

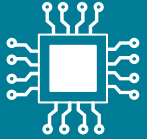


2023 Vitis™ Developer Webinar Series

AI Engine Architecture and Tools Overview

**Florent Werbrouck - Technical Marketing Manager
AI Engine and DSP**

Agenda for Today



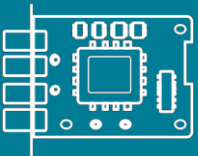
Introducing Versal™ Adaptive SoCs



Versal AI Engine Architecture Overview



AI Engine Programming Flow and Tools Introduction



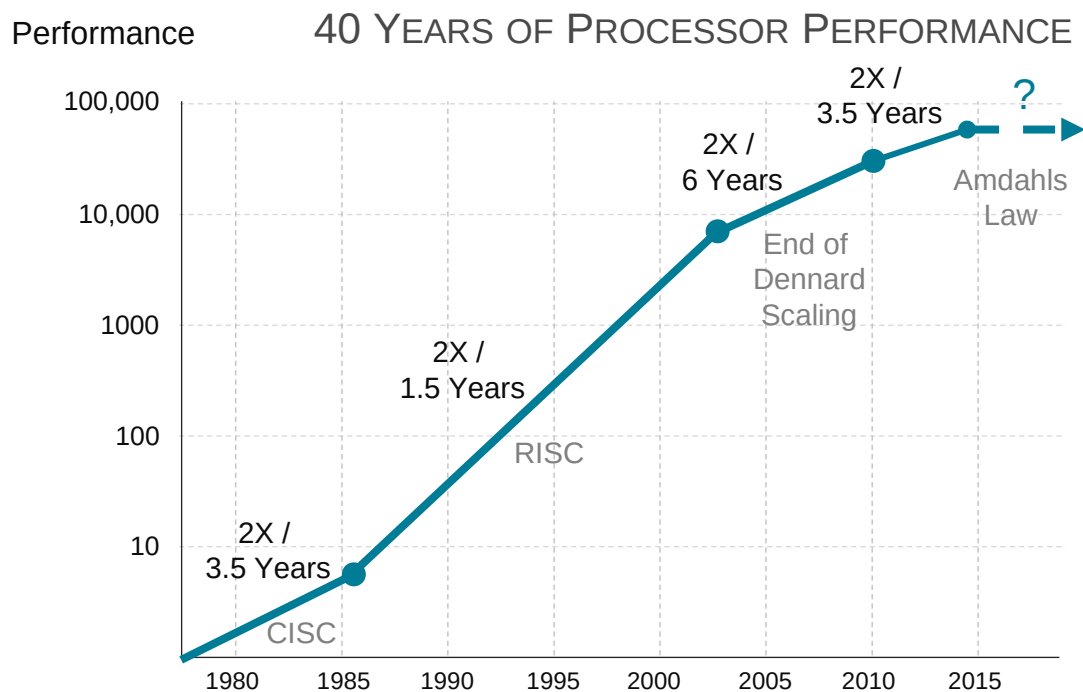
Resources to Get Started



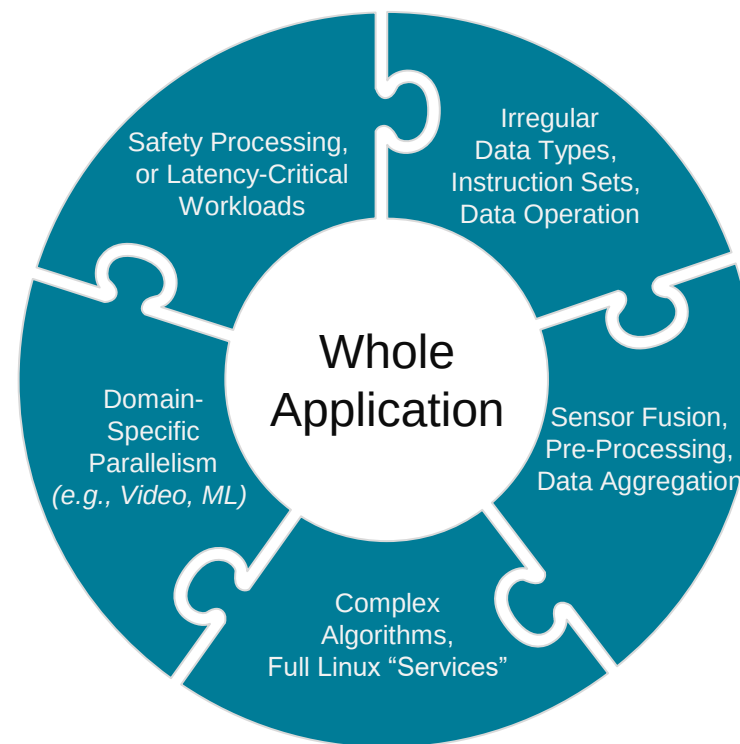
Introducing Versal™ Adaptive SoCs

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

Processing Architectures
are Not Scaling



A Single Architecture
Can't Do it Alone

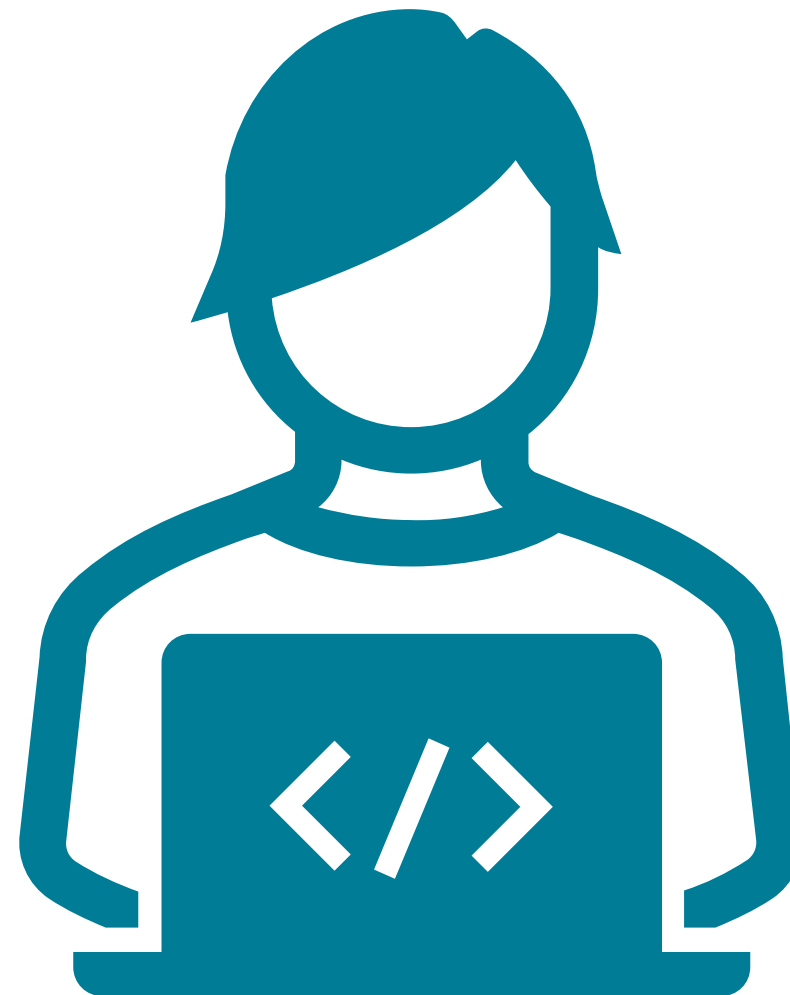


Today's Developer Needs:

Software programmability

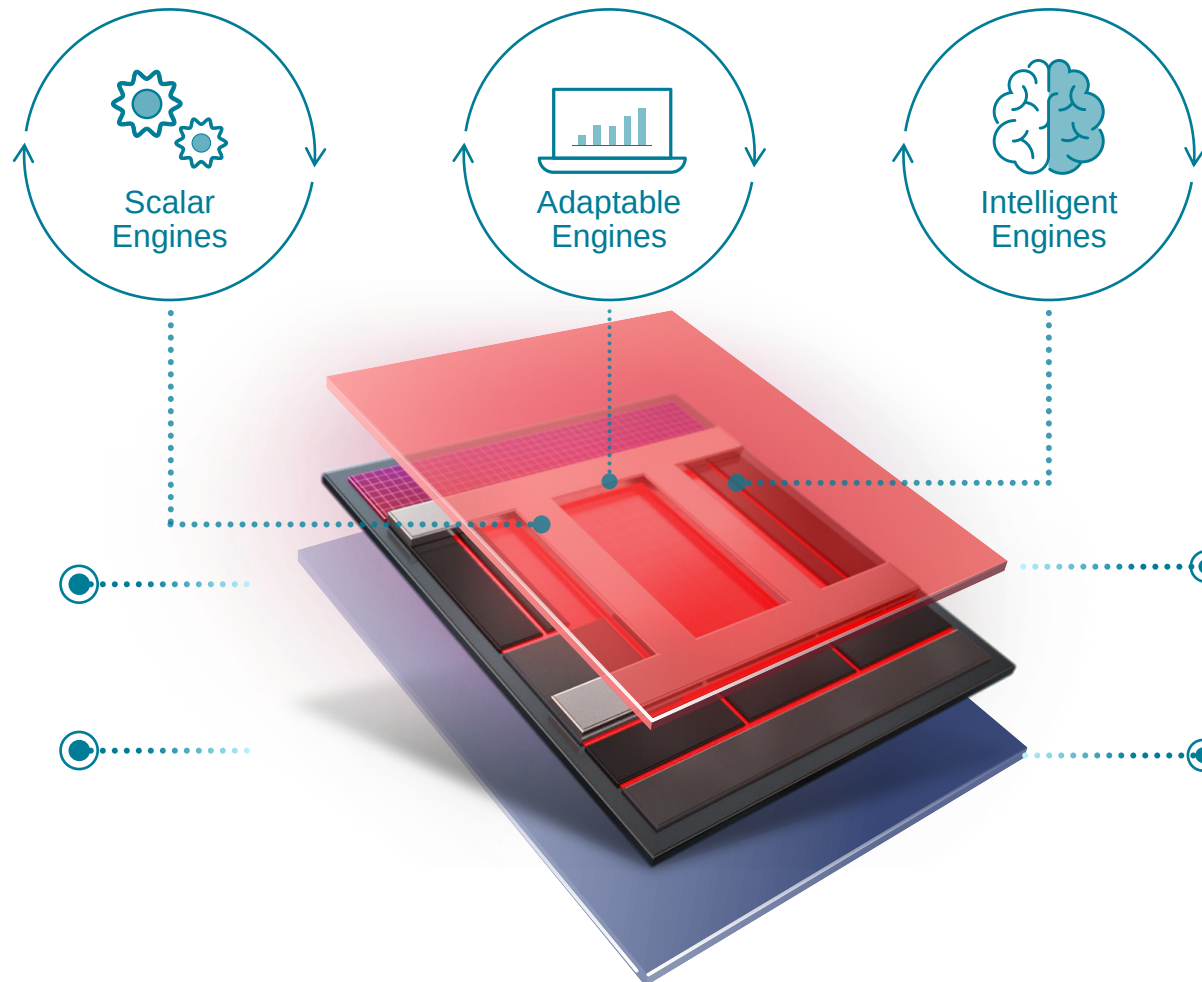
Performance for a diverse
range of applications

Adaptability to keep pace
with rapid innovation



New Class of Adaptive SoCs

COMPUTE ACCELERATION



AMD
VERSAL

PLATFORM

Development Tools
HW/SW Libraries
Runtime Stack

SW Programmable
Silicon Infrastructure

Enabling Data Scientists, SW Developers, HW Developers

AMD VERSAL

AI Edge
Series

AI Core
Series

Prime
Series

Premium
Series

HBM
Series

All Devices of the Series Include the AI Engine
Some Devices of the Series Include the AI Engine

Versal™ AI Engine Architecture Overview

Introduction to the AI Engine

First introduced in Versal™ AI Core series

Software programmable

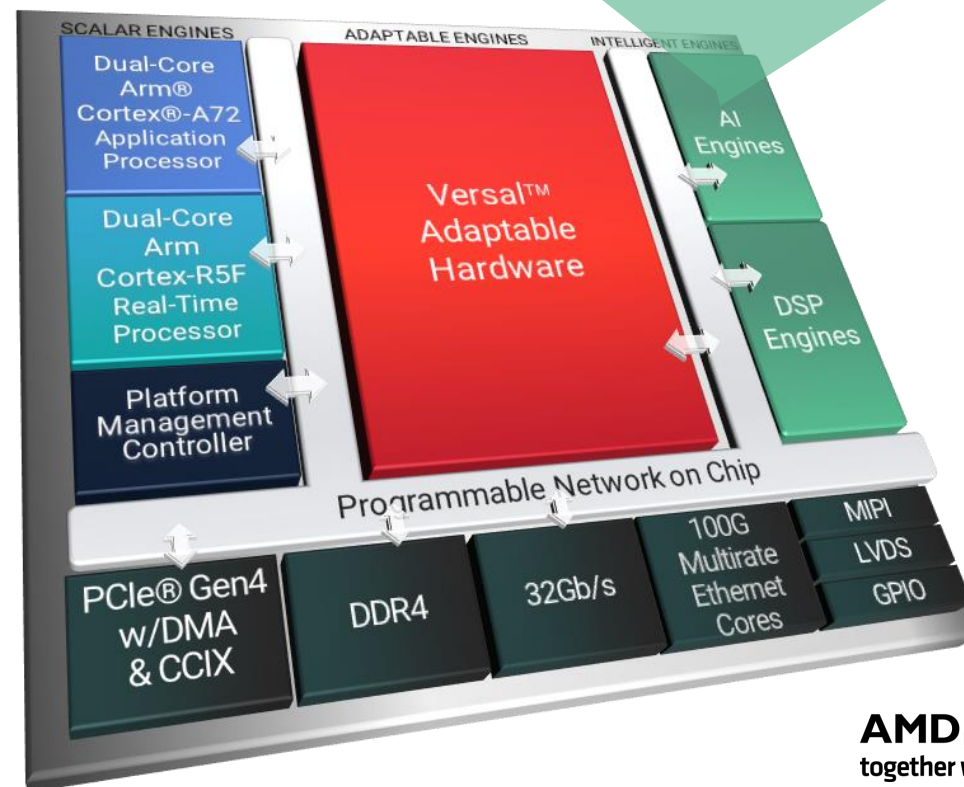
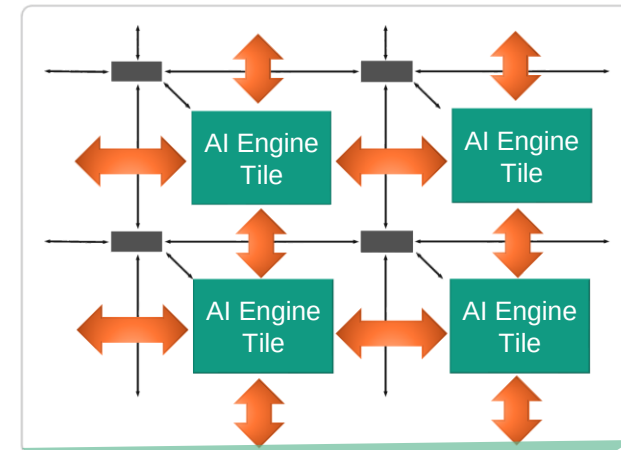
- VLIW processor running at 1GHz+
- Massively reduced development time vs traditional RTL

