SYNOPSYS®
Silicon to Software™

Creating Optimized AI SoC Architecture Using Virtual Prototyping

Mojin Kottarathil, Staff Applications Engineer

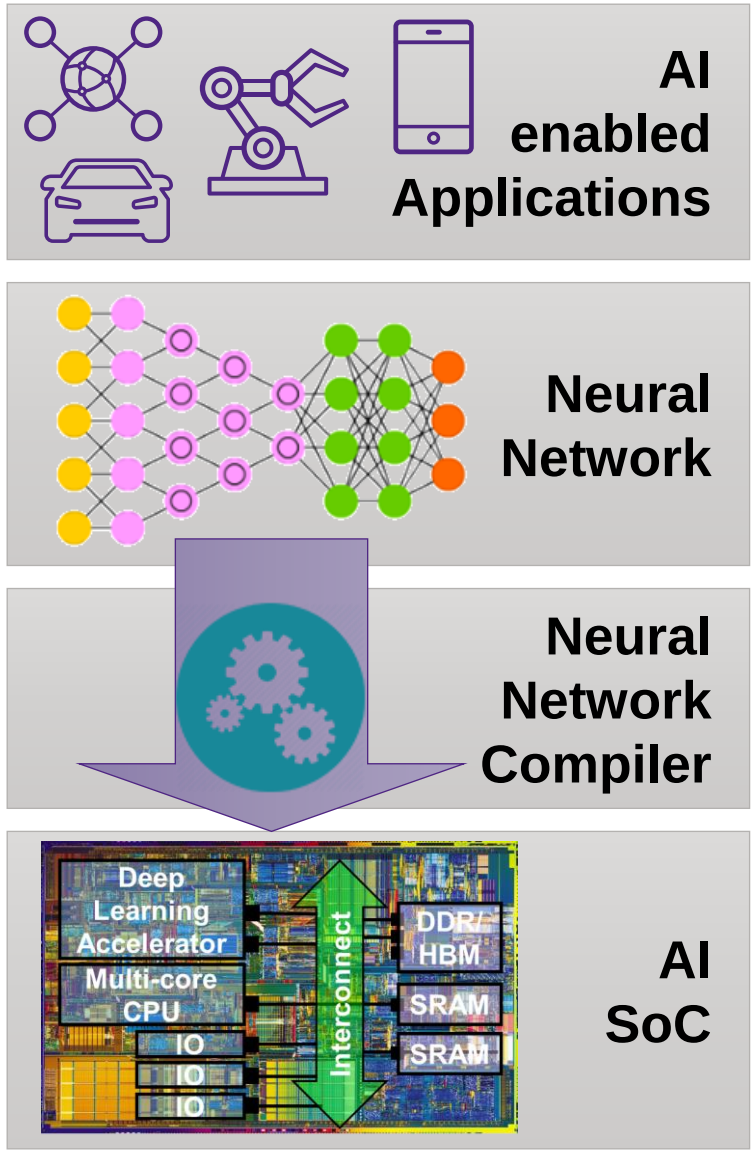
Synopsys ARC® Processor Summit 2022

Agenda

- Recent advancements in embedded AI applications and architectures
- Challenges in the design and verification of AI SoCs
- Synopsys Virtual Prototyping for early architecture analysis and optimization
- AI SoC platform case-study with ARC Processor IP
- How to get started

AI SoCs: A New Golden Age for Computer Architecture

- Applications becoming smart
  - autonomous vehicles, smart IoT, robots, etc.
  - AI moving to the client for better cost, latency, reliability
- Neural Networks are getting bigger
  - More accurate results, higher image size, complex NLP models
- Software is often the hardest part
  - Need optimizing compilers to map applications to custom chips
  - ResNet-50 is easy, real workloads are hard
- Moore's Law winds down - Domain-Specific Architectures gain
  - Custom accelerators/data-paths/instructions, SIMD
  - Many startups, semiconductors, super-scalers build AI SoCs



AI SoC Design Challenges

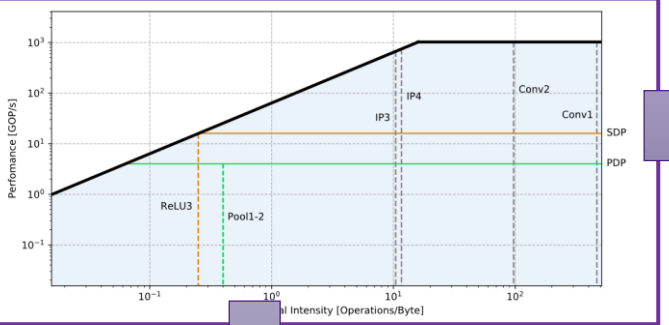
Brute-force Processing of Huge Data Sets

- **Choosing the right algorithm and architecture: CPU, vector DSP, ASIP, DNN accelerator**
  - DNN graphs are evolving fast, need short time to market and cannot optimize for one single graph
  - Joint design of AI algorithm, compiler and SoC architecture
  - Joint optimization of power, performance, accuracy, and cost
- **Highly parallel compute drives memory requirements**
  - E.g. in computer vision: higher resolution, higher frame-rate, more cameras
  - High on-chip and chip to chip bandwidth at low latency
  - High memory bandwidth requirements for parameters and layer to layer communication
- **Power & Performance analysis require realistic workloads to consider dynamic effects**
  - Scheduling of AI operators on parallel processing elements
  - Unpredictable interconnect and memory access latencies

Large Design Space Drives Differentiation by AI Algorithm & Architecture

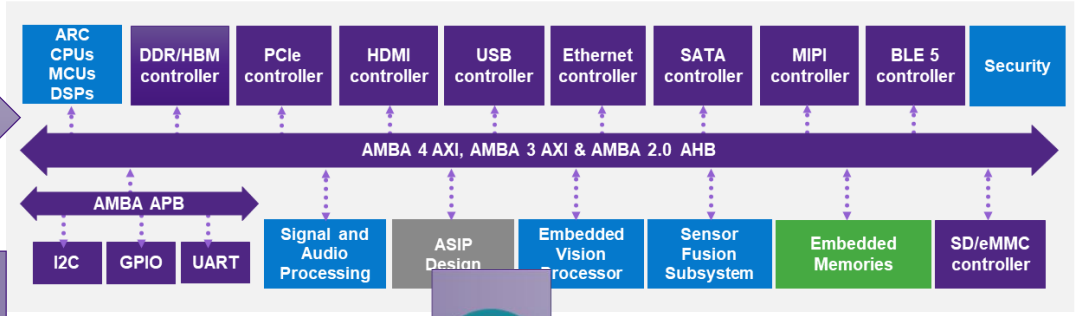
Shift Left Architecture Analysis of AI SoCs

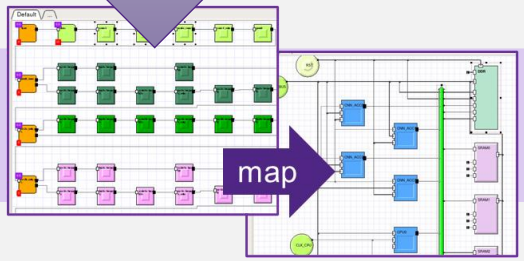
Analytical Performance Model

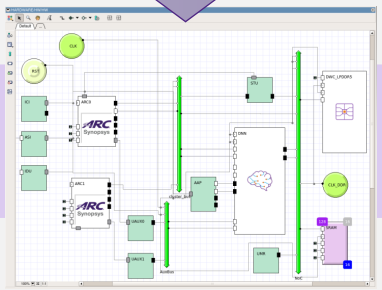


Architecture spec

Software







APM-based Workload Model
- Partitioning and exploration
- Interconnect/memory analysis

Fast Performance Model
- HW/SW co-optimization
- Performance/power analysis

RTL Emulation
- HW/SW co-verification
- Power characterization

RTL Prototyping
- HW/SW co-verification
- KPI validation

Model-based Architecture Simulation

RTL-based HW/SW co-verification

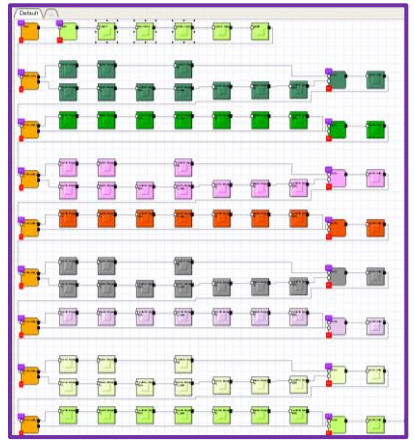
Use-cases for Architecture Analysis with Virtual Prototyping

Early architecture partitioning and exploration with workload models, calibrated from APM

- KPI capture and sensitivity analysis
- Traffic and application workload modeling
- HW/SW partitioning, architecture specification
- power/performance analysis

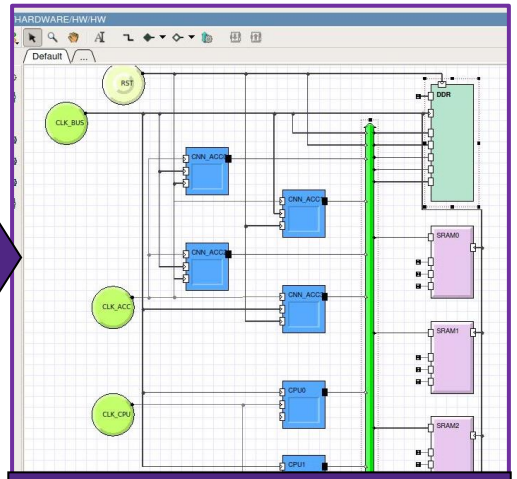
Performance optimization with Software

- KPI tracking and validation
- IP selection and benchmarking
- SoC performance validation
- L1/L2 cache & cache coherency optimization

