

# **Getting Started with Versal**

Software Application Engineer Brian Lay



© Copyright 2021 Xilinx

### Outline

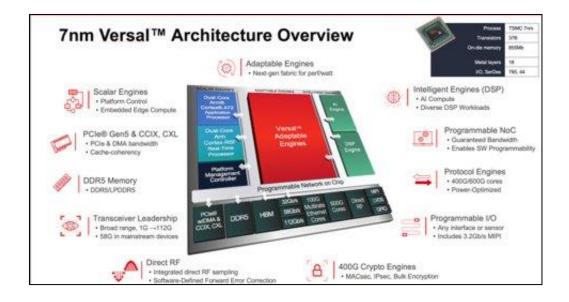
#### ACAP architecture and methodology

- Versal design flows overview
  - Traditional hardware design
  - Platforms
    - Fixed
    - Extensible
- Versal specific IP
- Simulation and Debug Flows
- Programming



### Versal Adaptive Compute Acceleration Platform (ACAP) Overview

- Revolutionary architecture designed to be completely SW programmable:
  - Shared DDR through NoC (no PS DDR)
  - PL Configuration through PMC
  - Debug through PMC
  - System Monitor through PMC
  - SEU through PMC (no more SEM IP)
  - CFI, AXI, NPI interfaces vs. CFI only
  - DRP (GT/MMCM ports) replaced with APB (PS<>AXI)
  - PCIe / CPM / GT-based IP sharing methodology (in new quad)
  - AXI interface for all Hard IP and Soft IP
  - SW-driven AIE processor (vs. HW design w/ DSP block)



# System Design Methodology – Vivado Flow

Paper	Traffic	Data Flow	Data Flow	Power Estimati	on System	Synthesis &
Algorithm	Analysis	Modelling (IPI)	Simulation	& Analysis	Simulation	Implementation
Develop a paper mapping of algorithm/ application to Versal	<ul> <li>Capture traffic flow for NoC</li> <li>Static analysis</li> <li>System C simulation</li> </ul>	<ul> <li>Connect: traffic generators, memories, performance monitors</li> <li>Configure: traffic generators, NoC connectivity, QoS requirements</li> <li>Elaborate: Design and export netlist</li> </ul>	<ul> <li>System Verilog simulation</li> <li>Gather and analyze statistics from performance monitors</li> </ul>	Leverage XPE with output from IP Integrator for accurate power analysis	<ul> <li>Full system simulation; replacing traffic generators</li> <li>Co-simulate with PS VIP and PL blocks</li> </ul>	<ul> <li>Take completed design through back-end tools</li> <li>Timing Closure</li> <li>PDI Generation</li> </ul>