Array of vector processors

- From 8 tiles on the smallest device (VE2002)
- Up to 472 tiles on the largest device (VP2802)

Terabytes/sec of interface bandwidth

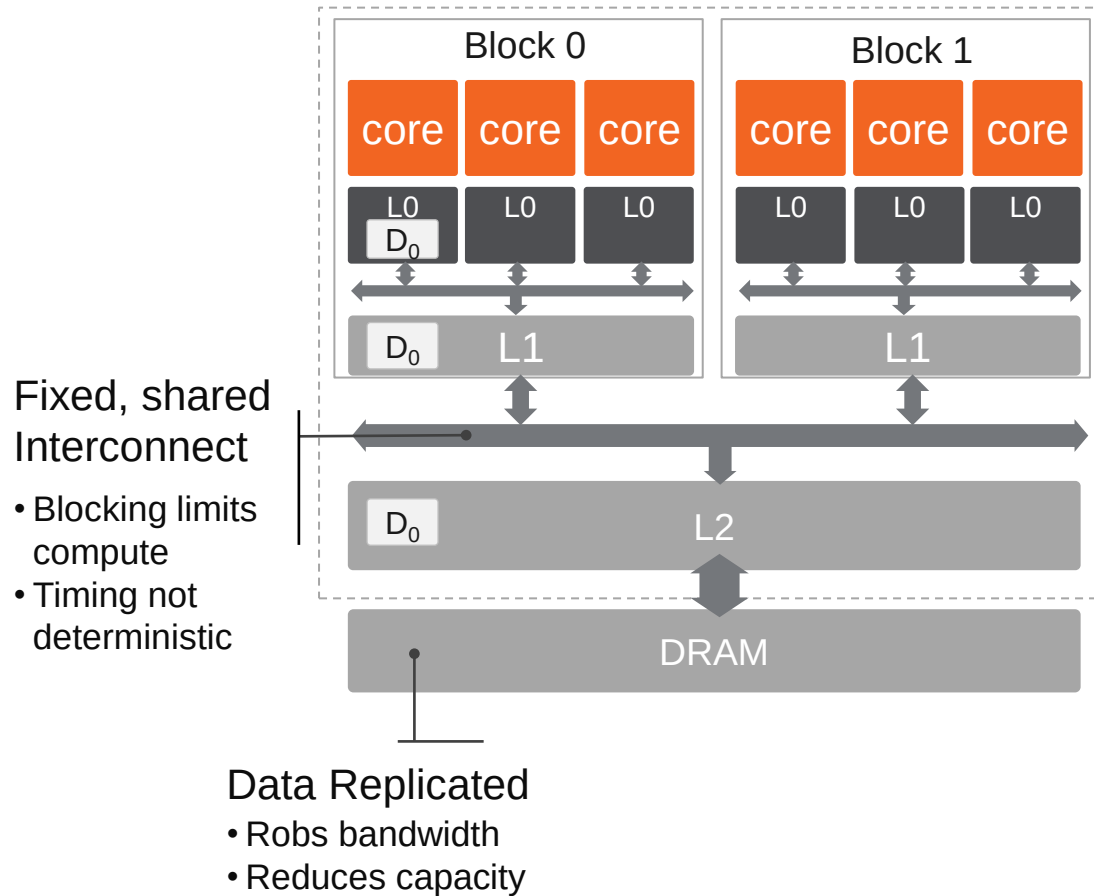
- Direct, massive throughput to adaptable hardware engines



AI Engine Reinvents Multi-Core Compute

Traditional Multi-Core

(Cache-based Architecture)

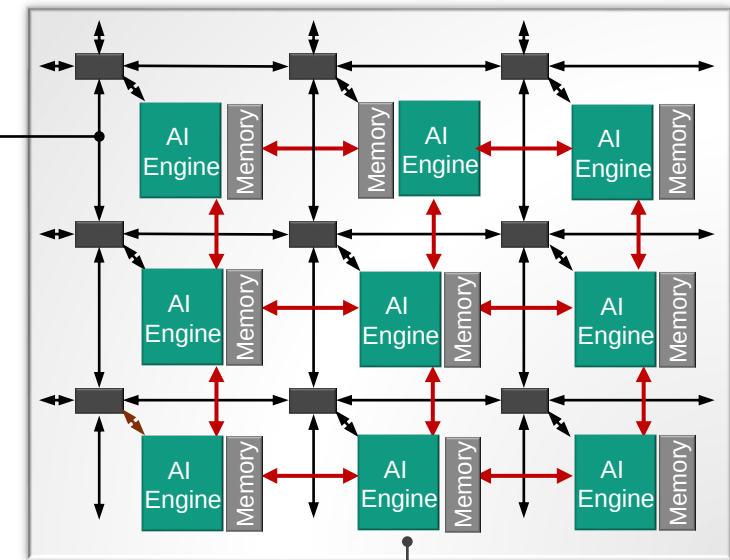


AI Engine Array

(Intelligent Engine)

Dedicated Interconnect

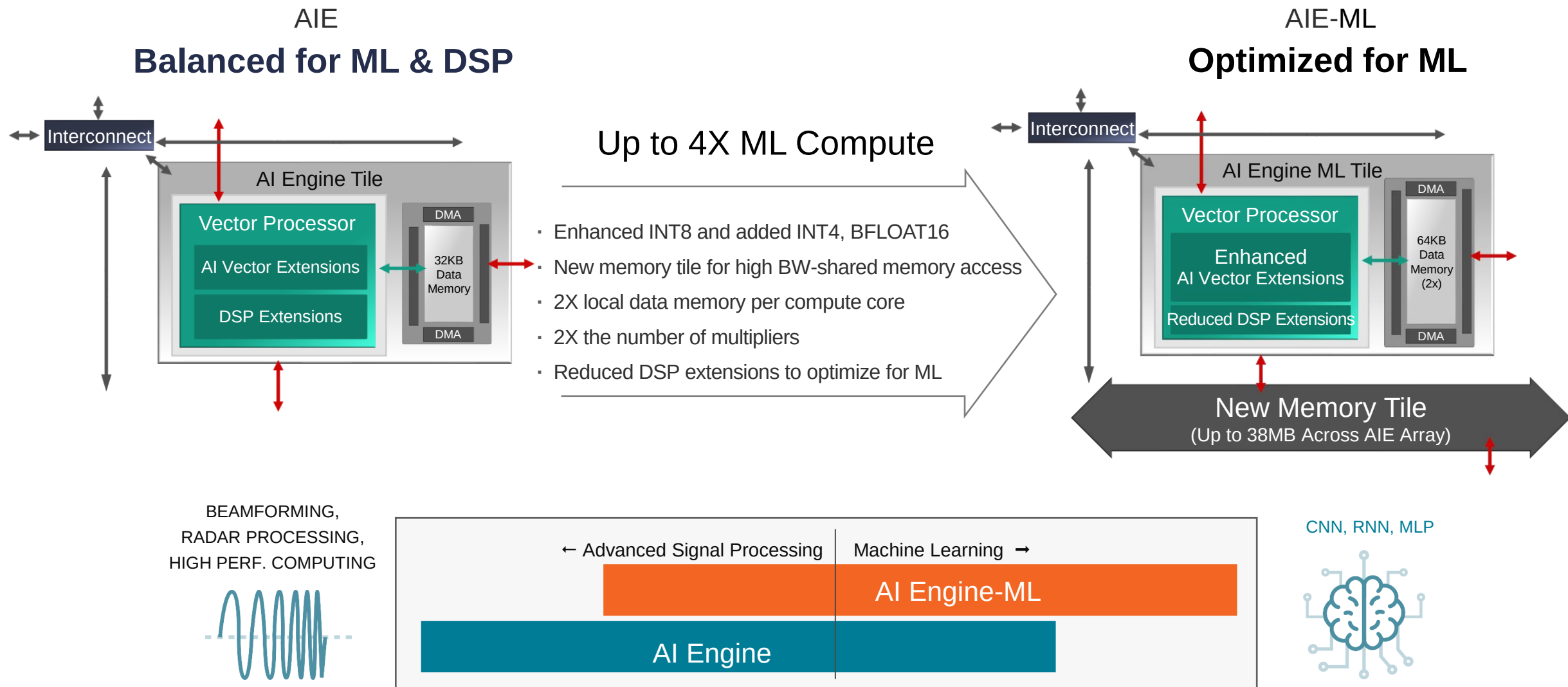
- Non-blocking
- Deterministic



Local, Distributed Memory

- No cache misses
- Higher bandwidth
- Less capacity required

Optimized **AIE-ML** for Up to 4X AI Compute



Versal™ Series with AI Engine

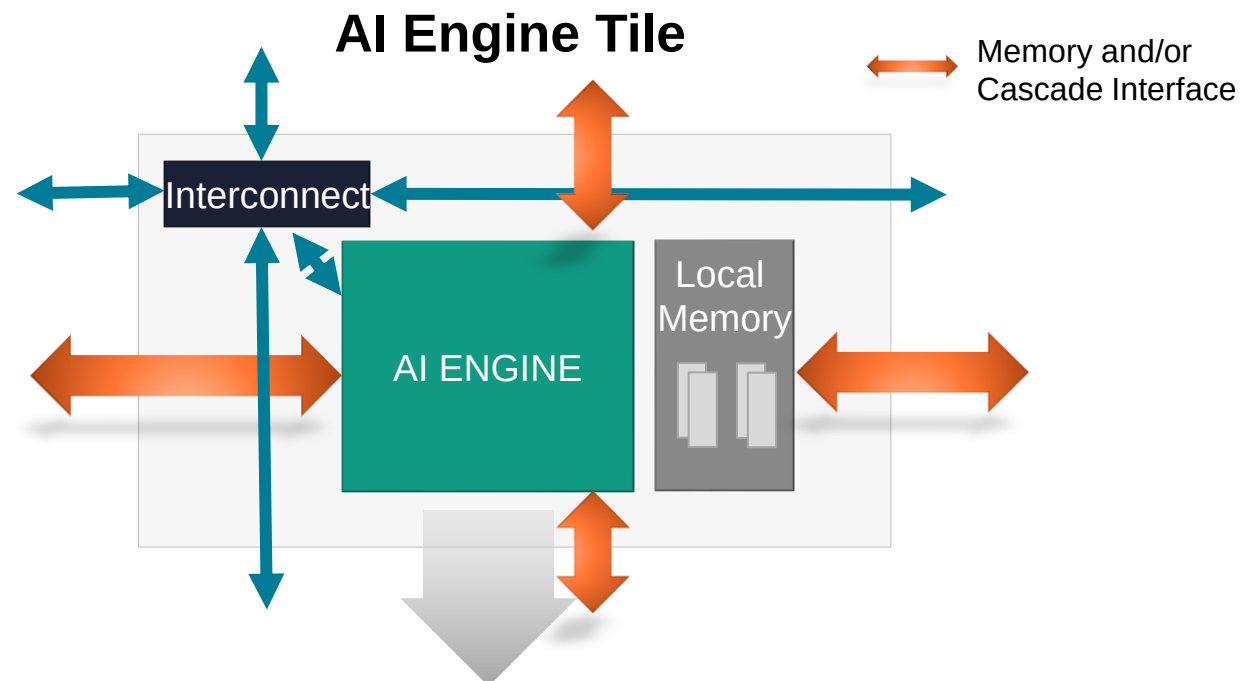
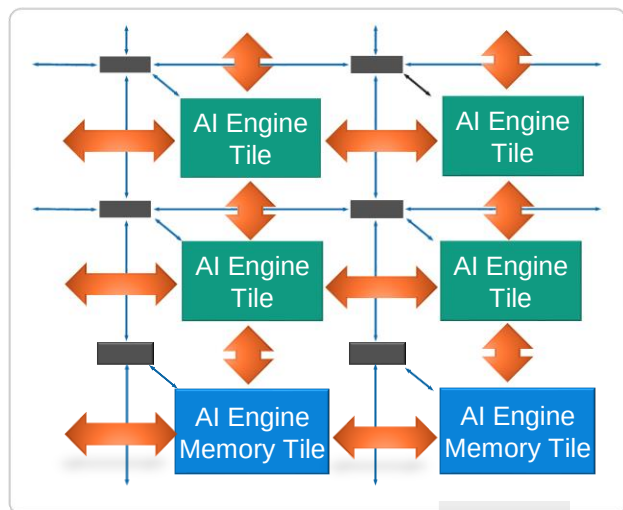
AMD VERSAL AI Core	VC1352	VC1502	VC1702	VC1802	VC1902	VC2602	VC2802
	AIE	AIE	AIE	AIE	AIE	AIE- ML	AIE-ML

