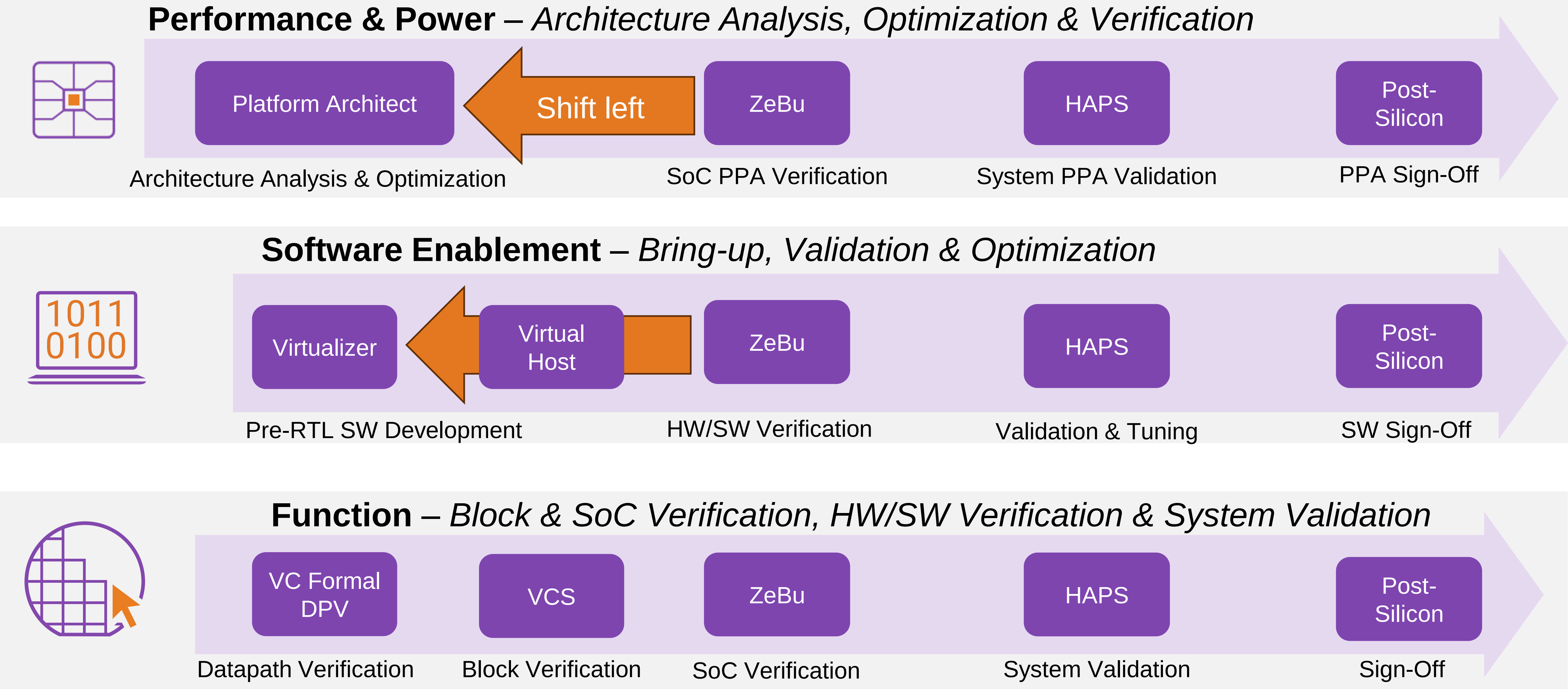
Application-level Hybrid Emulation for Software-Defined-Systems

Tim Kogel, Sr. Director, Technical Product Management
Malte Dörper, Principal, Product Management
Leonard Drucker, Scientist, Engineering
Synopsys, Inc.

SoC Design and Verification Flow



Performance & Power – *Architecture Analysis, Optimization & Verification*

Platform Architect

Shift left

ZeBu

HAPS

Post-Silicon

Architecture Analysis & Optimization

SoC PPA Verification

System PPA Validation

PPA Sign-Off

Software Enablement – *Bring-up, Validation & Optimization*

Virtualizer

Virtual Host

ZeBu

HAPS

Post-Silicon

Pre-RTL SW Development

HW/SW Verification

Validation & Tuning

SW Sign-Off

Function – *Block & SoC Verification, HW/SW Verification & System Validation*

VC Formal DPV

VCS

ZeBu

HAPS

Post-Silicon

Datapath Verification

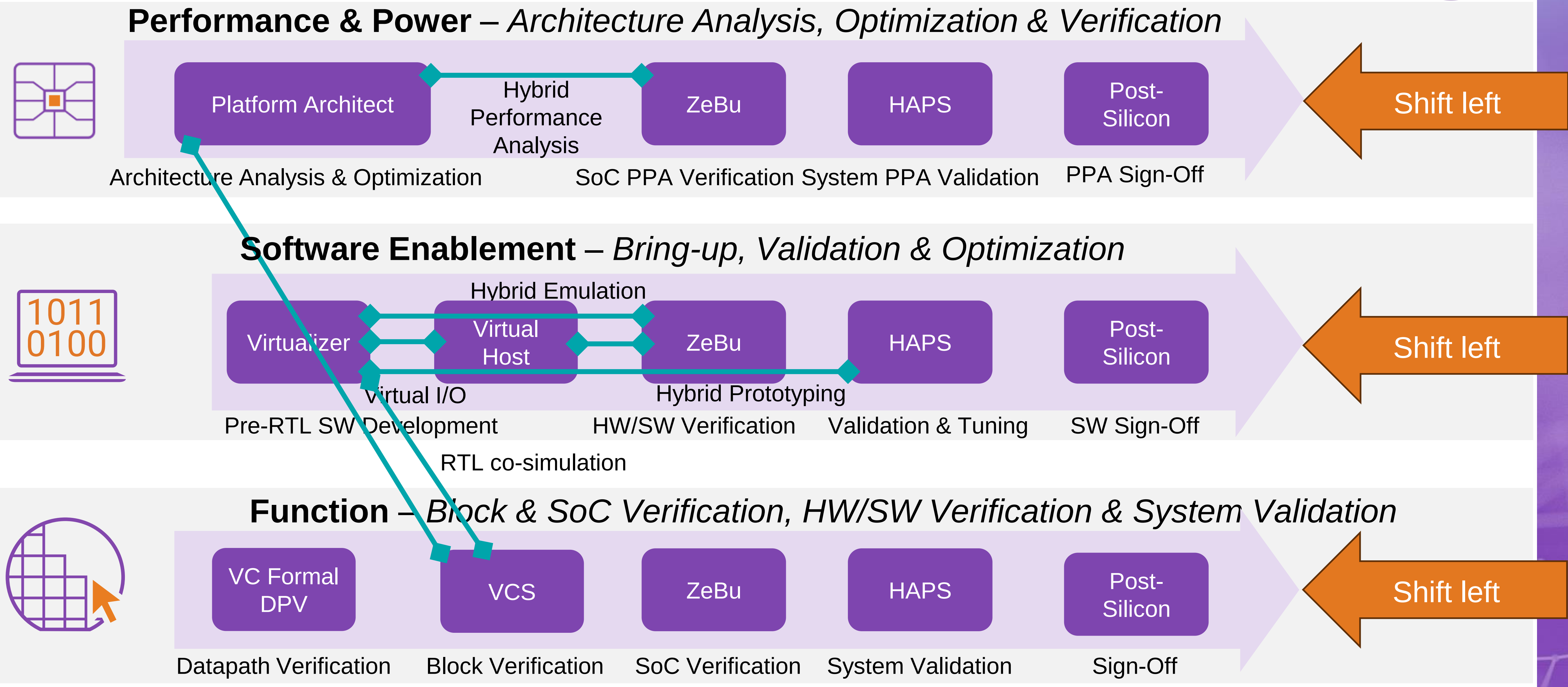
Block Verification

SoC Verification

System Validation

Sign-Off

Shift-left with Synopsys Hybrid Solutions



Performance & Power – *Architecture Analysis, Optimization & Verification*

Platform Architect — Hybrid Performance Analysis — ZeBu — HAPS — Post-Silicon

Architecture Analysis & Optimization | SoC PPA Verification | System PPA Validation | PPA Sign-Off

