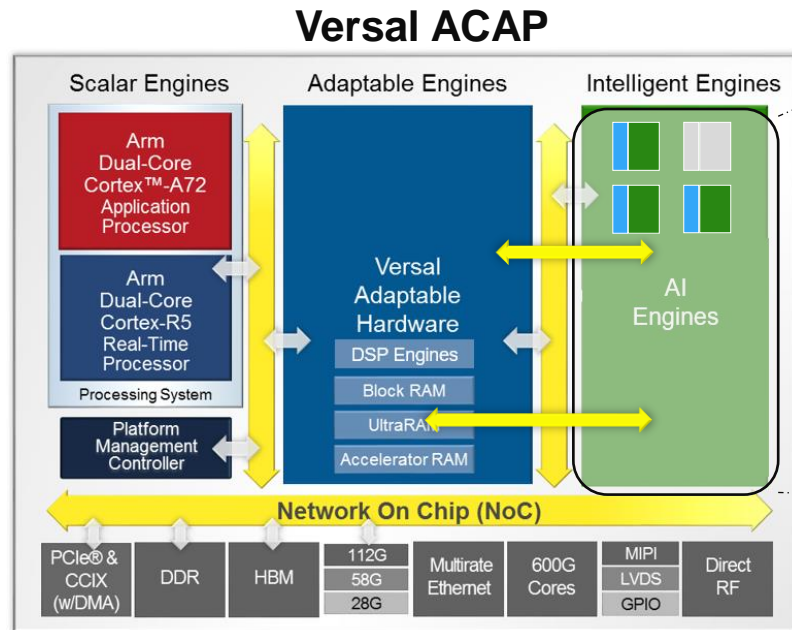
AI Engine: Terminology



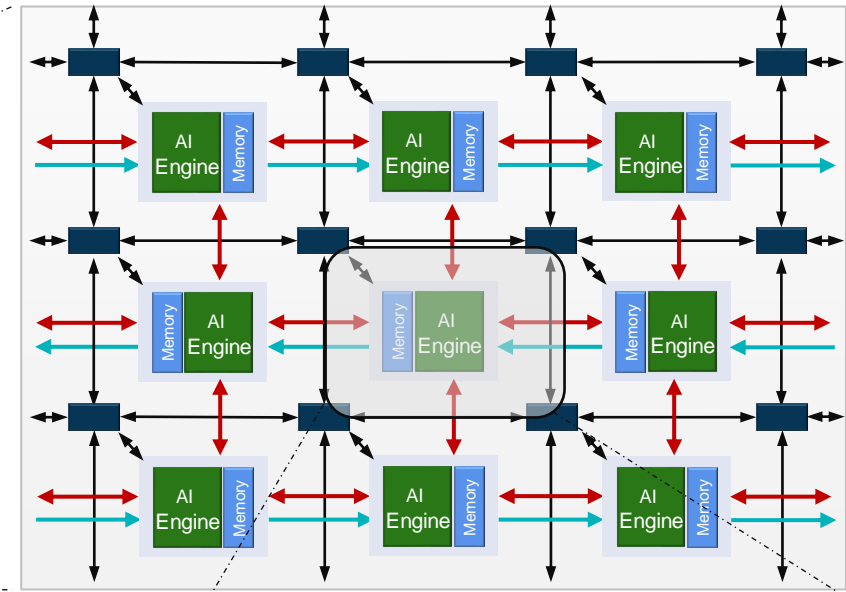
AI Engine Array



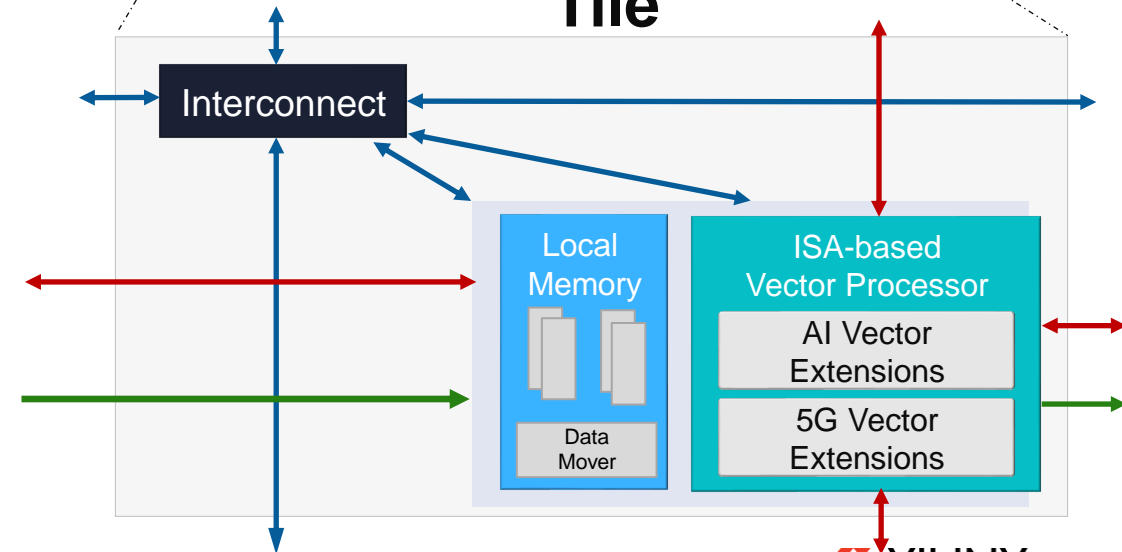
AI Engine: Terminology



AI Engine Array



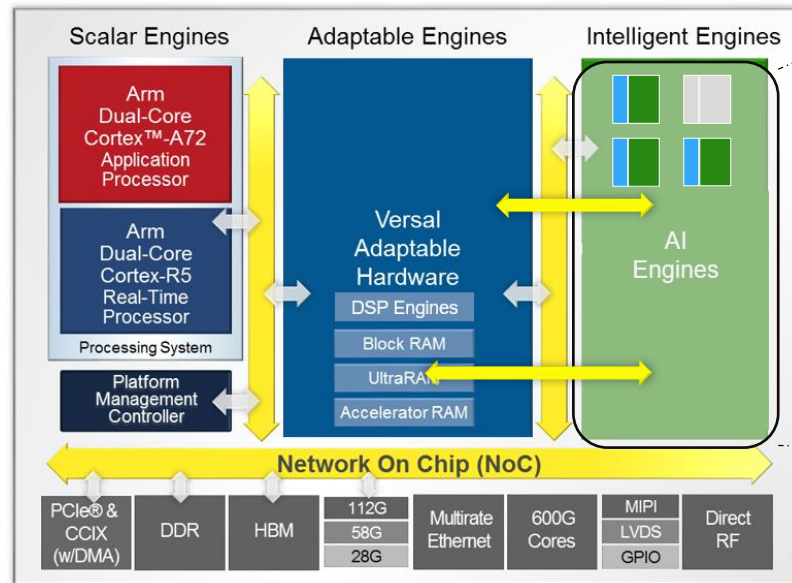
AI Engine Tile



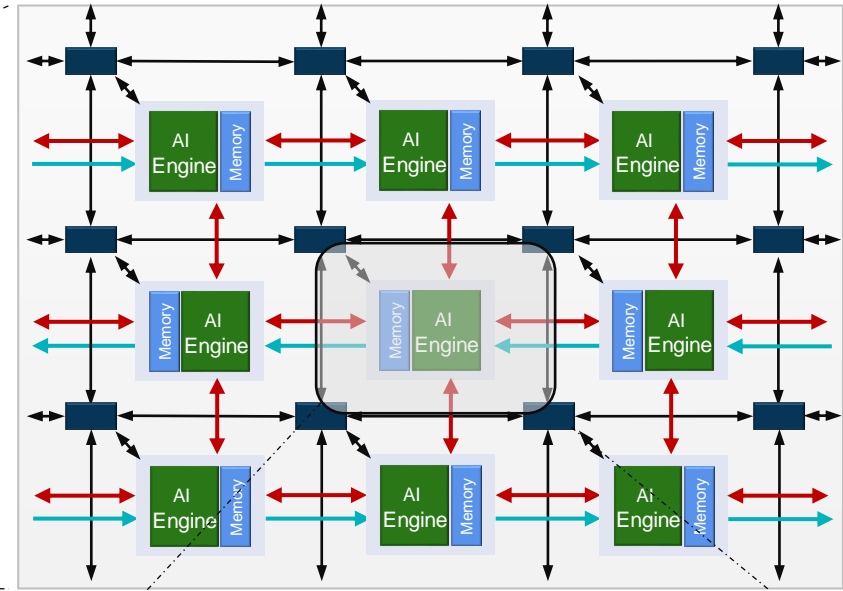
- Memory Interface
- Stream Interface
- Cascade Interface

AI Engine: Terminology

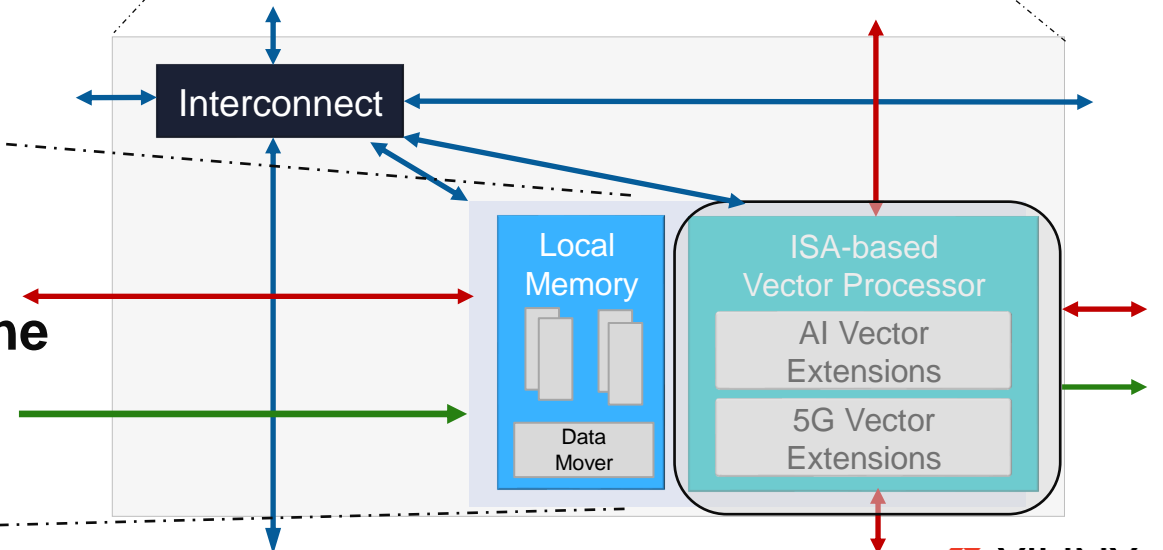
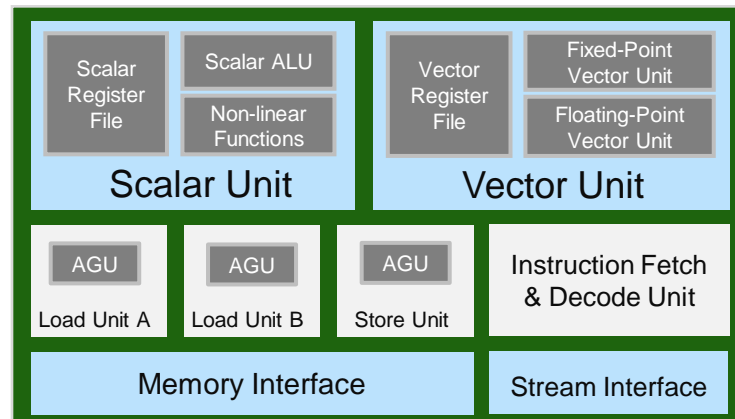
Versal ACAP