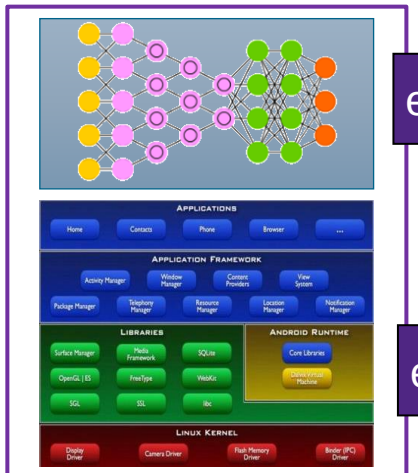


Workload model

map



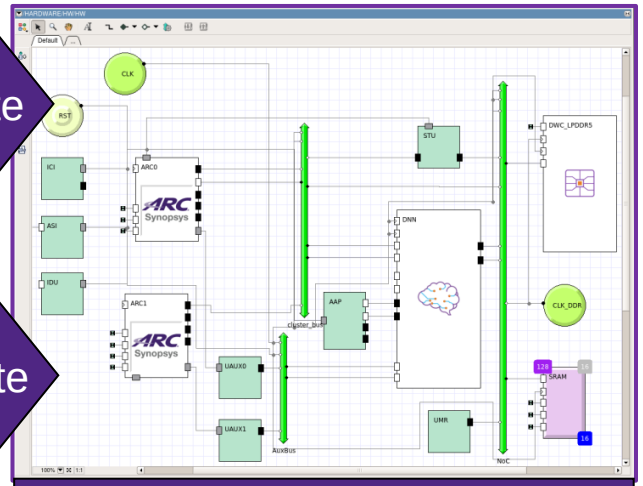
Hardware resource model



Application Software

execute

execute



Near cycle accurate Hardware model

Platform Architect Power and Performance Analysis Flow

Application specification

Software and traces

Model libraries

Power models and characterization

Parallel parameter sweep

Application workload

map

Hardware platform

Design space exploration

Workload trace and statistics

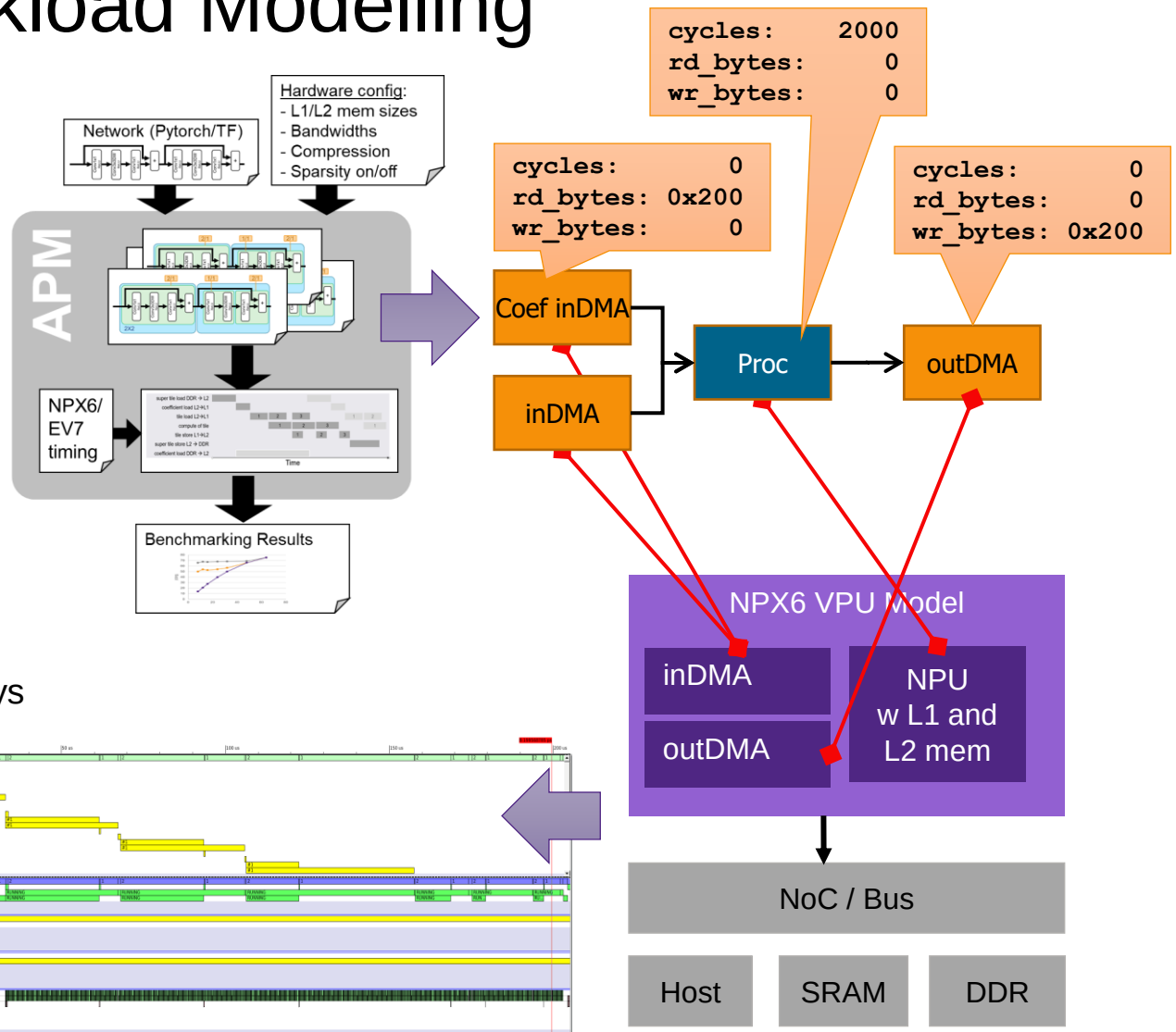
Root-cause analysis

Sensitivity analysis

ARC Processor Summit

Platform Architect Based Workload Modelling

- Analytic Performance Model (APM)
  - Used internally by Synopsys NPX System Architecture Team
- Workload Model generated from APM
  - Calibrated tasks for in-DMA, out-DMA, and processing
- SoC Platform Model
  - Accurate SystemC Transaction Level Models (TLM) of processing elements, interconnect and memory
- Map workload to NPX6 VPU (Virtual Processing Unit) model
  - Process VPUs has execution time of layer group
  - DMA execution times are based on actual bus and memory delays
- Analyze performance metrics
  - End-to-end performance
  - Workload activity
  - Utilization of resources
  - Interconnect metrics
    - Latency, Throughput
    - Contention, Outstanding transactions



ARC Processor Simulation Models

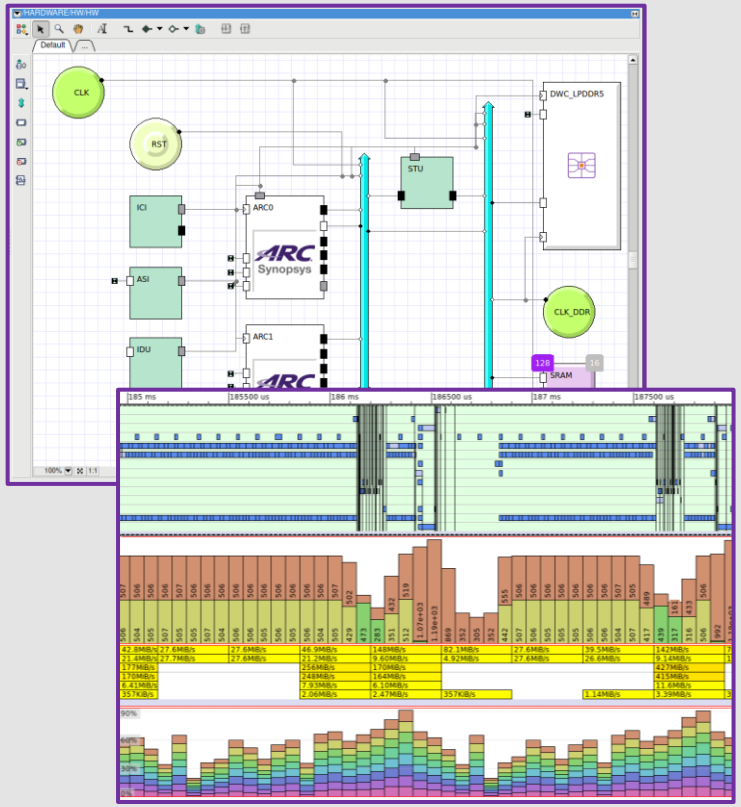
Support for building virtual prototypes

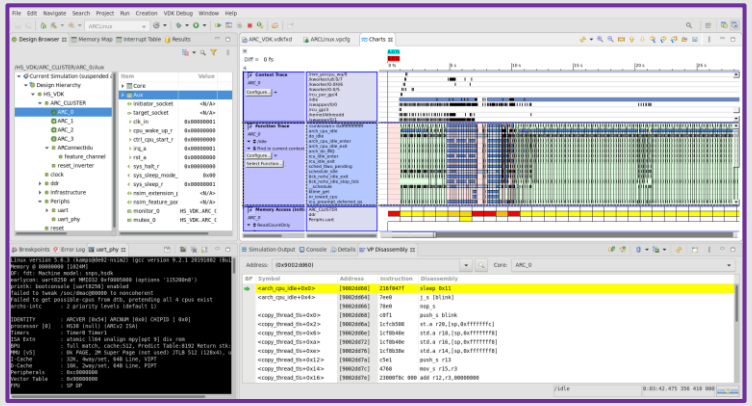
- nSIM NCAM has
  - SystemC wrapper
  - Model Libraries for Platform Architect and Virtualizer
    - For easy deployment in Synopsys Virtual Prototyping tools
    - Instrumented for debug and analysis

- Allows for easy creation of your own Virtual Platform

- Integration of MetaWare Debugger (mdb) into PA and Virtualizer
  - For debugging complete systems containing ARC IP models