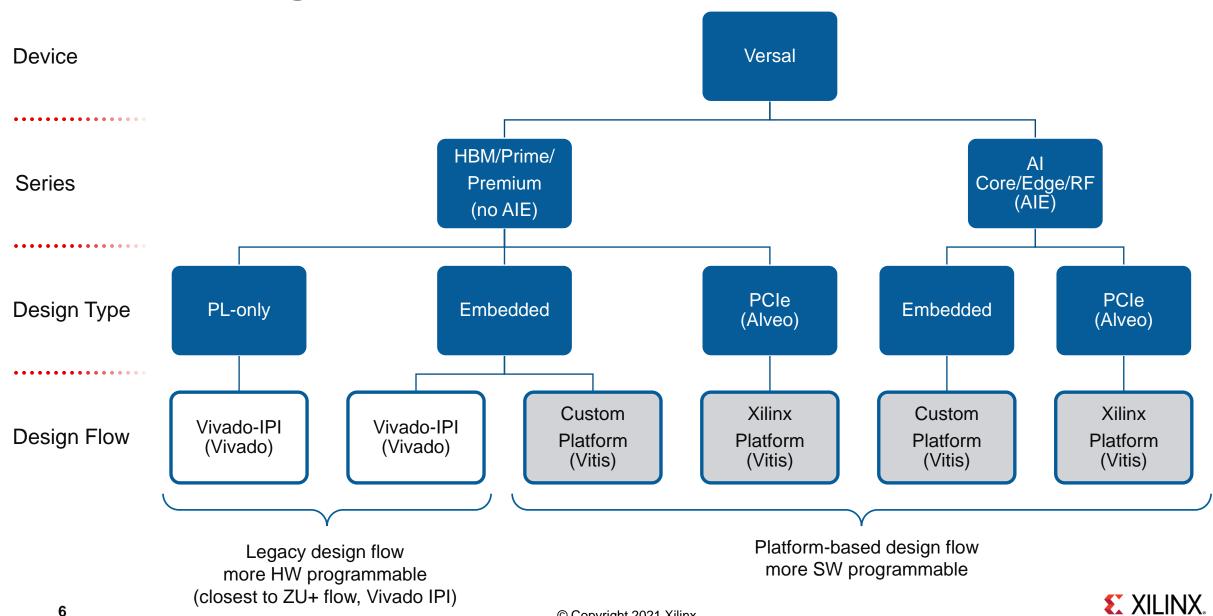
#### Leverage These Steps





# **Design Flows**





## **Versal Design Flow**

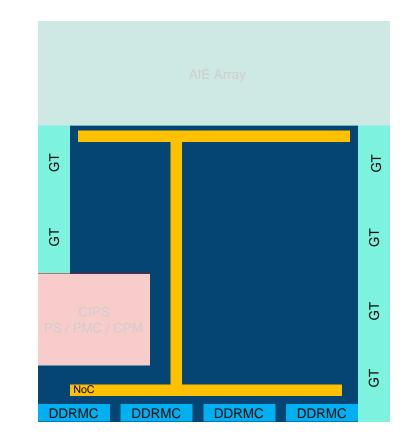
- PMC is required design component for all the flows •
- PLM (PMC software) provided by Xilinx ٠

© Copyright 2021 Xilinx

## Versal Design Flows (Vivado only)

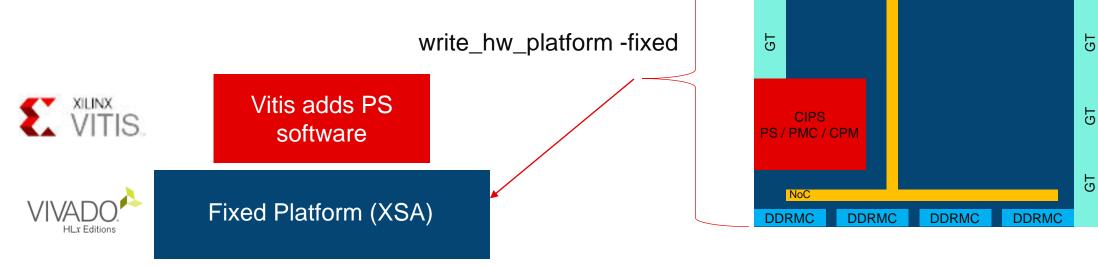
#### Hardware Design Flow

- Design uses fabric (+ NoC, DDR, GT, PCIe)
- Tools:
  - Vivado to create the PDI directly
- CIPS must be included in the design
- IPI will play a larger part in your design process



# Versal Design Flows (Vivado to Vitis)

- Traditional Embedded
  - Fabric + PS
  - Tools:
    - Vivado to create a fixed platform (XSA)
    - Vitis to program the PS





