

Creating Optimized AI SoC Architecture Using Virtual Prototyping

Mojin Kottarathil, Staff Applications Engineer
Synopsys ARC[®] Processor Summit 2022

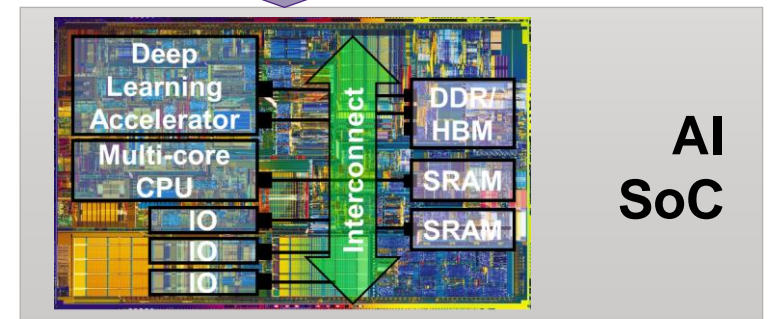
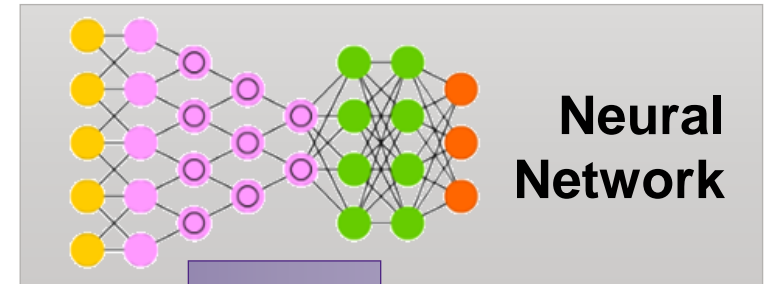
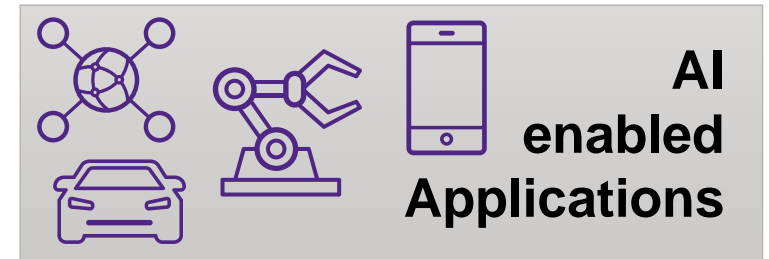


Agenda

- 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



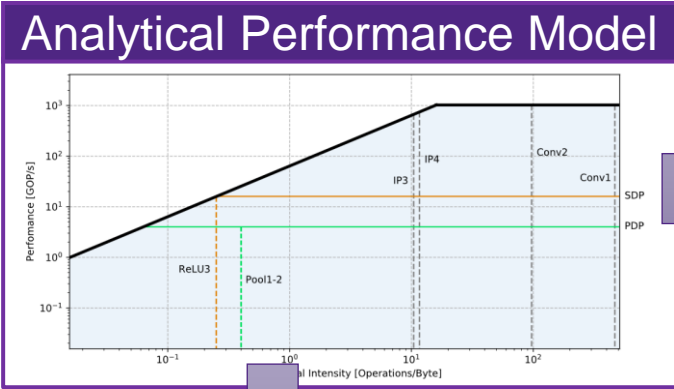
AI SoC Design Challenges

Brute-force Processing of Huge Data Sets

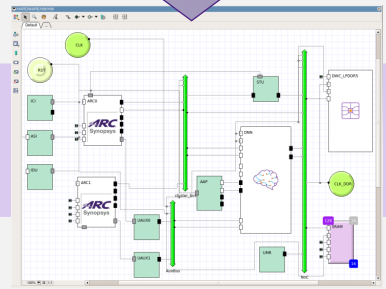
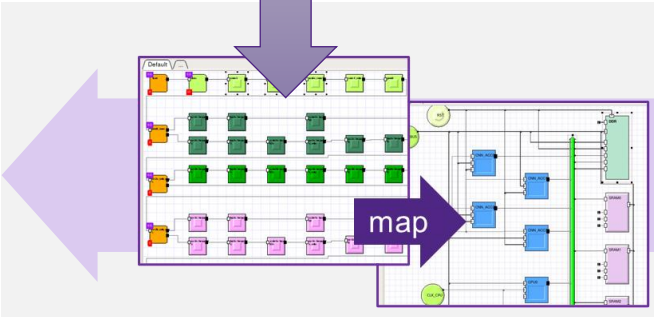
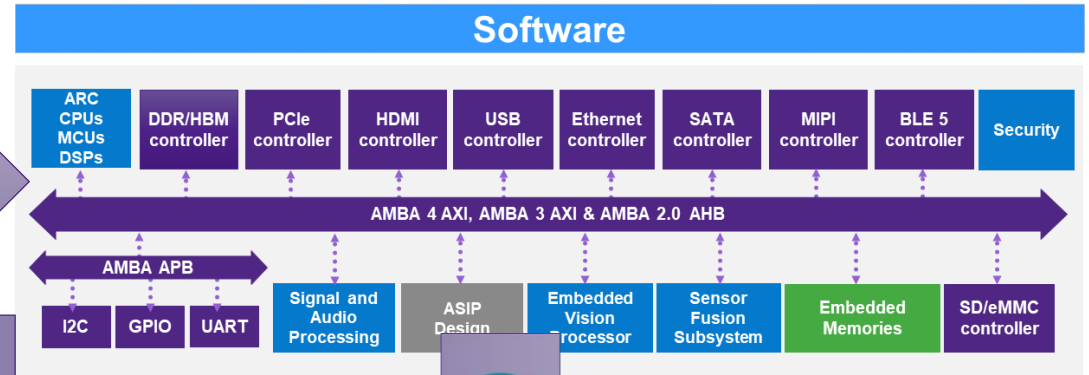
- **Choosing the right algorithm and architecture: CPU, vector DSP, ASIP, DNN accelerator**
 - DNN graphs are evolving fast, need short time to market and cannot optimize for one single graph
 - Joint design of AI algorithm, compiler and SoC architecture
 - Joint optimization of power, performance, accuracy, 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



Architecture spec



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

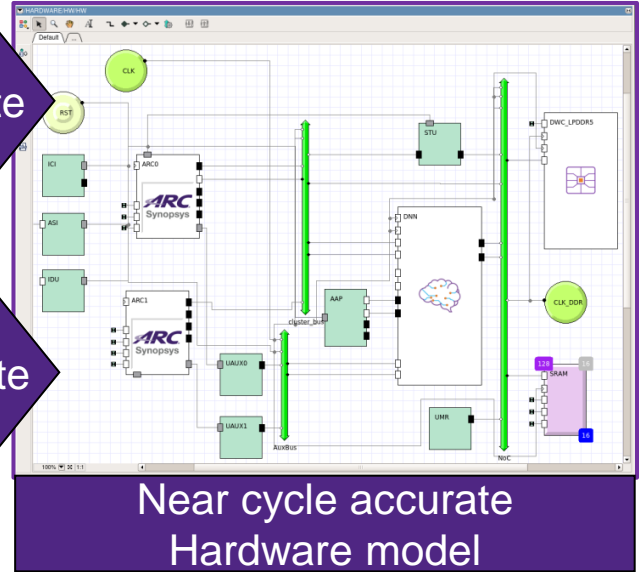
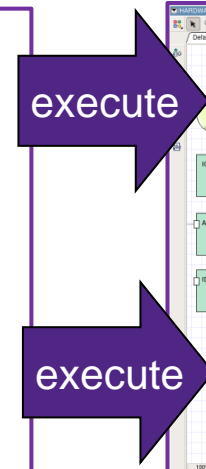
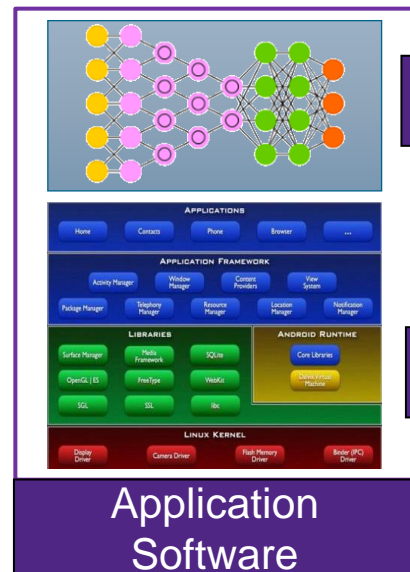
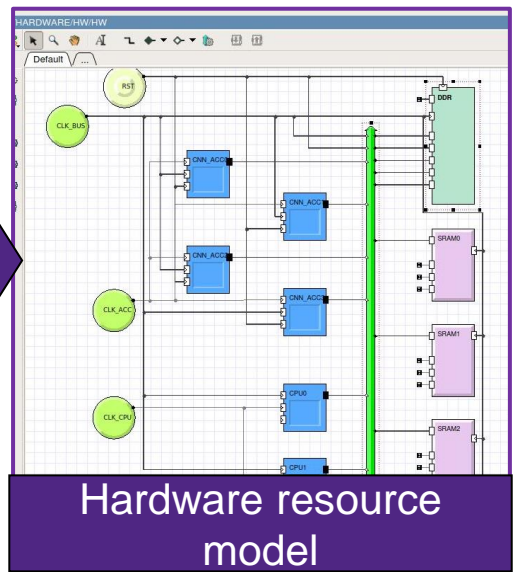
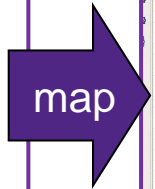
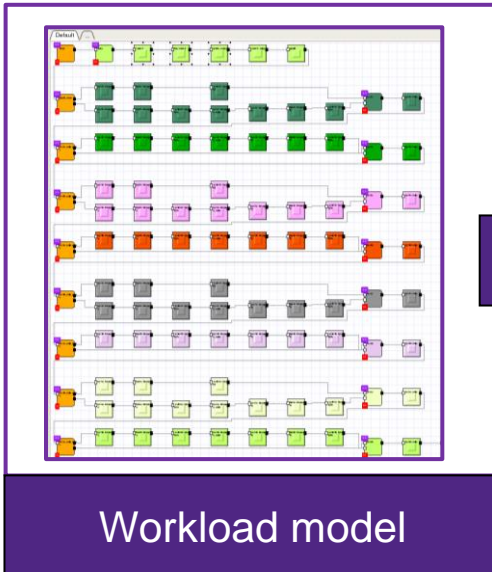
Use-cases for Architecture Analysis with Virtual Prototyping