AI Engine Array

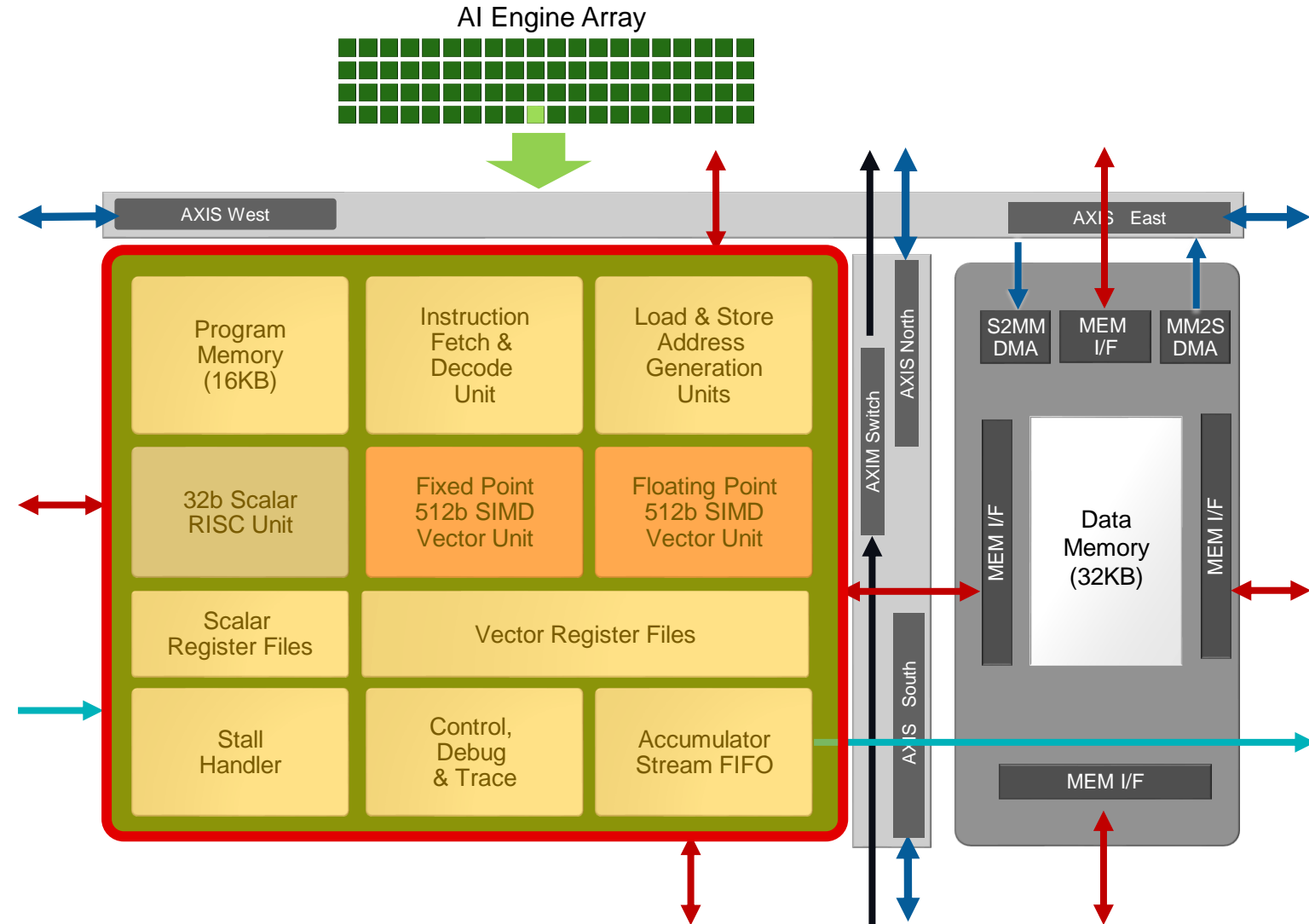
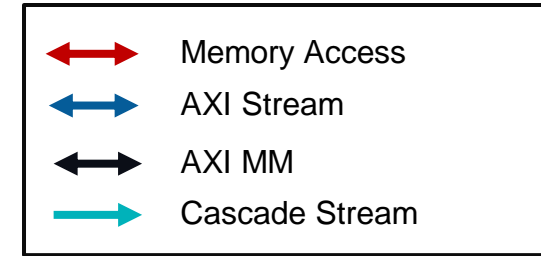


AI Engine



- Memory Interface
- Stream Interface
- Cascade Interface

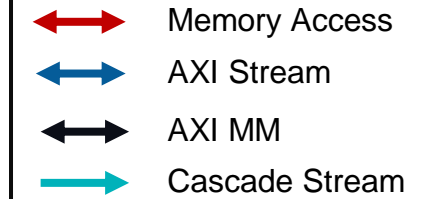
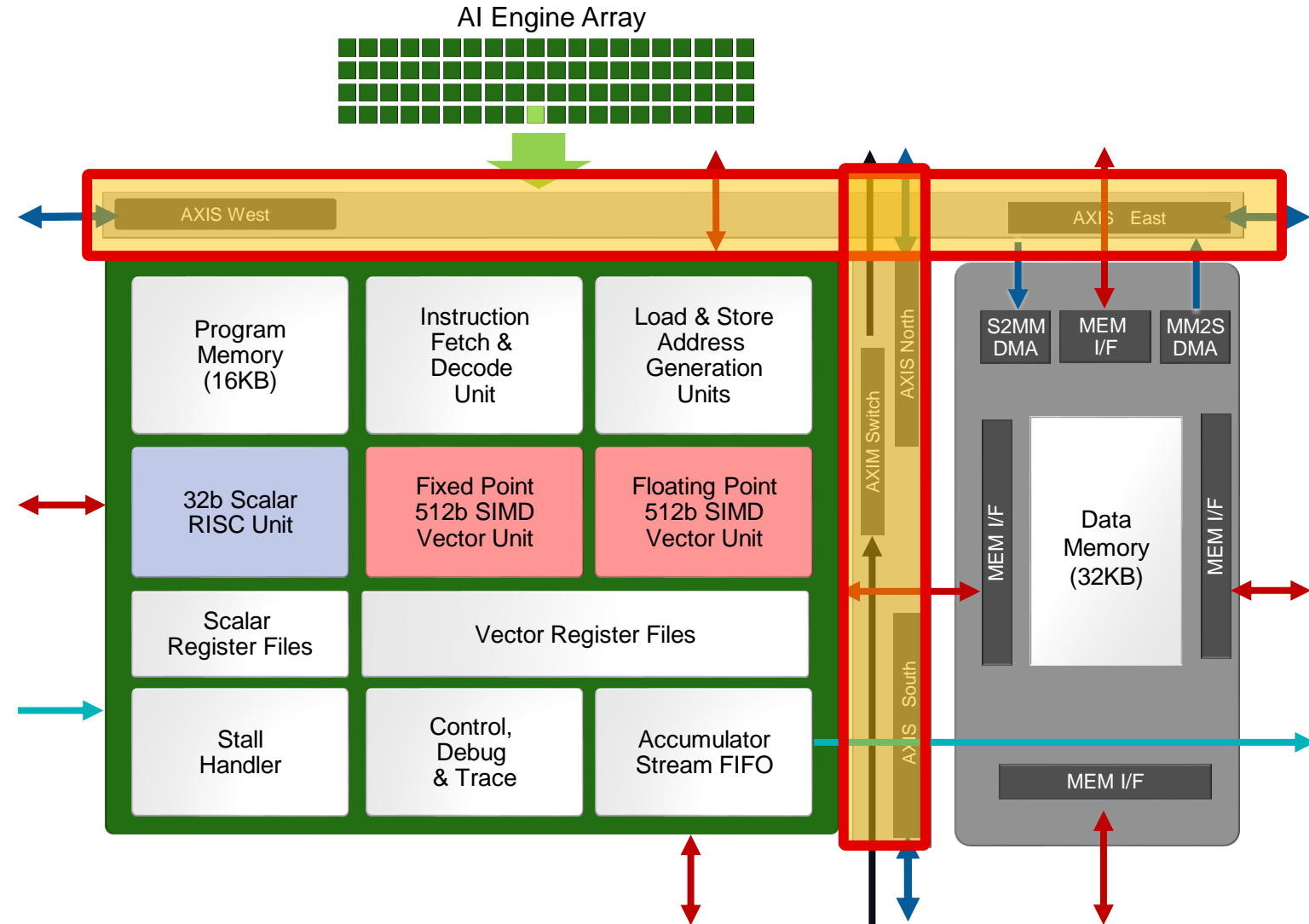
AI Engine: Tile-based Architecture