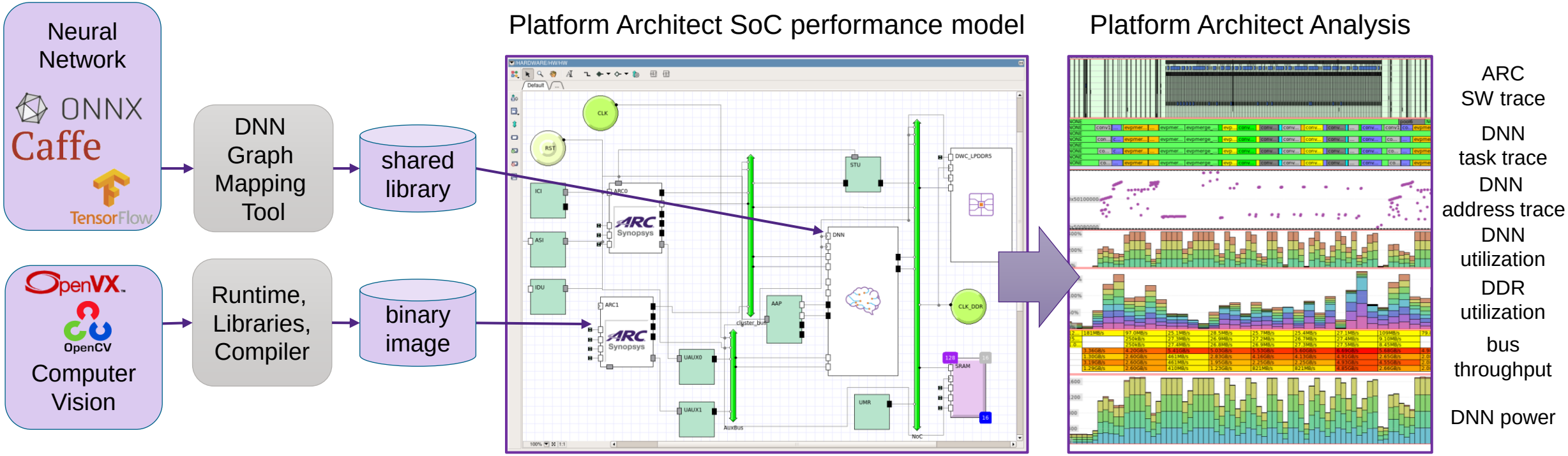
- Accurate model of ARC STU with non-blocking FT-AXI interfaces





ARC AI Fast Performance Model (FPM) in Platform Architect

Whitepaper "Performance Analysis Using ARC EV7x Fast Performance Model"



- Use MetaWare production build flow to compile DNN model and ARC Vector DSP binary image
- Use Platform Architect to execute application on cycle-approximate performance model in context of SoC platform
- Analyze AI application and SoC power and performance metrics,
  - e.g. Arc function profile, DNN trace, utilization, and address pattern, SoC bus and memory throughput and latency

Accuracy of FPM with FT interfaces in Platform Architect

Interconnect & memory models are crucial to achieve high accuracy for multi-core systems

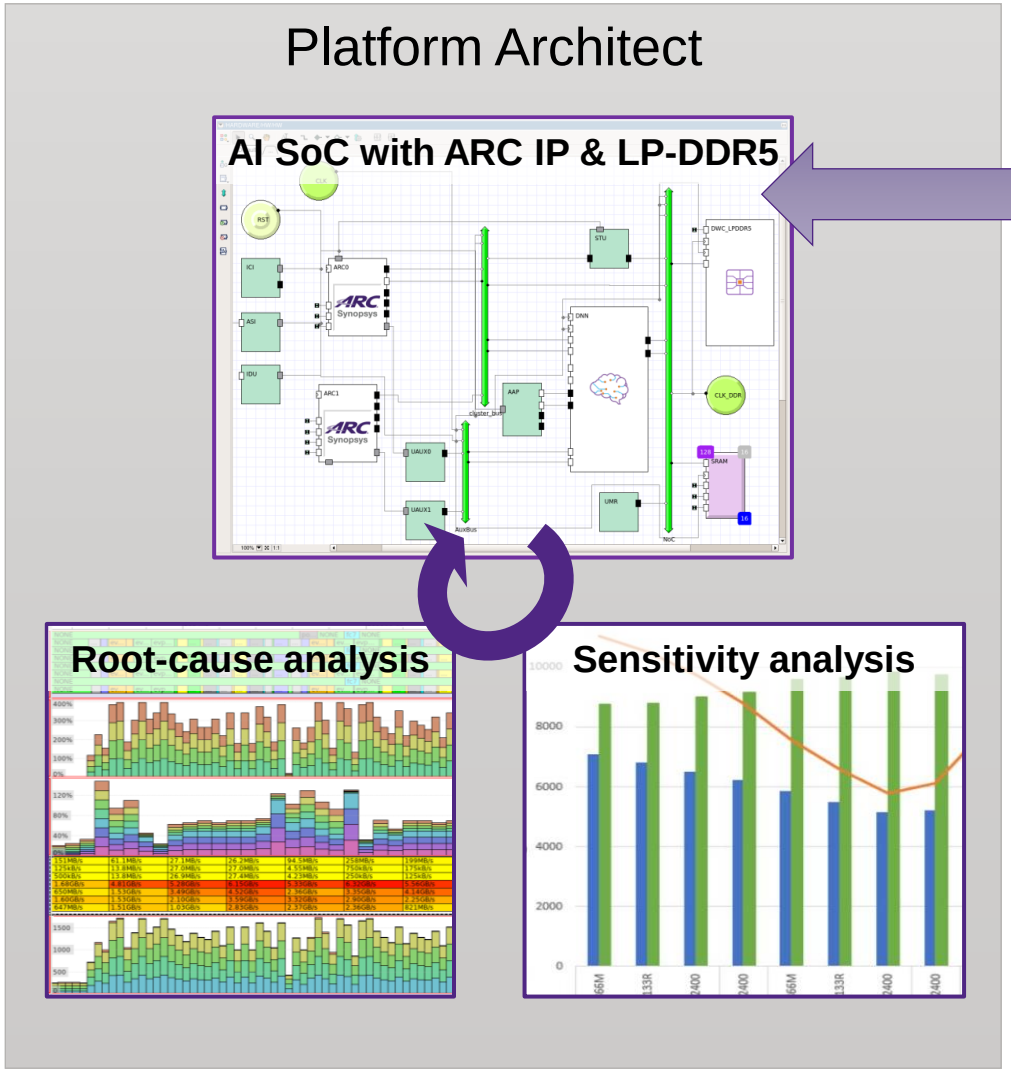
Neural Network Model	FPS ratio (single-core, 880 MACs)	FPS ratio (dual-core, 1760 MACs)
ResNet-50	101%	103%
Yolo-V2	101%	102%
Yolo-V3	100%	100%
MobileNet-SSD	104%	106%
MobileNet-V1	103%	106%
MobileNet-V2	102%	105%
OpenPose	100%	100%
SRGAN	104%	105%

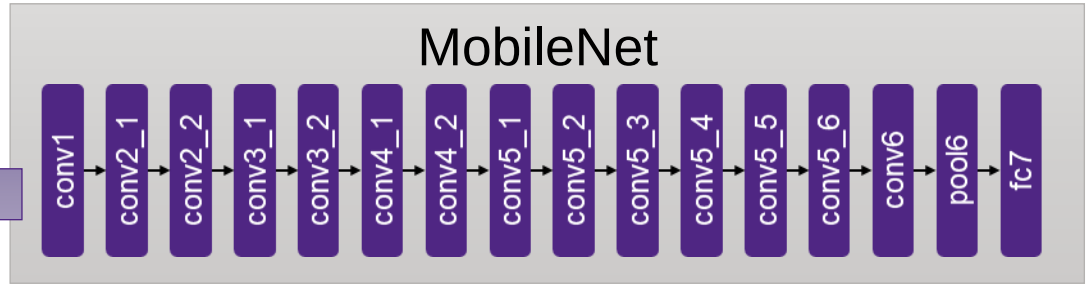
Table 1: ARC EV7x Processor FPM FPS as % of the hardware FPS. 100% means identical to hardware. >100% means an optimistic estimate.

AI SoC platform case-study with Fast Performance Model of ARC AI processor IP

- Capture an AI SoC platform with ARC AI processor IP, a Network-on-Chip, and DDR and SRAM memory hierarchy
- Analysis and optimization of IP-level and SoC architecture configurations

AI SoC Platform Case-study with ARC AI Subsystem





Goals:

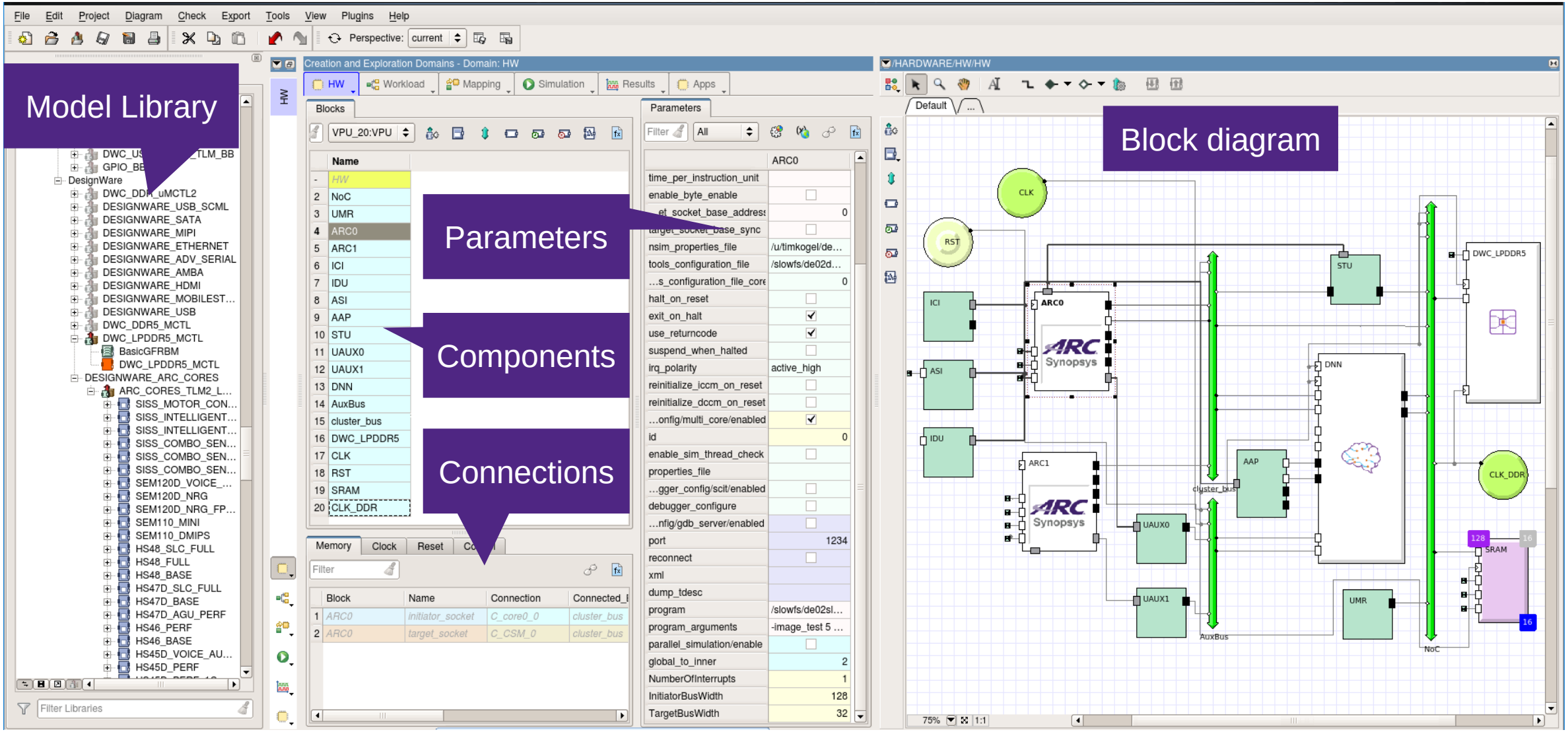
① 4 ms latency for inference of 5 frames