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

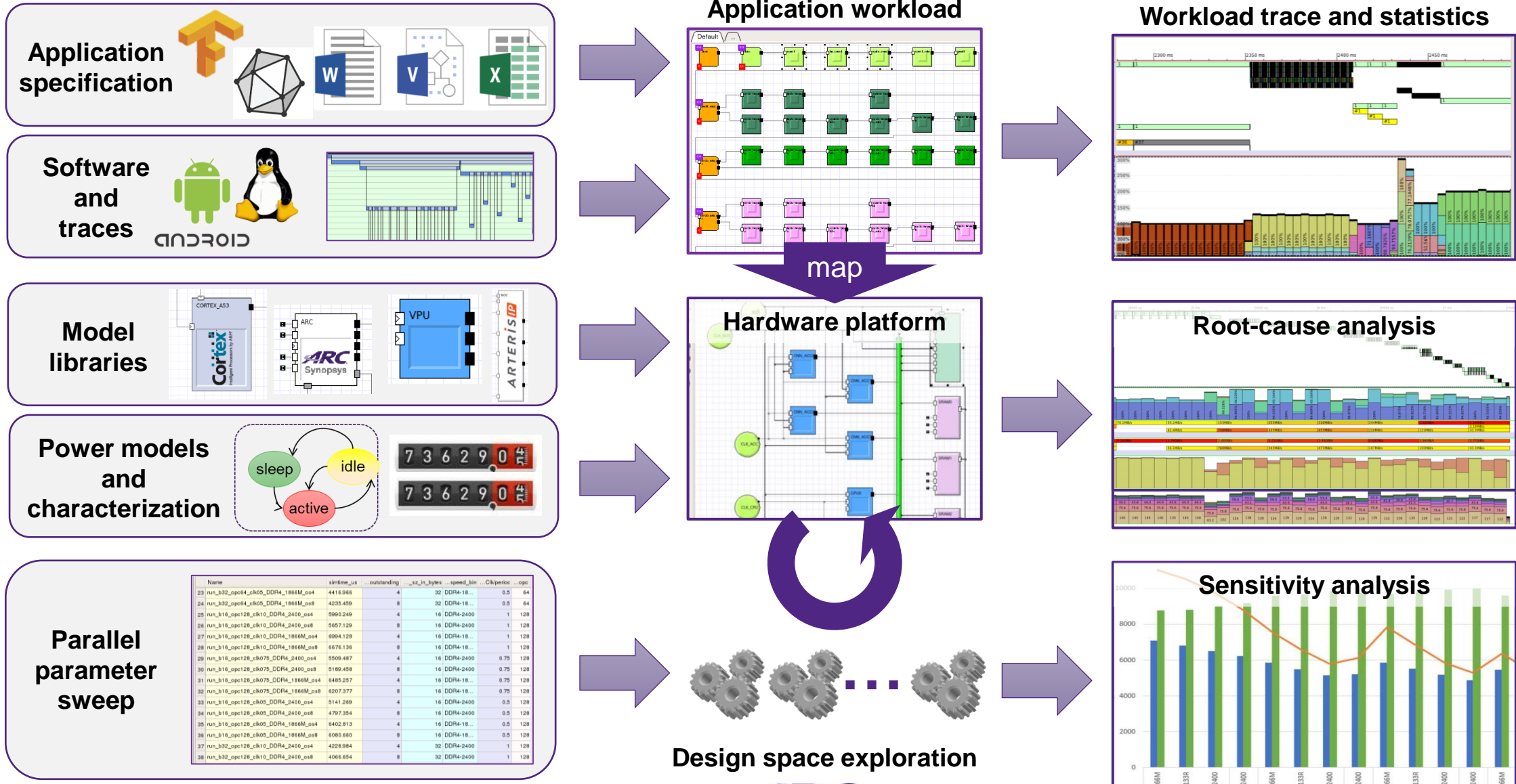
- KPI capture and sensitivity analysis
- Traffic and application workload modeling
- HW/SW partitioning, architecture specification
- power/performance analysis

Performance optimization with Software

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



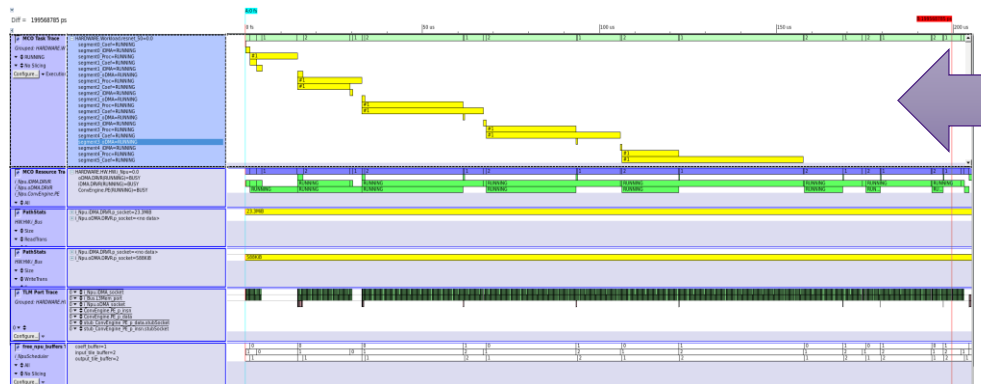
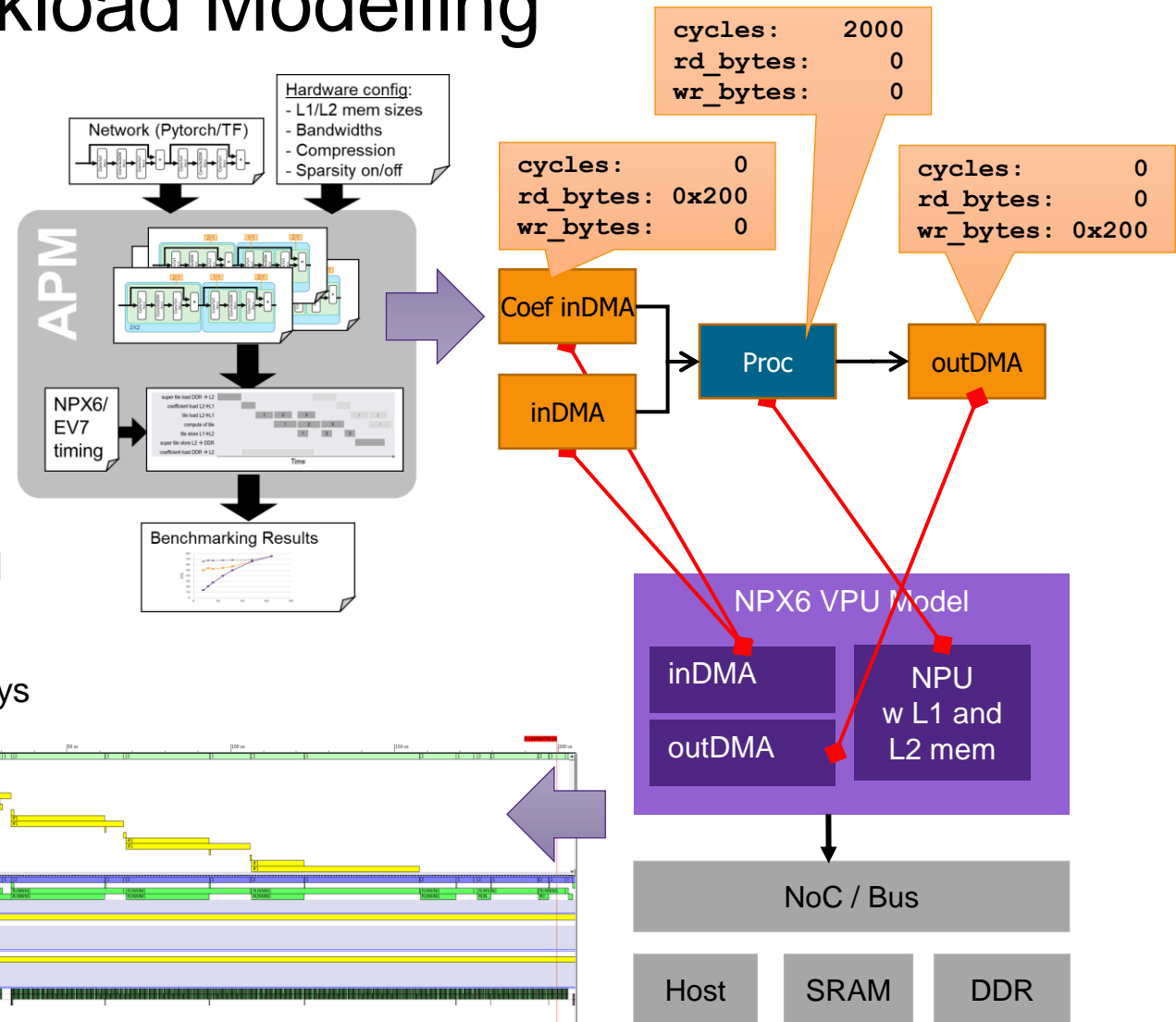
Platform Architect Power and Performance Analysis Flow