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

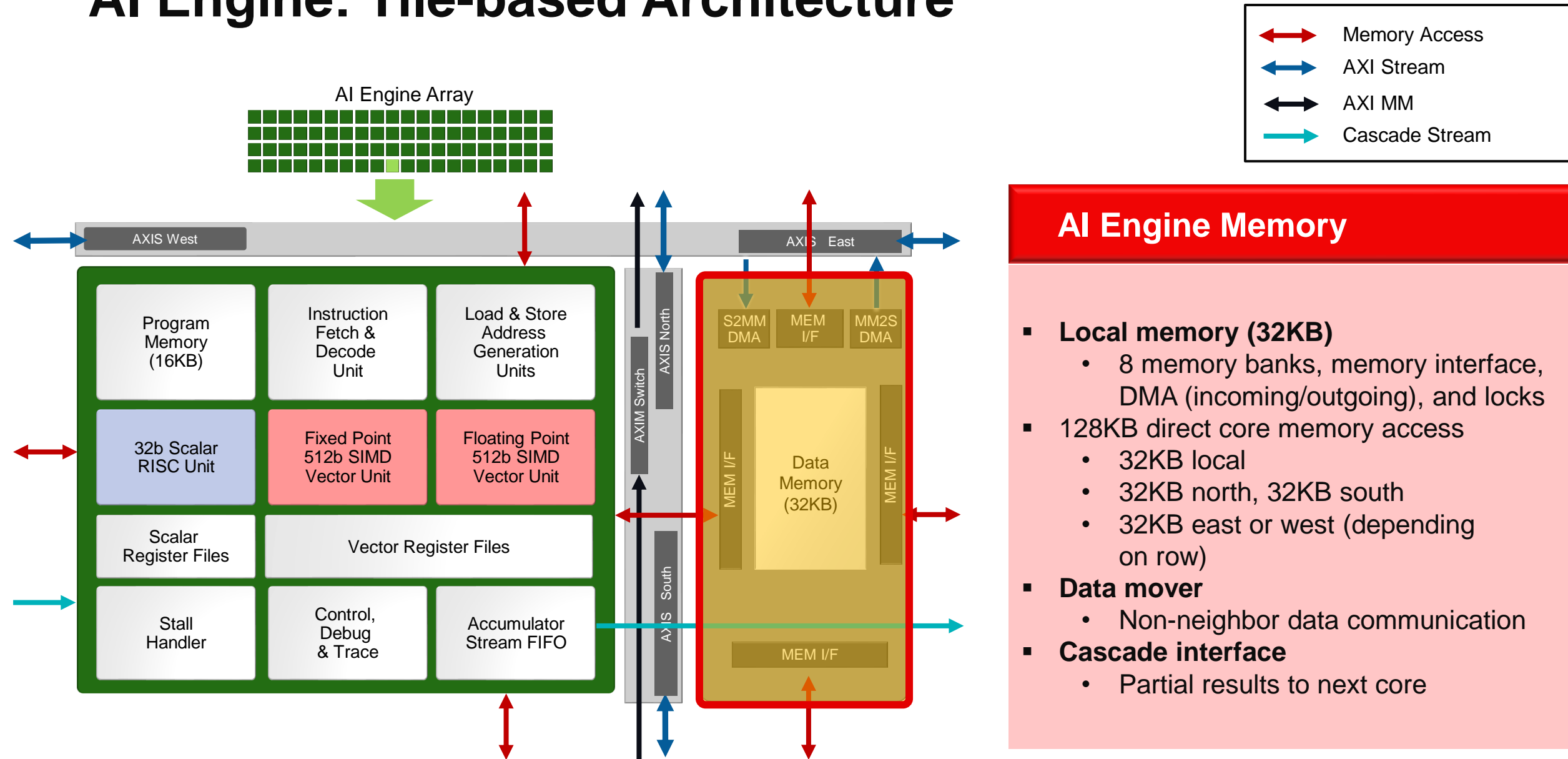
AI Engine: Tile-based Architecture



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

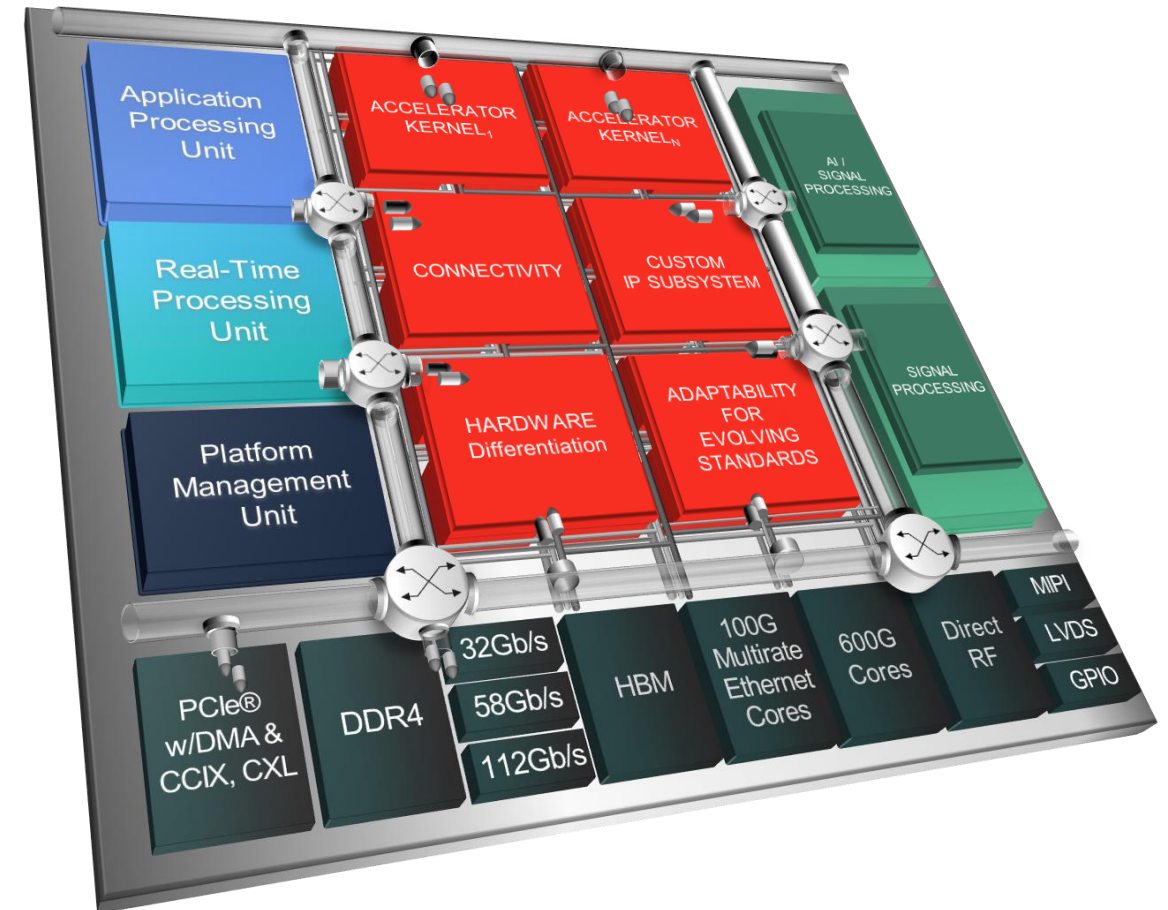
Programmable NoC: Bridging Engines & Hard IP

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

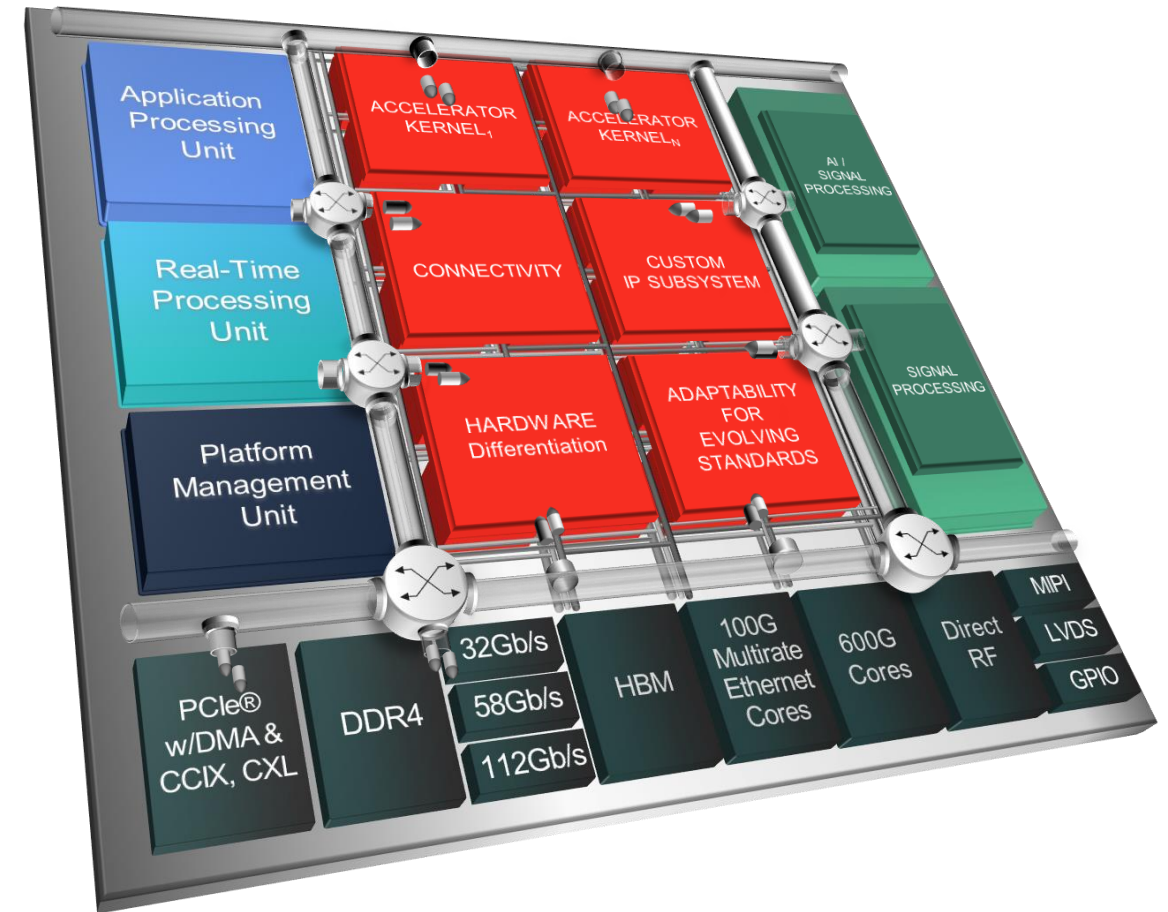
- Ability to intuitively compartmentalize the kernels with easy entry and exit points
- Ability to connect any master to any slave
- Global address maps are based on the NoC interconnect
- Configurable NoC is an AXI4-based network to route:
 - High bandwidth
 - Real time
 - Low latency
- Extends in both 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