② minimize DNN power and energy

Optimize Hardware configuration:

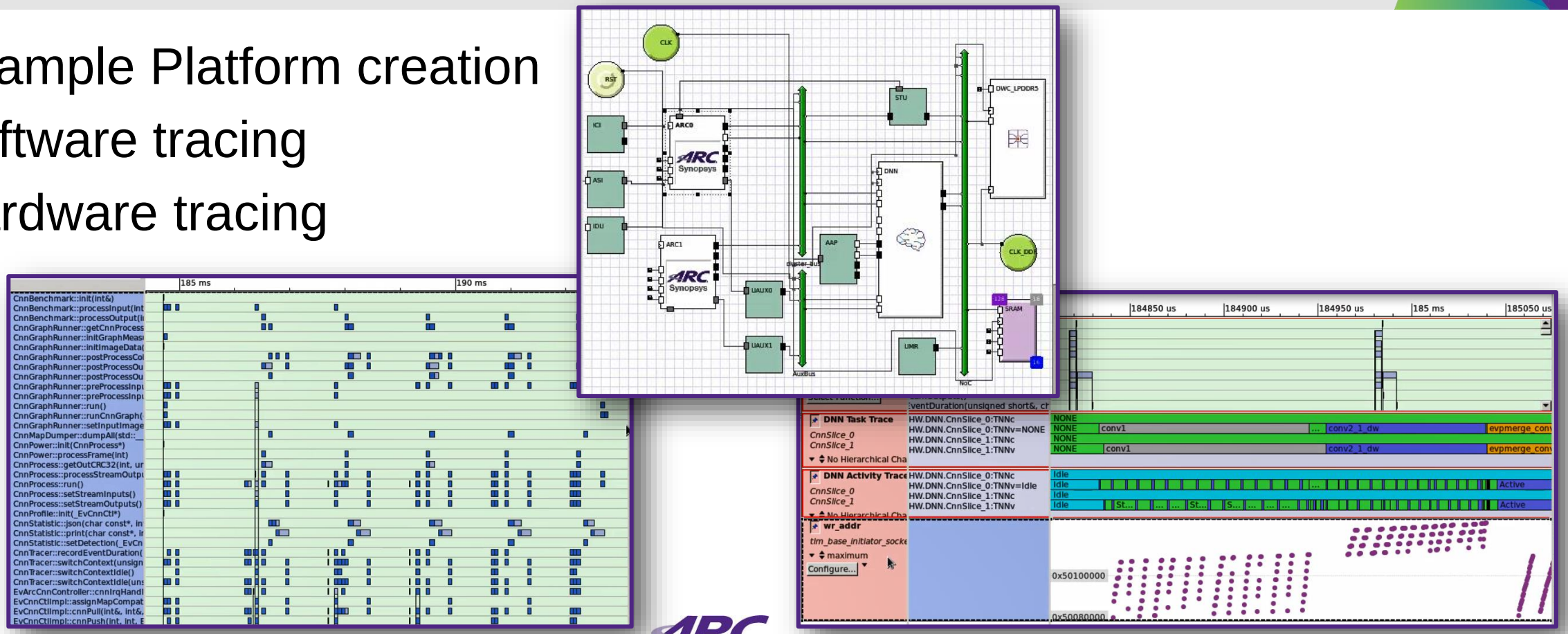
– IP configuration
– Speed of DDR memory
– Interconnect, buffers, transactions

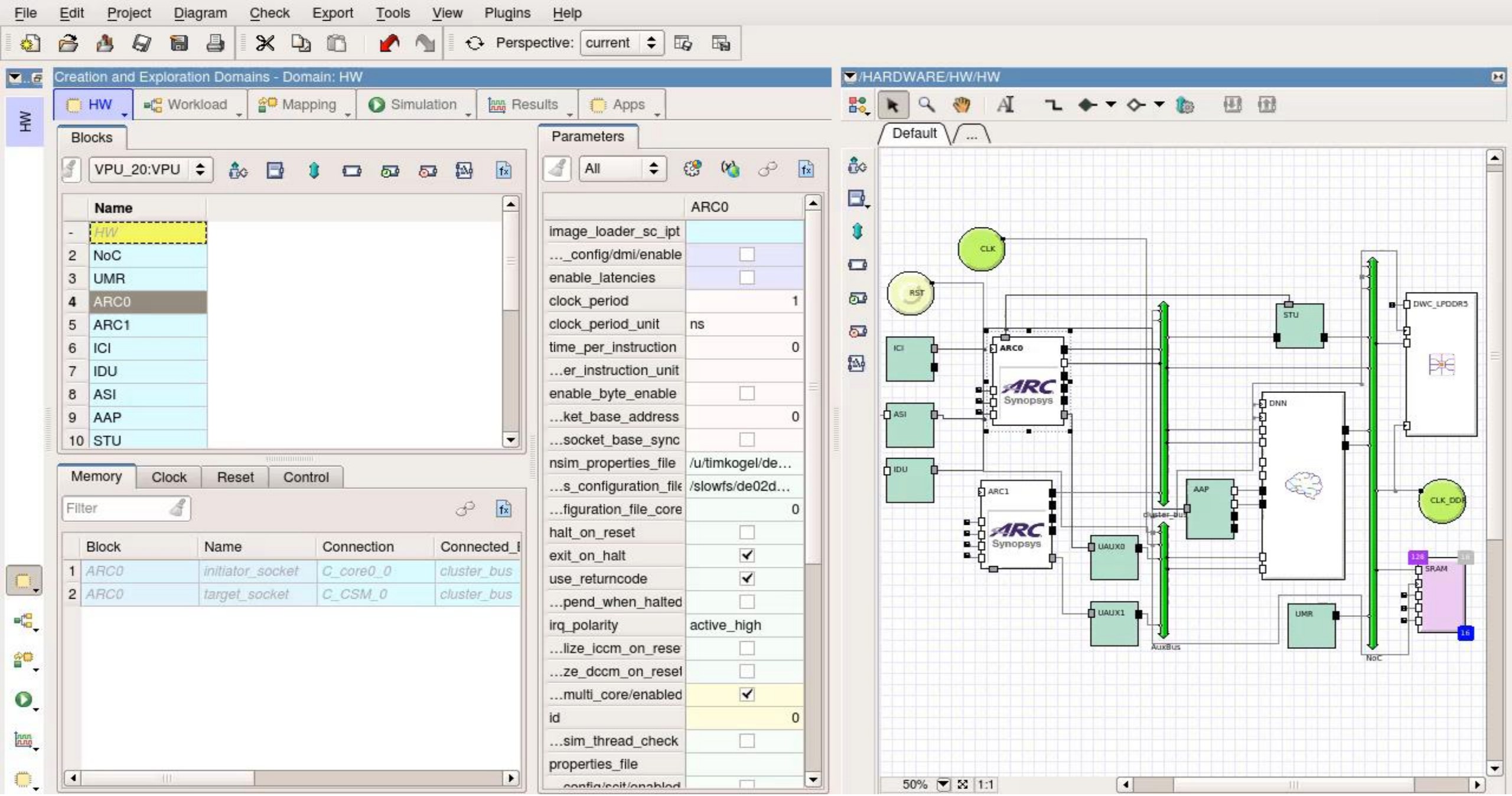
Platform Architect with ARC AI sub-system and DWC LPDDR5