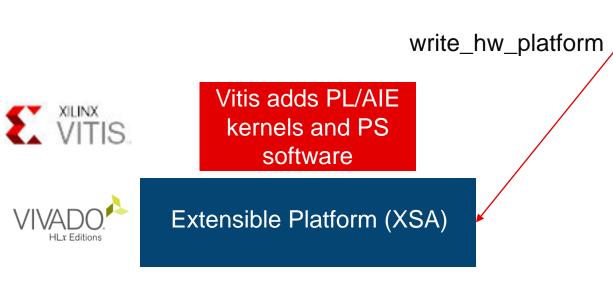
Ч

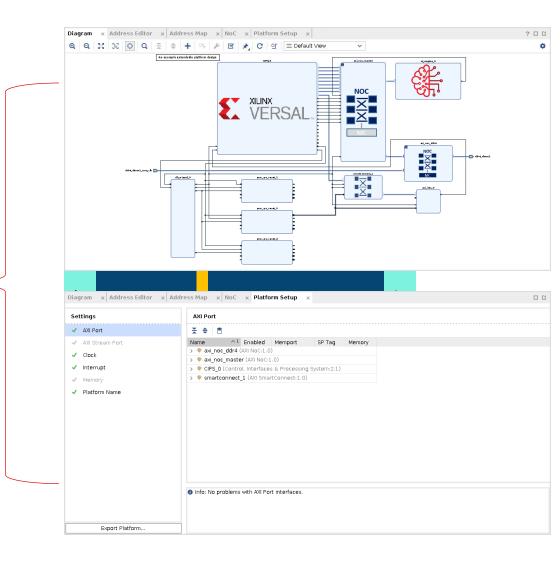
GЧ

# Versal Design Flows (Vivado and Vitis)

#### Acceleration

- Fabric + PS + AIE
- Tools:
  - Vivado to build extensible platform
  - Vitis to program, PS, AIE, PL accelerators









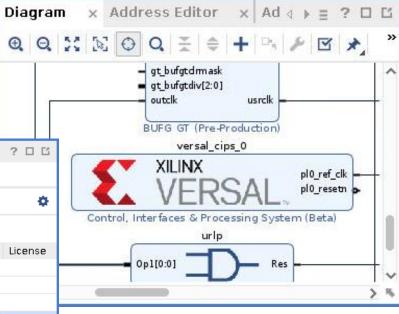
# **Versal Specific IP**



### **Versal IP- CIPS**

- CIPS- Control, Interfaces, and Processing System
- One IP contains covers many functions
  - PS
  - PMC
  - Debug
  - NoC
  - CPM
  - System monitors
  - SEM
  - Tamper

Cores   Inte	erfaces				
Q 🛨 🛊	¥. 4. 1	Ø 0			ø
<u>S</u> earch: Q-C	IPS	(1	match)		
Name		∧ 1	AXI4	Status	License
🗸 📄 Viva <b>do</b> F	Repository				
🗸 📄 Embe	dded Processing				
🗸 📄 Pro	ocessor				
		Drassesing Custors	AVIA AVIA Churches	Pre-Production	Included
Ŧ	Control, Interfaces & I	Processing System	AXI4, AXI4-Stream	Pre-Production	included
¢ Contails	Control, interfaces &	Processing System	Axi4, Axi4-stream	Pre-Production	included >
¢ Details				Pre-Production	included >
<	Control, Interfaces & Control, Interface 2.1			Pre-Production	included >
Cetails Name:	Control, Interface			Pre-Production	>



**E** XILINX.

## **Versal IP- CIPS Configuration**

Control,	Interfaces	& Processing	System (2.1)
----------	------------	--------------	--------------

🚯 Documentation 🛛 🚍 IP Location

Component Name versal\_cips\_0

	Security Related Tamper Event	Zeroize BBRAM	AM Tamper Response		
Home Boot Mode	□ JTAG Toggle		SYS INTERRUPT	7	
Debug	Temperature Alarm		SYS INTERRUPT	-	
PS-PMC	Voltage Alarm For VCC_PSLP		SYS INTERRUPT	~	
> CPM4 ~ Device Integrity	Voltage Alarm For VCC_PSFP		SYS INTERRUPT	~	
Sysmon Configuration	Voltage Alarm For VCC_PMC Or VCCAUX_PMC		SYS INTERRUPT	-	
XilSEM Library Configuration Tamper	Voltage Alarm For VCC_SOC		SYS INTERRUPT		
	Voltage Alarm For VCCINT Or VCCAUX Or VCC_RAM		SYS INTERRUPT		
	Voltage Alarm For VCC0_503		SYS INTERRUPT	~	
	Glitch Detector		SYS INTERRUPT	~	
	Tamper Trigger Register		SYS INTERRUPT	7	
	External from MIO		SYS INTERRUPT	~	