AMD VERSAL AI Edge	VE2002	VE2102	VE2202	VE2302	VE1752	VE2602	VE2802
	AIE-ML	AIE-ML	AIE-ML	AIE-ML	AIE	AIE- ML	AIE-ML

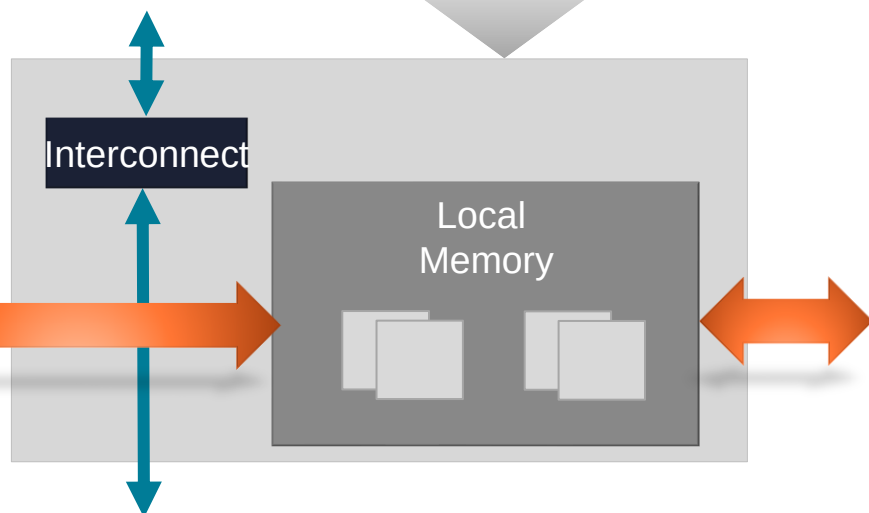
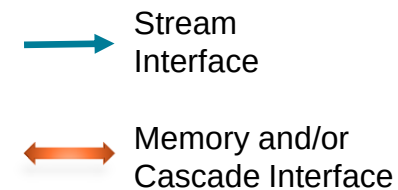
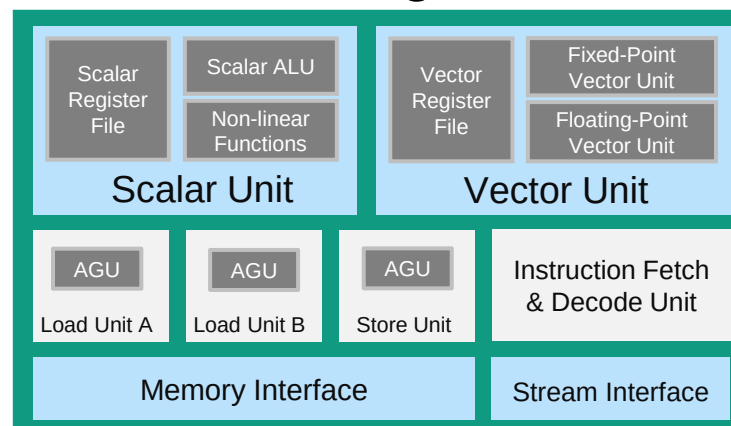
AMD VERSAL Premium	VP2502	VP2802
	AIE	AIE

AI Engine: Terminology

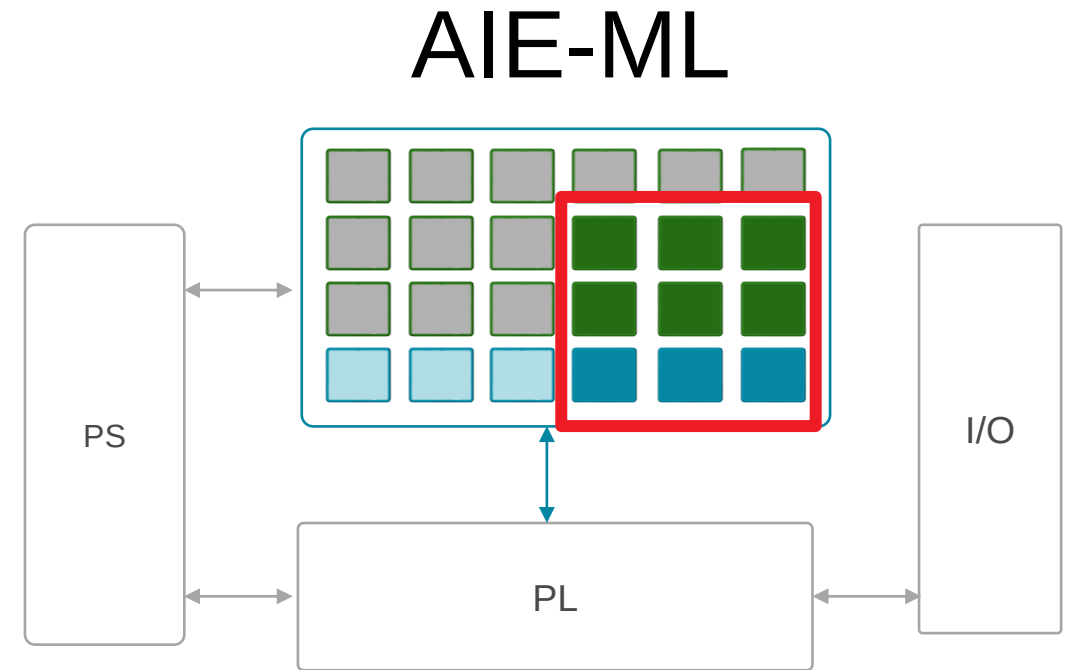
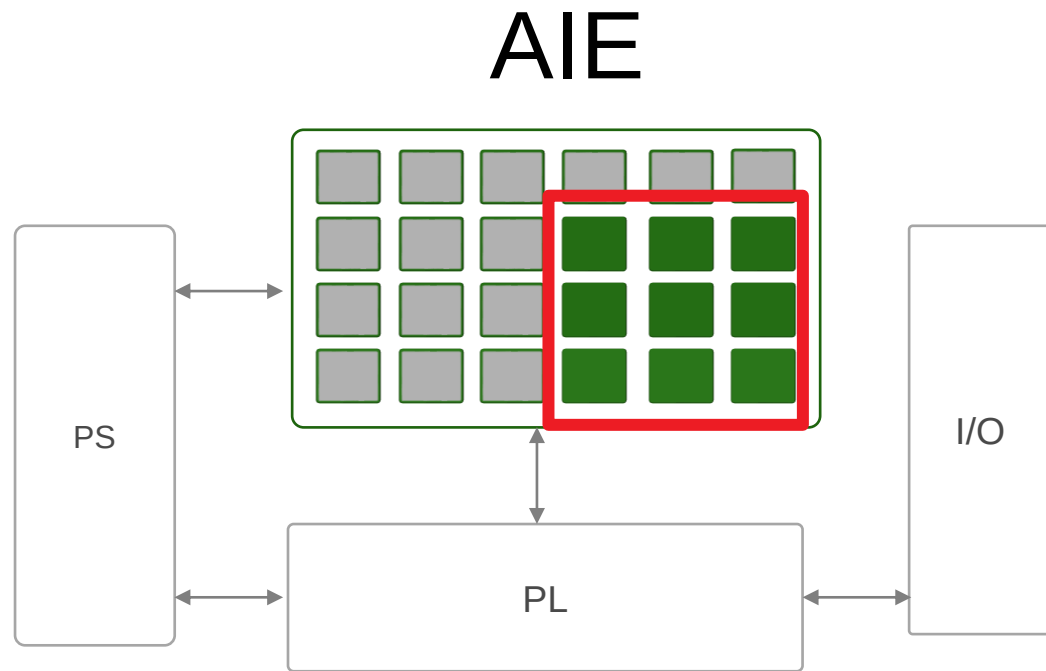
AI Engine Array



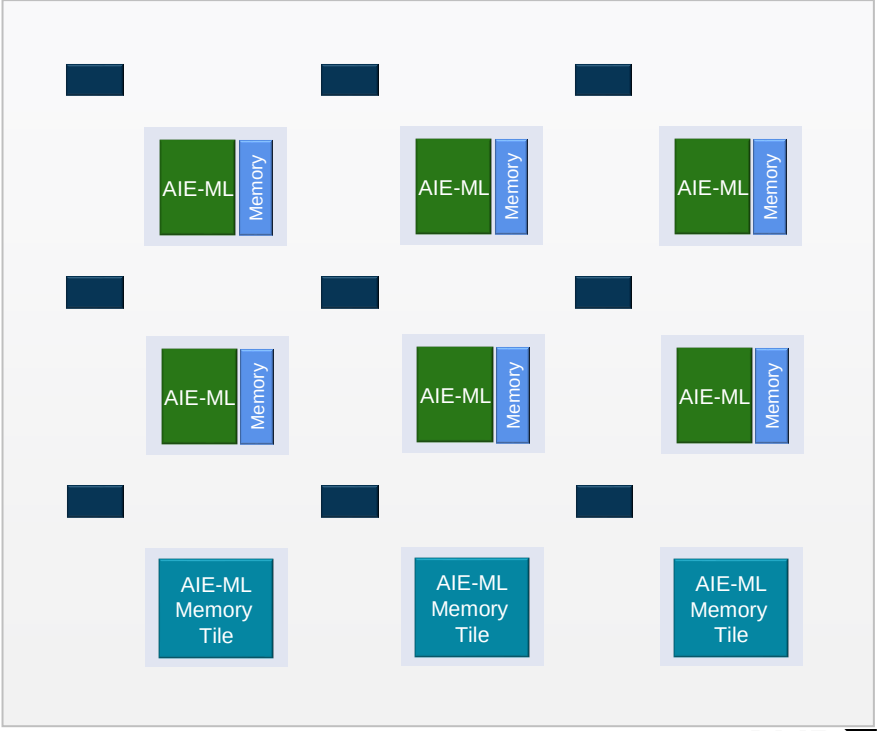
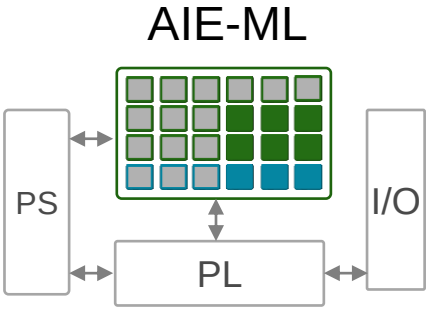
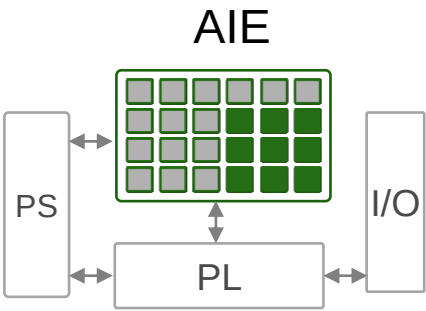
AI Engine



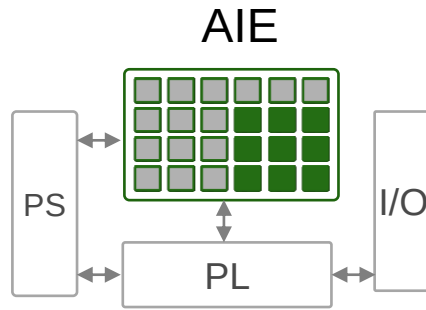
AI Engine: Array Architecture



AI Engine: Array Architecture – Memory

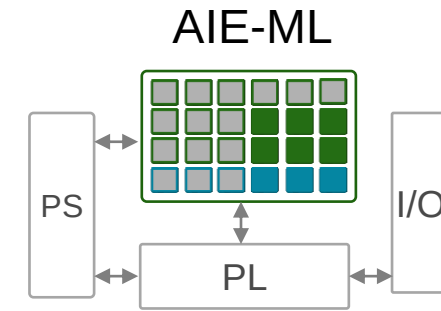


AI Engine: Array Architecture – Memory



Placement of the local memories depends on the row

AI Engine: Array Architecture – Memory

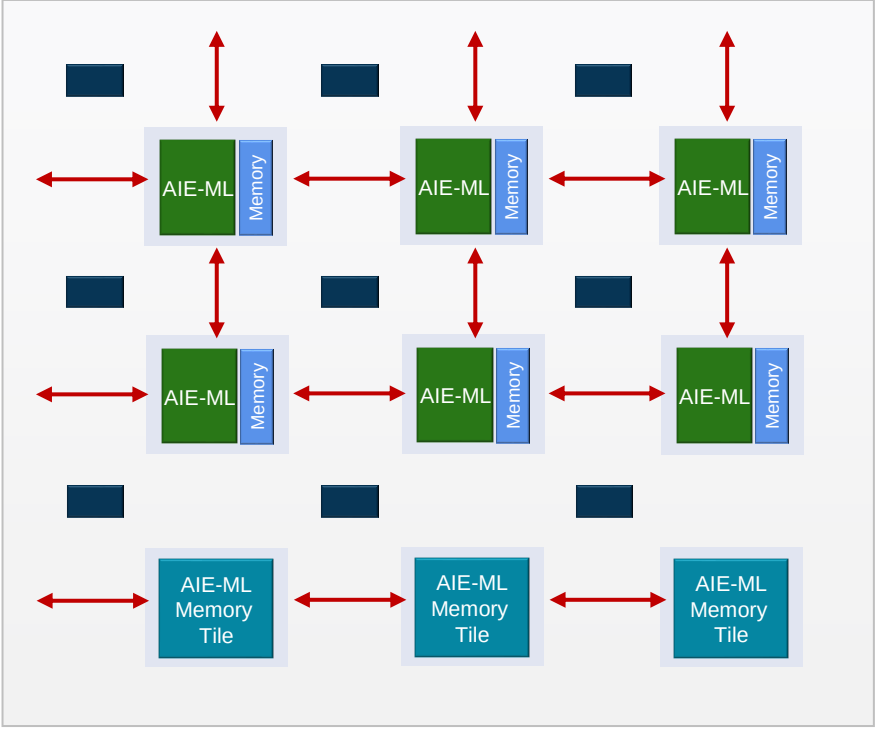
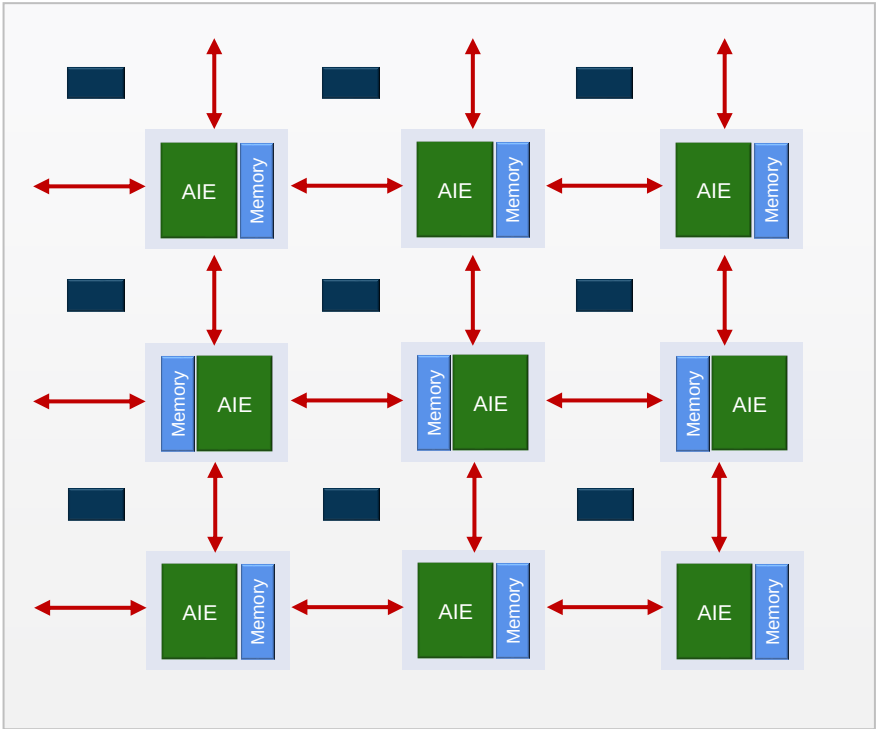
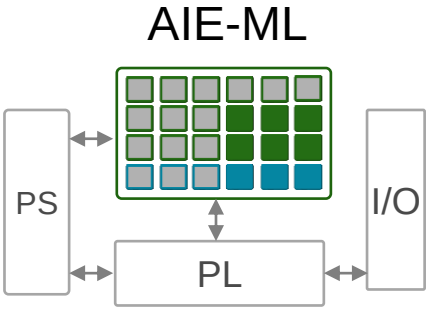
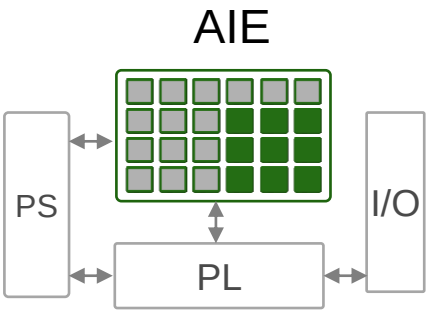