Video 1: Platform creation and tracing

- Example Platform creation
- Software tracing
- Hardware tracing



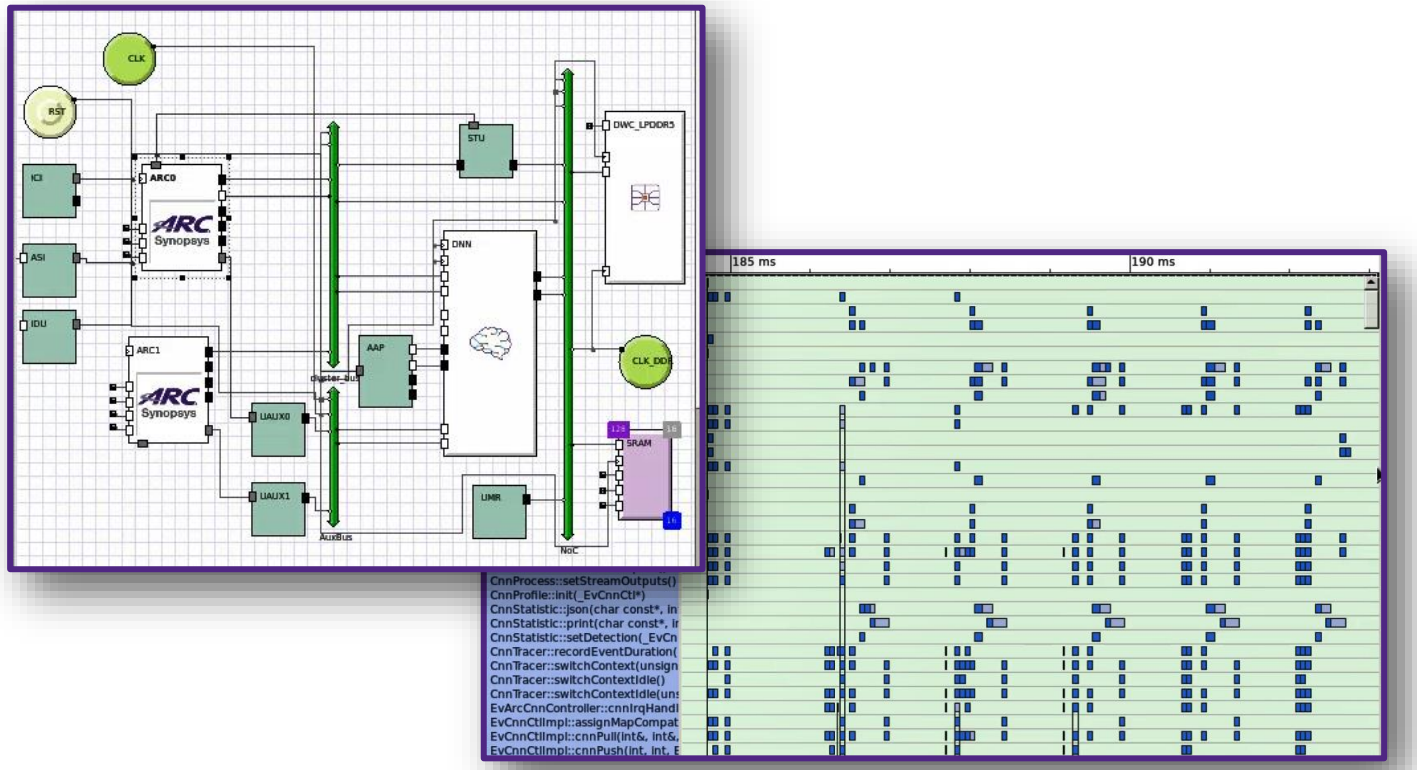


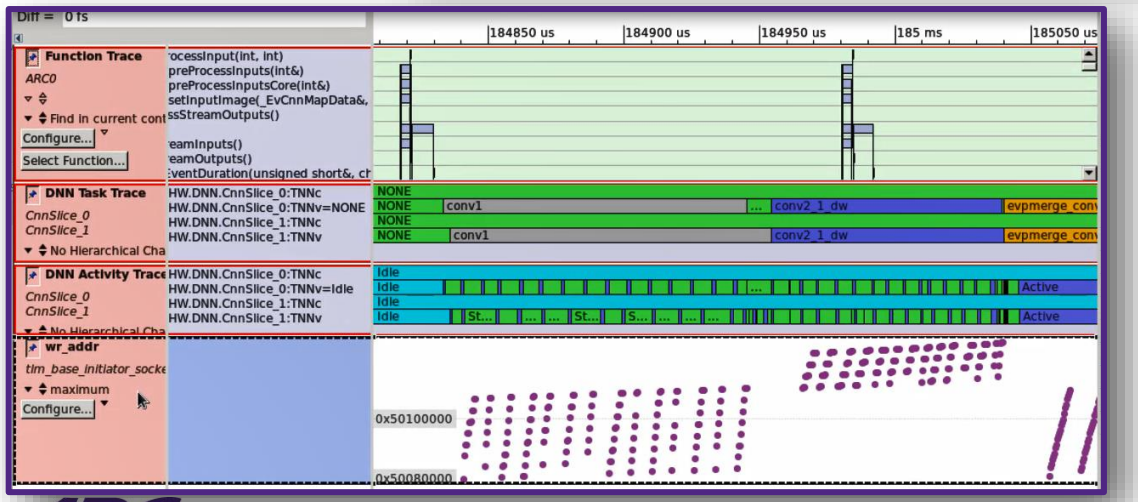
What We Just Learned

Platform creation and tracing

We learned how to:

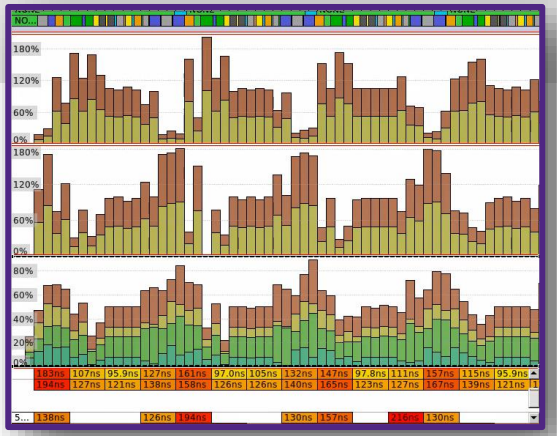
- ✓ Create demo platform with ARC Fast Performance Model and DesignWare LPDDR5 memory controller
- ✓ Use ARC VPX Function Trace to analyze Software activity
- ✓ Correlate Software trace with Hardware traces from DNN accelerator and interconnect