Shift left

Software Enablement – *Bring-up, Validation & Optimization*

Virtualizer — Hybrid Emulation — Virtual Host — ZeBu — HAPS — Post-Silicon

Virtual I/O | Hybrid Prototyping

Pre-RTL SW Development | HW/SW Verification | Validation & Tuning | SW Sign-Off

Shift left

RTL co-simulation

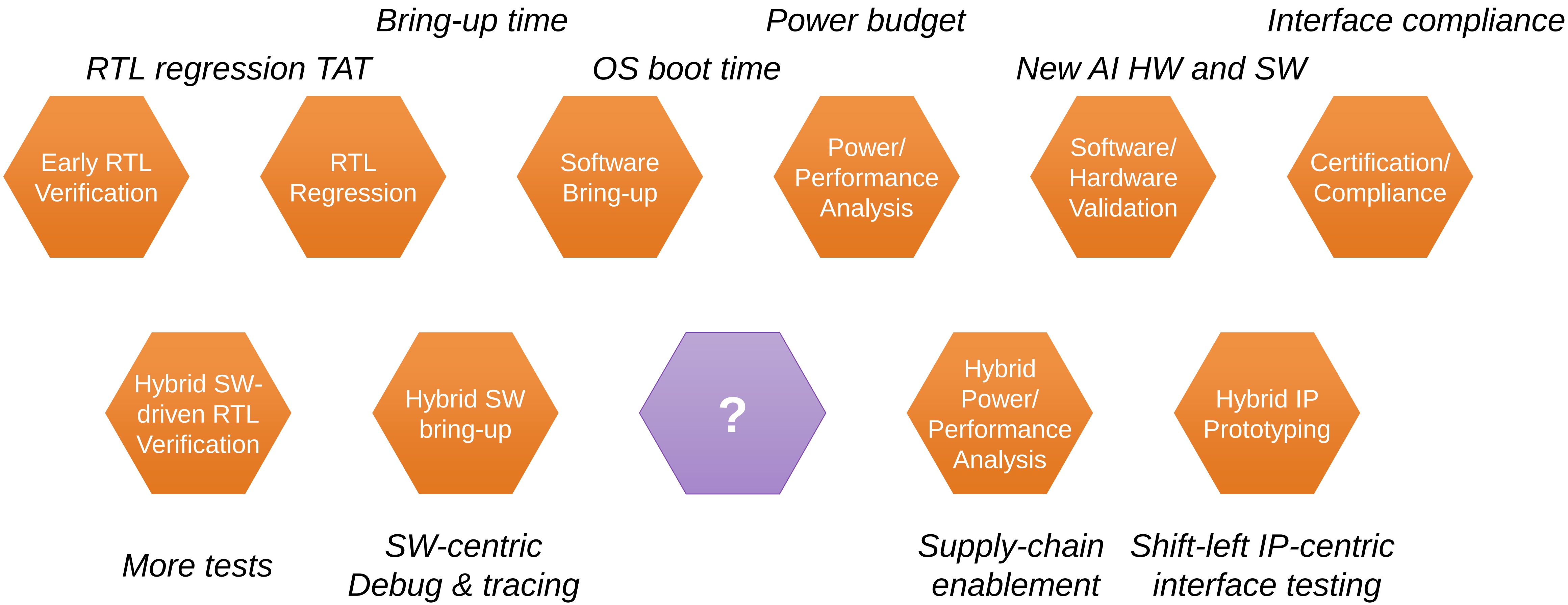
Function – *Block & SoC Verification, HW/SW Verification & System Validation*

VC Formal DPV — VCS — ZeBu — HAPS — Post-Silicon

Datapath Verification | Block Verification | SoC Verification | System Validation | Sign-Off

Shift left

Synopsys HW-Assisted Verification Use Cases



Bring-up time *Power budget* *Interface compliance*

RTL regression TAT *OS boot time* *New AI HW and SW*

Early RTL Verification

RTL Regression

Software Bring-up

Power/ Performance Analysis

Software/ Hardware Validation

Certification/ Compliance

Hybrid SW-driven RTL Verification

Hybrid SW bring-up

?

Hybrid Power/ Performance Analysis

Hybrid IP Prototyping

More tests

SW-centric Debug & tracing

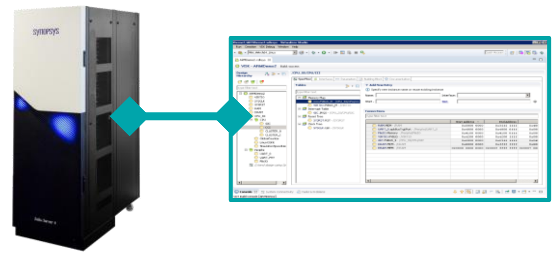
Supply-chain enablement

Shift-left IP-centric interface testing

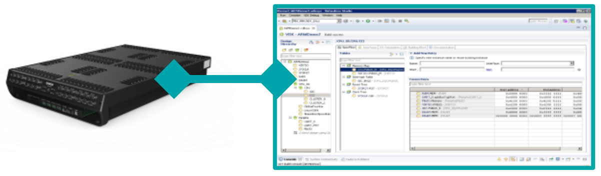
Hybrid Emulation and Prototyping Use-Cases

- Early Driver/Firmware/Application development
  - Complete SoC model using VDK and synthesizable RTL IPs
  - Power and performance validation over billions of application cycles
- Software driven System Validation with Real-world IO
  - Early Driver/Firmware/Application development
  - Modular and scalable validation from IP to system level
- Architecture and Performance Analysis
  - Efficient architecture analysis and smart performance monitoring
  - Offline and online analysis of performance data collected on ZeBu
  - Faster PPA optimization, helping shift left verification cycle

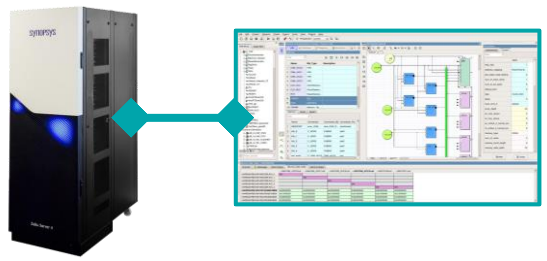
ZeBu – Virtualizer Hybrid



HAPS – Virtualizer Hybrid



ZeBu - Platform Architect Hybrid



Virtualizer Overview

VDK – Virtualizer Development Kit

- Pre-RTL virtual model of hardware board
- ARMv8 & v9 Starting point VDKs available
- Fast and efficient debug and test

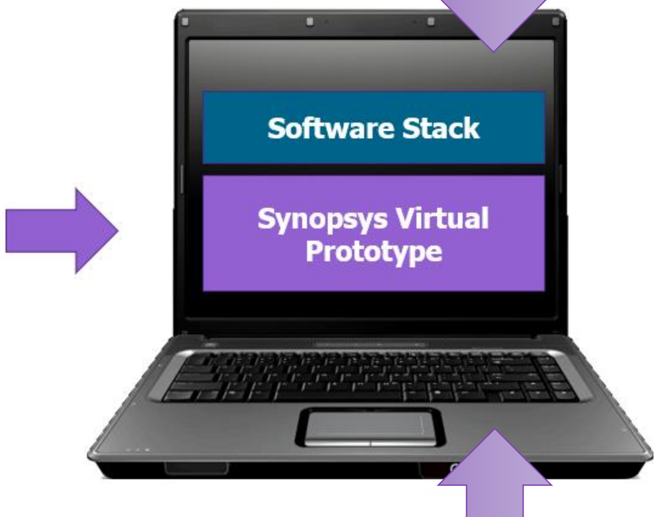
Virtualizer Studio

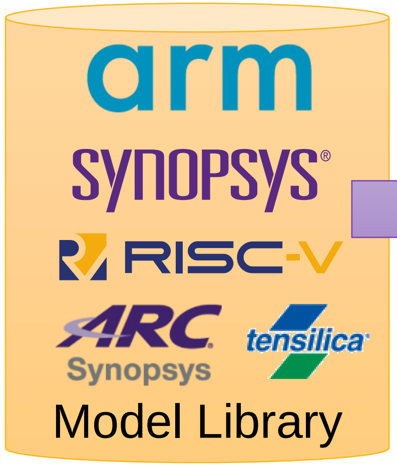
- Efficient model and VDK creation
- Largest model library

Embedded SW Debuggers

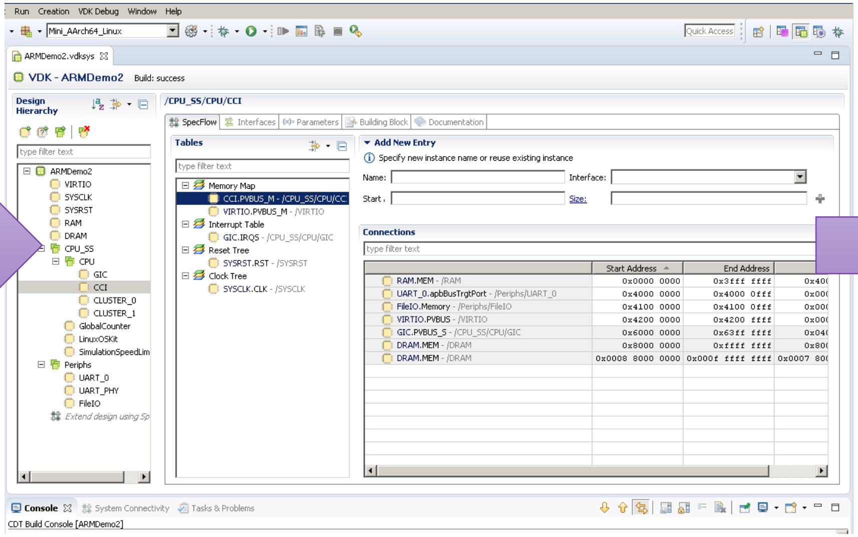
LAUTERBACH DEVELOPMENT TOOLS · Green Hills SOFTWARE · IAR SYSTEMS · ARM DS Development Tools · ARC Synopsys · CEVA · tensilica

Development board



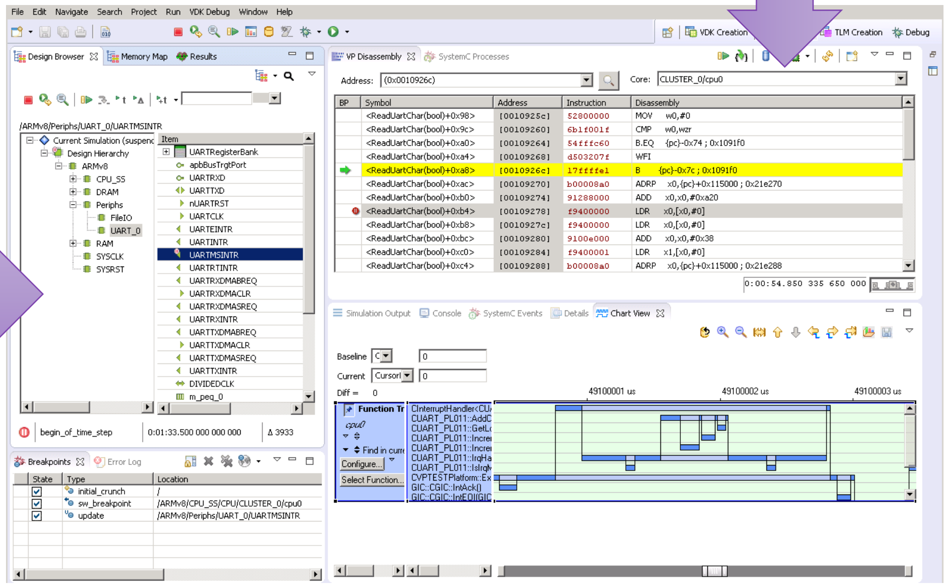


Model Library



Connections

	Start Address	End Address	
RAM.MEM - /RAM	0x0000 0000	0x3fff ffff	0x400
UART_0.apbBusTrgtPort - /Periphs/UART_0	0x4000 0000	0x4000 0fff	0x000
FileIO.Memory - /Periphs/FileIO	0x4100 0000	0x4100 0fff	0x000
VIRTIO.PVBUS - /VIRTIO	0x4200 0000	0x4200 ffff	0x000
GIC.PVBUS_S - /CPU_SS/CPU/GIC	0x6000 0000	0x63ff ffff	0x040
DRAM.MEM - /DRAM	0x8000 0000	0xffff ffff	0x800
DRAM.MEM - /DRAM	0x0008 8000 0000	0x000f ffff ffff	0x0007 800