All lines are identical
Memory is always on the East

One or two rows of memory tiles
At the bottom of the array



AI Engine: Array Architecture – Memory Interface



AIE



- **Its own 32KB local memory**
- **32KB on the North**
- **32KB on the South**
- **32KB on the West**

or on the East depending on the row

The AIE on the edges of the array have access to one or two fewer memory modules

AI Engine: Array Architecture – Memory Interface

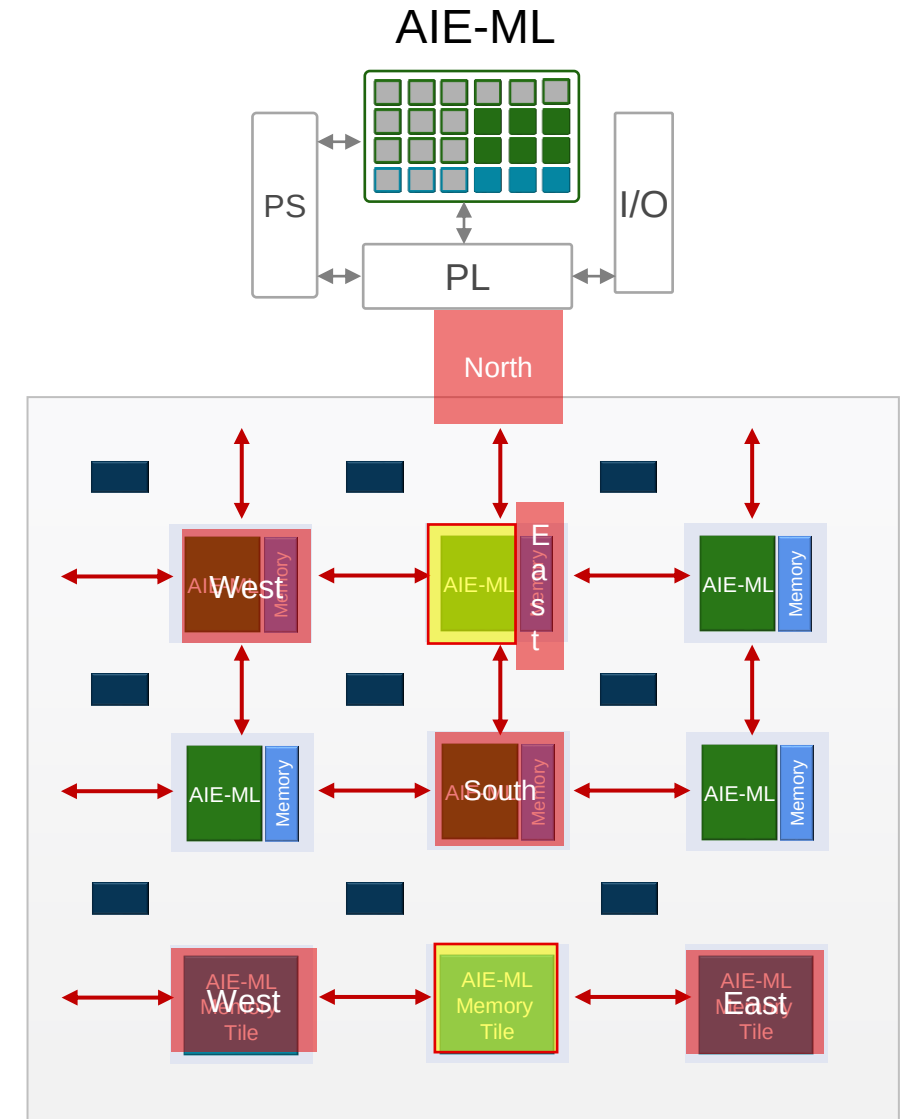
Each AIE-ML core has direct access to 256KB memory:

- Its own 64KB local memory
- 64KB on the North
- 64KB on the South
- 64KB on the West

The DMAs in the memory tiles core have direct access to:

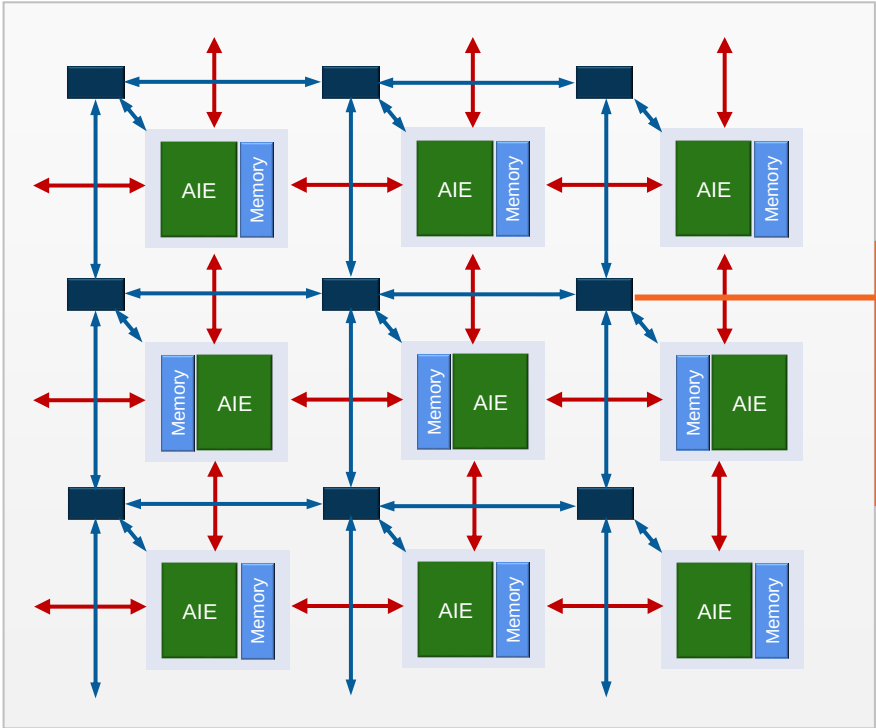
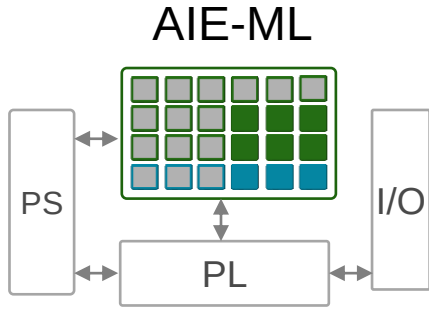
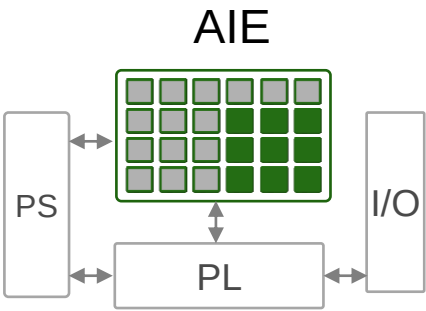
- The local memory
- The memory of the memory tiles on the East and West

The AIE-ML on the edges of the array have access to one or two fewer memory modules



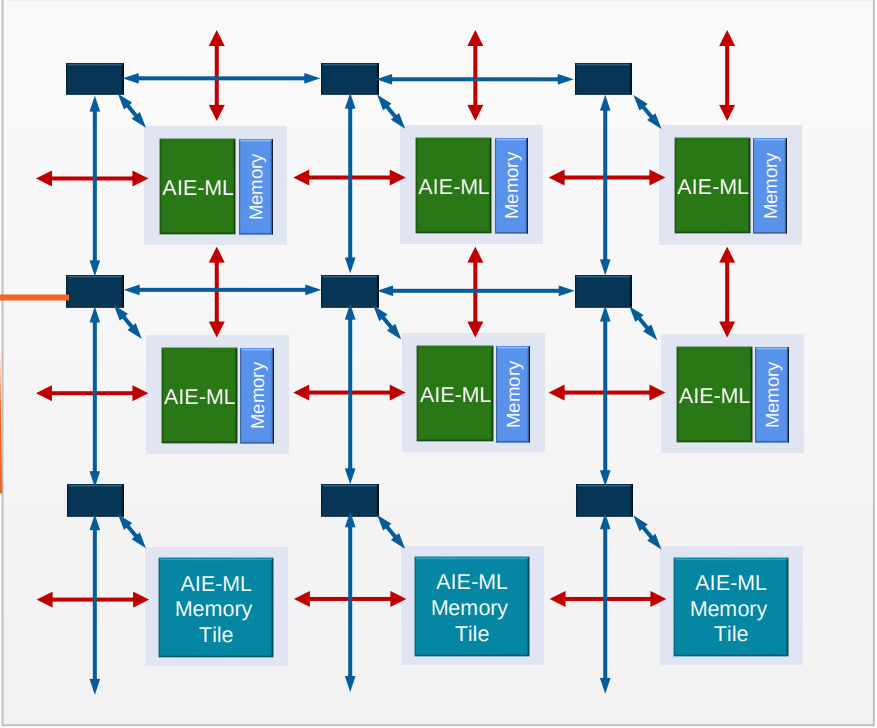
AI Engine: Array Architecture – Interconnect

↔ Direct Memory Interface
↔ Stream Interface



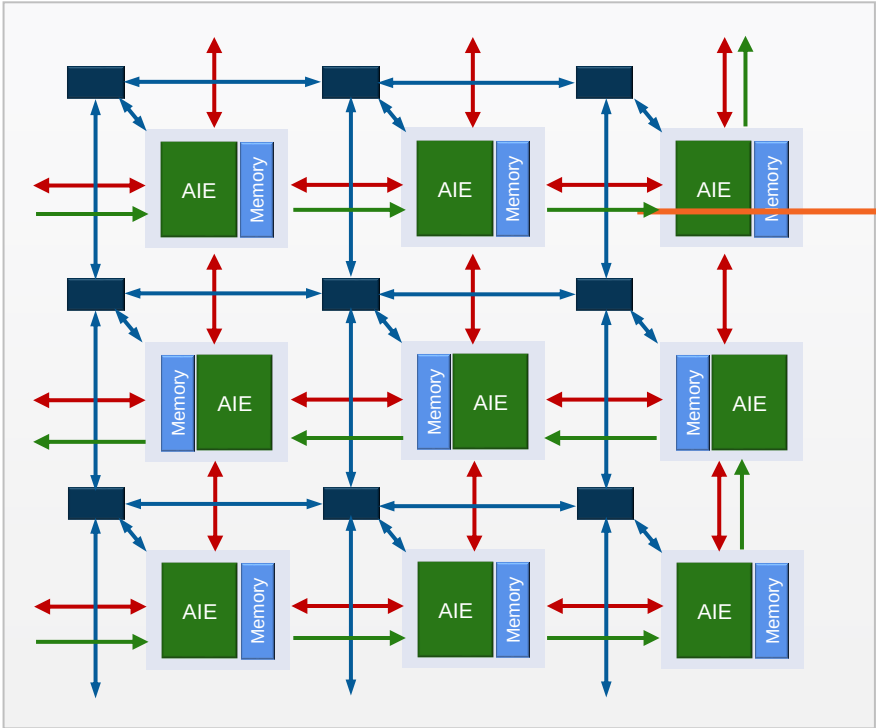
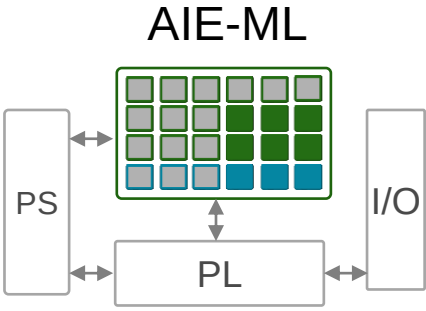
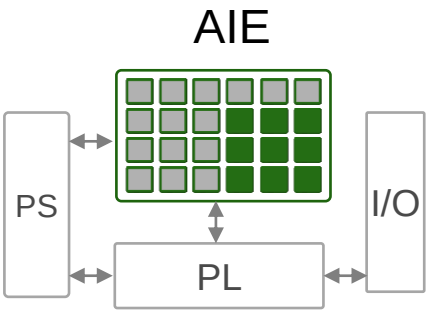
Flexible and dedicated interconnect

- Up to 200+ GB/s bandwidth per tile
- Communication between tiles
- Access to PL and memory in the PL