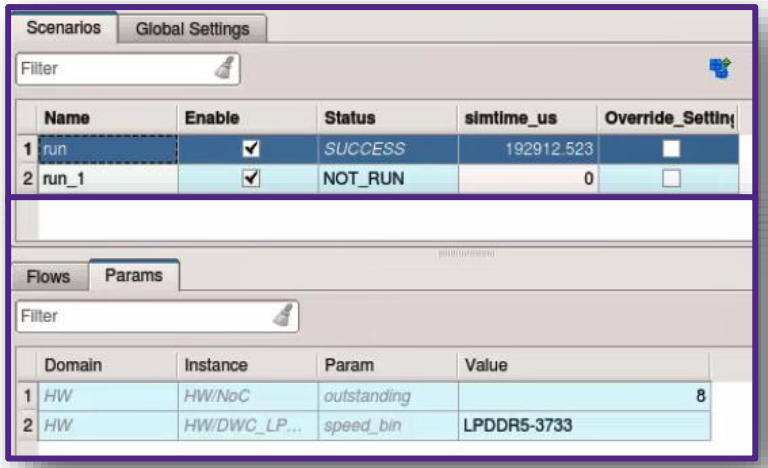


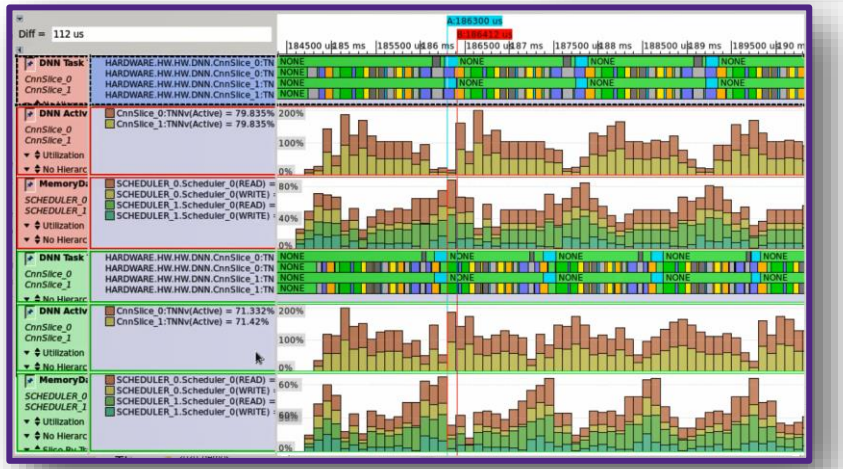


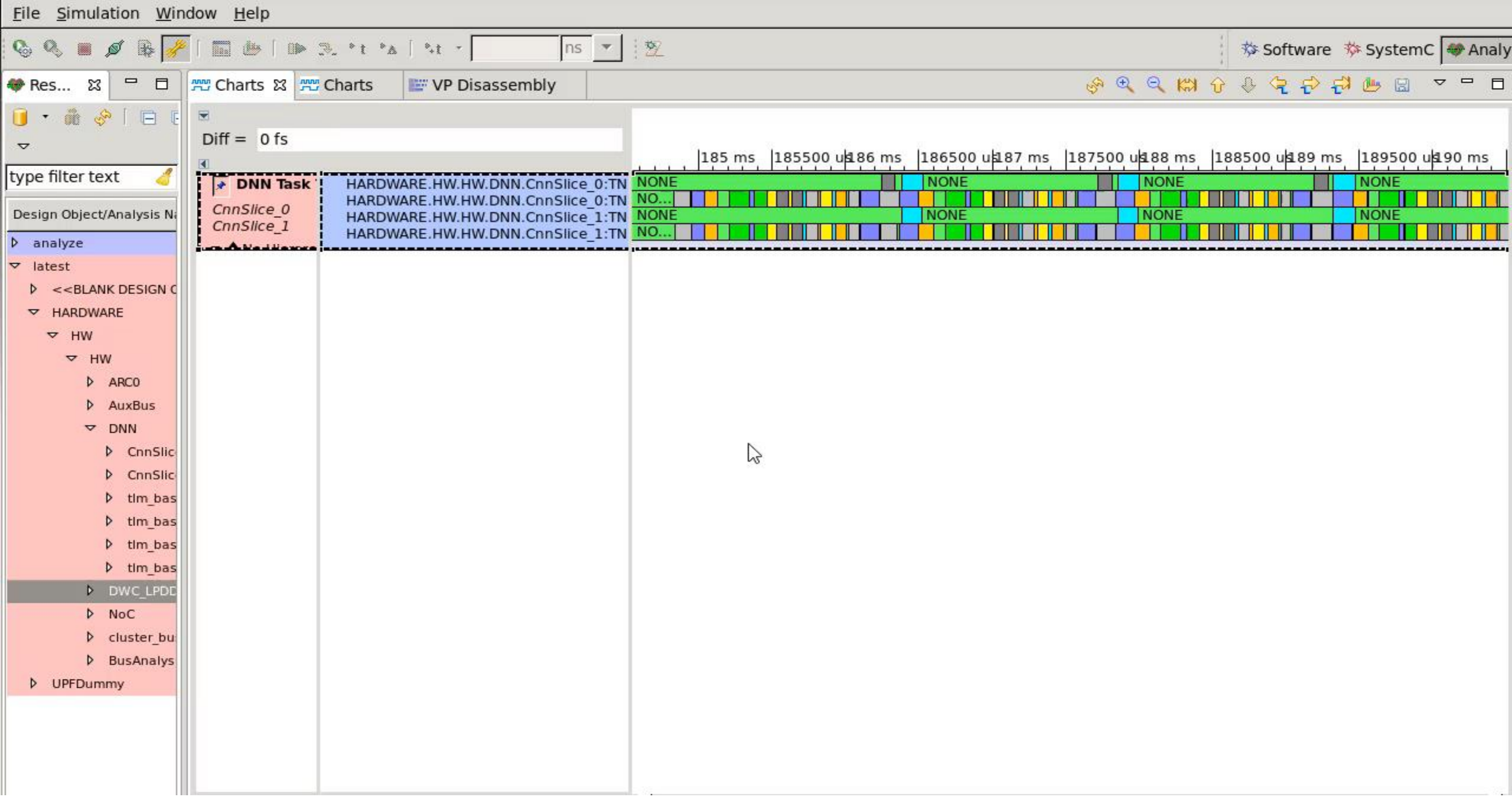
Video 2: Performance Analysis

- Performance analysis of initial result
- Change architecture configuration
- Compare results from different simulations







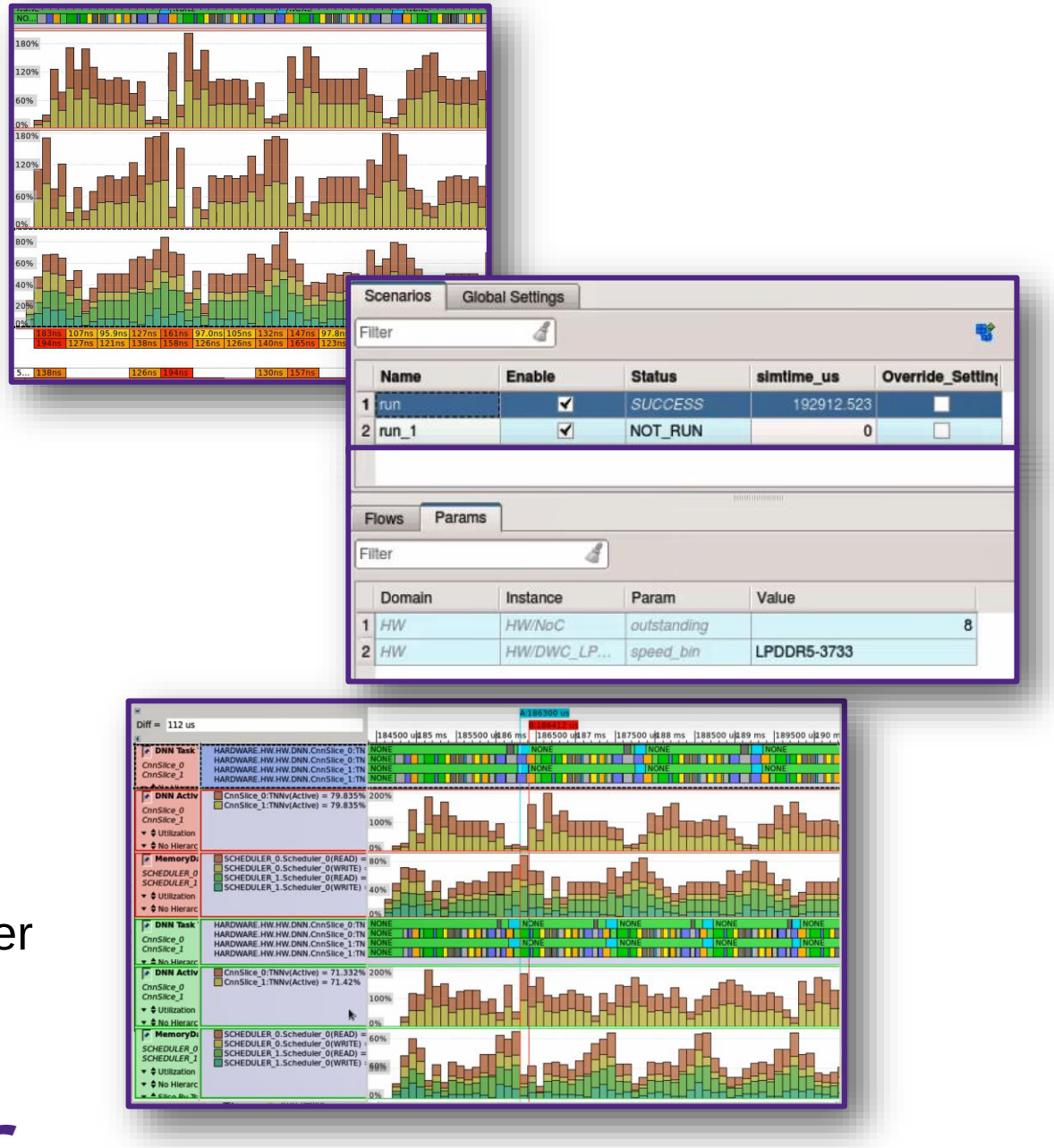


What We Just Learned

Performance Analysis

We learned how to:

- ✓ Analyze activity and stall cycles of ARC AI accelerator, correlate DNN activity with interconnect and LPDDR analysis views
- ✓ Change bus and LPDDR5 controller configuration to increase memory bandwidth
- ✓ Compare results from multiple runs, new results show diminishing returns from higher memory bandwidth



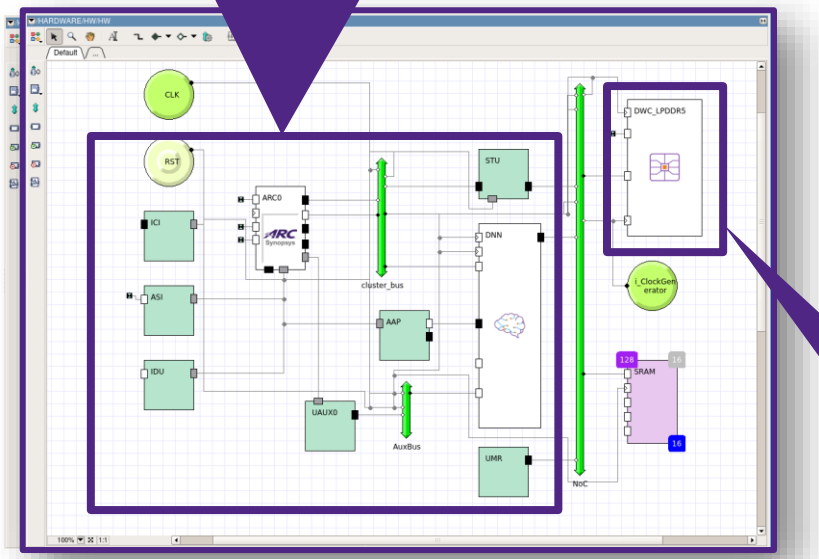
AI SoC Block Diagram in Platform Architect

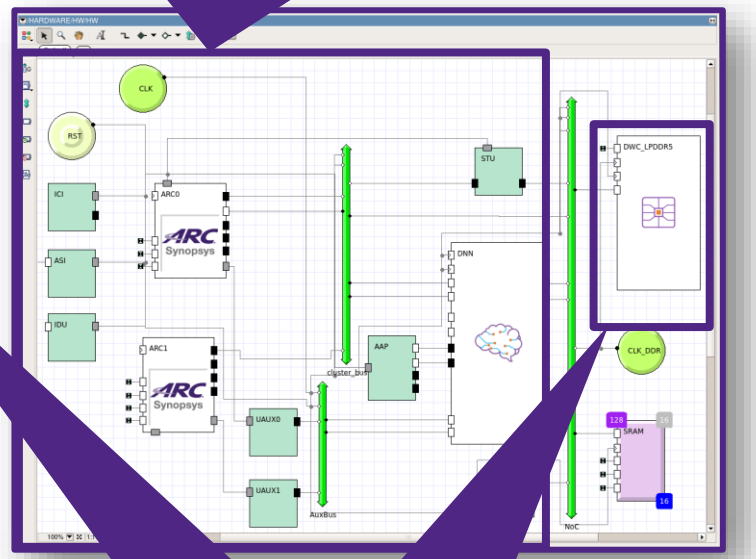
Scaling AI Sub-system and LPDRR5 memory controller

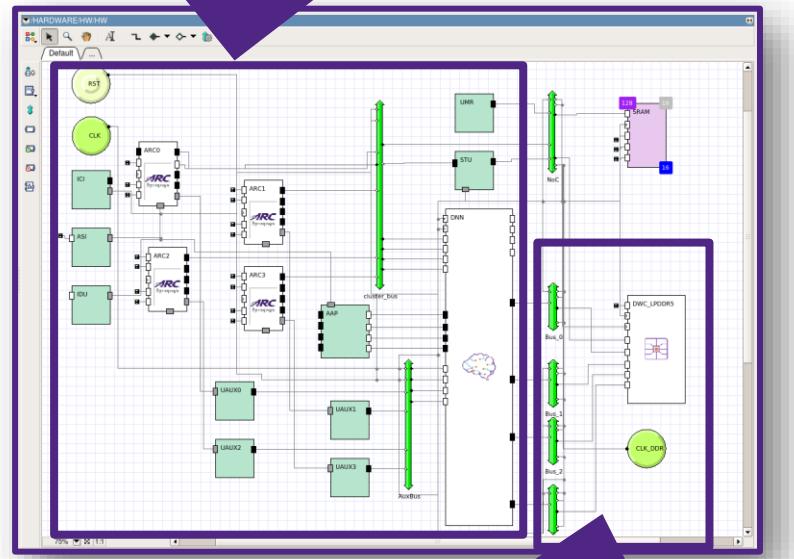
Single-core sub-system
- 1 ARC VPX cores
- 1 DNN slice