Address: (0x0010926c) Core: CLUSTER_0/cpu0

BP	Symbol	Address	Instruction	Disassembly
	<ReadUartChar(bool)+0x98>	[0010925c]	52800000	MOV w0,#0
	<ReadUartChar(bool)+0x9c>	[00109260]	6b1f001f	CMP w0,wzr
	<ReadUartChar(bool)+0xa0>	[00109264]	54fffc60	B.EQ {pc}-0x74 ; 0x1091f0
	<ReadUartChar(bool)+0xa4>	[00109268]	d503207f	WFI
➡	<ReadUartChar(bool)+0xa8>	[0010926c]	17ffffe1	B {pc}-0x7c ; 0x1091f0
	<ReadUartChar(bool)+0xac>	[00109270]	b00008a0	ADRP x0,{pc}+0x115000 ; 0x21e270
	<ReadUartChar(bool)+0xb0>	[00109274]	91288000	ADD x0,x0,#0xa20
●	<ReadUartChar(bool)+0xb4>	[00109278]	f9400000	LDR x0,[x0,#0]
	<ReadUartChar(bool)+0xb8>	[0010927c]	f9400000	LDR x0,[x0,#0]
	<ReadUartChar(bool)+0xbc>	[00109280]	9100e000	ADD x0,x0,#0x38
	<ReadUartChar(bool)+0xc0>	[00109284]	f9400001	LDR x1,[x0,#0]
	<ReadUartChar(bool)+0xc4>	[00109288]	b00008a0	ADRP x0,{pc}+0x115000 ; 0x21e288

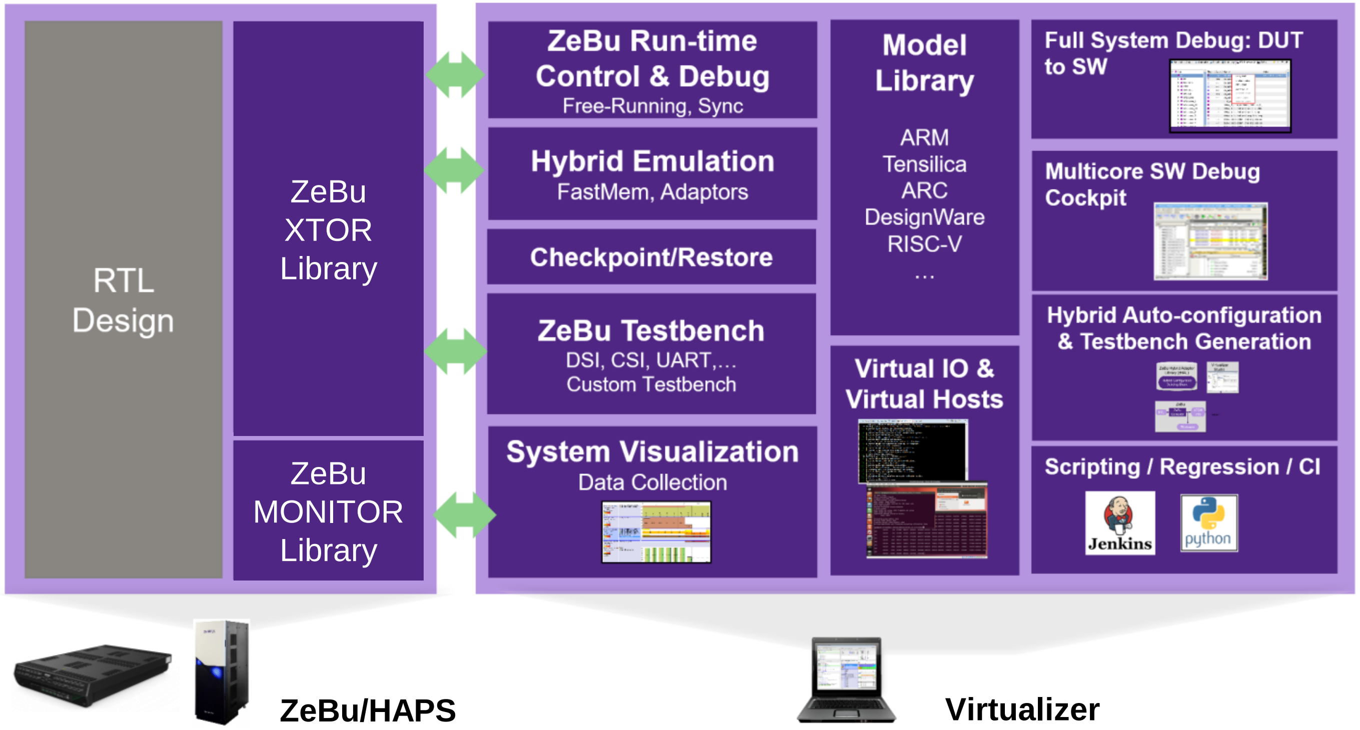
0:00:54.850 335 650 000

State	Type	Location
☑	initial_crunch	/
☑	sw_breakpoint	/ARMv8/CPU_SS/CPU/CLUSTER_0/cpu0
☑	update	/ARMv8/Periphs/UART_0/UARTMSINTR

Synopsys Hybrid Technologies

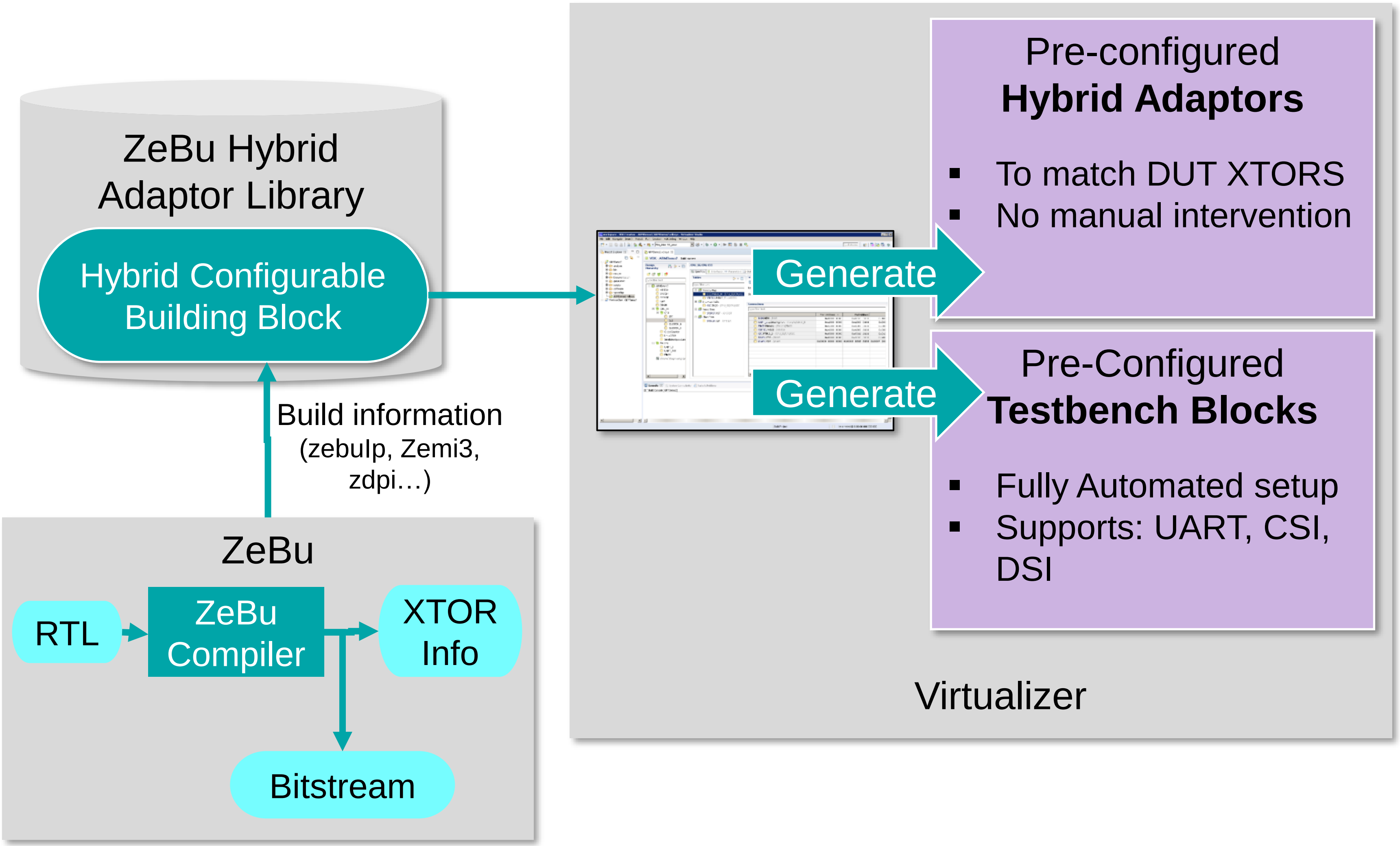
integrated with Virtual, Emulation & Prototyping tools