Name	simtime_us	...outstanding	...sz_in_bytes	...speed_bin	...Clk/period	...opc
23 run_b32_opc64_clk05_DDR4_1866M_0s4	4416366	4	32	DOR4-18...	0.5	84
24 run_b32_opc64_clk05_DDR4_1866M_0s8	4235459	8	16	DOR4-18...	0.5	84
25 run_b18_opc128_clk10_DDR4_2400_0s4	5990249	4	16	DOR4-2400	1	129
26 run_b18_opc128_clk10_DDR4_2400_0s8	5657129	8	16	DOR4-2400	1	129
27 run_b18_opc128_clk10_DDR4_1866M_0s4	6994128	4	16	DOR4-18...	1	129
28 run_b18_opc128_clk10_DDR4_1866M_0s8	6676136	8	16	DOR4-18...	1	129
29 run_b18_opc128_clk075_DDR4_2400_0s4	5509487	4	16	DOR4-2400	0.75	129
30 run_b18_opc128_clk075_DDR4_2400_0s8	5189458	8	16	DOR4-2400	0.75	129
31 run_b18_opc128_clk075_DDR4_1866M_0s4	6485257	4	16	DOR4-18...	0.75	129
32 run_b18_opc128_clk075_DDR4_1866M_0s8	6207377	8	16	DOR4-18...	0.75	129
33 run_b18_opc128_clk05_DDR4_2400_0s4	5141289	4	16	DOR4-2400	0.5	129
34 run_b18_opc128_clk05_DDR4_2400_0s8	4797354	8	16	DOR4-2400	0.5	129
35 run_b18_opc128_clk05_DDR4_1866M_0s4	6402813	4	16	DOR4-18...	0.5	129
36 run_b18_opc128_clk05_DDR4_1866M_0s8	6080660	8	16	DOR4-18...	0.5	129
37 run_b32_opc128_clk10_DDR4_2400_0s4	4228984	4	32	DOR4-2400	1	129
38 run_b32_opc128_clk10_DDR4_2400_0s8	4066654	8	32	DOR4-2400	1	129

Platform Architect Based Workload Modelling

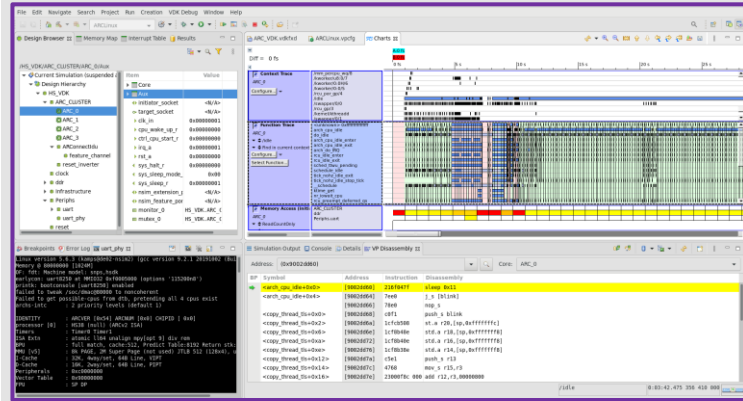
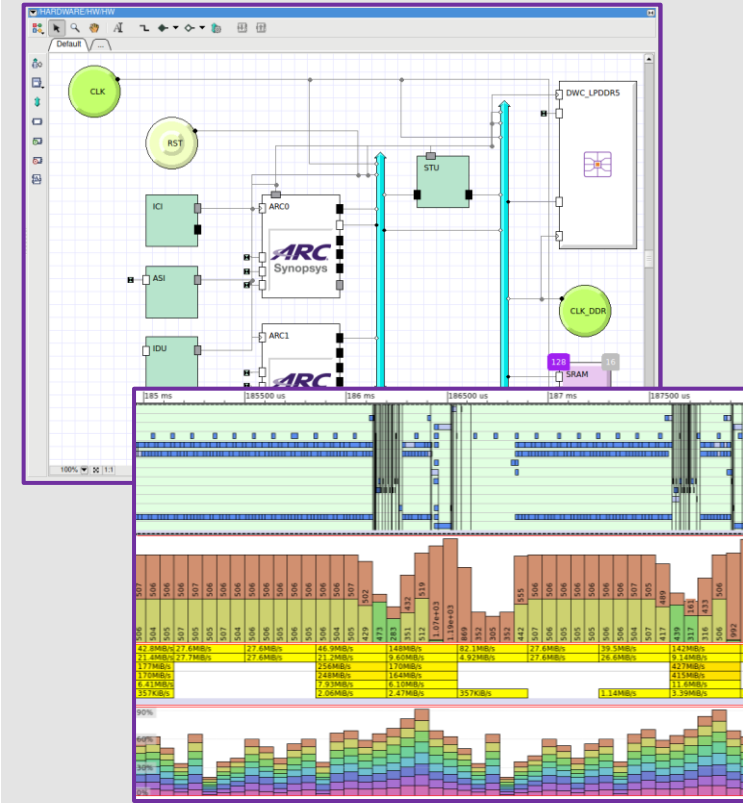
- Analytic Performance Model (APM)
 - Used internally by Synopsys NPX System Architecture Team
- Workload Model generated from APM
 - Calibrated tasks for in-DMA, out-DMA, and processing
- SoC Platform Model
 - Accurate SystemC Transaction Level Models (TLM) of processing elements, interconnect and memory
- 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