Dual-core sub-system
- 2 ARC VPX cores
- 2 DNN slices

Quad-core sub-system
- 4 ARC VPX cores
- 4 DNN slices





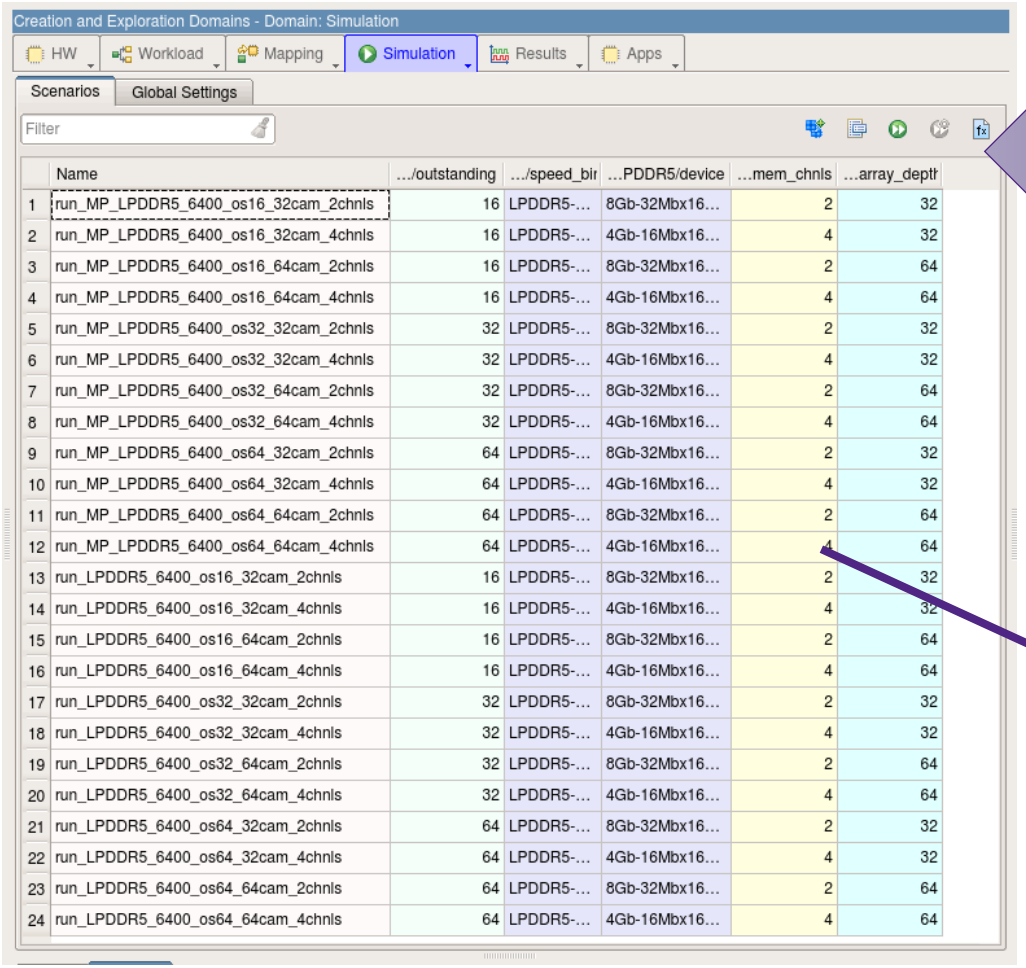


DesignWare LPDDR5 Memory Controller

- Multi-port LPDDR5 Mctrl
- parallel AXI bus fabric

AI SoC Architecture Sweep

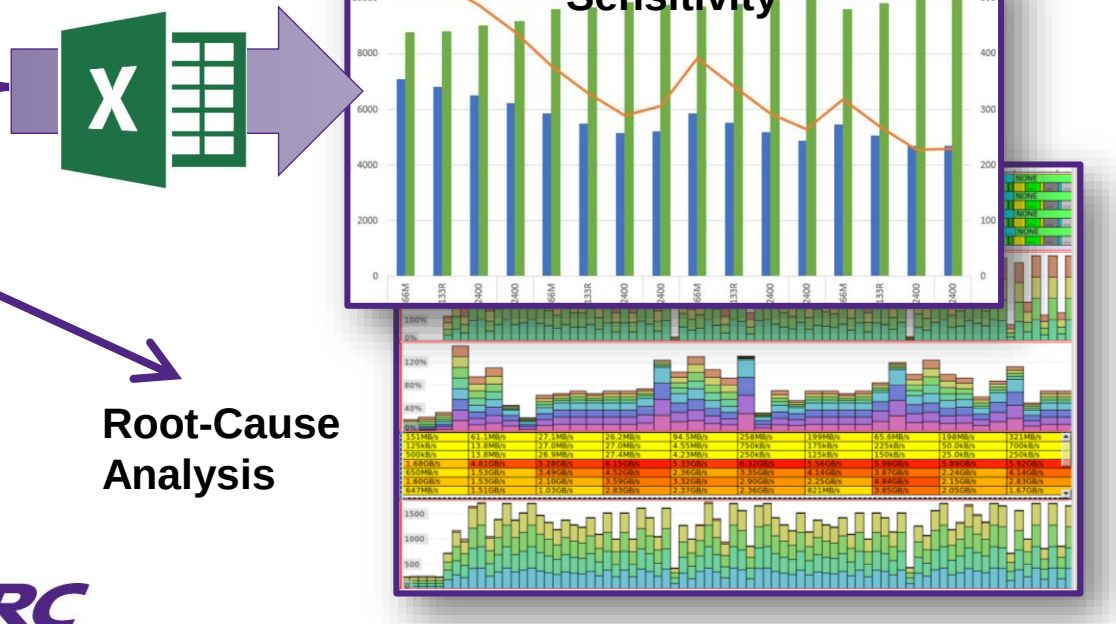
Goal: 4 ms inference latency, minimize power & energy



	Name	.../outstanding	.../speed_bir	...PDDR5/device	...mem_chnls	...array_depth
1	run_MP_LPDDR5_6400_os16_32cam_2chnls	16	LPDDR5-...	8Gb-32Mbx16...	2	32
2	run_MP_LPDDR5_6400_os16_32cam_4chnls	16	LPDDR5-...	4Gb-16Mbx16...	4	32
3	run_MP_LPDDR5_6400_os16_64cam_2chnls	16	LPDDR5-...	8Gb-32Mbx16...	2	64
4	run_MP_LPDDR5_6400_os16_64cam_4chnls	16	LPDDR5-...	4Gb-16Mbx16...	4	64
5	run_MP_LPDDR5_6400_os32_32cam_2chnls	32	LPDDR5-...	8Gb-32Mbx16...	2	32
6	run_MP_LPDDR5_6400_os32_32cam_4chnls	32	LPDDR5-...	4Gb-16Mbx16...	4	32
7	run_MP_LPDDR5_6400_os32_64cam_2chnls	32	LPDDR5-...	8Gb-32Mbx16...	2	64
8	run_MP_LPDDR5_6400_os32_64cam_4chnls	32	LPDDR5-...	4Gb-16Mbx16...	4	64
9	run_MP_LPDDR5_6400_os64_32cam_2chnls	64	LPDDR5-...	8Gb-32Mbx16...	2	32
10	run_MP_LPDDR5_6400_os64_32cam_4chnls	64	LPDDR5-...	4Gb-16Mbx16...	4	32
11	run_MP_LPDDR5_6400_os64_64cam_2chnls	64	LPDDR5-...	8Gb-32Mbx16...	2	64
12	run_MP_LPDDR5_6400_os64_64cam_4chnls	64	LPDDR5-...	4Gb-16Mbx16...	4	64
13	run_LPDDR5_6400_os16_32cam_2chnls	16	LPDDR5-...	8Gb-32Mbx16...	2	32
14	run_LPDDR5_6400_os16_32cam_4chnls	16	LPDDR5-...	4Gb-16Mbx16...	4	32
15	run_LPDDR5_6400_os16_64cam_2chnls	16	LPDDR5-...	8Gb-32Mbx16...	2	64
16	run_LPDDR5_6400_os16_64cam_4chnls	16	LPDDR5-...	4Gb-16Mbx16...	4	64
17	run_LPDDR5_6400_os32_32cam_2chnls	32	LPDDR5-...	8Gb-32Mbx16...	2	32
18	run_LPDDR5_6400_os32_32cam_4chnls	32	LPDDR5-...	4Gb-16Mbx16...	4	32
19	run_LPDDR5_6400_os32_64cam_2chnls	32	LPDDR5-...	8Gb-32Mbx16...	2	64
20	run_LPDDR5_6400_os32_64cam_4chnls	32	LPDDR5-...	4Gb-16Mbx16...	4	64
21	run_LPDDR5_6400_os64_32cam_2chnls	64	LPDDR5-...	8Gb-32Mbx16...	2	32
22	run_LPDDR5_6400_os64_32cam_4chnls	64	LPDDR5-...	4Gb-16Mbx16...	4	32
23	run_LPDDR5_6400_os64_64cam_2chnls	64	LPDDR5-...	8Gb-32Mbx16...	2	64
24	run_LPDDR5_6400_os64_64cam_4chnls	64	LPDDR5-...	4Gb-16Mbx16...	4	64