- **Complete tool stack**, integration between ZeBu, HAPS-100, Platform Architect & Virtualizer
- **Advanced technologies** FastMem server, Checkpoint / Restore
- **High productivity** and **performance** for authoring & runtime
- **Large set** of pre-integrated **models** (Arm FastModels, ARC, Tensilica, CEVA & other 3rd party models)
- Integrated **System Level Debug** with Software and Hardware debuggers



Fast Creation of Hybrid Emulation Setups

Authoring Automation



ZeBu Hybrid Adaptor Library

Hybrid Configurable Building Block

Build information (zebuIp, Zemi3, zdpi…)

ZeBu

RTL

ZeBu Compiler

XTOR Info

Bitstream

Generate

Generate

Pre-configured **Hybrid Adaptors**

- To match DUT XTORS
- No manual intervention

Pre-Configured **Testbench Blocks**

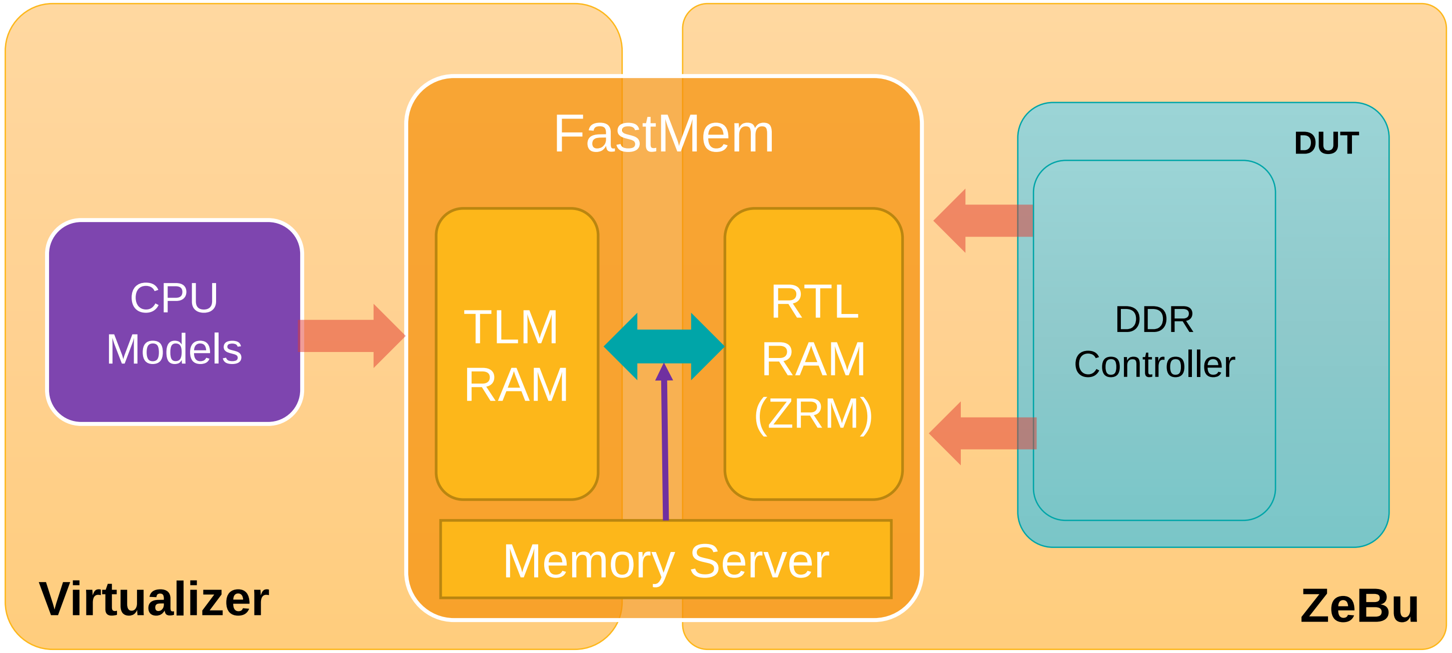
- Fully Automated setup
- Supports: UART, CSI, DSI

Virtualizer

- Faster Hybrid setup
- Configure both hybrid adaptors & ZeBu testbenches
- Correct by construction

Accelerated Memory Execution

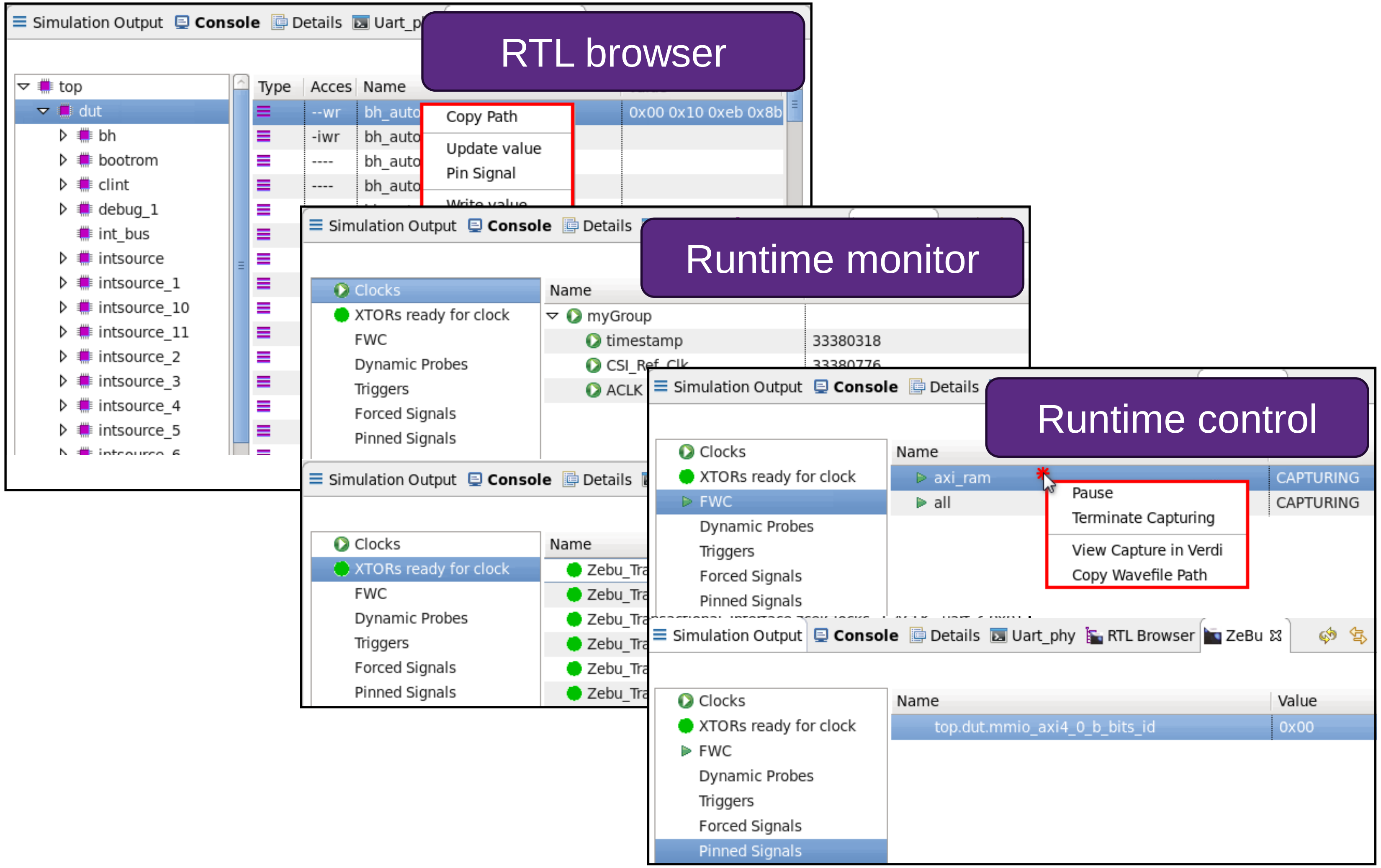
Advanced memory sharing technology



- Accelerates sharing of hybrid memory regions
  - Memory segmented into pages, with page ownership tracking
  - Automatic coherency of shared mem regions
  - Efficient data move
  - Speedup: ~5-10x
- Supports address interleaving & multi-bank
- Supports multiple use-cases
  - FastMem
  - Distributed FastMem
  - Extra large FastMemX