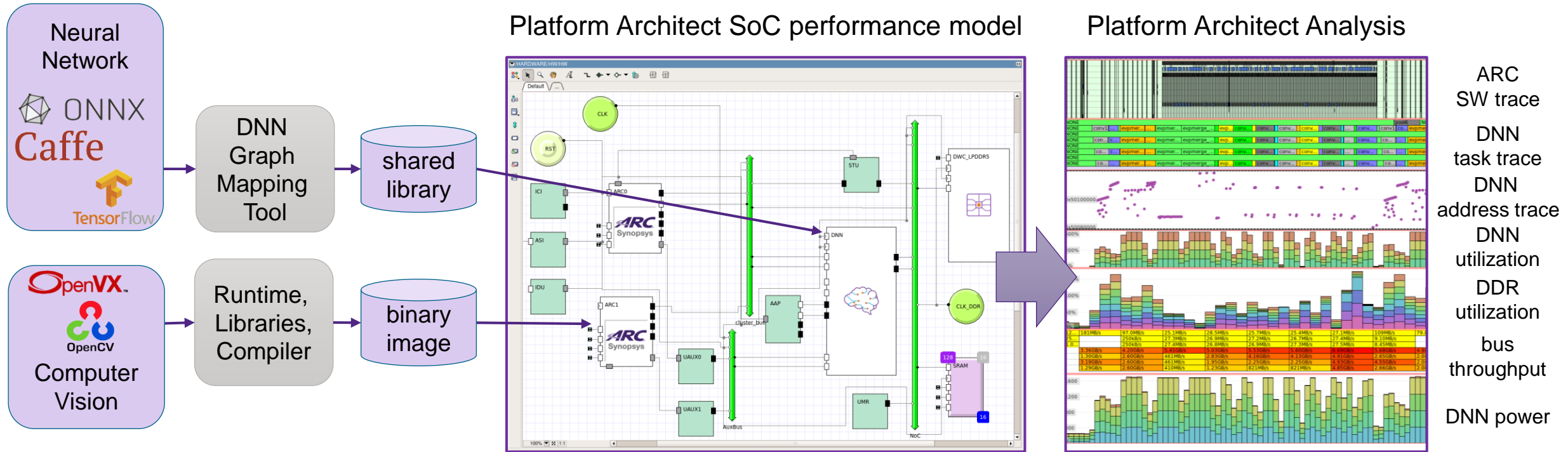
Support for building virtual prototypes

- 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%

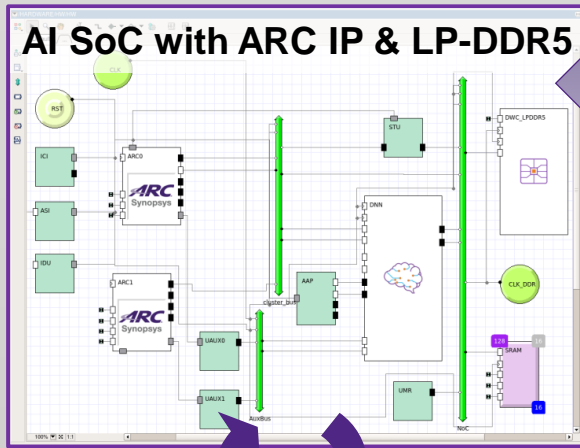
Table 1: ARC EV7x Processor FPM FPS as % of the hardware FPS. 100% means identical to hardware. >100% means an optimistic estimate.

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

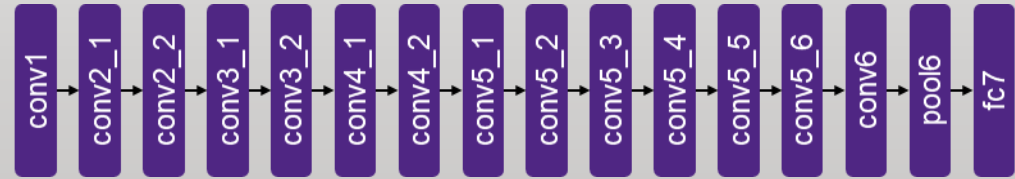
- Capture an AI SoC platform with ARC AI processor IP, a Network-on-Chip, and DDR and SRAM memory hierarchy
- Analysis and optimization of IP-level and SoC architecture configurations

AI SoC Platform Case-study with ARC AI Subsystem

Platform Architect



MobileNet



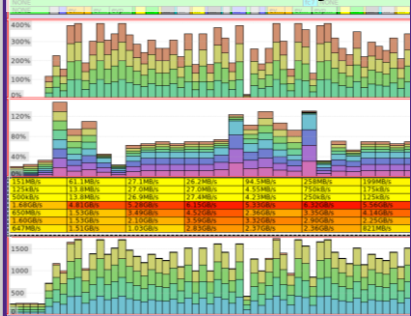
Goals:

- ① 4 ms latency for inference of 5 frames
- ② minimize DNN power and energy

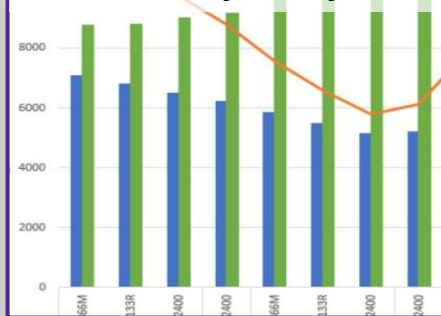
Optimize Hardware configuration:

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

Root-cause analysis



Sensitivity analysis



Platform Architect with ARC AI sub-system and DWC LPDDR5

Model Library

- DESIGNWARE
- DESIGNWARE_USB_SCML
- DESIGNWARE_SATA
- DESIGNWARE_MIPI
- DESIGNWARE_ETHERNET
- DESIGNWARE_ADV_SERIAL
- DESIGNWARE_AMBA
- DESIGNWARE_HDMI
- DESIGNWARE_MOBILEST...
- DESIGNWARE_USB
- DWC_DDR5_MCTL
- DWC_LPDDR5_MCTL
- BasicGFRBM
- DWC_LPDDR5_MCTL
- DESIGNWARE_ARC_CORES
- ARC_CORES_TLM2_L...
- SISS_MOTOR_CON...
- SISS_INTELLIGENT...
- SISS_INTELLIGENT...
- SISS_COMBO_SEN...
- SISS_COMBO_SEN...
- SISS_COMBO_SEN...
- SISS_COMBO_SEN...
- SEM120D_VOICE_...
- SEM120D_NRG
- SEM120D_NRG_FP...
- SEM110_MINI
- SEM110_DMIPS
- HS48_SLC_FULL
- HS48_FULL
- HS48_BASE
- HS47D_SLC_FULL
- HS47D_BASE
- HS47D_AGU_PERF
- HS46_PERF
- HS46_BASE
- HS45D_VOICE_AU...
- HS45D_PERF

Parameters

Name	Value
time_per_instruction_unit	
enable_byte_enable	<input type="checkbox"/>
target_socket_base_address	0
target_socket_base_sync	<input type="checkbox"/>
nsim_properties_file	/u/timkogel/de...
tools_configuration_file	/slowfs/de02d...
...s_configuration_file_core	0
halt_on_reset	<input type="checkbox"/>
exit_on_halt	<input checked="" type="checkbox"/>
use_returncode	<input checked="" type="checkbox"/>
suspend_when_halted	<input type="checkbox"/>
irq_polarity	active_high
reinitialize_iccm_on_reset	<input type="checkbox"/>
reinitialize_dccm_on_reset	<input type="checkbox"/>
...onfig/multi_core/enabled	<input checked="" type="checkbox"/>
id	0
enable_sim_thread_check	<input type="checkbox"/>
properties_file	
...gger_config/scit/enabled	<input type="checkbox"/>
debugger_configure	<input type="checkbox"/>
...nfig/gdb_server/enabled	<input type="checkbox"/>
port	1234
reconnect	<input type="checkbox"/>
xml	
dump_tdesc	
program	/slowfs/de02si...
program_arguments	-image_test 5 ...
parallel_simulation/enable	<input type="checkbox"/>
global_to_inner	2
NumberOfInterrupts	1
InitiatorBusWidth	128
TargetBusWidth	32

Connections

Block	Name	Connection	Connected_I
1	ARC0	initiator_socket	C_core0_0 cluster_bus
2	ARC0	target_socket	C_CSM_0 cluster_bus

Block diagram

The block diagram illustrates the hardware architecture. It features two ARC Synopsys processors (ARC0 and ARC1) connected to a central cluster bus. Key components include:

- ARC0** and **ARC1**: The main processing units.
- ICL**, **ASI**, **IDU**: Interconnect and control blocks.
- UAX0**, **UAX1**: User Auxiliary blocks.
- AAP**: Address Access Port.
- UUR**: User Register.
- STU**: System Trace Unit.
- DWC_LPDDR5**: On-chip memory controller.
- CLK**, **RST**: Clock and Reset sources.
- SRAM**: On-chip SRAM (128x16).
- CLK_DDR**: External clock source.
- cluster_bus**, **AuxBus**, **NoC**: Various interconnects and network-on-chip.

Video 1: Platform creation and tracing

- Example Platform creation
- Software tracing
- Hardware tracing

The image displays a comprehensive view of platform creation and tracing. It includes a software execution trace on the left, a hardware block diagram in the center, and hardware tracing data on the right. The software trace shows a sequence of functions such as `CnnBenchmark::init`, `CnnGraphRunner::getProcess`, and `CnnProcess::run` with associated timing markers. The hardware diagram illustrates the internal components of the ARC Synopsys processor, including the ARC0 and ARC1 cores, local interconnects (LIAUX0, LIAUX1), and various peripheral blocks like the DNN accelerator, STU, LMBR, and memory controllers. The hardware tracing on the right provides a detailed look at the DNN task execution, showing activity traces for different slices and a data visualization of the neural network's internal state.