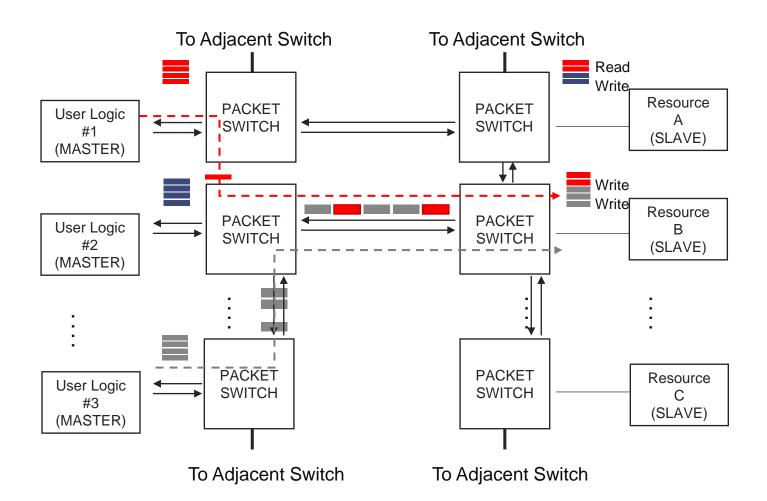
**Re-customize IP** 



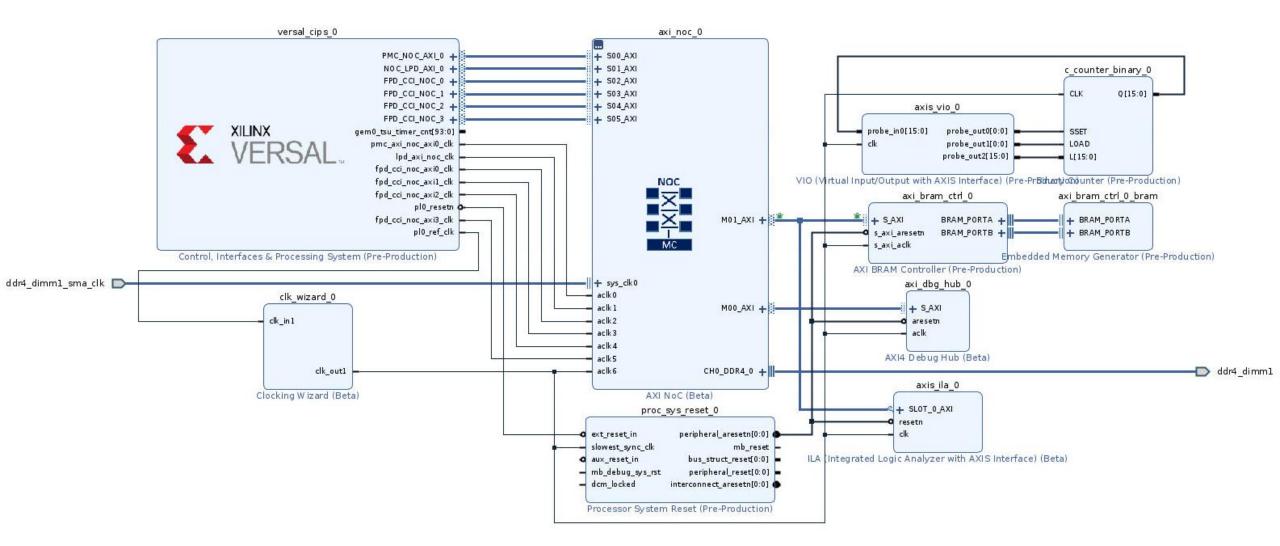
 $\odot$   $\odot$   $\otimes$ 

# Versal IP – NoC (Network on Chip)

- Shared connectivity to move packetized data around the SoC
- Facilitates communication between
  - Processing system
  - DDR
  - AI Engines
  - Programmable logic
  - Any other hardened components