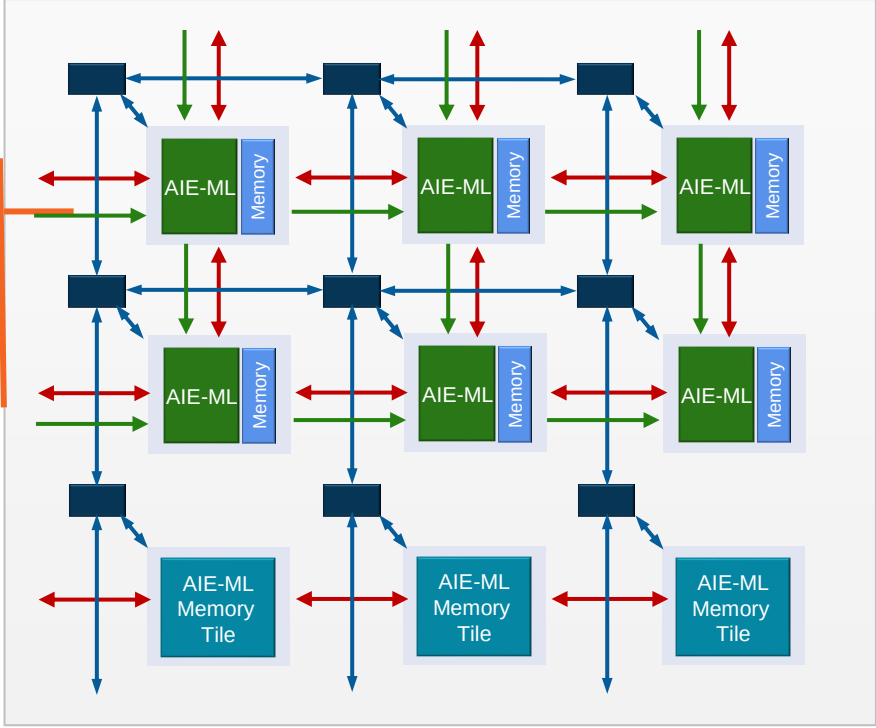
AI Engine: Array Architecture – Cascade Interface

- ↔ Direct Memory Interface
- ↔ Stream Interface
- Cascade Interface

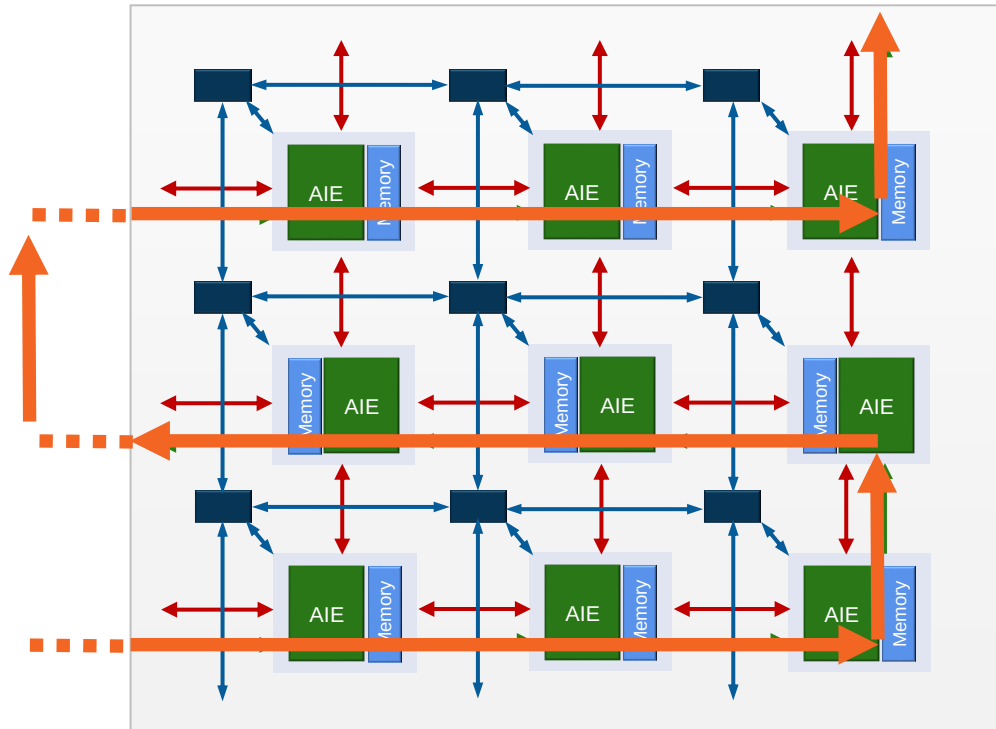
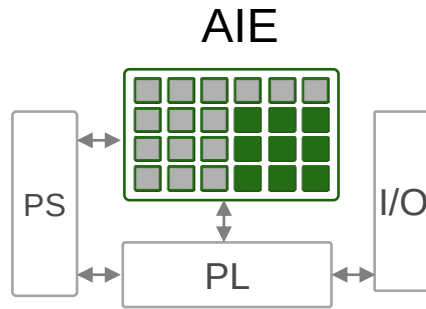
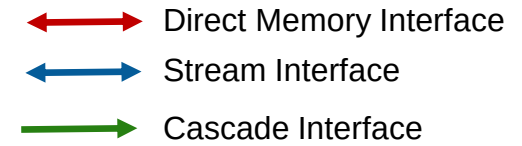


Cascade Interface

Sending partial result to the next AI Engine

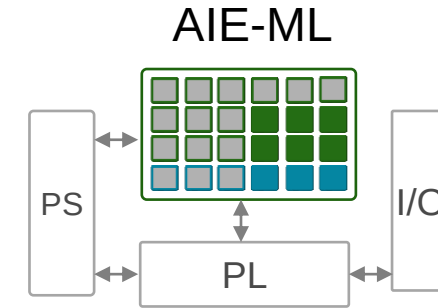
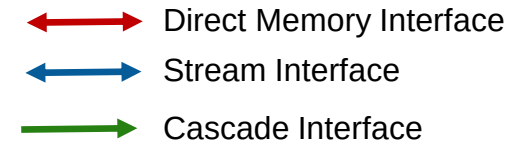


AI Engine: Array Architecture – Cascade Interface



- The cascade streams travel from tile to tile in horizontal manner from West to East or East to West depending on the row
- As a cascade stream reaches the edge at one end, it is connected to the input of the tile above it

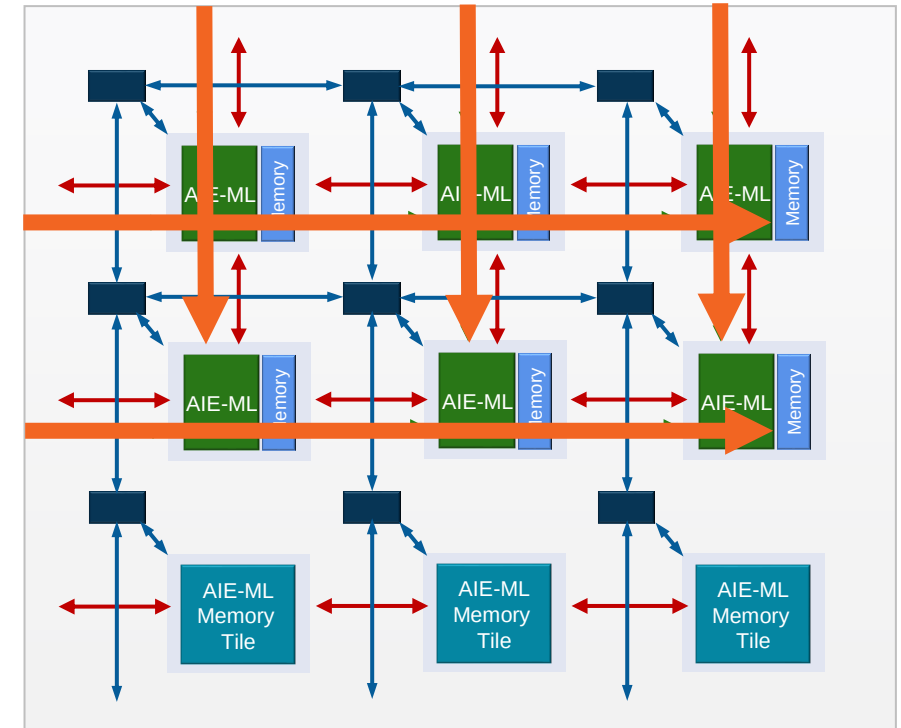
AI Engine: Array Architecture – Cascade Interface



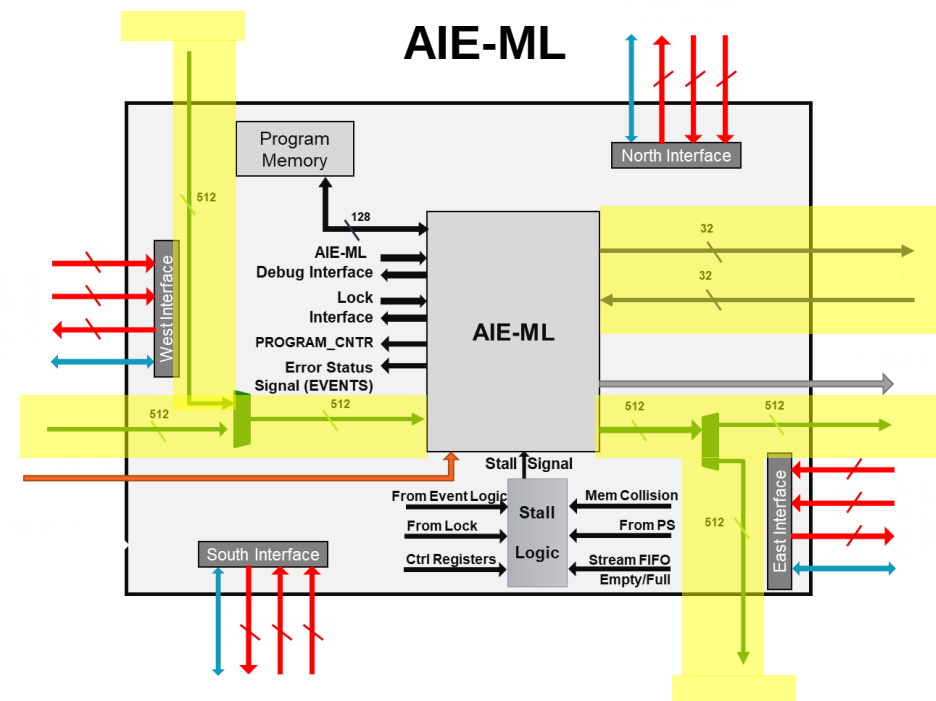
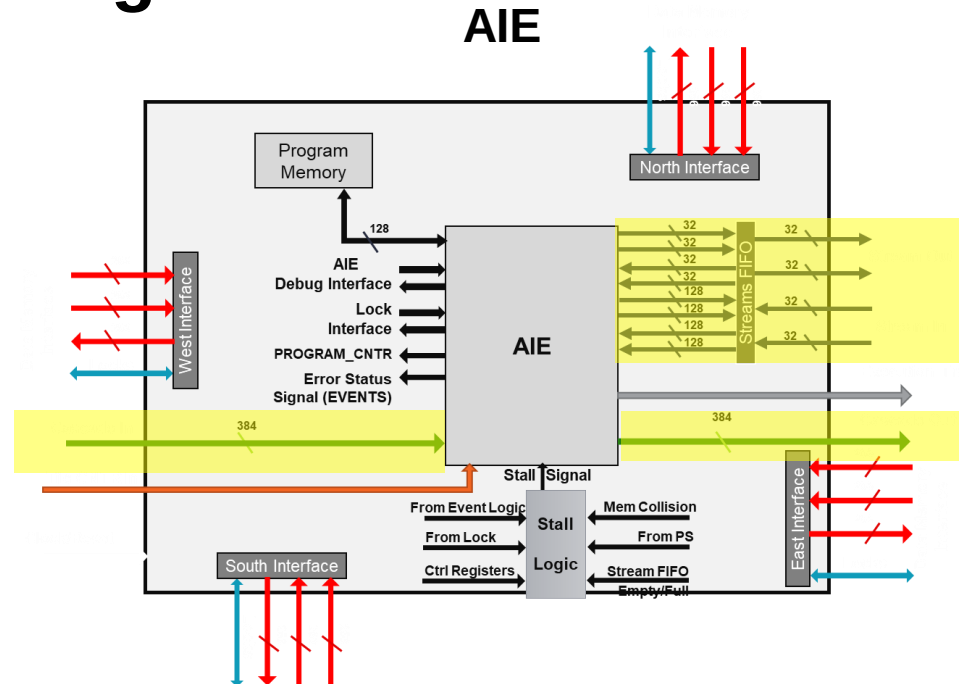
The AIE-ML array has cascade connections:

- In horizontal direction, from West to East
- And in vertical direction, from North to South