Blocks

VPU_20:VPU

Name
- HW
2 NoC
3 UMR
4 ARC0
5 ARC1
6 ICI
7 IDU
8 ASI
9 AAP
10 STU

Memory Clock Reset Control

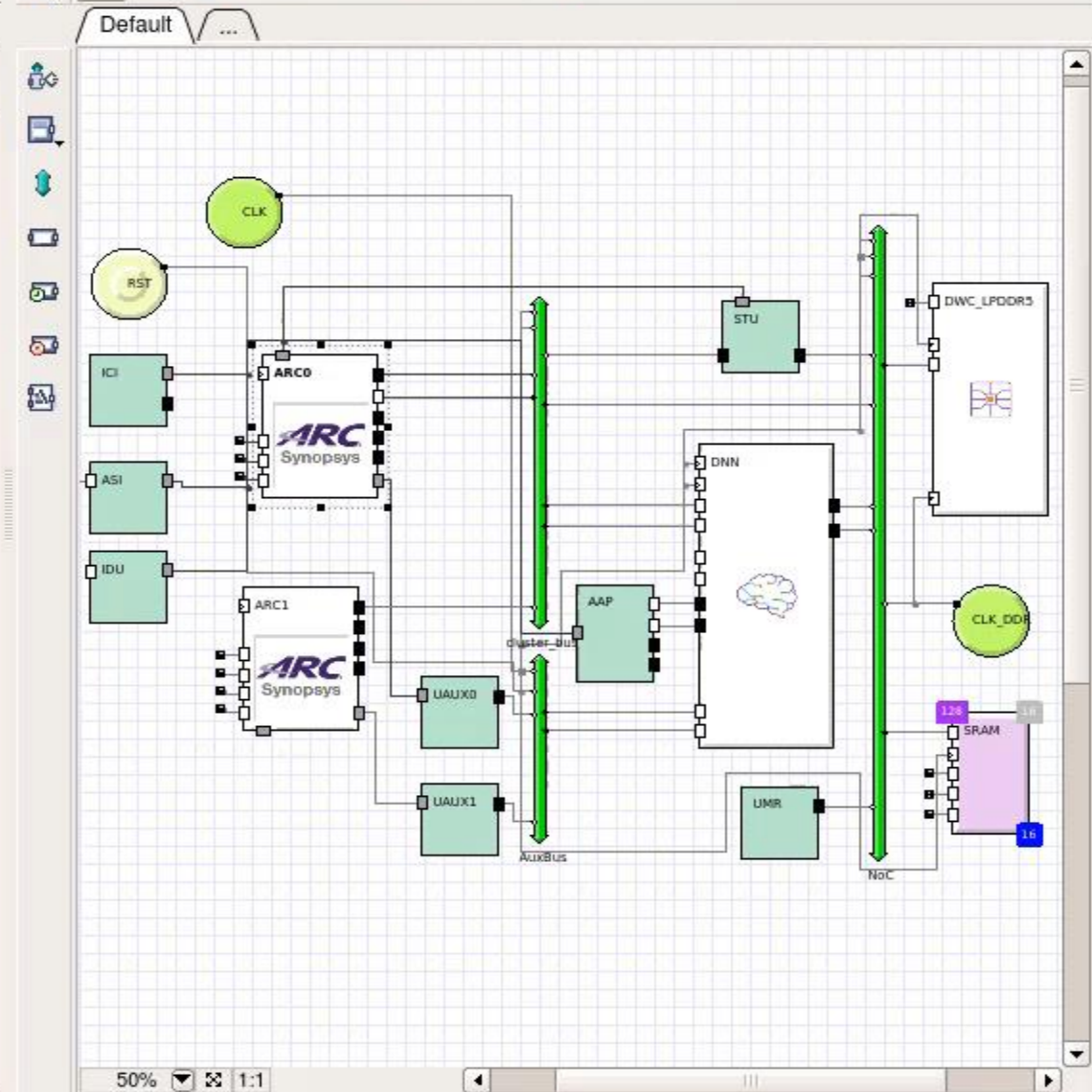
Filter

Block	Name	Connection	Connected	
1	ARC0	initiator_socket	C_core0_0	cluster_bus
2	ARC0	target_socket	C_CSM_0	cluster_bus

Parameters

All

ARC0	
image_loader_sc_ipt	
..._config/dmi/enable	<input type="checkbox"/>
enable_latencies	<input type="checkbox"/>
clock_period	1
clock_period_unit	ns
time_per_instruction	0
...er_instruction_unit	
enable_byte_enable	<input type="checkbox"/>
...ket_base_address	0
...socket_base_sync	<input type="checkbox"/>
nsim_properties_file	/u/timkogel/de...
...s_configuration_file	/slowfs/de02d...
...figuration_file_core	0
halt_on_reset	<input type="checkbox"/>
exit_on_halt	<input checked="" type="checkbox"/>
use_returncode	<input checked="" type="checkbox"/>
...pend_when_halted	<input type="checkbox"/>
irq_polarity	active_high
...lize_iccm_on_rese	<input type="checkbox"/>
...ze_dccm_on_rese	<input type="checkbox"/>
...multi_core/enabled	<input checked="" type="checkbox"/>
id	0
...sim_thread_check	<input type="checkbox"/>
properties_file	
...config/exit/enabled	<input type="checkbox"/>

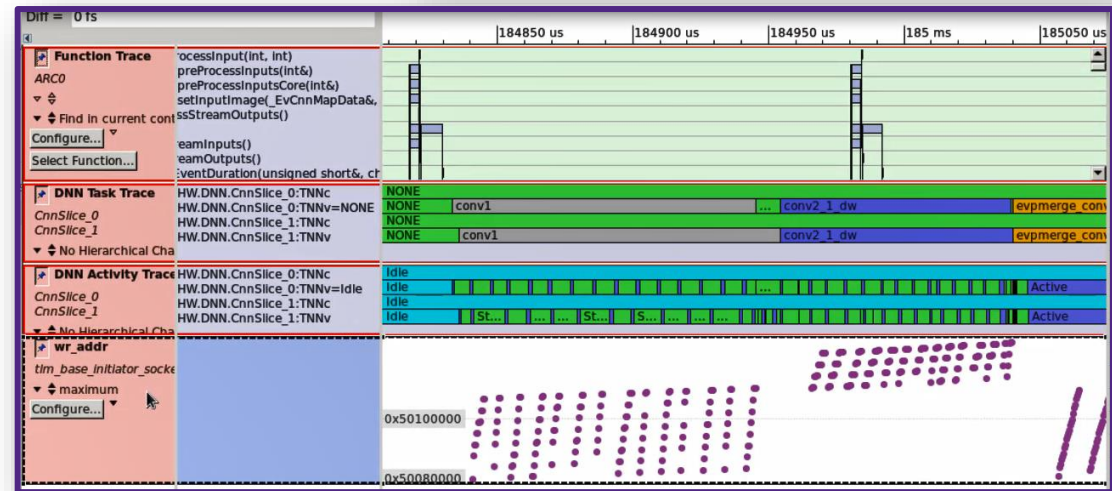
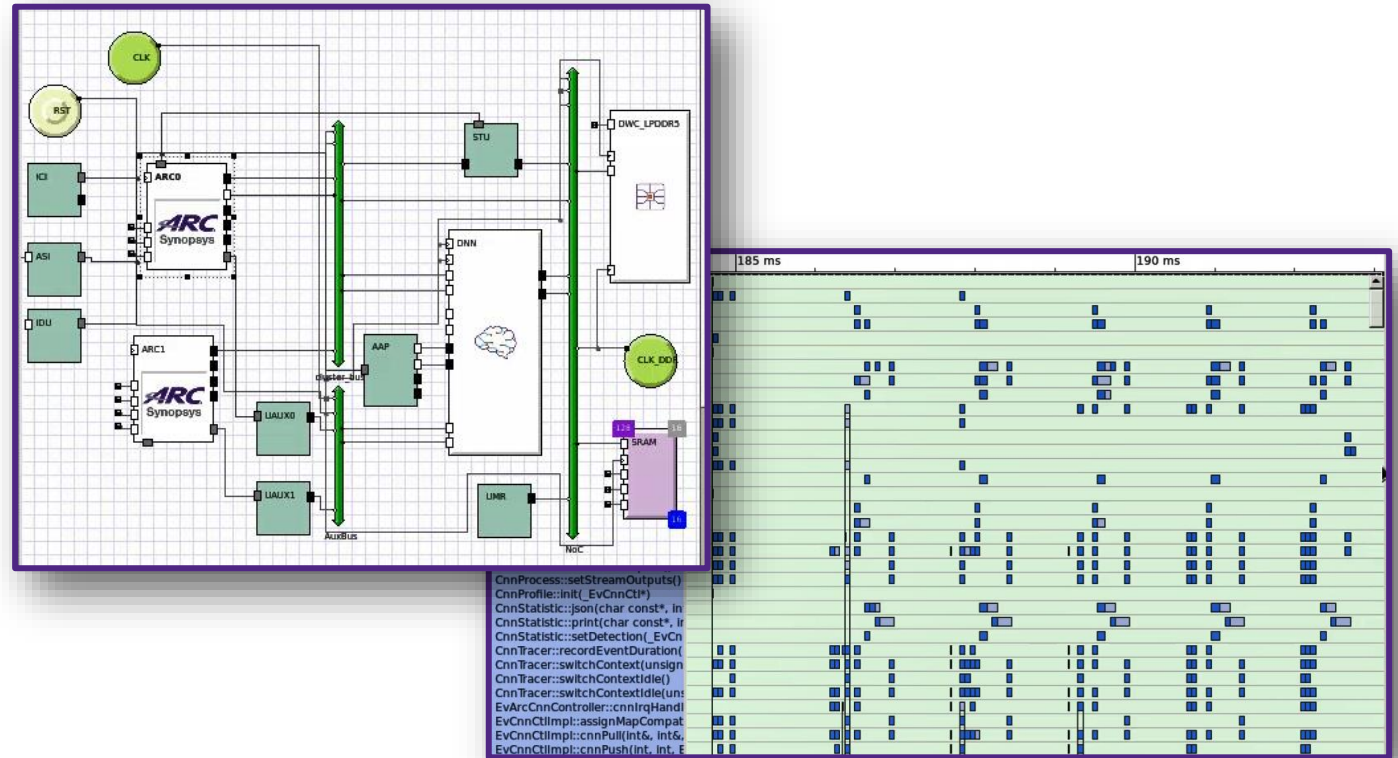