Higher Hybrid Debug Productivity

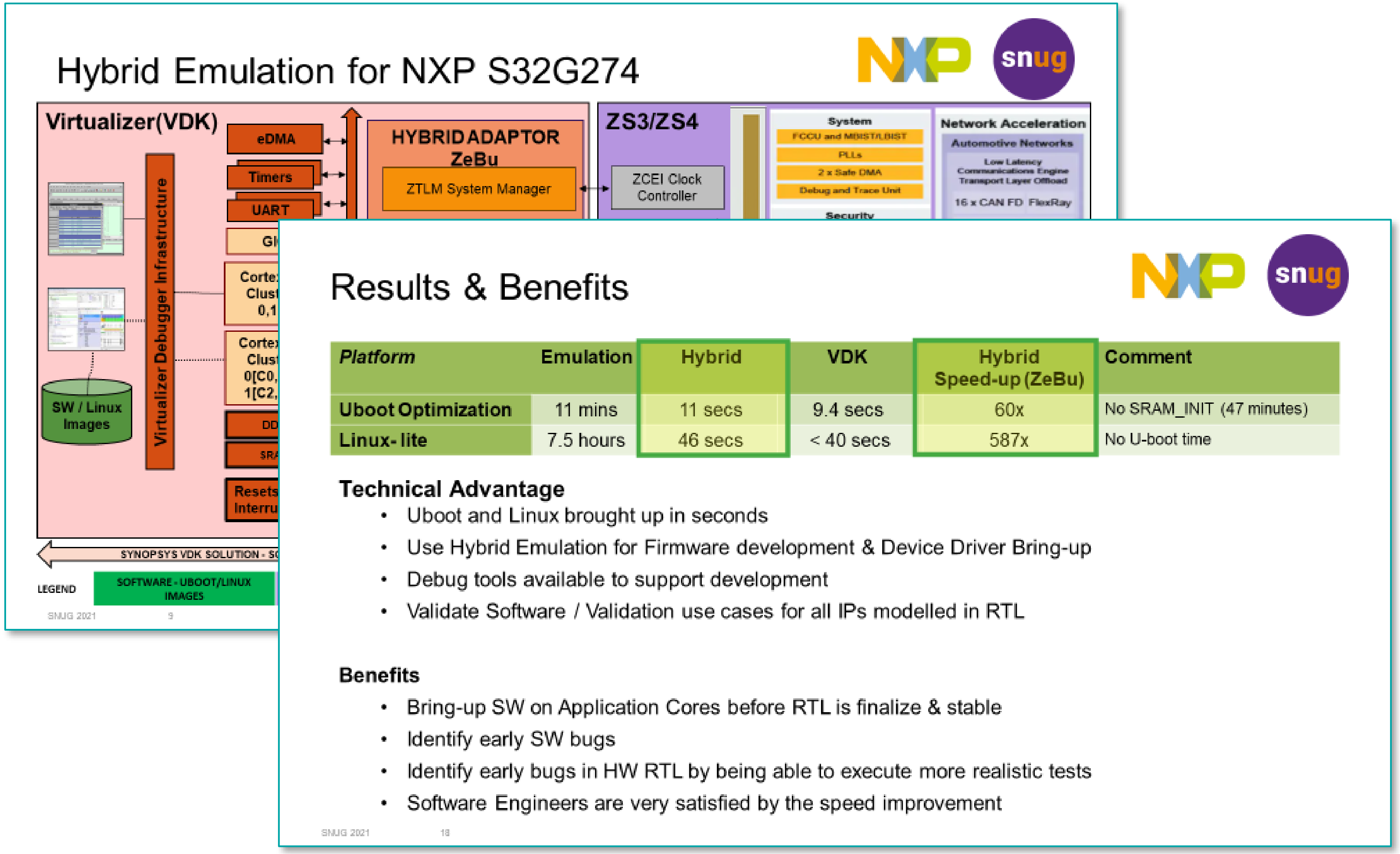
ZeBu views in Virtualizer Studio



- RTL browser
  - Access to all DUT signals
- Runtime monitor
  - System Clock Status
  - XTORs Readiness Status
- Runtime control
  - Waveform dump (FWC, QiWC, Dynamic Probe)
  - Triggers
  - Forces

Hybrid Emulation for faster SW Development

Presented by NXP and Synopsys at SNUG World 2021



Hybrid Emulation for NXP S32G274

Results & Benefits

Platform	Emulation	Hybrid	VDK	Hybrid Speed-up (ZeBu)	Comment
Uboot Optimization	11 mins	11 secs	9.4 secs	60x	No SRAM_INIT (47 minutes)
Linux- lite	7.5 hours	46 secs	< 40 secs	587x	No U-boot time

Technical Advantage

- Uboot and Linux brought up in seconds
- Use Hybrid Emulation for Firmware development & Device Driver Bring-up
- Debug tools available to support development
- Validate Software / Validation use cases for all IPs modelled in RTL

Benefits

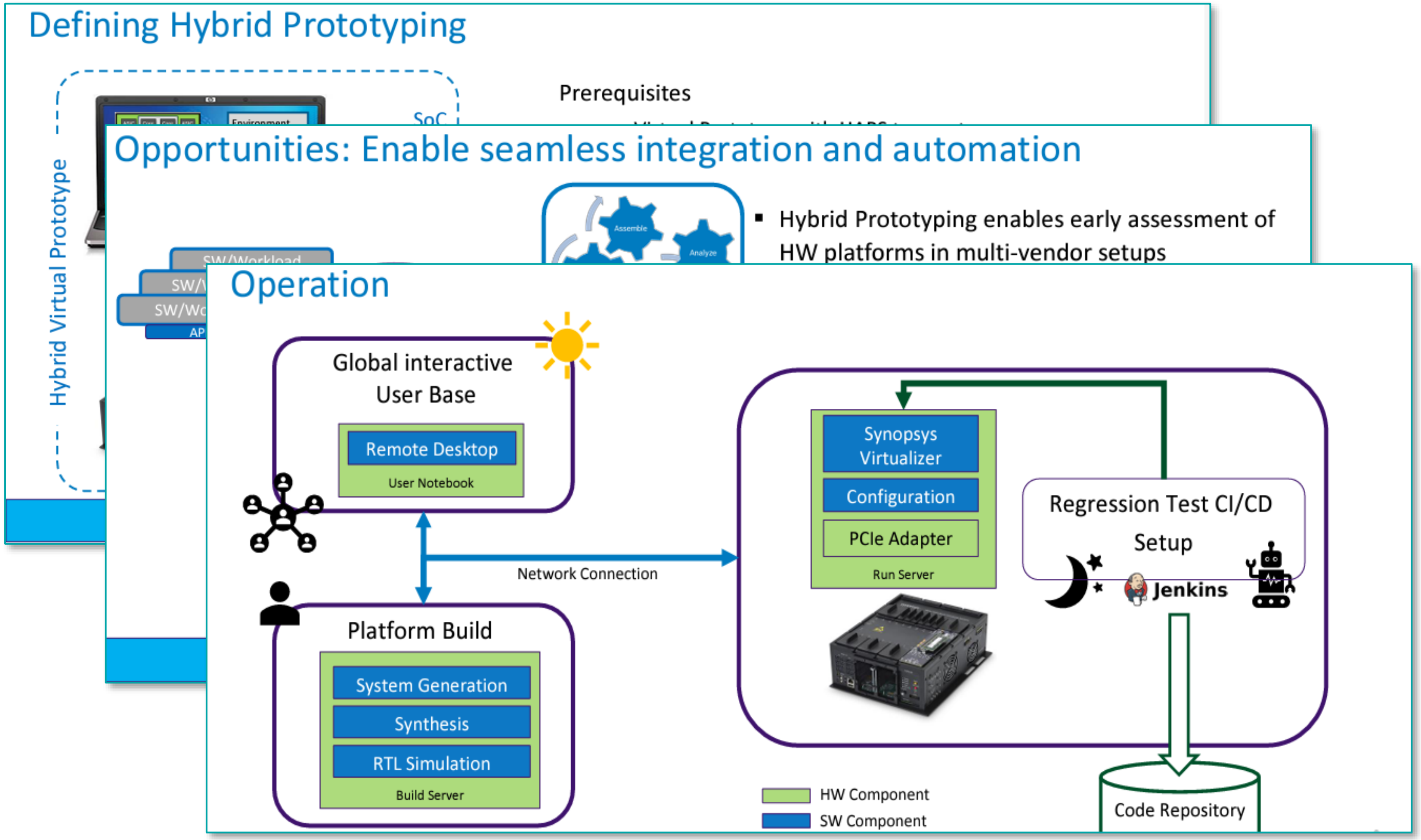
- Bring-up SW on Application Cores before RTL is finalize & stable
- Identify early SW bugs
- Identify early bugs in HW RTL by being able to execute more realistic tests
- Software Engineers are very satisfied by the speed improvement

SNUG 2021 18

- Increasing demand for software bring-up on emulator for complex embedded multicore SoC
- Hybrid emulation technology provides shift-left approach and helps to reduce risk and cost
- Fast setup with large library of models and transactors in Virtualizer and ZeBu
- Hybrid FastMem technology enables high simulation speed for shared memory access

Automotive Supply Chain Enablement

Presented by BOSCH at Synopsys VP-Day 2023



Defining Hybrid Prototyping

Prerequisites

Opportunities: Enable seamless integration and automation

- Hybrid Prototyping enables early assessment of HW platforms in multi-vendor setups

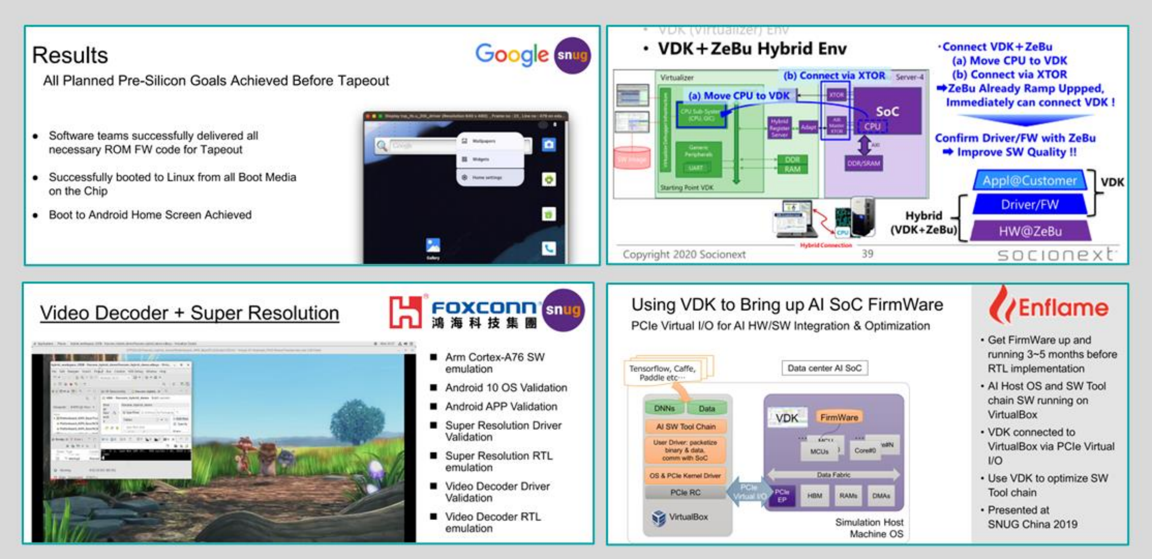
Operation

- Enabled testing of all critical IPs coming from Tier-1 and Tier-2
- Verify critical functionality of the HW, timing properties, and performance on IP and system level
- Ensuring correct system integration and sanity
- Avoided costly HW fixes and late SW taskforces
- Early understanding of the specification
- Flexible support for regression and interactive use globally