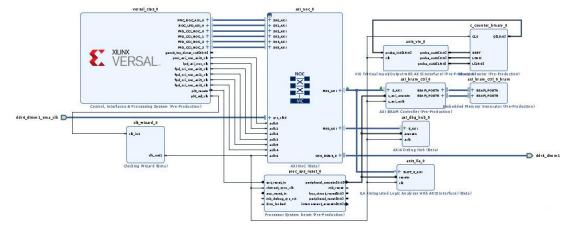


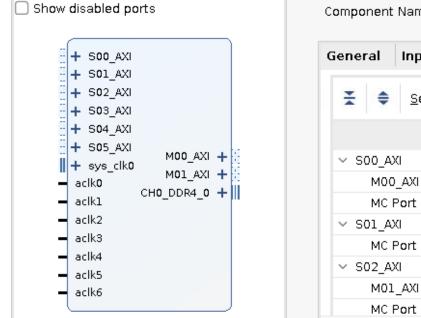
### Versal IP – Using the NoC (Network on Chip)





#### Versal IP – Using the NoC (Network on Chip)



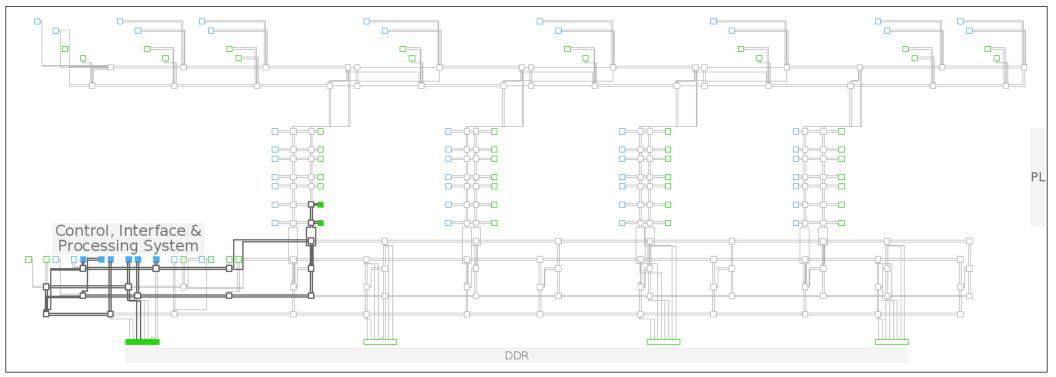


Component Name	axi noc 0

eneral Inputs	3 Outputs	Conr	nectivity QoS	D	DR Basic	DDR Mem	ory DDR A	ud ∢ →
¥ ♦ Searc	:h: Q-				Gbps	Advance	ed Run No	C DRC:
	Read Traffi Class	с	Write Traffic Class			Bandwidth Read (MB/s)	Bandwidth Write (MB/s)	
~ S00_AXI	BEST_EFFORT	~	BEST_EFFORT	$\sim$				
M00_AXI					yes	5 🖉	5 🖉	
MC Port 0					yes	5 🖉	5 🖉	
✓ S01_AXI	BEST_EFFORT	~	BEST_EFFORT	~				
MC Port 0					yes	5 🖉	5 🖉	
✓ S02_AXI	BEST_EFFORT	~	BEST_EFFORT	~				
M01_AXI					yes	5 🖉	5 🖉	
MC Port 0					yes	5 🖉	5 🖉	

## Versal IP – NoC (Network on Chip)

- The NoC physically spans the SoC
  - Blue NMUs (NoC master units)
  - Green NSUs (NoC slave units)
  - White NPS (NoC packet switch)