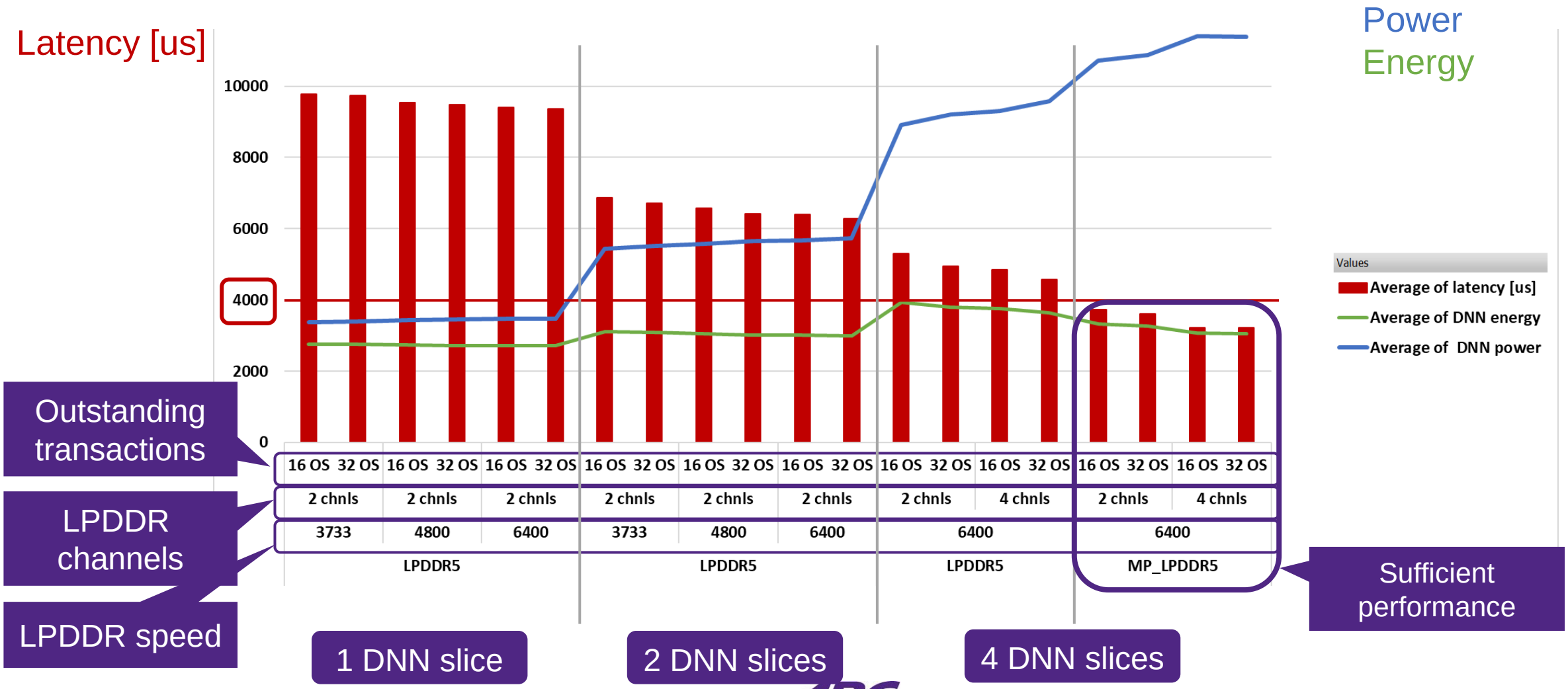
Sweep parameters

- AI configuration: 1, 2, 4 DNN slices
- Outstanding transactions: 16, 32, 64
- LPDDR5 memory speed: 3733, 4800, 6400
- Interconnect/LPDDR controller: single port, multi-port
- LPDDR controller scheduler queue: 32, 64
- LPDDR channels: 2, 4



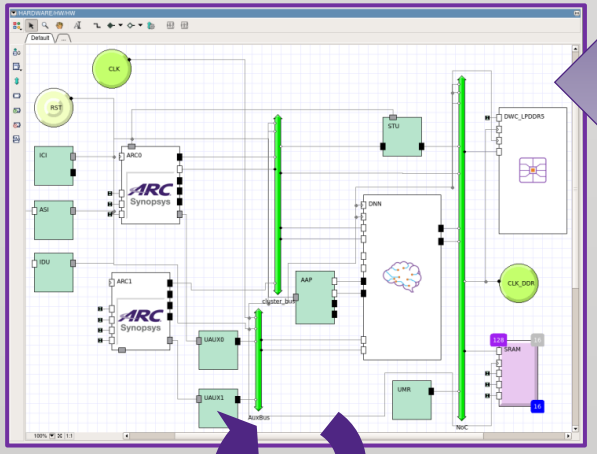
Analysis and Optimization of Architecture Configurations

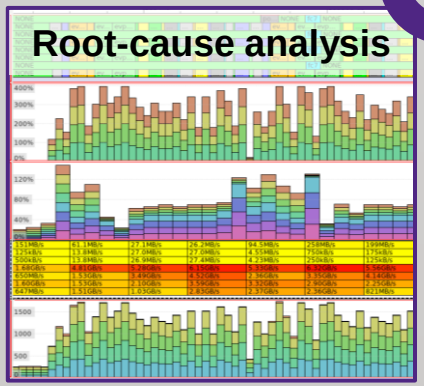
Inference latency for 5 frames vs. DNN power and energy consumption

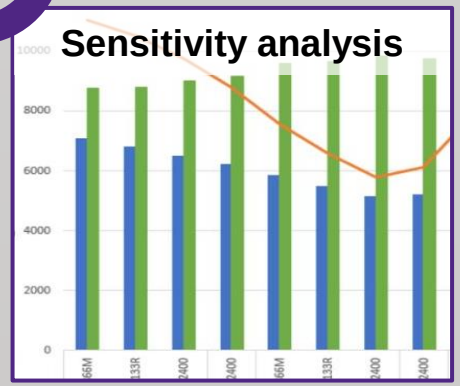


Example Summary

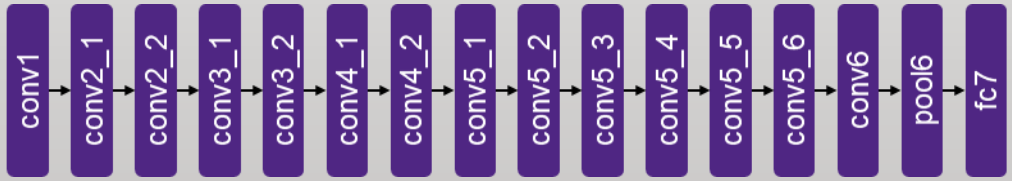
Platform Architect







MobileNet



Goals:

① 4 ms latency for inference of 5 frames

② minimize DNN power and energy

Optimized Hardware configuration:

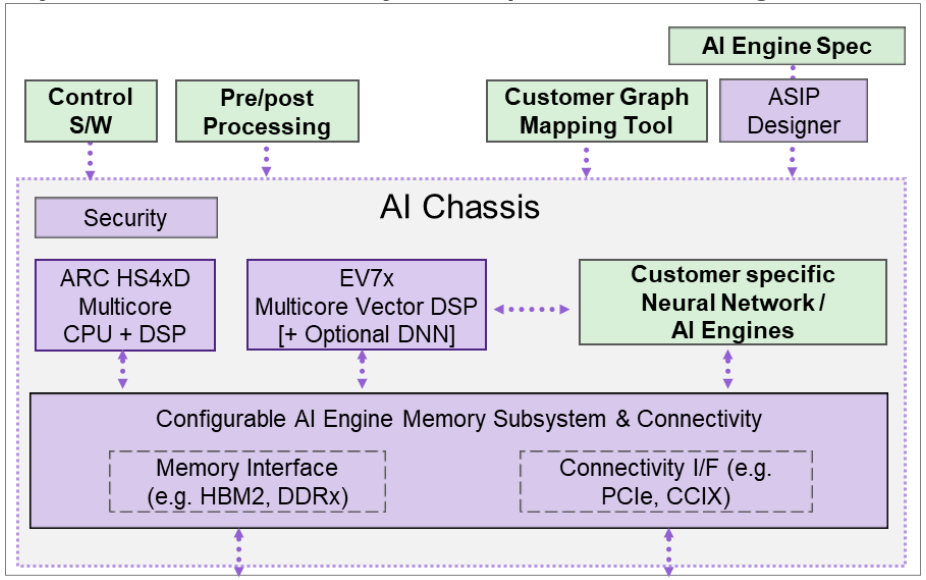
- AI configuration:1, 2, **4** DNN slices
- Outstanding transactions: 16, **32**, 64
- LPDDR memory speed: 3733, 4800, **6400**
- Interconnect/LPDDR controller: single port, **multi-port**
- LPDDR controller scheduler queue: **32**, 64

How To Get Started?

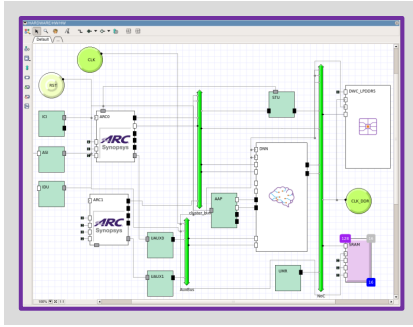
Faster Development of AI SoCs with Synopsys IP, tools, and services

Deep Knowledge in:

– AI Frameworks, AI & CNN Graphs, Graph Compression, and Mapping Tools
– Class leading CNN, State of the art Vector DSP, & ASIP capabilities
– Leading edge processor IP and SW (ARC)
– Mastery of key support IP (HBM, PCIe, DDR, MIPI)
– Foundry Process, Memory Compilers and Logic Libraries



Architecture Exploration & Optimization



Platform Architect

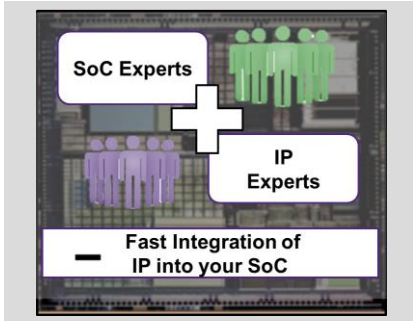
- Exploration and optimization flows
- Power and performance analysis
- Tooling for model creation and platform assembly
- Rich model library

Verification, Emulation & Prototyping

ZeBu/HAPS

- SoC verification
- Software development & bring-up
- Hybrid emulation
- Power & performance analysis
- AI benchmarks

Services



Services

- Architectural tradeoffs
- IP subsystems
- ASIP design
- System verification
- Early Software development

Thank You!

- Further resources
  - Landing page: DesignWare IP for Artificial Intelligence
  - Landing page: Platform Architect

- Further questions
  - Mojin.Kottarathil@synopsys.com

SYNOPSYS®
Silicon to Software™

Thank You