Programmable NoC: Bridging Engines & Hard IP

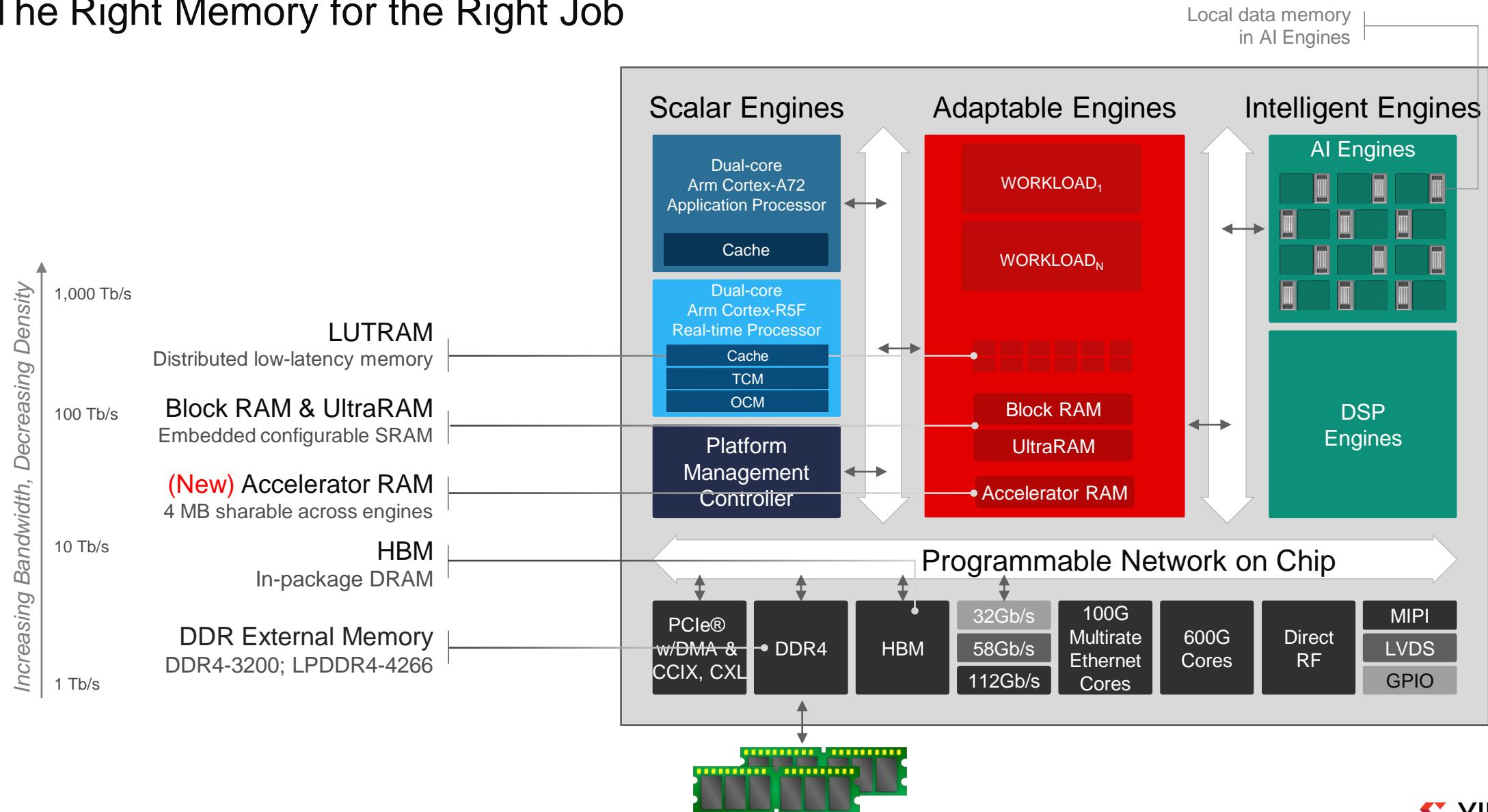
Programming Framework

- 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>



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>



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

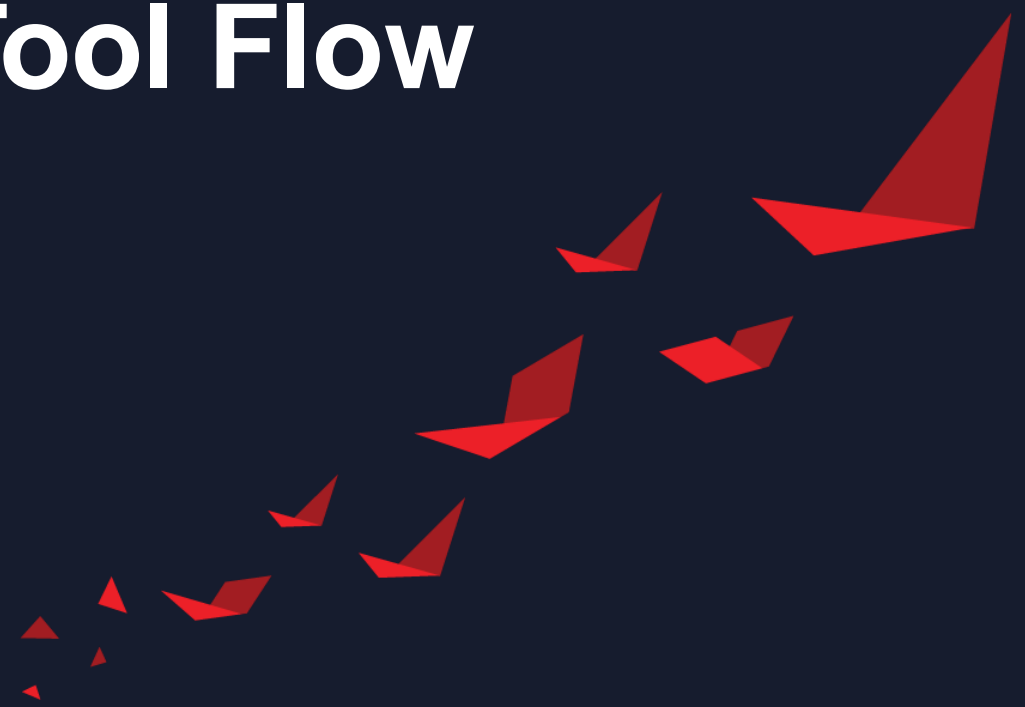
	VC1352	VC1502	VC1702	VC1802	VC1902	VC2602	VC2802
AI Engines	128	198	304	300	400	0	0
AI Engines-ML	0	0	0	0	0	152	304
AIE/AIE-ML Data Memory (Mb)	32	50	76	75	100	76	152
AIE-ML Shared Memory (Mb)	0	0	0	0	0	304	304
DSP Engines	928	1,032	1,312	1,600	1,968	984	1,312
System Logic Cells	539,840	814,520	981,120	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)

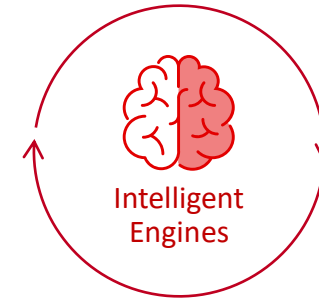
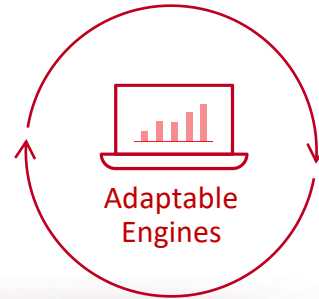
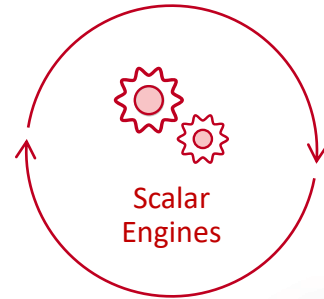


Versal ACAP: Design Tool Flow



Adaptive Compute Acceleration Platform (ACAP)

COMPUTE ACCELERATION



Customizable memory hierarchies

NoC to move data

ADAPTIVE

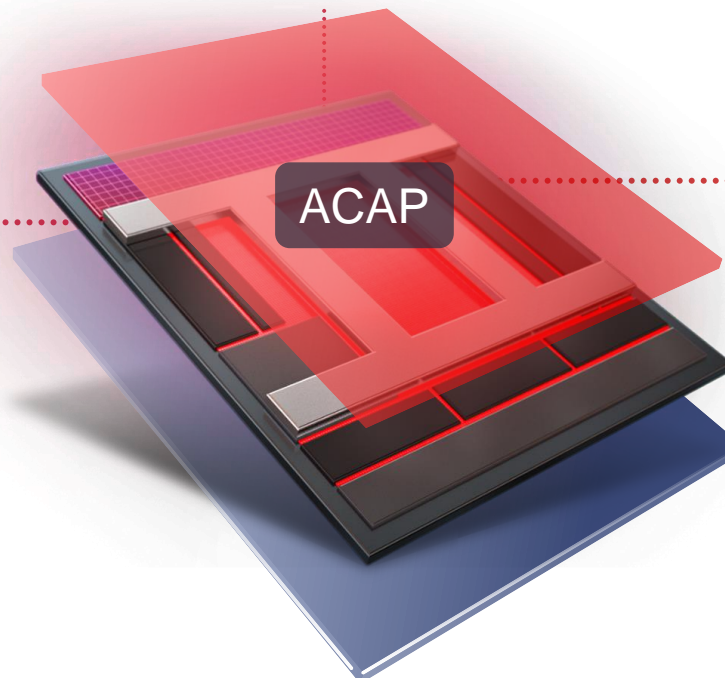
Future Proof for
New Algorithms

Support for Common
Standards

PLATFORM

Single Heterogeneous Platform
for HW/SW Engineers

Frameworks, Libraries, SW,
Runtime Stack



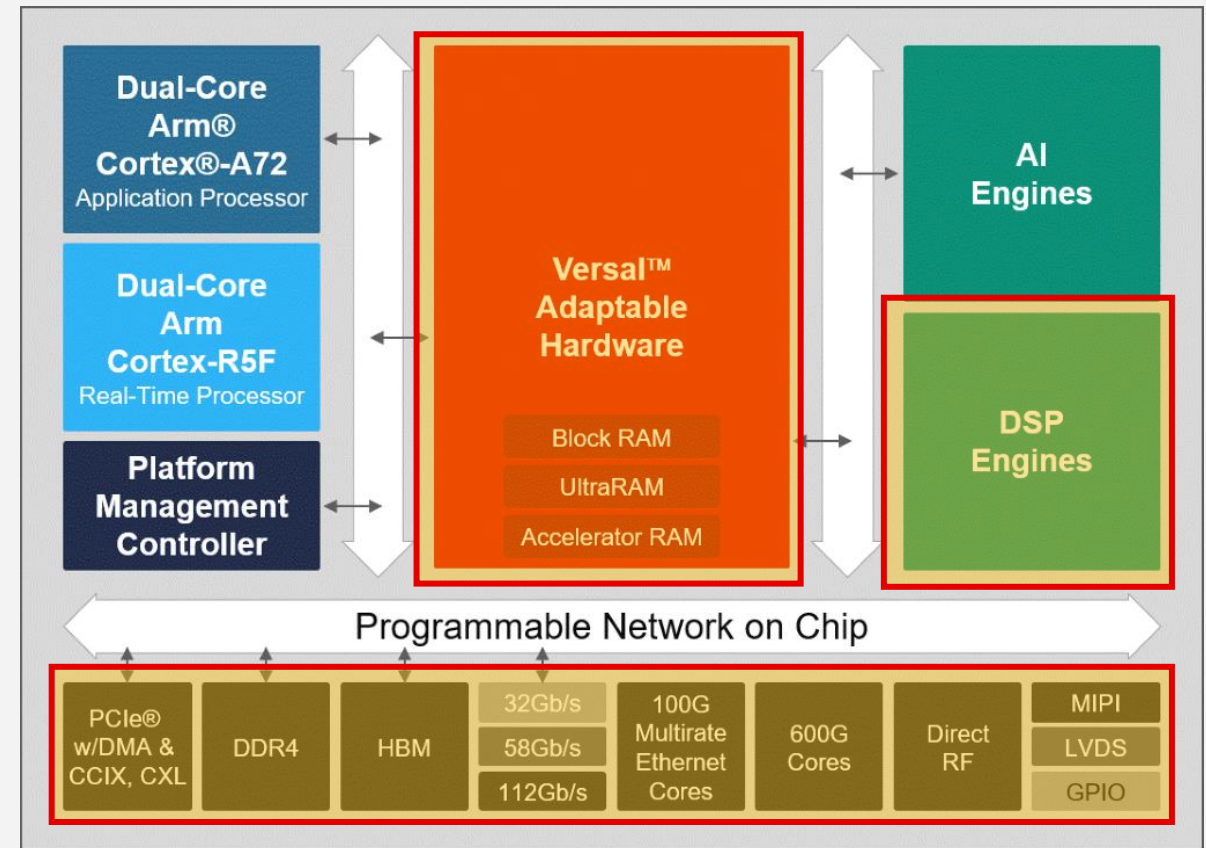
Enabling Data Scientists, Software Developers, Hardware Developers