State-Of-The-Art Hybrid

Key Use-cases deployed over 10+ Years

Use-case	Components
SW Bring-up and Development	**Scope:** IP/Subsystem **ZeBu:** IP RTL **VDK:** Core + System Components
System Bring-up	**Scope:** SoC **ZeBu:** System Components RTL **VDK:** Core
Performance Benchmarking	**Scope:** IP/Subsystem **ZeBu:** IP RTL + Core RTL **VDK:** Core + System Components
Power Profiling	**Scope:** IP/Subsystem **ZeBu:** IP RTL + Power Intent (UPF) **VDK:** Core + System Components **Empower:** Dynamic Power Profile



Results

All Planned Pre-Silicon Goals Achieved Before Tapeout

- Software teams successfully delivered all necessary ROM FW code for Tapeout
- Successfully booted to Linux from all Boot Media on the Chip
- Boot to Android Home Screen Achieved

- VDK (Virtualizer) Env
- VDK+ZeBu Hybrid Env

- Connect VDK+ZeBu
  (a) Move CPU to VDK
  (b) Connect via XTOR

➡ZeBu Already Ramp Uppped, Immediately can connect VDK !

Confirm Driver/FW with ZeBu
➡ Improve SW Quality !!

Copyright 2020 Socionext 39

Video Decoder + Super Resolution

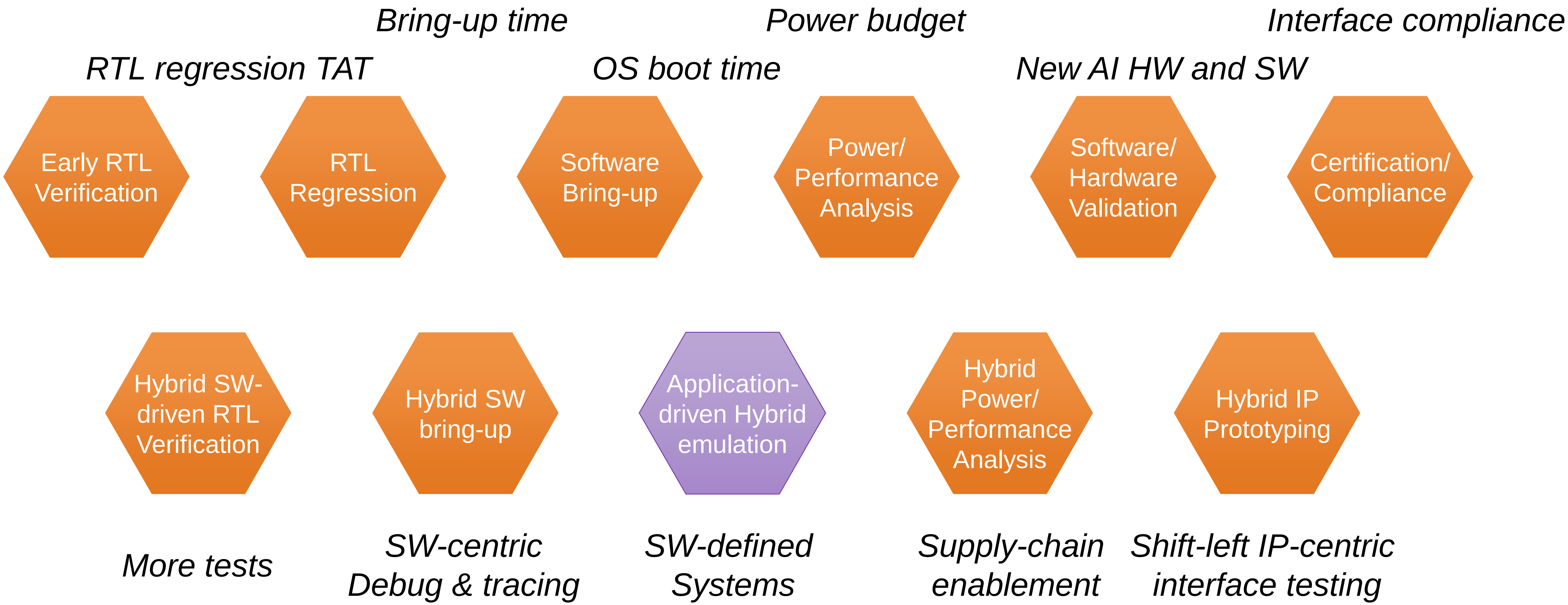
- Arm Cortex-A76 SW emulation
- Android 10 OS Validation
- Android APP Validation
- Super Resolution Driver Validation
- Super Resolution RTL emulation
- Video Decoder Driver Validation
- Video Decoder RTL emulation

Using VDK to Bring up AI SoC FirmWare

PCIe Virtual I/O for AI HW/SW Integration & Optimization

- Get FirmWare up and running 3~5 months before RTL implementation
- AI Host OS and SW Tool chain SW running on VirtualBox
- VDK connected to VirtualBox via PCIe Virtual I/O
- Use VDK to optimize SW Tool chain
- Presented at SNUG China 2019

Synopsys HW-Assisted Verification Use Cases



Bring-up time *Power budget* *Interface compliance*

RTL regression TAT *OS boot time* *New AI HW and SW*

- Early RTL Verification
- RTL Regression
- Software Bring-up
- Power/ Performance Analysis
- Software/ Hardware Validation
- Certification/ Compliance

- Hybrid SW-driven RTL Verification
- Hybrid SW bring-up
- Application-driven Hybrid emulation
- Hybrid Power/ Performance Analysis
- Hybrid IP Prototyping

More tests *SW-centric Debug & tracing* *SW-defined Systems* *Supply-chain enablement* *Shift-left IP-centric interface testing*

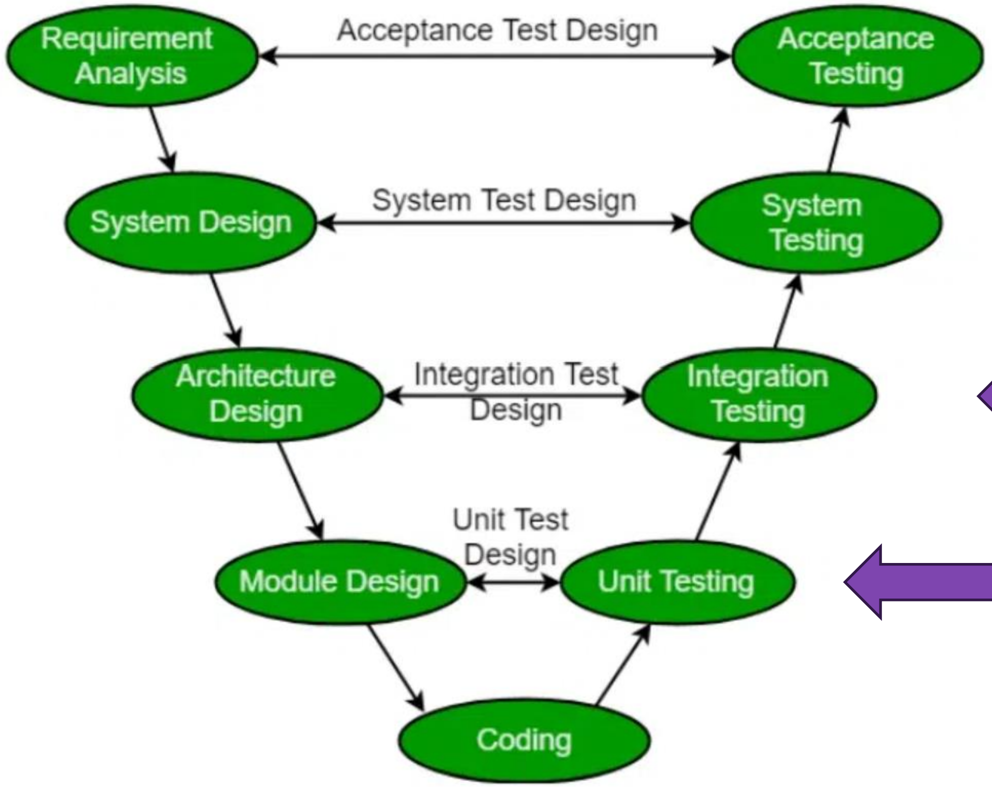
New Requirement: Pre-silicon Application Benchmarking

State-of-the-art Hybrid

OS & driver bring-up, some power monitors

New Hybrid Requirements

“I want to be able to run my application SW benchmarks”
“I need pre-silicon analysis of SW-driven power & performance”