The cascade start points and end points are tied off at the array edges



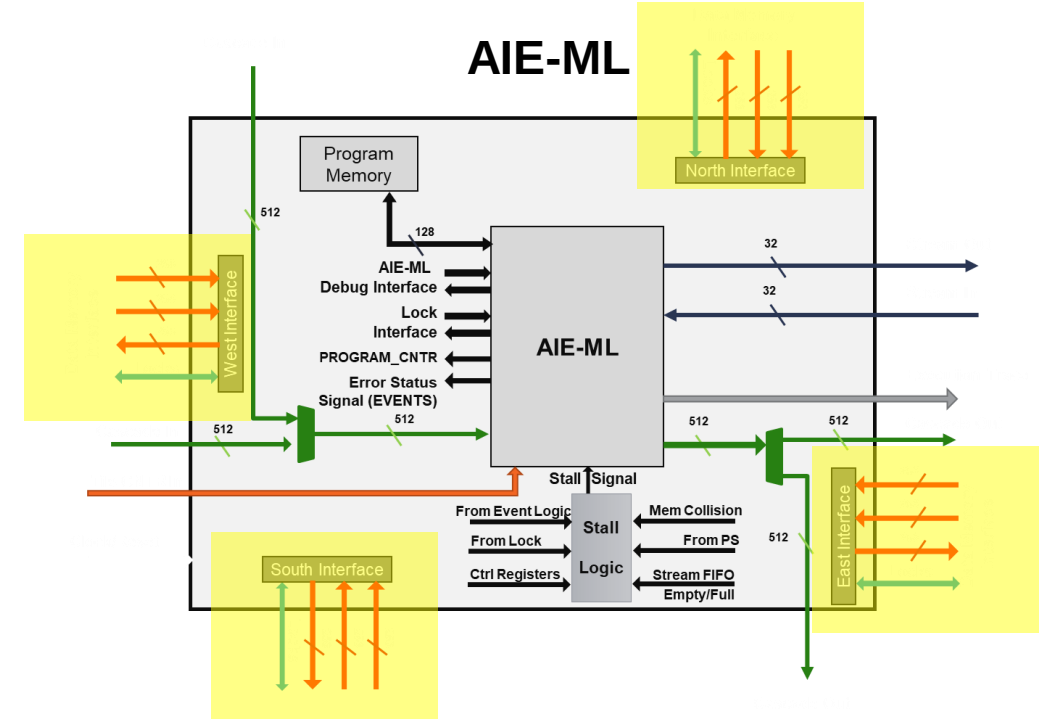
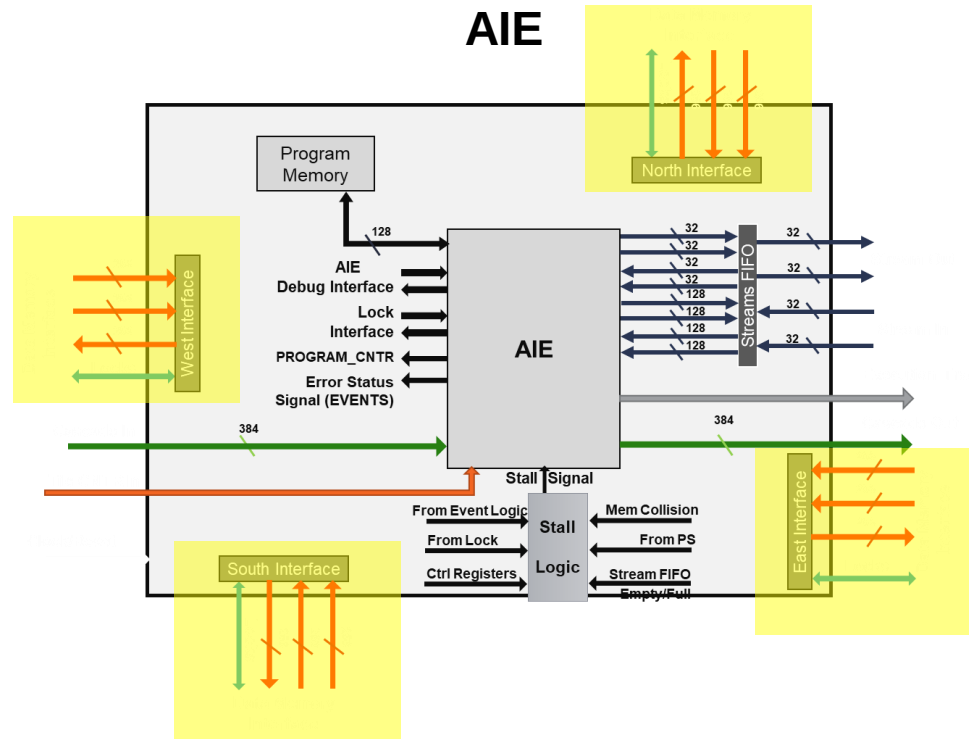
AI Engine: Interfaces



Two differences regarding the core interfaces between AIE and AIE-ML:

- Cascade interface direction and size
 - Vertical direction added in AIE-ML
 - Size increased to 512 bits in AIE-ML vs 384 bits in AIE
- Streaming interface
 - Reduced to one incoming 32-bit data stream and one outgoing 32-bit data stream in AIE-ML
 - Streams FIFO in AIE, allowing 128-bit data over 4 cycles and non-blocking transactions

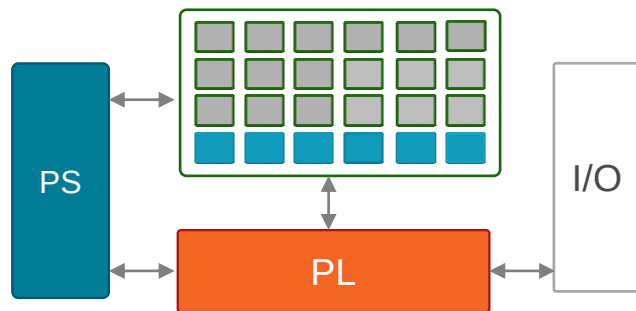
AI Engine: Interfaces – Data Memory Interface



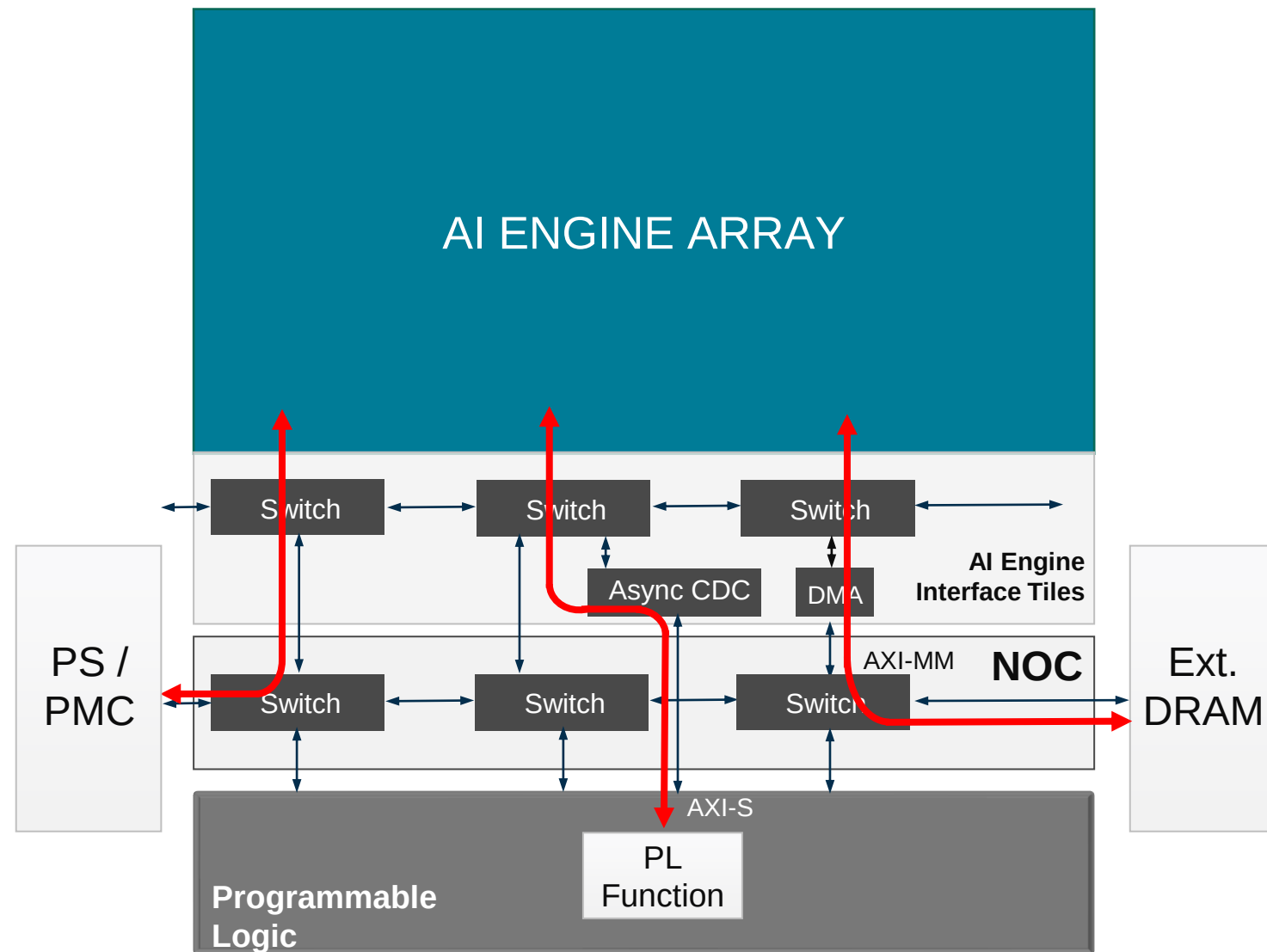
Data Memory Interface

- Two 256-bit-wide load units and one 256-bit-wide store unit
- Direct access to neighboring memories
- Lock interface issues lock signals to manage shared resources

AI Engine Integration with Versal™ Adaptive SoC



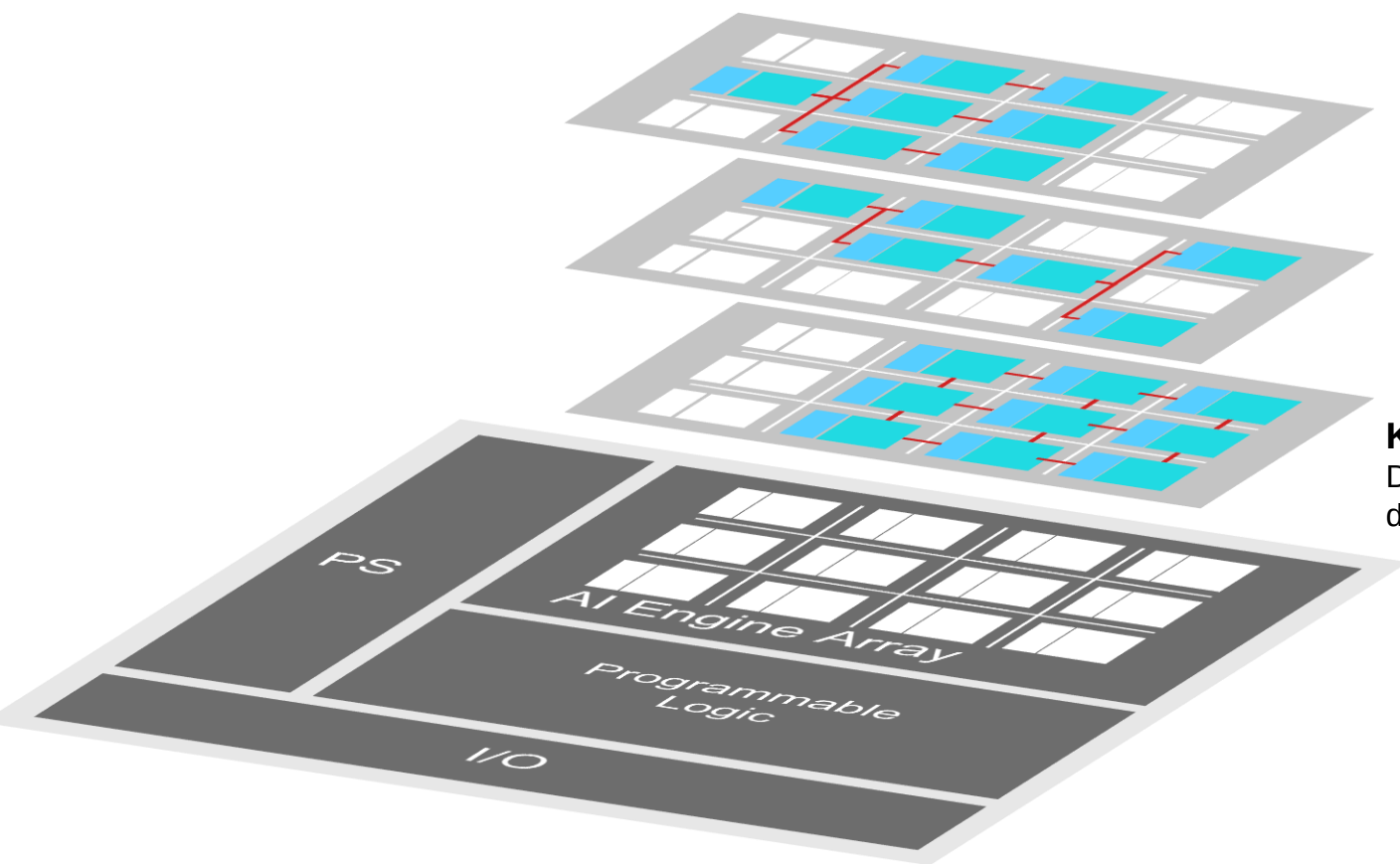
- TB/s of interface bandwidth
 - AI Engine to programmable logic
 - AI Engine to NOC
- Leveraging NOC connectivity
 - PS manages config / debug / trace
 - AI Engine to DRAM (no PL required)



AI Engine Programing Flow and Tools Introduction

Software Programmable: Any Developer

Programming Abstraction Levels



Architecture Overlay

- Abstracts the AI Engine (black box)
 - Automatic connectivity and kernel programming
- Example: Vitis™ AI

Data Flow with AMD libraries

- Use AMD libraries for data flow

Kernel Program Data flow with user- defined libraries

- User has full control
- Write kernel code
- Control data flow and connectivity

Programming the AI Engine

Programming the AI Engine array requires understanding of:



Algorithm to be implemented



Capabilities of the AI Engines



Overall data flow between individual functional units

AI Engine array supports three levels of parallelism:

SIMD

Multiple elements to be computed in parallel

Instruction Level

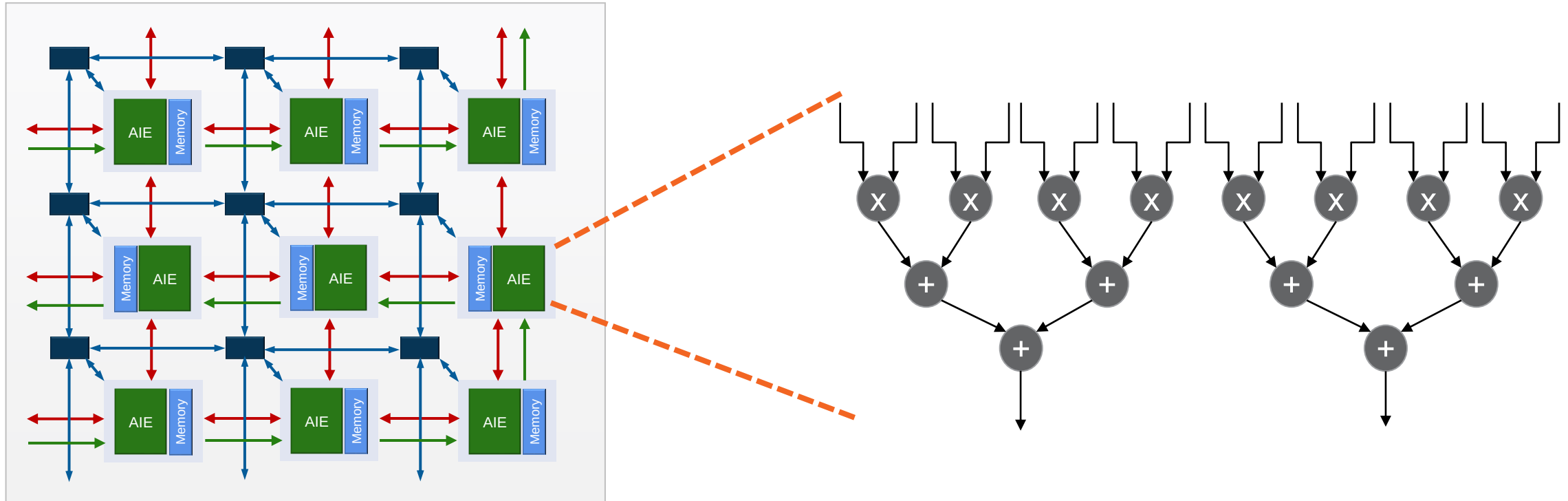
Multiple instructions to be executed in a single clock cycle