Different Capabilities Means Different Tools

Tools for Development PL and Its Associated Resources



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

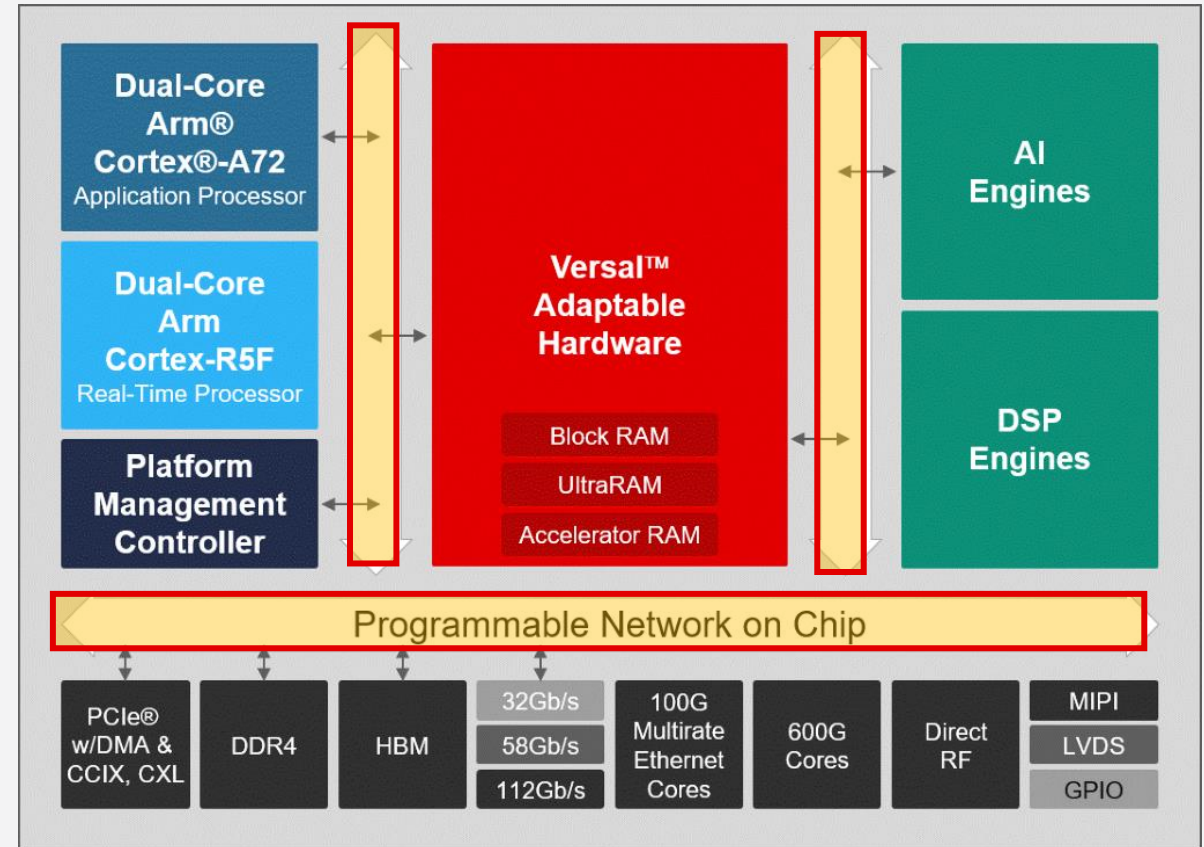


Different Capabilities Means Different Tools

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

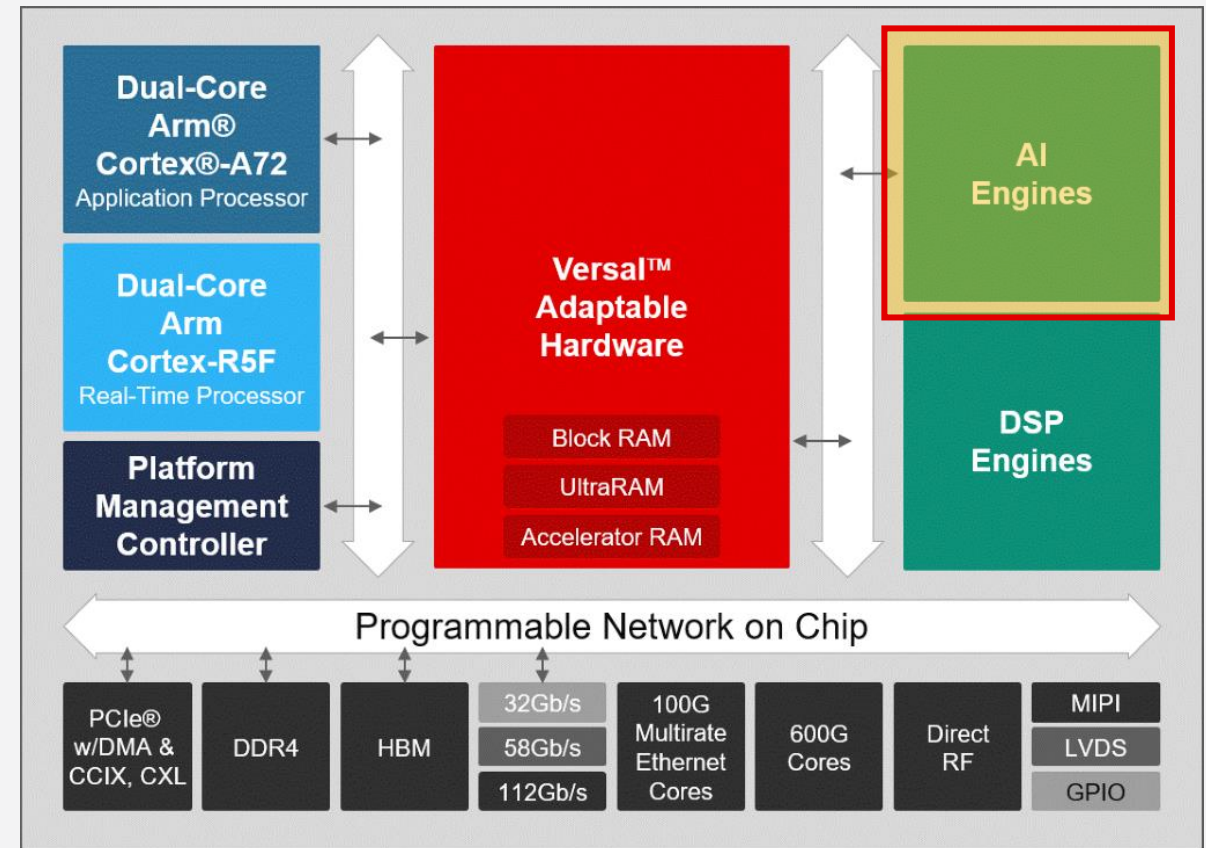


Different Capabilities Means Different Tools

Tools for Development Adaptable Intelligent Engines

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

aiecompiler

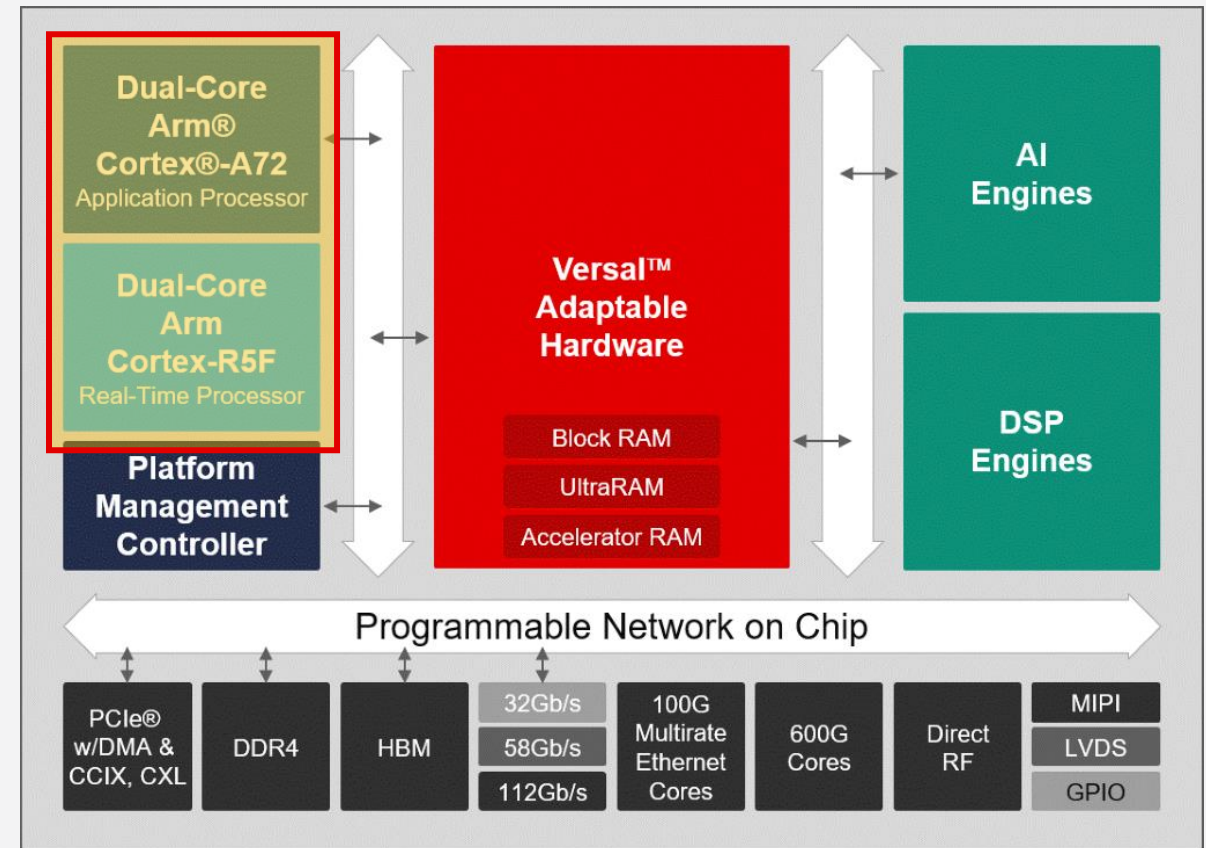


Different Capabilities Means Different Tools