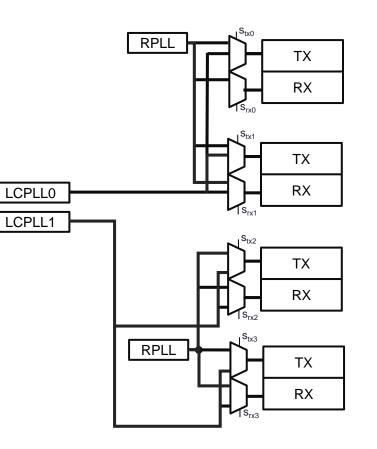




### **GT Wizard Features**

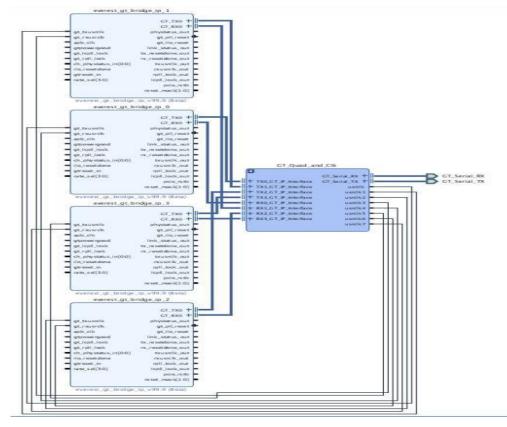
#### • GTs inherit values from the parent IPs

- Makes it easier to split up the GT Quad
- Configure the GT through the parent without ever having to open the Wizard
- Parent IP is packaged without the GT
- Flow is primarily IPI based
  - RTL flow is possible but multi-IP flow not integrated into wizard
  - Allows more intelligent sharing of PLL resources, validation of use cases at run time
- Pin Planning now part of Vivado, not IP generation
  - New GT tab similar to the Memory flow
  - Only full Quad pin planning allowed, no lane swapping allowed



### **GT Wizard Features**

- Resource sharing is now a lot easier
  - Split up the Quad as you see fit
  - Let the tools give you a best fit
- Simplify rate changes
  - GT wizard can generate multiple ELF configuration files through GT Quad customization options
  - DRP-like functionality is natively incorporated into the Quad (APB AXI bus)
- Bridge IP
  - For third party solutions
  - Acts as an interface between GT Quad and custom logic