Multi-Core

Up to 472 AI Engines can execute in parallel

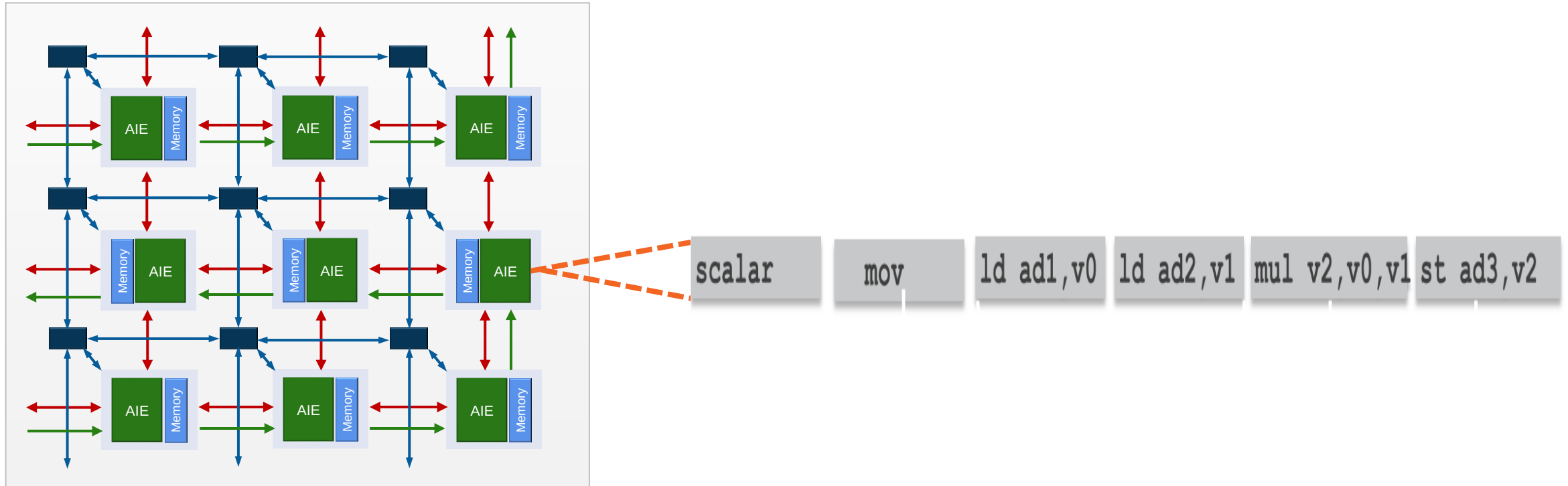
AI Engine Array Supports Three Levels of Parallelism

SIMD (single instruction, multiple data) – Multiple compute operations in parallel



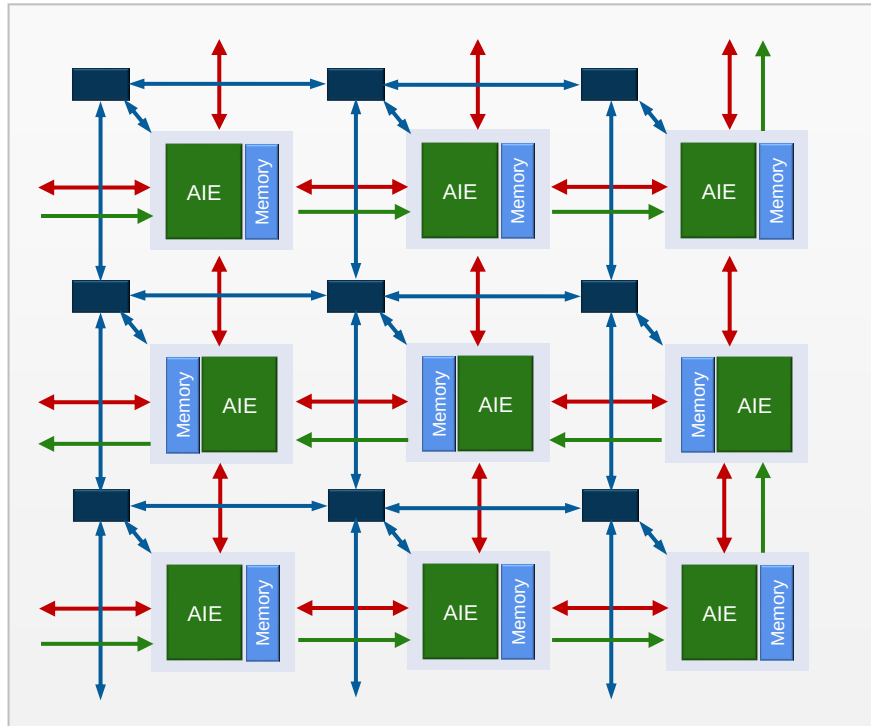
AI Engine Array Supports Three Levels of Parallelism

VLIW (very long instruction word) – Multiple operations in the same cycle



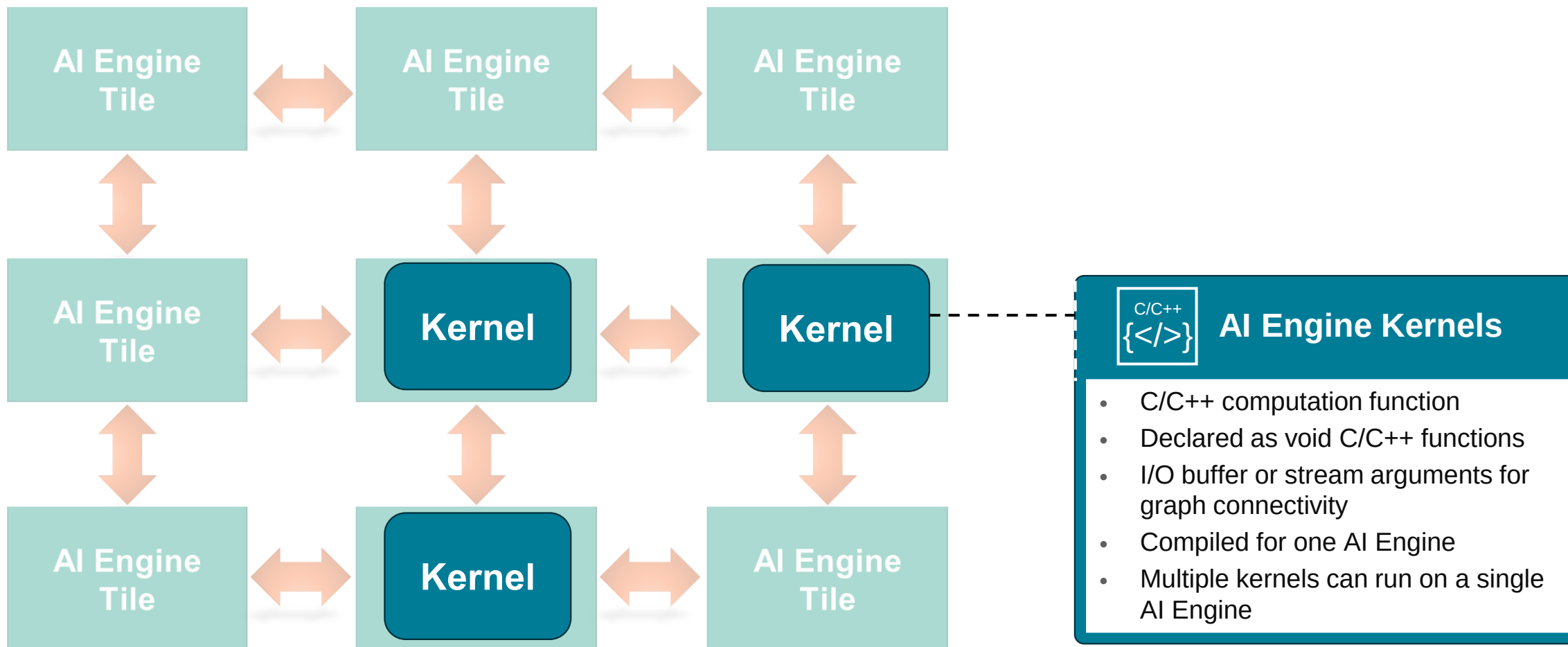
AI Engine Array Supports Three Levels of Parallelism