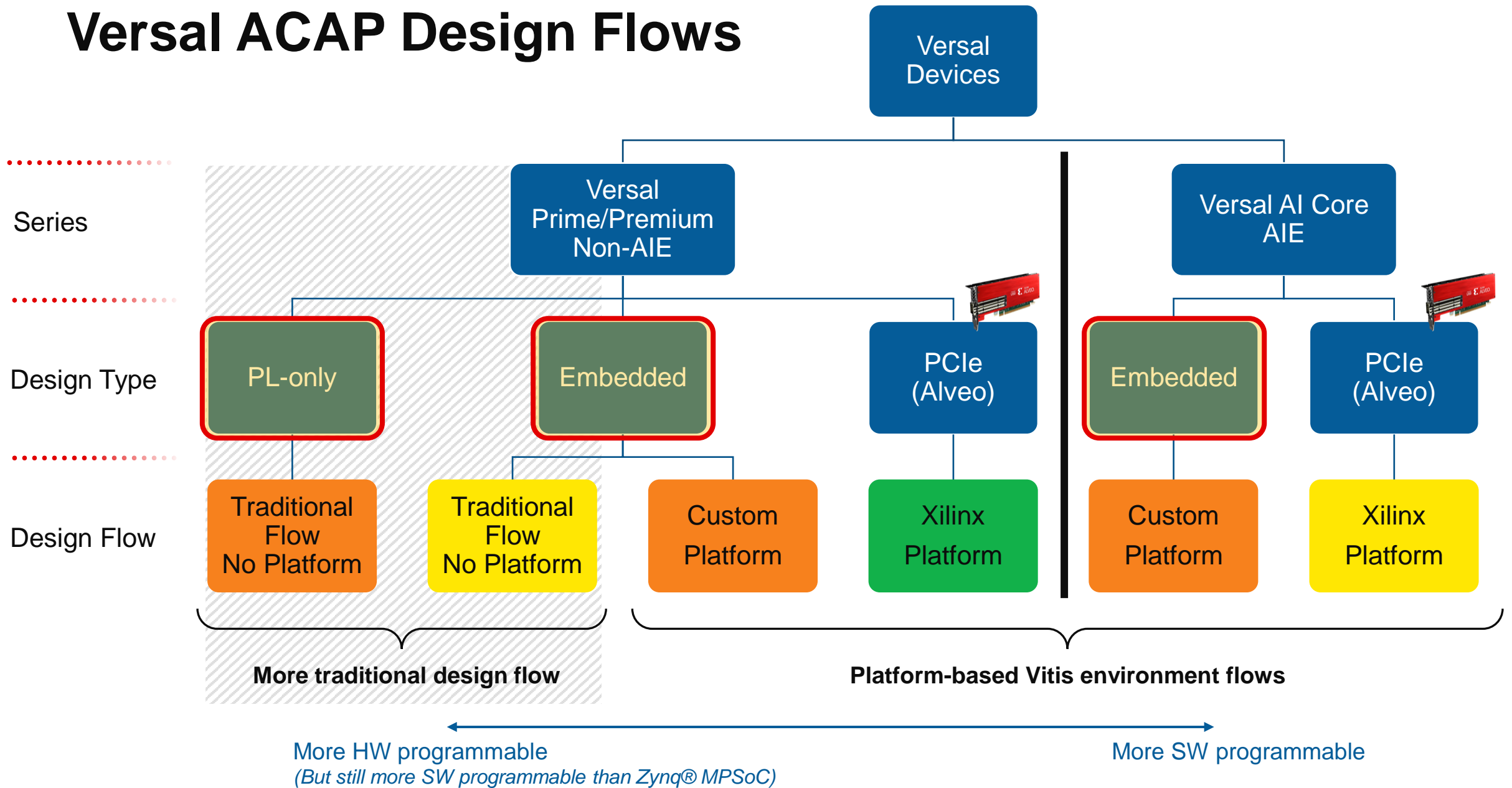
Tools for Development Scalar Processors



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



Versal ACAP Design Flows



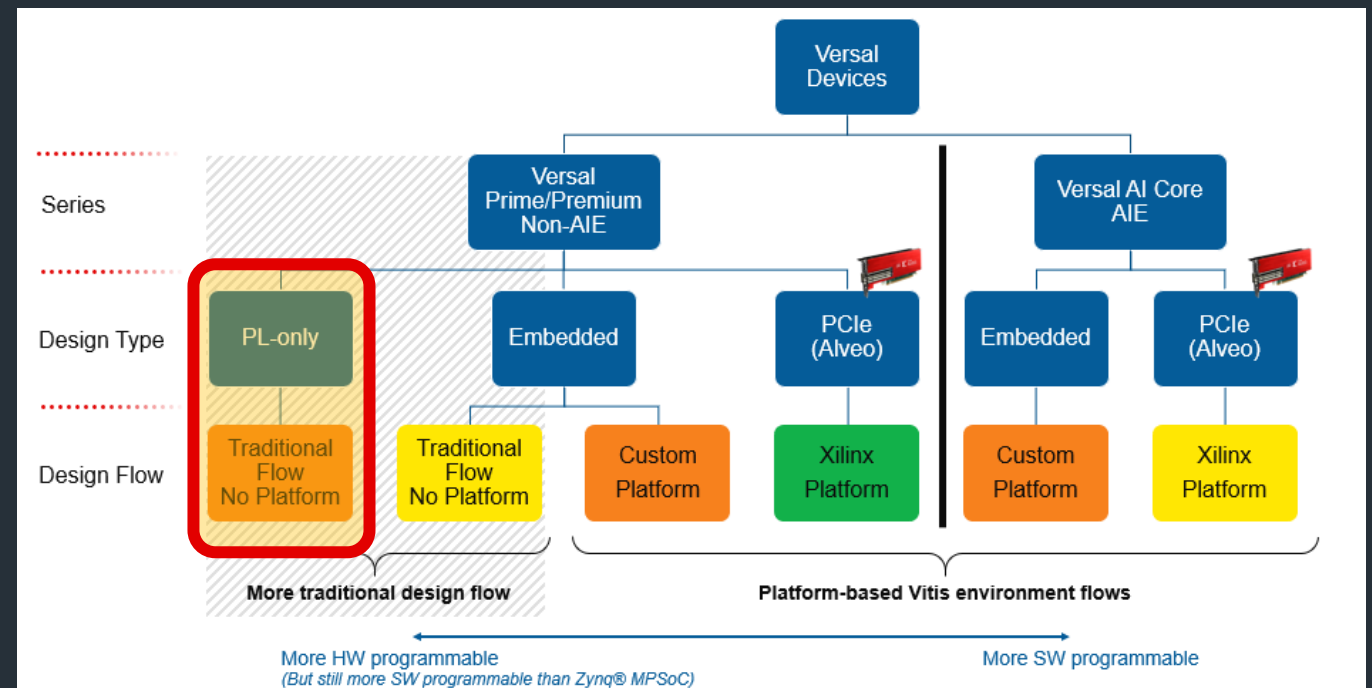
Versal ACAP Design Flows

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

Embedded System

Comprises of:

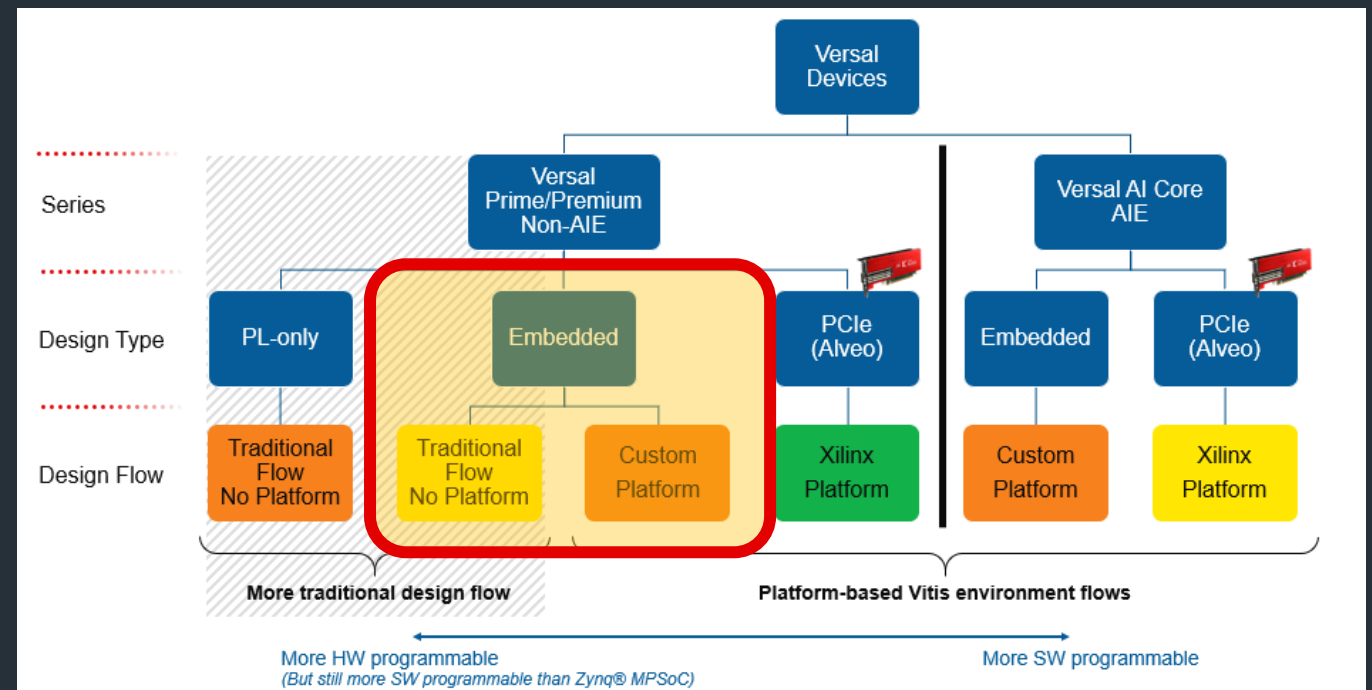
- Embedded processor
- Acceleration logic built into the traditional PL

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

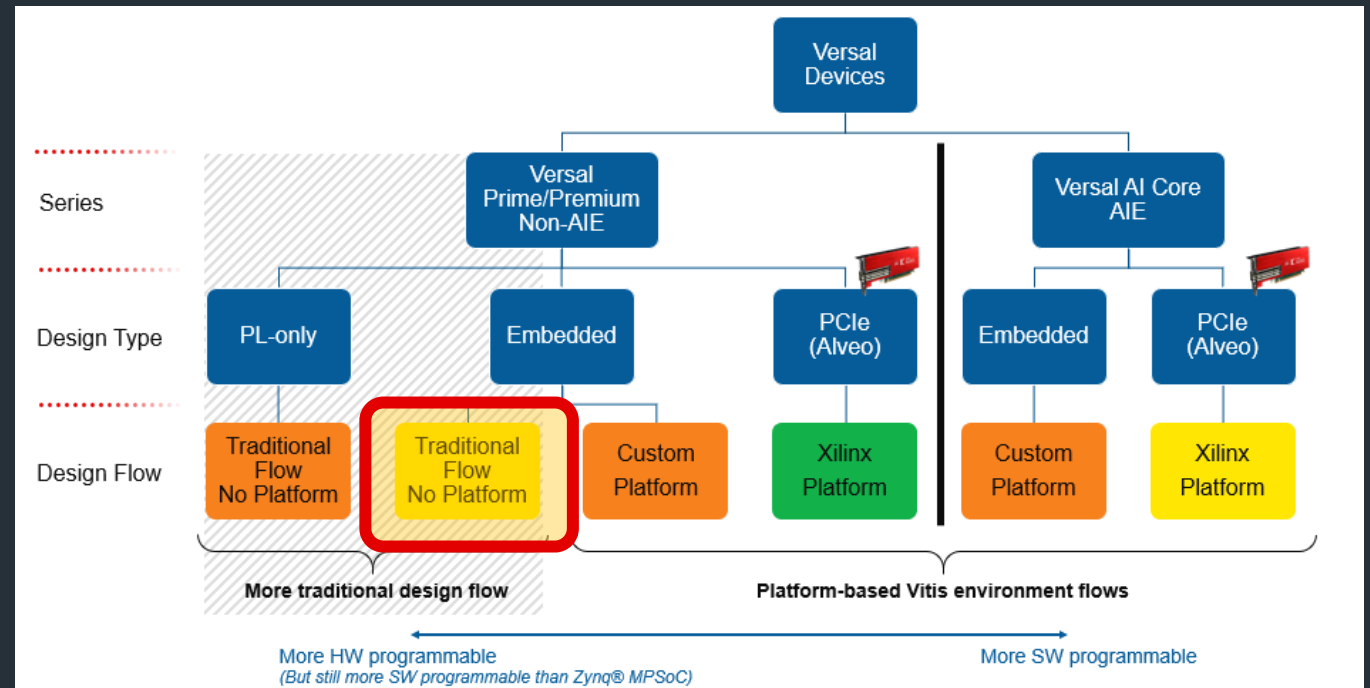


Versal ACAP Design Flows

Embedded System

Traditional flow:

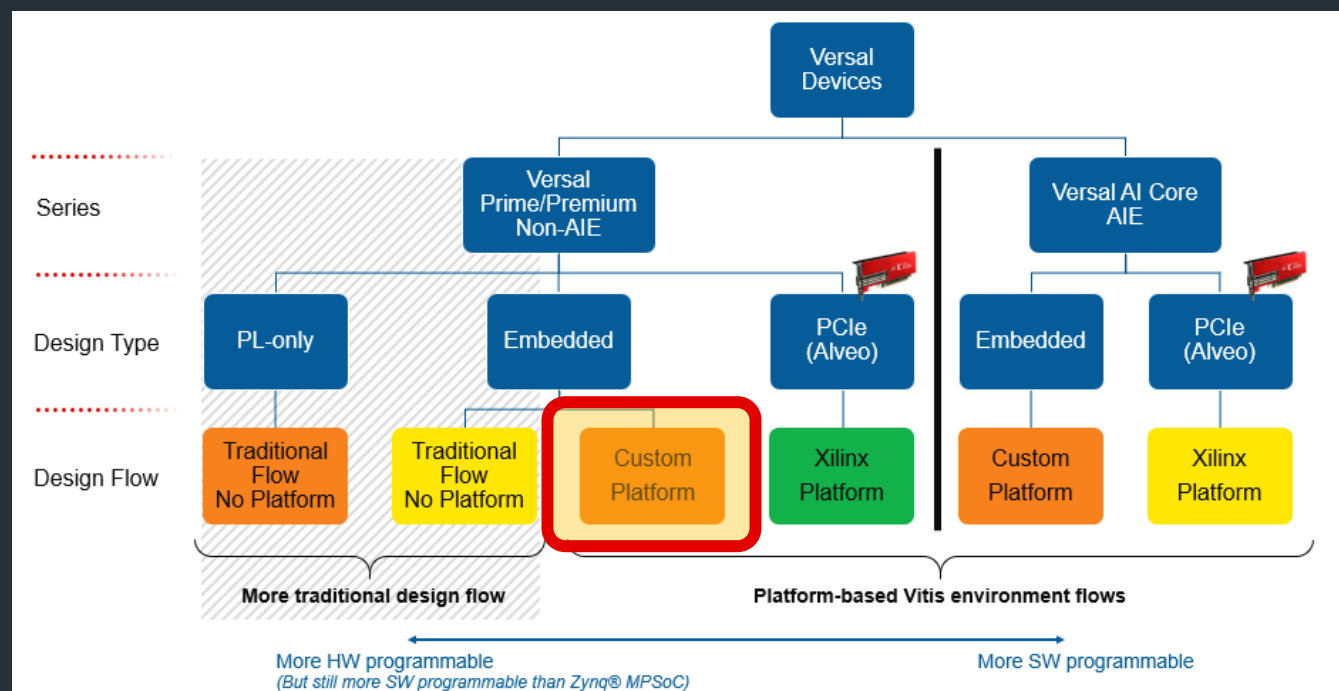
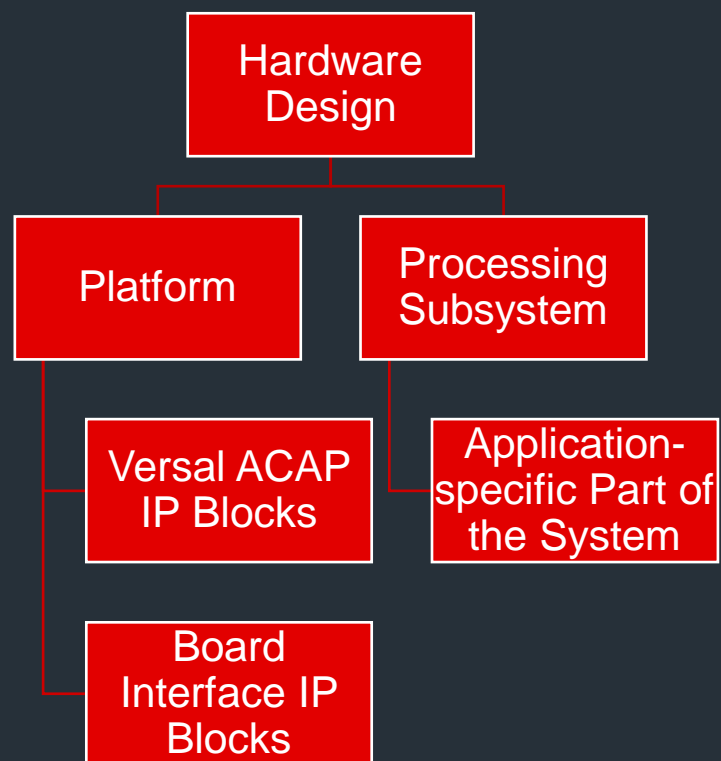
- No platform
- Hardware design considered a *fixed platform*
 - Vivado tools create the programmable logic on the device
 - Vitis tools cannot modify the PL



Versal ACAP Design Flows

Embedded System

Platform-based flow:



Versal ACAP Design Flows

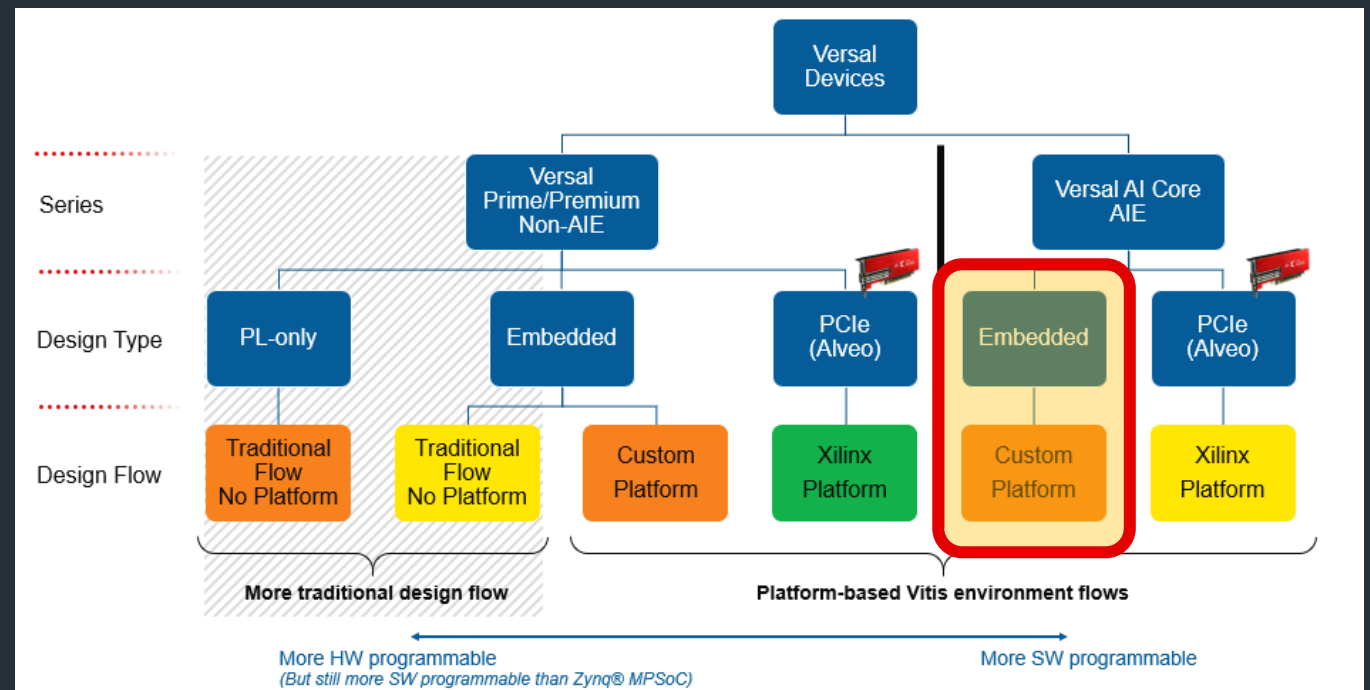
Embedded AI Engine System

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

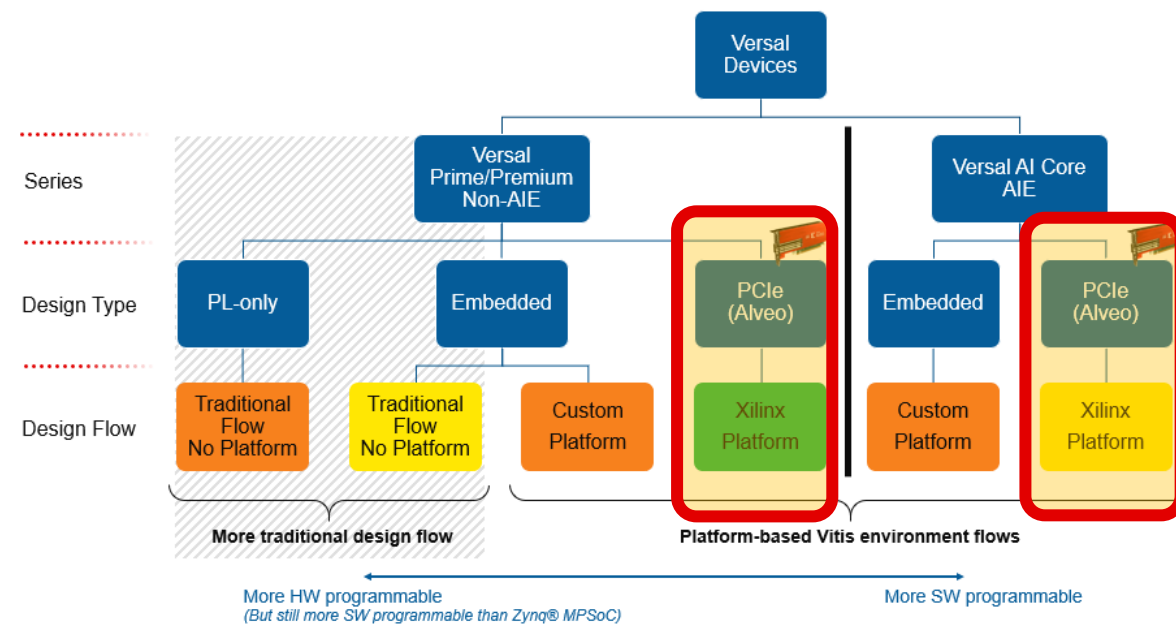


Versal ACAP Design Flows

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

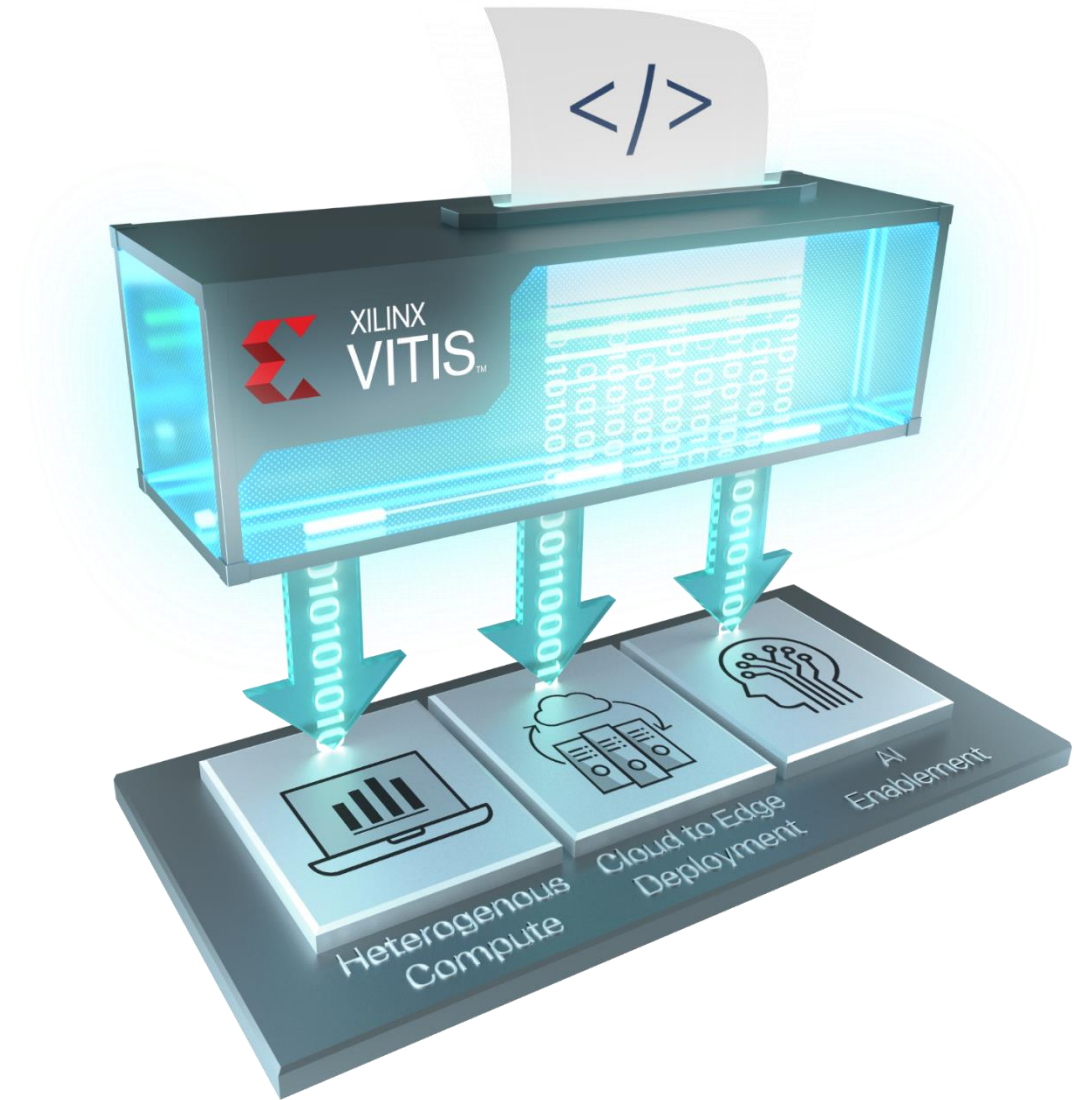


Vitis Unified Software Platform



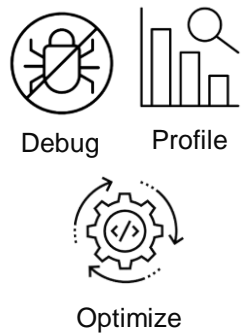
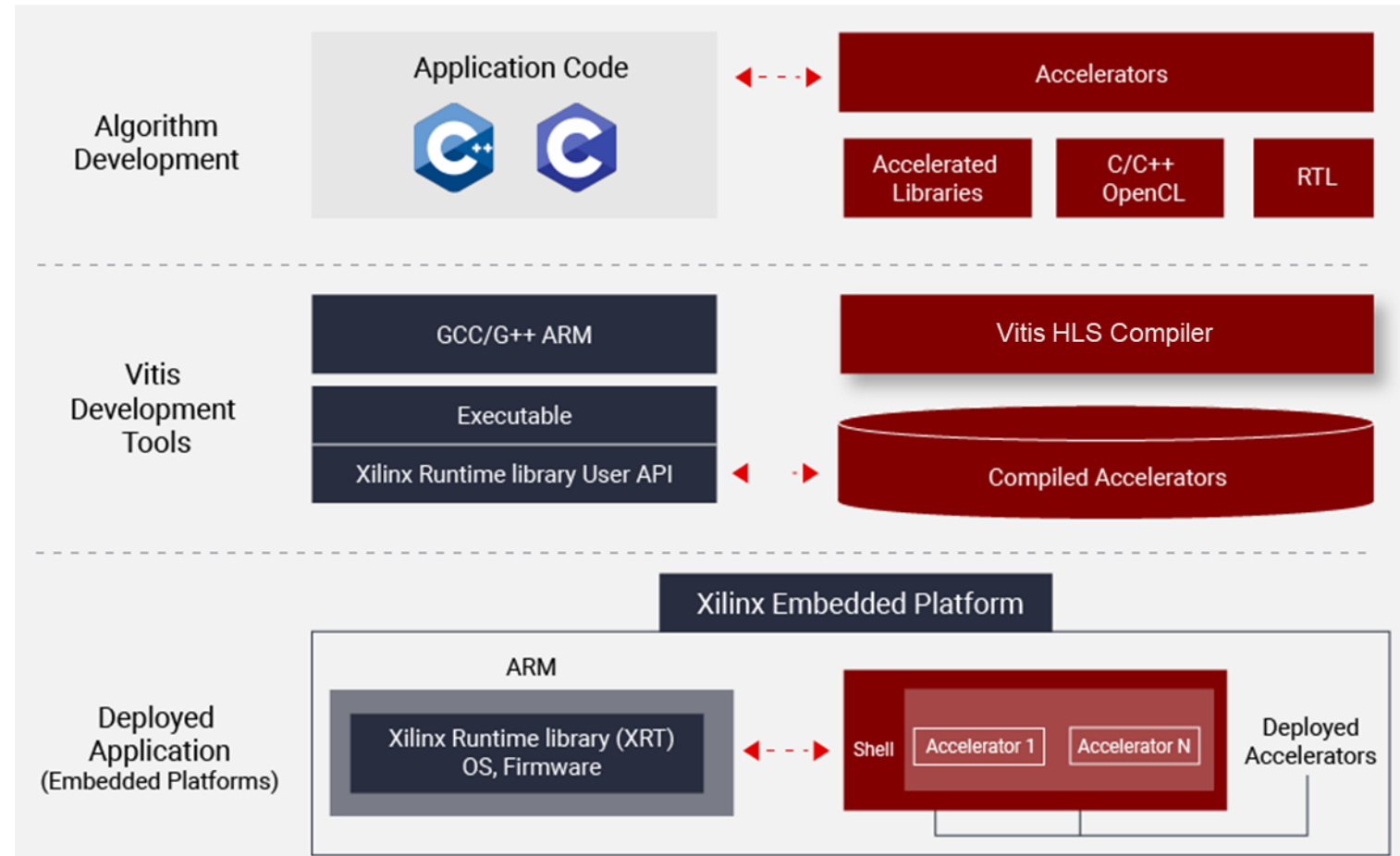
Unified Software Platform

- 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



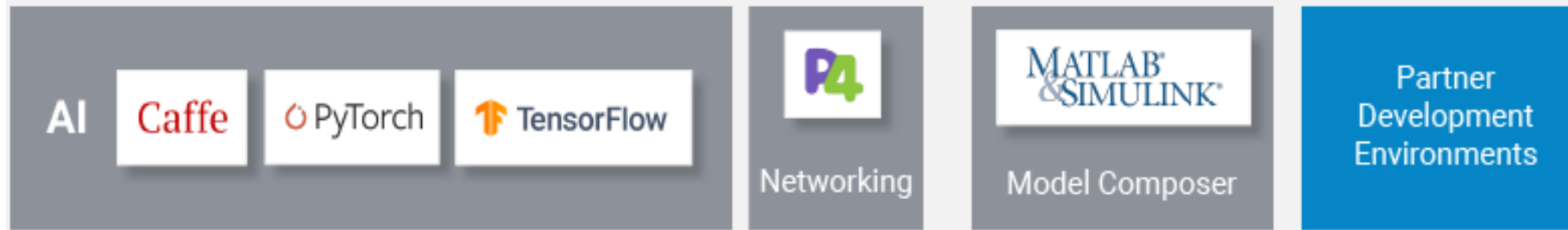
GUI and Command-line Support
Eclipse Open-source Standard

Vitis Unified Solution Stack

For Heterogenous Compute, Edge to Cloud

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

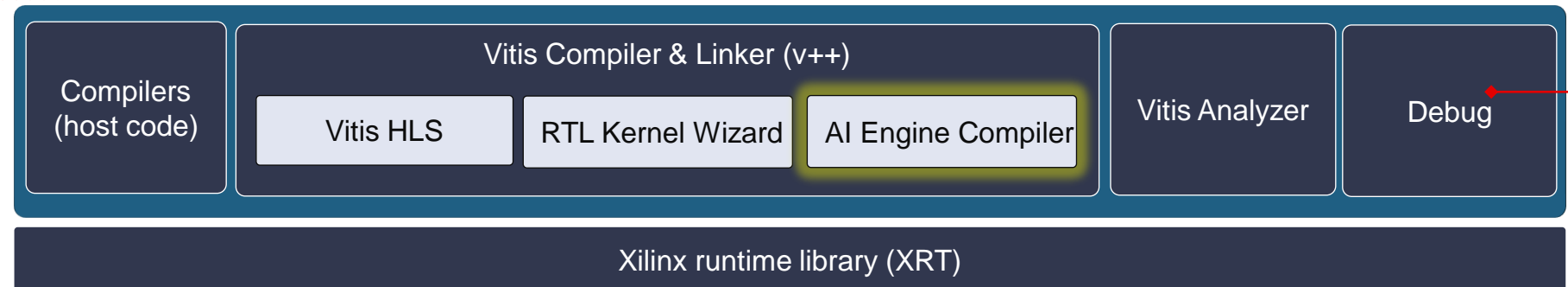
Domain-specific
development
environments



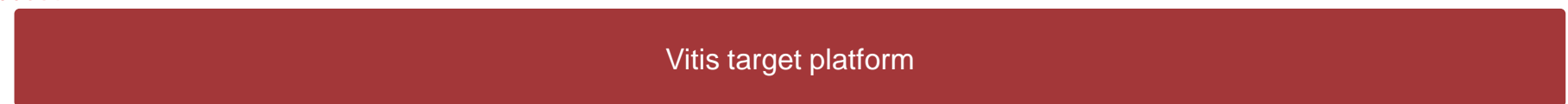
Vitis accelerated
libraries



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



Thank You