Application benchmarking

IP-level SW testing

OS & driver bring-up

Leverage Available Software before Silicon

Industry changes supporting Shift-Left

- Common use of open-source for OS stacks
  - Products use open-source operating systems and software stacks
  - Linux is prevalent, Android is prevalent in automotive, mobile, consumer
  - Early access to SW stacks → earlier system validation
- Freely available application Software
  - Apps and benchmarks are established in many markets
  - Easily added to pre-silicon environments
- Multitude of product configurations
  - Products designed for reconfigurability to adjust to customer
  - Many SKUs defined by SW
  - Pre-Silicon testing needs to include ability to change

androidauto

SOAFEE

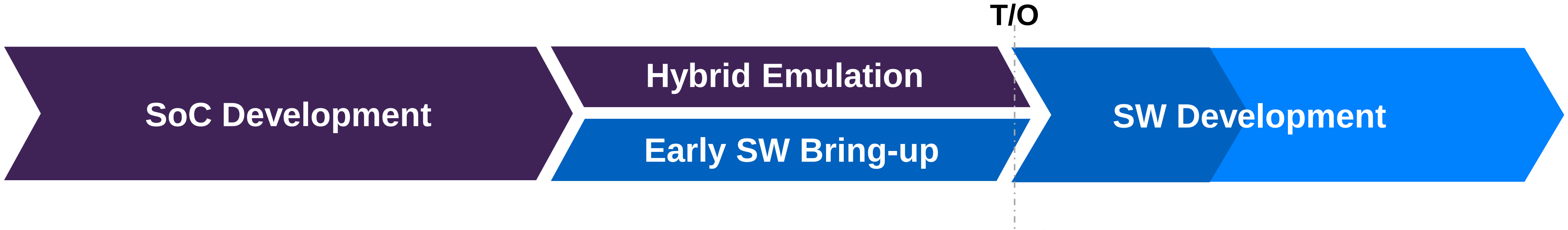
AUTOSAR

SDV™
Eclipse Software Defined Vehicle

3DMARK®

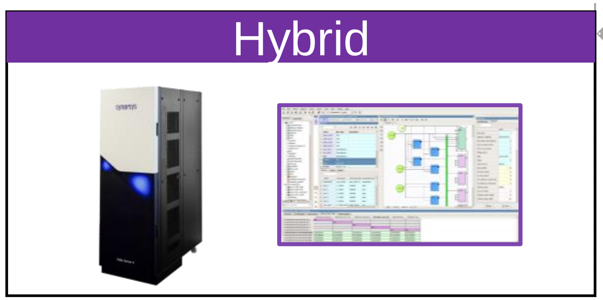
Gap for Software Defined Systems

Shift Left but coming short of running enough SW

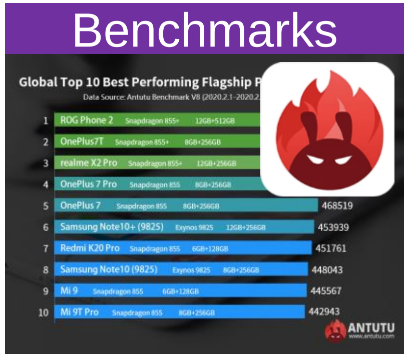


Hybrid combines fastest speeds for Software – via virtual CPU sub-systems, with fastest speeds for RTL

The world is moving to Software Defined Systems



Shift Left

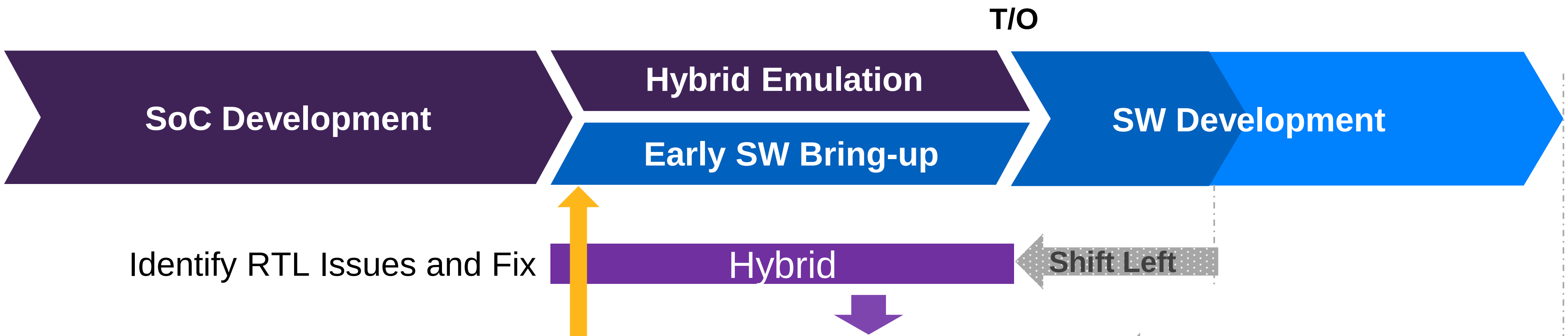


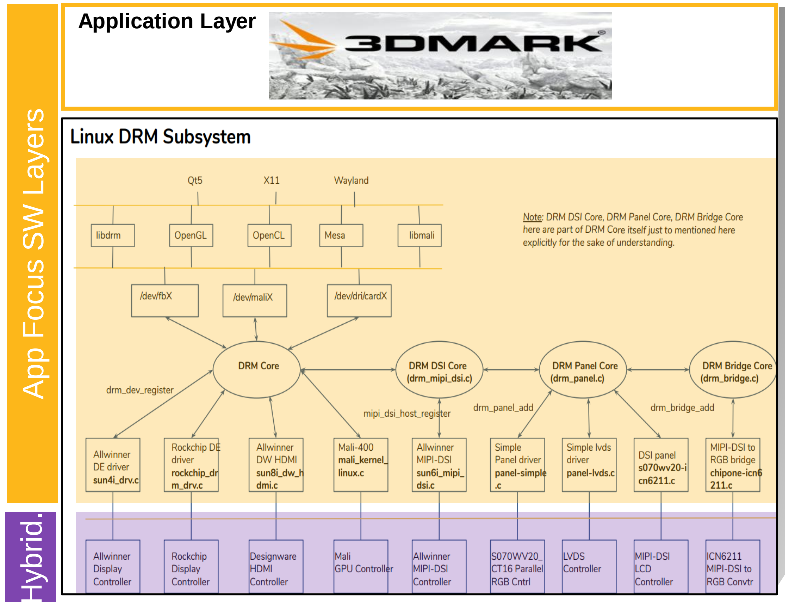
Too late to improve RTL

Software Defined Systems → Hardware managed by **Software**

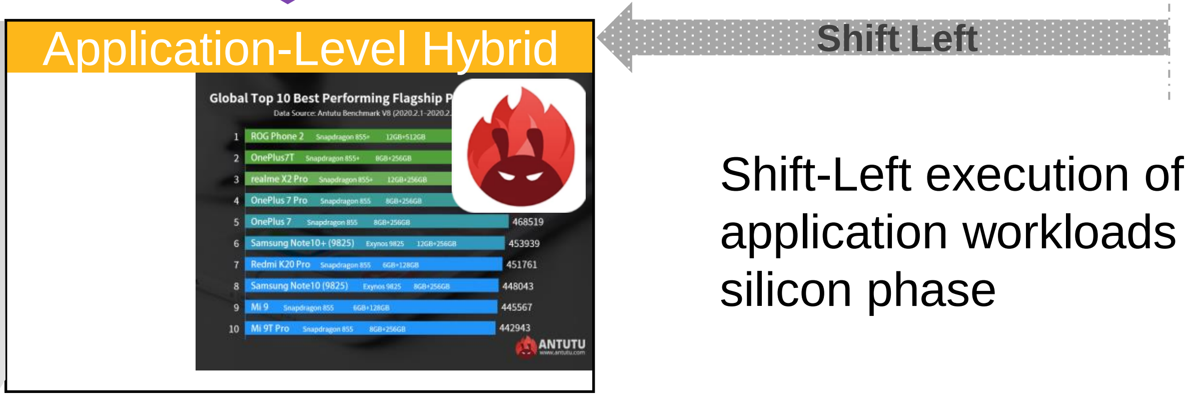
Emerging “Application-Level” Hybrid Emulation

Run full apps to gain performance and power insights




App Focus SW Layers
Hybrid.
Application Layer
3DMARK
Linux DRM Subsystem
Qt5
X11
Wayland
libdrm
OpenGL
OpenCL
Mesa
libmali
/dev/fbX
/dev/maliX
/dev/dri/cardX
DRM Core
DRM DSI Core (drm_mipi_dsi.c)
DRM Panel Core (drm_panel.c)
DRM Bridge Core (drm_bridge.c)
drm_dev_register
mipi_dsi_host_register
drm_panel_add
drm_bridge_add
Allwinner DE driver sun4i_drv.c
Rockchip DE driver rockchip_drm_drv.c
Allwinner DW HDMI sun8i_dw_hdmi.c
Mali-400 mali_kernel_linux.c
Allwinner MIPI-DSI sun6i_mipi_dsi.c
Simple Panel driver panel-simple.c
Simple lvds driver panel-lvds.c
DSI panel s070wv20-icn6211.c
MIPI-DSI to RGB bridge chipone-icn6211.c
Allwinner Display Controller
Rockchip Display Controller
Designware HDMI Controller
Mali GPU Controller
Allwinner MIPI-DSI Controller
S070WV20_CT16 Parallel RGB Cntrl
LVDS Controller
MIPI-DSI LCD Controller
ICN6211 MIPI-DSI to RGB Convtr

Note: DRM DSI Core, DRM Panel Core, DRM Bridge Core here are part of DRM Core itself just to mentioned here explicitly for the sake of understanding.



Global Top 10 Best Performing Flagship P

Data Source: Antutu Benchmark V8 (2020.2.1-2020.2.