What We Just Learned

Platform creation and tracing

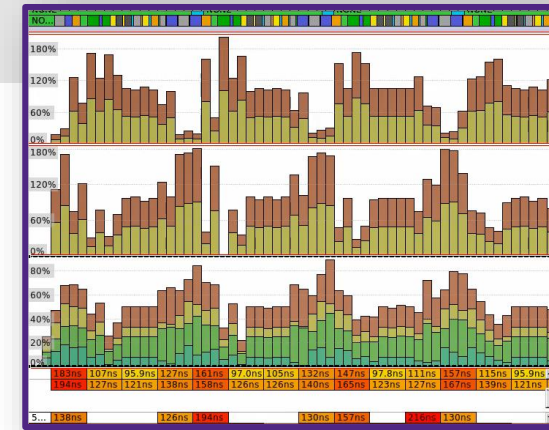
We learned how to:

- ✓ 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



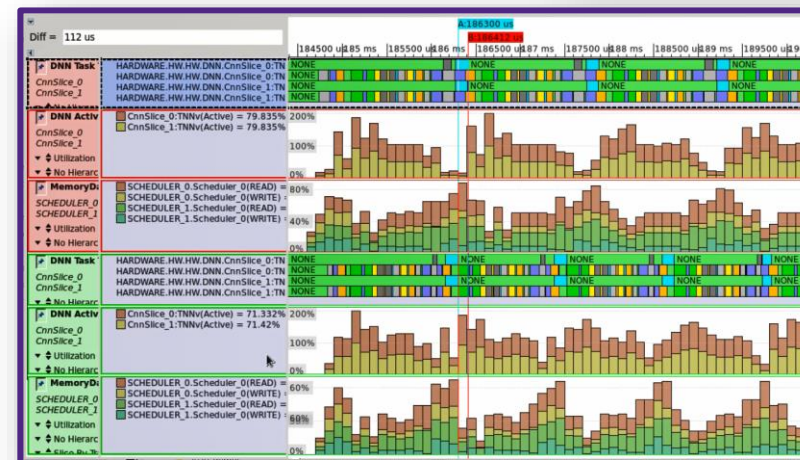
Video 2: Performance Analysis

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



Name	Enable	Status	simtime_us	Override_Setting
1 run	<input checked="" type="checkbox"/>	SUCCESS	192912.523	<input type="checkbox"/>
2 run_1	<input checked="" type="checkbox"/>	NOT_RUN	0	<input type="checkbox"/>

Domain	Instance	Param	Value
1 HW	HW/NoC	outstanding	8
2 HW	HW/DWC_LP...	speed_bin	LPDDR5-3733



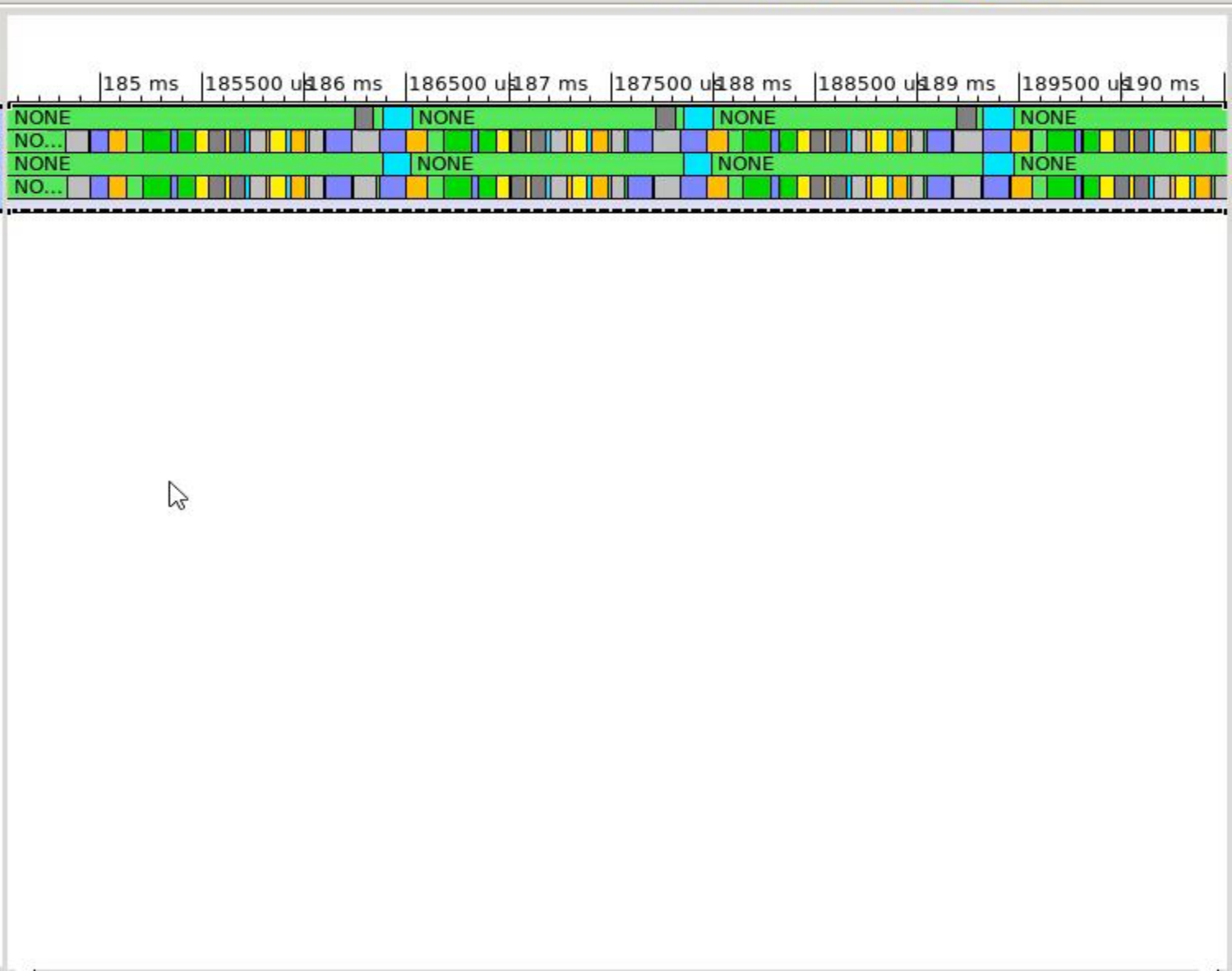
type filter text

Design Object/Analysis N...

- analyze
- latest
 - <<BLANK DESIGN C...
 - HARDWARE
 - HW
 - HW
 - ARCO
 - AuxBus
 - DNN
 - CnnSlic
 - CnnSlic
 - tlm_bas
 - tlm_bas
 - tlm_bas
 - tlm_bas
 - DWC_LPDD
 - NoC
 - cluster_bu
 - BusAnalys
 - UPFDummy

Diff = 0 fs

DNN Task	HARDWARE.HW.HW.DNN.CnnSlice_0:TN
CnnSlice_0	NO...
CnnSlice_1	HARDWARE.HW.HW.DNN.CnnSlice_1:TN
	NO...