Multi-core – Multiple AI Engines computing in parallel

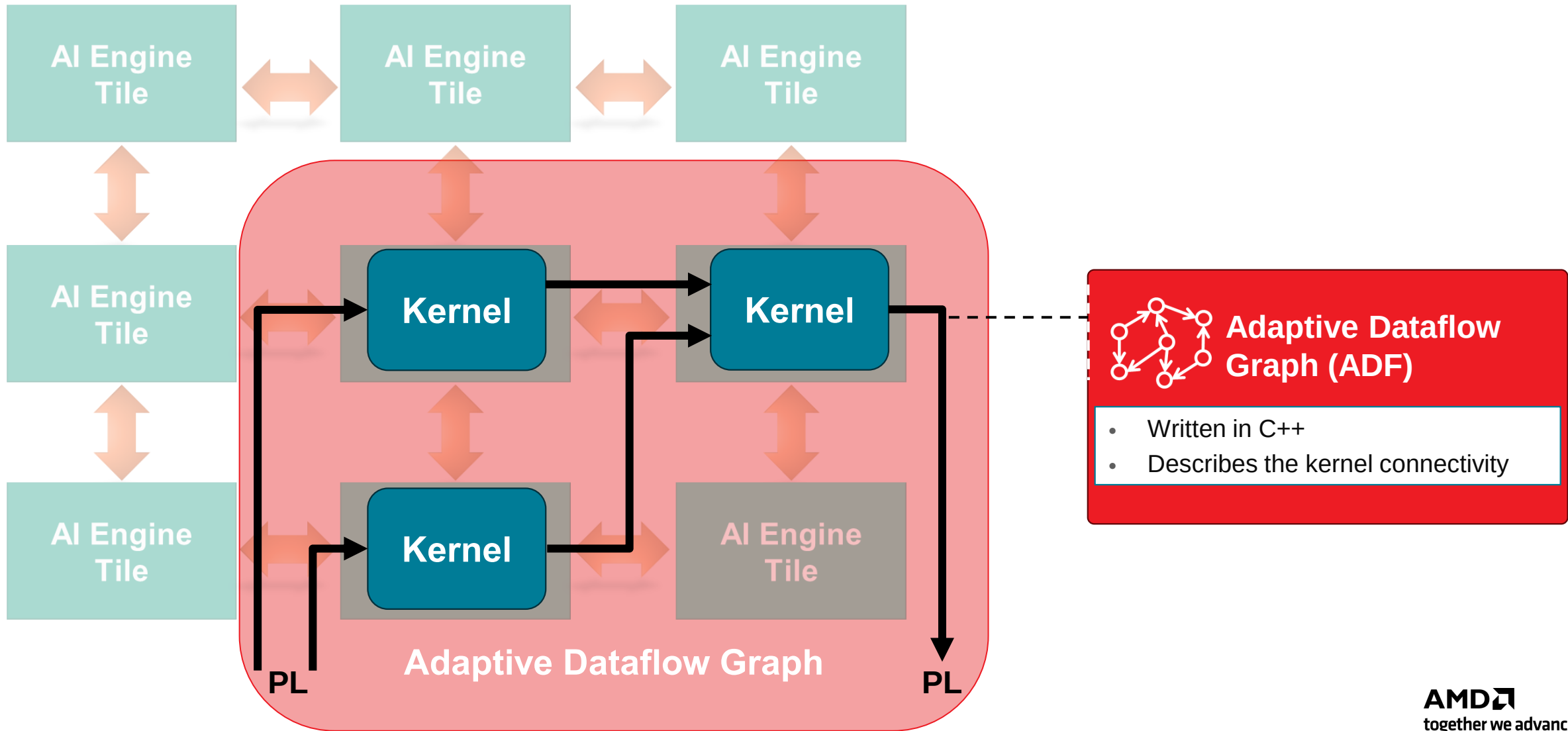


Compute Problem

Programming the AI Engine: Kernels



Programming the AI Engine: Adaptive Dataflow Graph

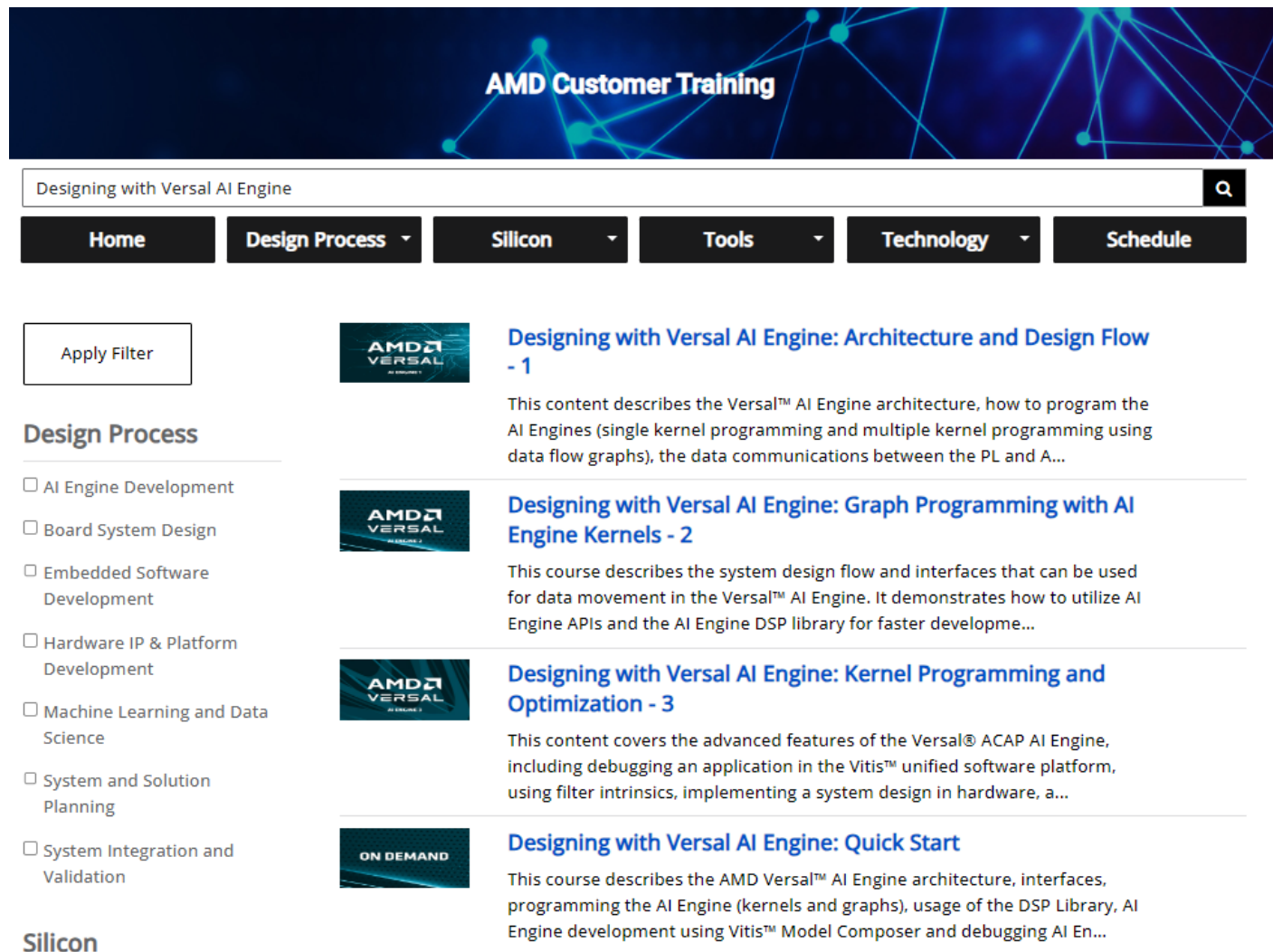




Vitis™ Unified IDE for AI Engine Development Tool Demo

Resources to Get Started

AI Engine On-Demand Training



The screenshot shows the AMD Customer Training website. At the top is a dark blue header with the text 'AMD Customer Training' and a network diagram. Below the header is a search bar containing the text 'Designing with Versal AI Engine' and a magnifying glass icon. Under the search bar is a navigation menu with buttons: 'Home', 'Design Process' (with a dropdown arrow), 'Silicon' (with a dropdown arrow), 'Tools' (with a dropdown arrow), 'Technology' (with a dropdown arrow), and 'Schedule'. On the left side, there is a filter section with a button 'Apply Filter'. Below this, under the heading 'Design Process', there is a list of checkboxes: 'AI Engine Development', 'Board System Design', 'Embedded Software Development', 'Hardware IP & Platform Development', 'Machine Learning and Data Science', 'System and Solution Planning', and 'System Integration and Validation'. Below this list is the heading 'Silicon'. The main content area displays four search results, each with an AMD Versal logo icon, a title, and a description. The results are: 1. 'Designing with Versal AI Engine: Architecture and Design Flow - 1' (description: This content describes the Versal™ AI Engine architecture, how to program the AI Engines (single kernel programming and multiple kernel programming using data flow graphs), the data communications between the PL and A...), 2. 'Designing with Versal AI Engine: Graph Programming with AI Engine Kernels - 2' (description: This course describes the system design flow and interfaces that can be used for data movement in the Versal™ AI Engine. It demonstrates how to utilize AI Engine APIs and the AI Engine DSP library for faster developme...), 3. 'Designing with Versal AI Engine: Kernel Programming and Optimization - 3' (description: This content covers the advanced features of the Versal® ACAP AI Engine, including debugging an application in the Vitis™ unified software platform, using filter intrinsics, implementing a system design in hardware, a...), and 4. 'Designing with Versal AI Engine: Quick Start' (description: This course describes the AMD Versal™ AI Engine architecture, interfaces, programming the AI Engine (kernels and graphs), usage of the DSP Library, AI Engine development using Vitis™ Model Composer and debugging AI En...).

<https://xilinxprod-catalog.netexam.com/Search?searchText=Designing+with+Versal+AI+Engine>

AI Engine Documentation

AMD Versal Adaptive SoC Design Process Documentation



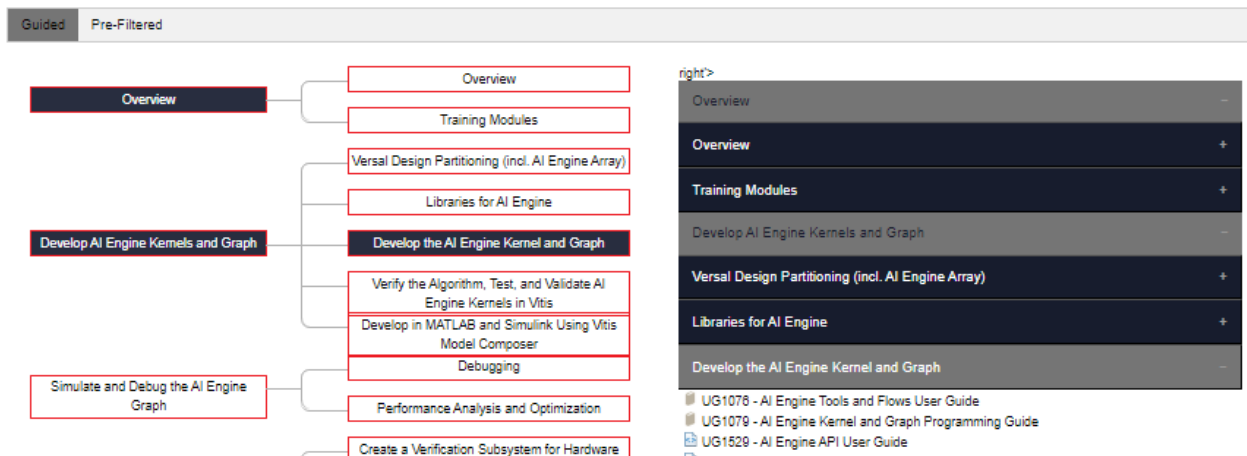
AI Engine Development

Provides step-by-step guidance for creating AI Engine applications and integrating with a Versal Adaptive SoC system.

AMD provides a breadth of documentation, resources, and methodologies to accelerate your development on the Versal architecture. If you're not sure where to begin with Versal Adaptive SoCs:

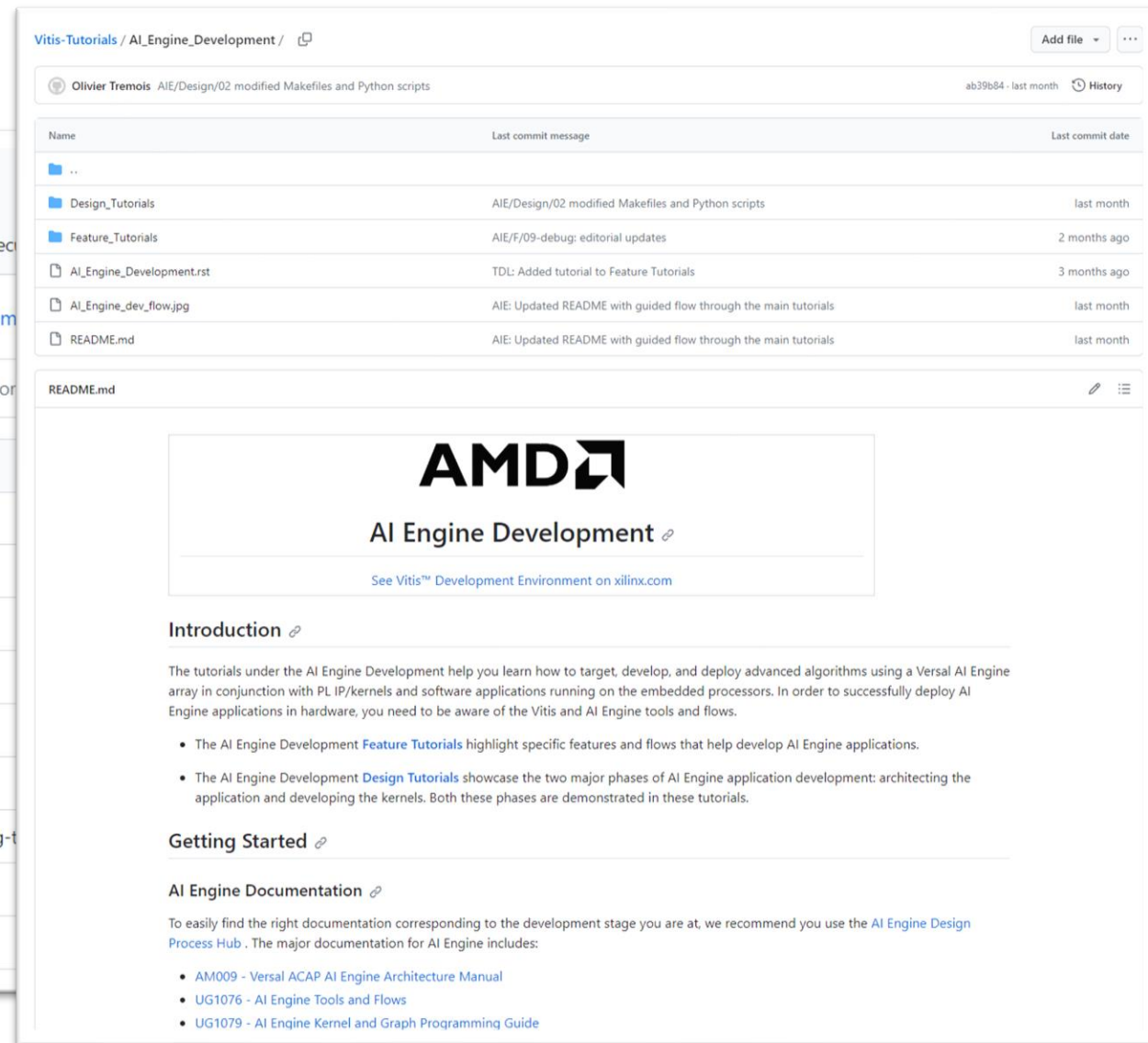
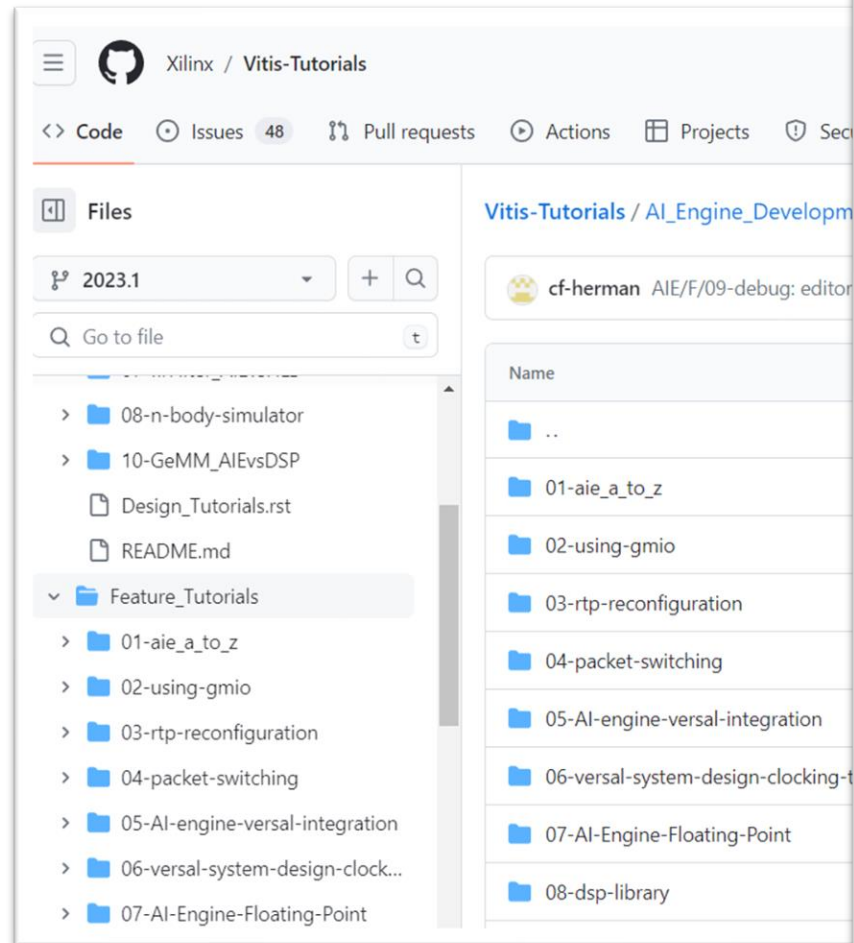
- The **AMD Versal Design Flow Assistant** is an interactive guide to help you create a development strategy
- These seven AMD Versal Design Process Hubs provide a visual and streamlined reference to all AMD Versal documentation, by design process

For the complete list of AMD Versal training courses, see [General AMD Versal Training](#).



<https://docs.xilinx.com/p/ai-engine-development>

AI Engine Tutorials



https://github.com/Xilinx/Vitis-Tutorials/tree/2023.1/AI_Engine_Development/

AI Engine Support

AMD

Adaptive SoC & FPGA Support

Search...

SEARCH

LOGIN

Home

FORUMS

KNOWLEDGE BASE

BLOGS

ABOUT OUR COMMUNITY

ADVANCED SEARCH

AI ENGINE ARCHITECTURE & TOOLS

VITIS

DISCUSSIONS

ARTICLES

Sort by:

Top Questions

New Vitis Unified IDE is out now! Try It Out and Share Your Feedback

AI Engine Architecture & Tools · nathan xu · June 27, 2023 at 8:07 AM

9031

Do you know about our Superuser Program? Keep reading to learn more and earn a new badge!

Announcements · stacias · May 17, 2023 at 1:33 AM

9.62K360

Versal ACAP AI Engine (AIE) - Useful Resources

AI Engine Architecture & Tools · florentw · December 17, 2020 at 12:17 PM

5.46K73

DESCRIPTION

Discuss topics on Versal AI Engine architectures, AI Engine tools, APIs and AI engine based systems guidance and debug.

16

462

65

FOLLOWERS

POSTS

ARTICLES

31 MEMBERS ARE DISCUSSING THIS TOPIC.

ASK A QUESTION

<https://support.xilinx.com/>

AMD

together we advance_

41

DISCLAIMER AND ATTRIBUTIONS

DISCLAIMER

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18

©2023 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Versal, Vitis, and combinations thereof are trademarks of Advanced Micro Devices, Inc. PCIe and PCI Express are registered trademarks of PCI-SIG Corporation. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