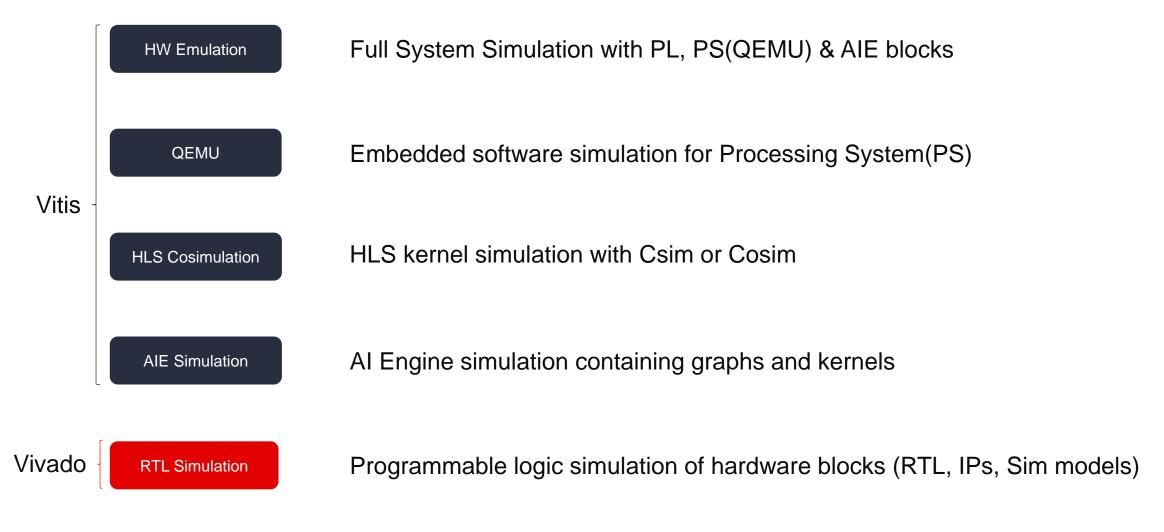




# **Simulation and Debug**



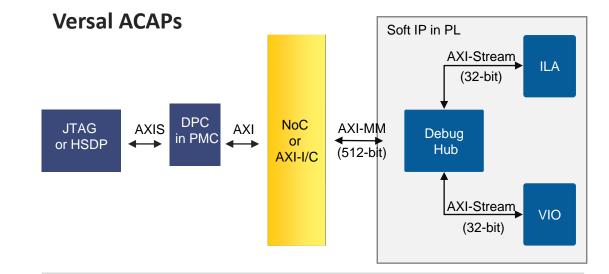
#### Simulation

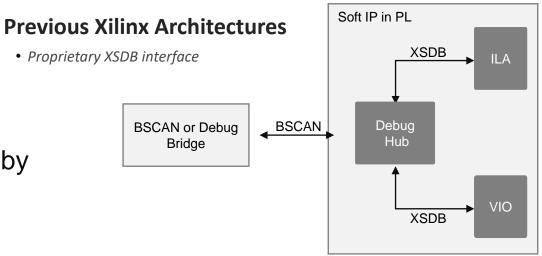




# **Versal IP - HW Debug**

- Versal Debug Cores Use AXI-Streaming Infrastructure
- Familiar Debug IP
  - Integrated Logic Analyzer (AXIS-ILA)
  - Virtual Input/Output (AXIS-VIO)
  - Memory Calibration Debug Interface
- New Debug IP
  - PCI Express Link Debug
  - Hardened Integrated Bit Error Ratio Test (IBERT)
- HSDP (hardened part of XPIPE)- performance benefits – loading the linux kernel into memory by JTAG is slow (12.8Gb Smart link plus 10Gb vs. JTAG 100Mb )









# **Device Programming**

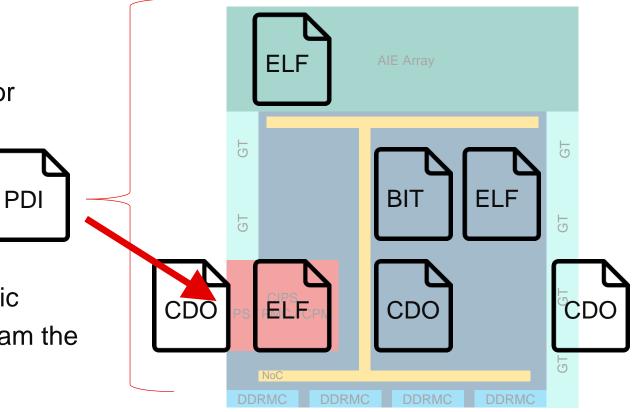


## **Programmable Device Image Demystified**

- A PDI is essentially the Versal equivalent of a "bitstream"
- PDI contents
  - CDO files Configuration register writes for hardened IP
  - BIT file Fabric CFrame data
  - ELF files AIE/PS/uB software

#### PMC uses

- Cframe interface (CFI) to program the fabric
- NoC programming interface (NPI) to program the NoC interconnect
- AXI interface to load the AIE



#### **Conclusion / Resources**

Three recommended design entry flows used for Versal

- New Versal Hard IP
- Simulation and debug
- PDI
- Resources
  - AM011 Versal Technical Reference Manual
  - UG1273 Versal ACAP Design Guide
  - PG352 CIPS IP
  - PG313 NoC IP
  - PG331 Transceiver Wizard
  - Versal online documentation



Versal ACAP Design Proce	ess Documentation		
Xilinx documentation is organized around a set of us process of interest for more information.	ser design processes to help you find relevant conte	ent for your design needs. The high-level design proc	esses are displayed below. Click on the design
System and Solution Planning: Provides guidance for and AI Engine.	or identifying the components, performance, I/O, an	d data transfer requirements at a system level. Inclu	des application mapping for the solution to PS, PI
System and Solution	Board System Design	Hardware, IP, and	Embedded Software
System Integration and	⊣oo ⊣oo		
Guided Pre-Fil	tered		
Overview	+	Overview	
Identify Xilinx Device for the Design	+	Ider	tify performance and power requirements
Map Applications (to PS, PL and AI Engine)	+	for the Design	tify and define memory requirements
		Map Applications (to PS, PL, and Al Engine)	and data transfer)
Versal Product Page	Design Hubs	Home Page	Send Feedback