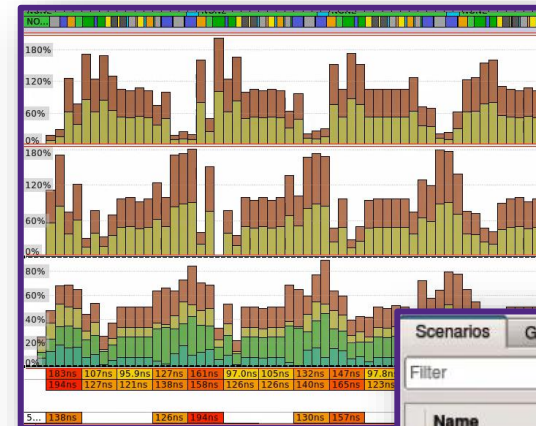


What We Just Learned

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



Name	Enable	Status	simtime_us	Override_Setting
1 run	<input checked="" type="checkbox"/>	SUCCESS	192912.523	<input type="checkbox"/>
2 run_1	<input checked="" type="checkbox"/>	NOT_RUN	0	<input type="checkbox"/>

Domain	Instance	Param	Value
1 HW	HW/NoC	outstanding	8
2 HW	HW/DWC_LP...	speed_bin	LPDDR5-3733

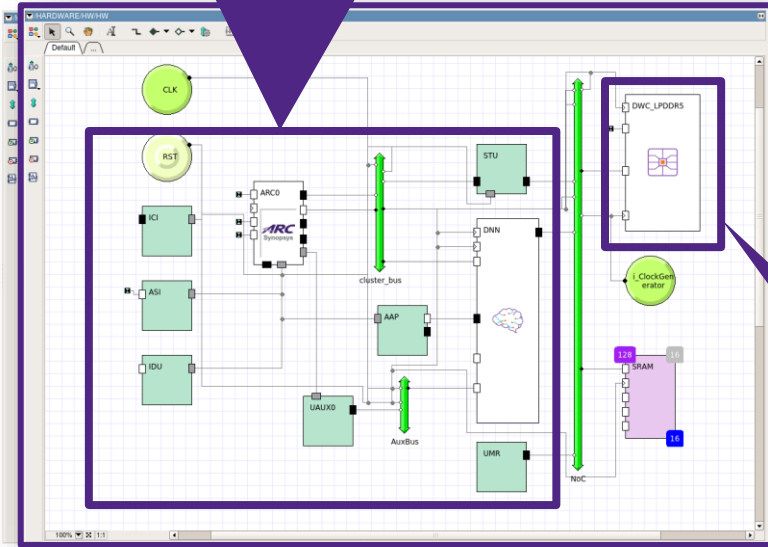


AI SoC Block Diagram in Platform Architect

Scaling AI Sub-system and LPDDR5 memory controller

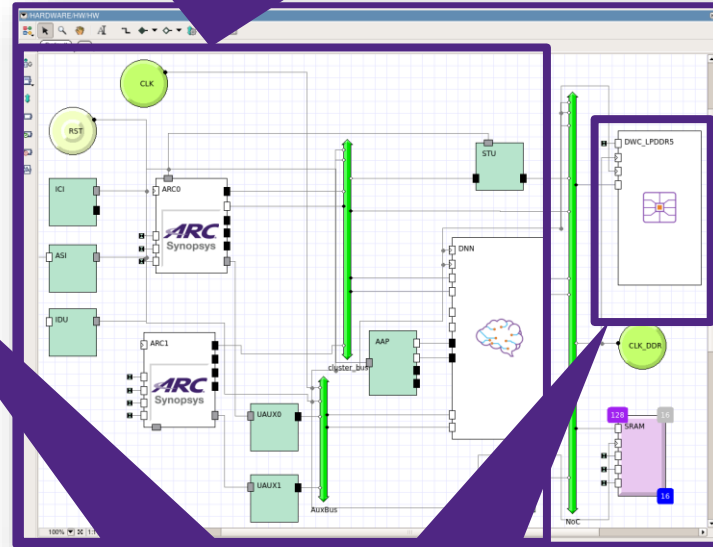
Single-core sub-system

- 1 ARC VPX cores
- 1 DNN slice



Dual-core sub-system

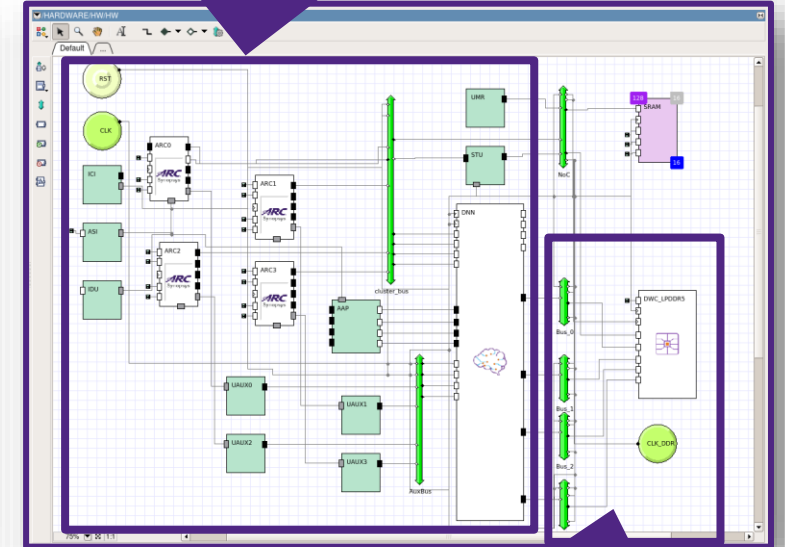
- 2 ARC VPX cores
- 2 DNN slices



DesignWare LPDDR5
Memory Controller

Quad-core sub-system

- 4 ARC VPX cores
- 4 DNN slices



- Multi-port LPDDR5 Mctrl
- parallel AXI bus fabric

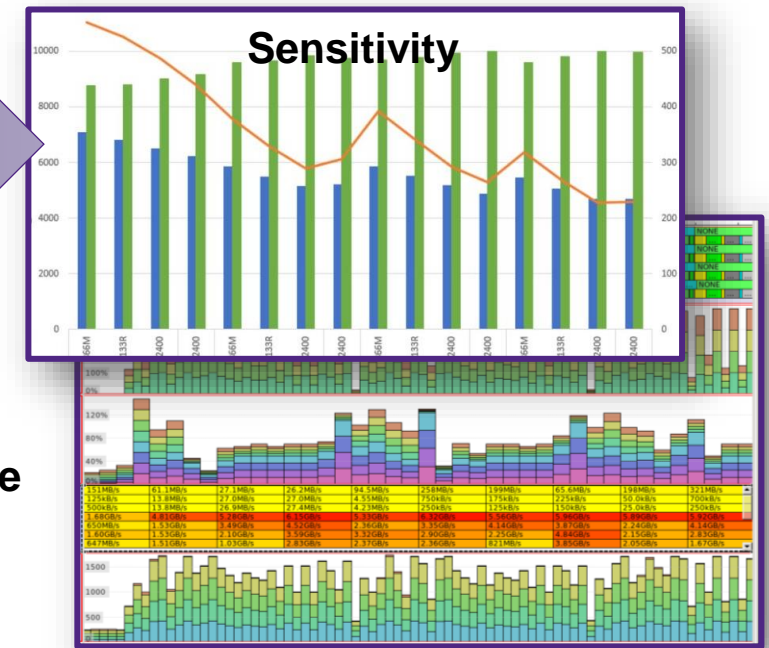
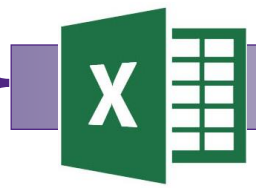
AI SoC Architecture Sweep

Goal: 4 ms inference latency, minimize power & energy

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

Name	.../outstanding	.../speed_bfr	...LPDDR5/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



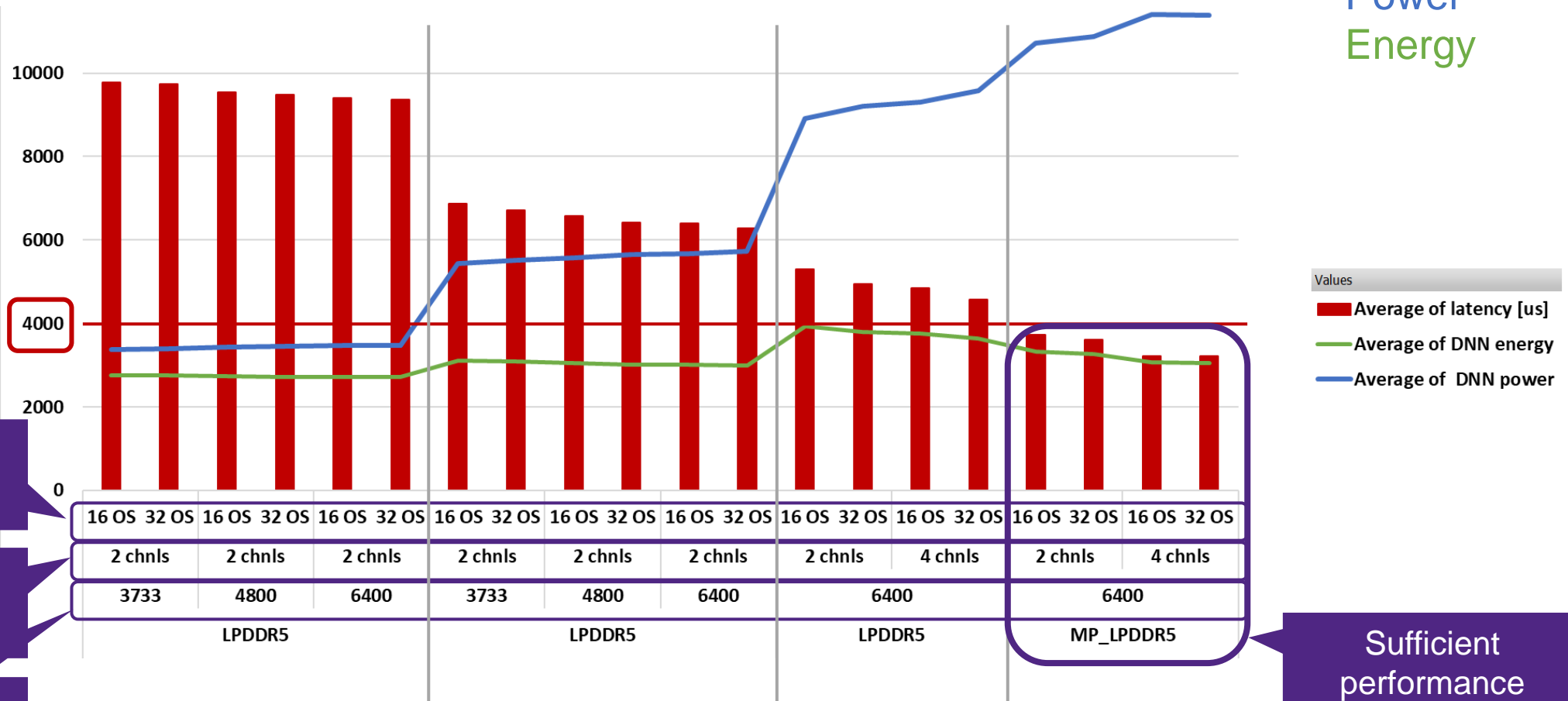
Root-Cause Analysis

Analysis and Optimization of Architecture Configurations

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

Latency [us]

Power
Energy



Outstanding transactions

LPDDR channels

LPDDR speed

1 DNN slice

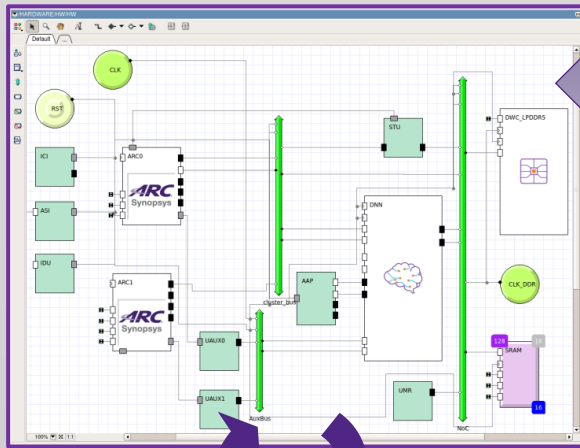
2 DNN slices

4 DNN slices

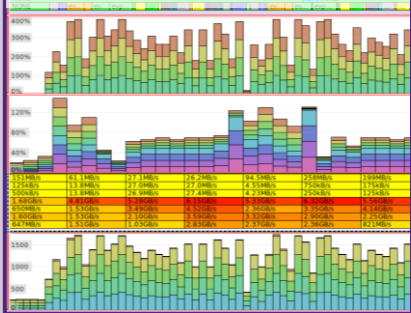
Sufficient performance

Example Summary

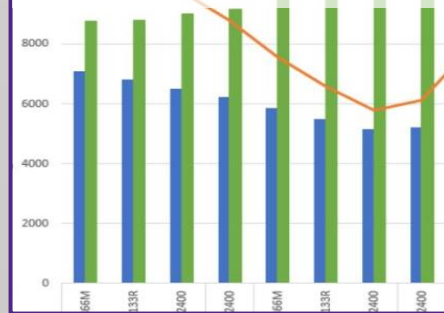
Platform Architect



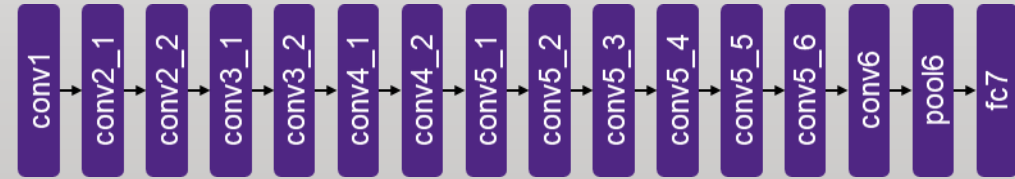
Root-cause analysis



Sensitivity analysis



MobileNet



Goals:

- ① 4 ms latency for inference of 5 frames
- ② minimize DNN power and energy

Optimized Hardware configuration:

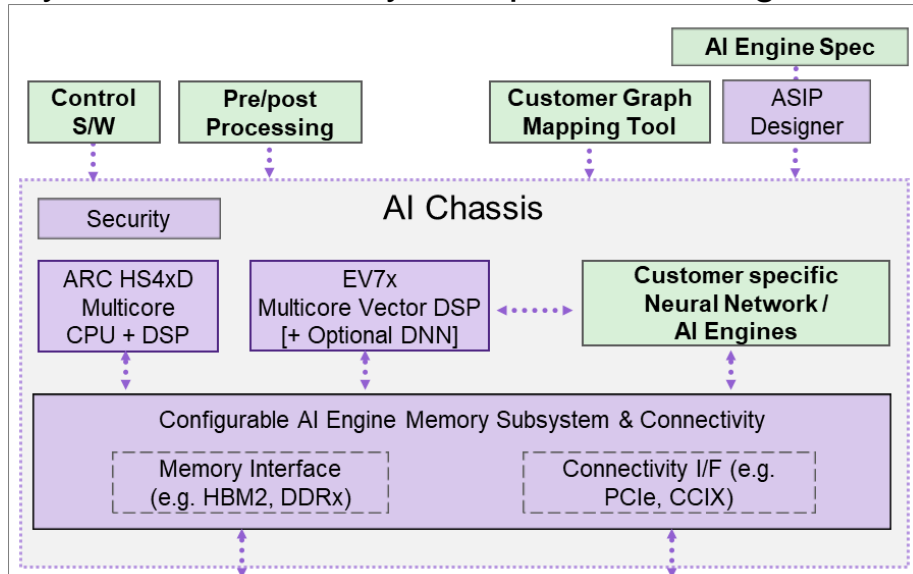
- AI configuration: 1, 2, **4** DNN slices
- Outstanding transactions: 16, **32**, 64
- LPDDR memory speed: 3733, 4800, **6400**
- Interconnect/LPDDR controller: single port, **multi-port**
- LPDDR controller scheduler queue: **32** 64

How To Get Started?

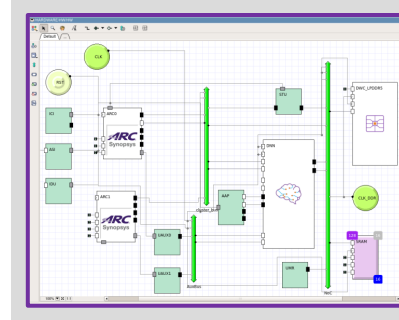
Faster Development of AI SoCs with Synopsys IP, tools, and services

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



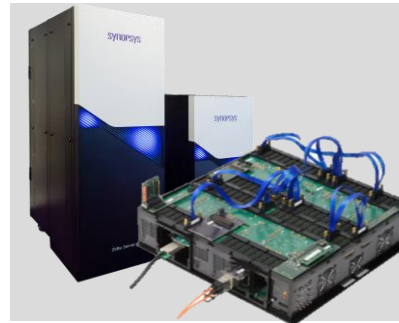
Architecture Exploration & Optimization



Platform Architect

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

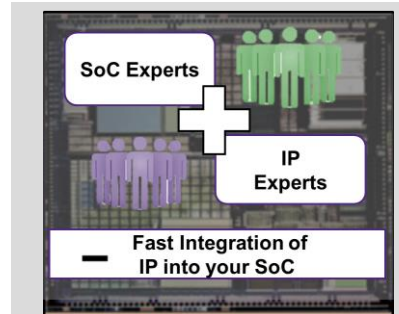
Verification, Emulation & Prototyping



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

Thank You