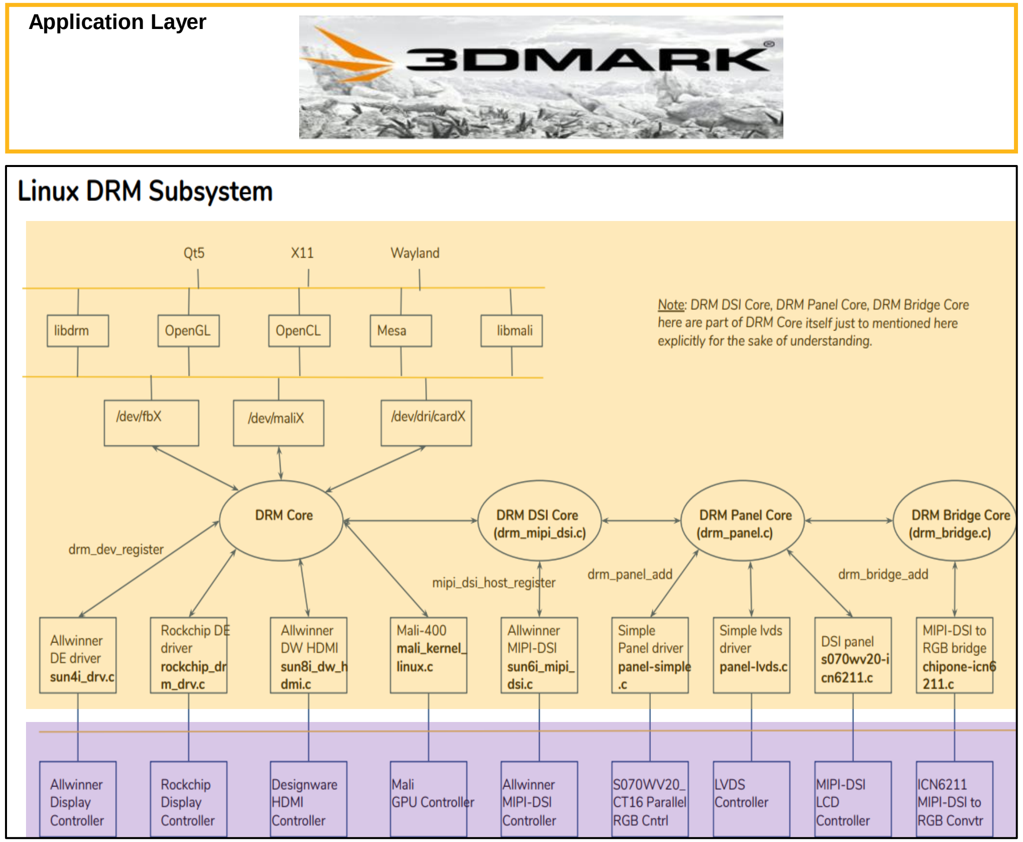
1	ROG Phone 2	Snapdragon 855+	12GB+512GB	
2	OnePlus7T	Snapdragon 855+	8GB+256GB	
3	realme X2 Pro	Snapdragon 855+	12GB+256GB	
4	OnePlus 7 Pro	Snapdragon 855	8GB+256GB	
5	OnePlus 7	Snapdragon 855	8GB+256GB	468519
6	Samsung Note10+ (9825)	Exynos 9825	12GB+256GB	453939
7	Redmi K20 Pro	Snapdragon 855	6GB+128GB	451761
8	Samsung Note10 (9825)	Exynos 9825	8GB+256GB	448043
9	Mi 9	Snapdragon 855	6GB+128GB	445567
10	Mi 9T Pro	Snapdragon 855	8GB+256GB	442943

Shift-Left execution of realistic application workloads to the pre-silicon phase

Ability to measure power & performance bottlenecks and locate hotspots

The Transition to “Software Defined Systems”

And its impact on Hybrid Emulation



Application Layer

Linux DRM Subsystem

3DMARK
Qt5
X11
Wayland
libdrm
OpenGL
OpenCL
Mesa
libmali
/dev/fbX
/dev/maliX
/dev/dri/cardX
DRM Core
DRM DSI Core (drm_mipi_dsi.c)
DRM Panel Core (drm_panel.c)
DRM Bridge Core (drm_bridge.c)
drm_dev_register
mipi_dsi_host_register
drm_panel_add
drm_bridge_add
Allwinner DE driver sun4i_drv.c
Rockchip DE driver rockchip_dr m_drv.c
Allwinner DW HDMI sun8i_dw_h dmi.c
Mali-400 mali_kernel_ linux.c
Allwinner MIPI-DSI sun6i_mipi_ dsi.c
Simple Panel driver panel-simple .c
Simple lvds driver panel-lvds.c
DSI panel s070wv20-i cn6211.c
MIPI-DSI to RGB bridge chipone-icn6 211.c
Allwinner Display Controller
Rockchip Display Controller
Designware HDMI Controller
Mali GPU Controller
Allwinner MIPI-DSI Controller
S070WV20_ CT16 Parallel RGB Cntrl
LVDS Controller
MIPI-DSI LCD Controller
ICN6211 MIPI-DSI to RGB Convtr

Note: DRM DSI Core, DRM Panel Core, DRM Bridge Core here are part of DRM Core itself just to mentioned here explicitly for the sake of understanding.

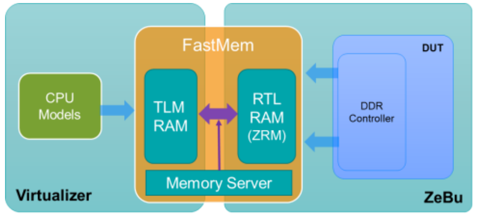
- The state-of-the-art hybrid (noted as **purple boxes**) needs to be enhanced with SW (noted as **yellow** boxes).
- For the hybrid to be useful for benchmarking software defined systems, all the relevant SW needs to be available

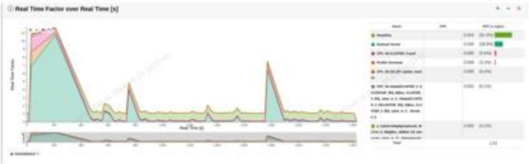
- Latest performance advances in virtual and emulation enable execution of application-level workloads
- Observability capabilities in Virtual and Hybrid enable correlation of SW to power and performance bottlenecks

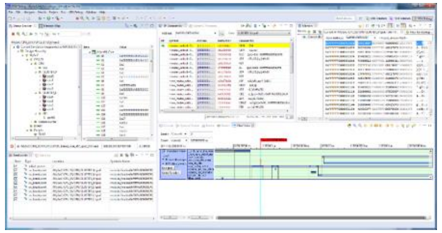
Achieve Speed and Insight

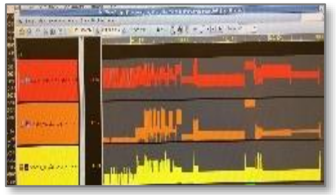
Application-Level Hybrid Emulation with Synopsys Virtualizer and ZeBu

	Speed	Insight
Virtualizer	Execution speed 100's – 1000's MIPS	Capture function traces across end-user apps, to identify SW functions with adverse impacts on system.
ZeBu	Execution speed 3MHz – 10's MHz; from emulation to prototyping	Capture power across billions of cycles, to find "anomalies“, e.g. high power with low performance.





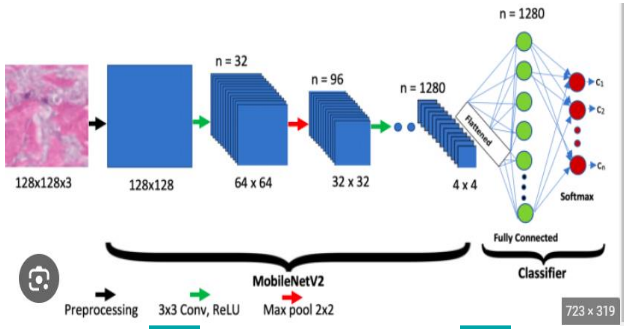


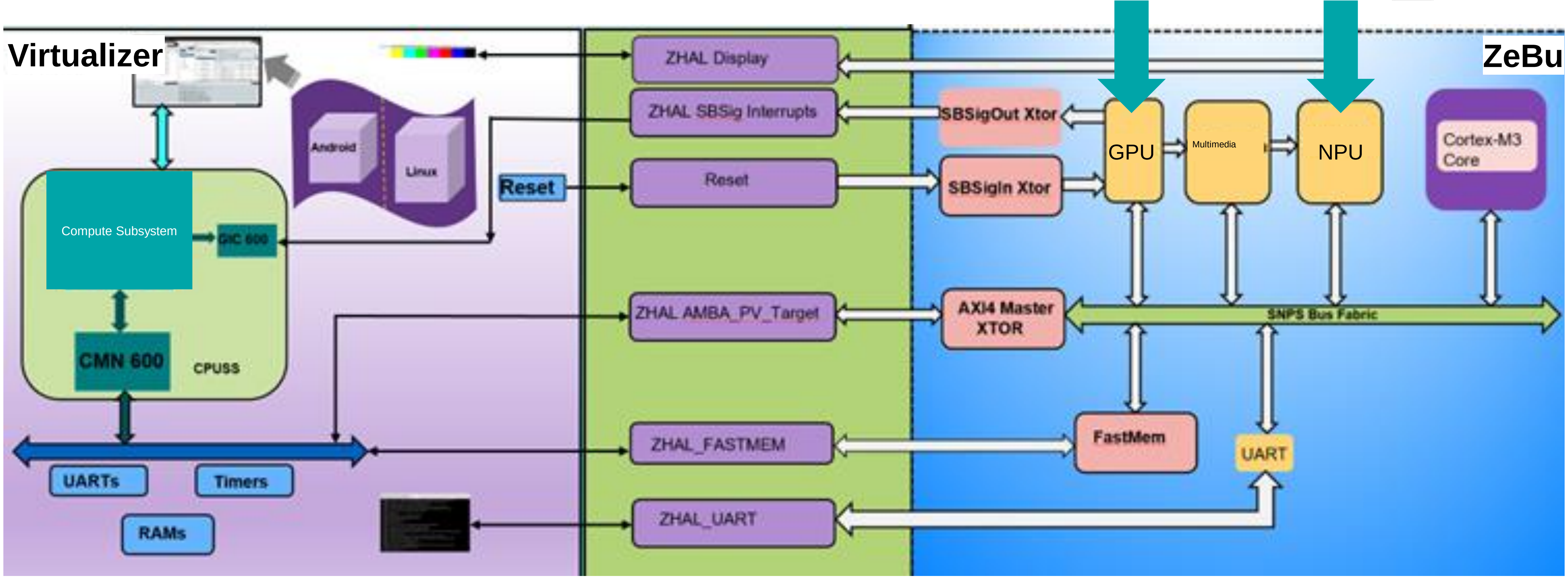


Billions cycles power profile

Architecture Inferencing Case Study

AI model drives HW requirements (compute, memory)

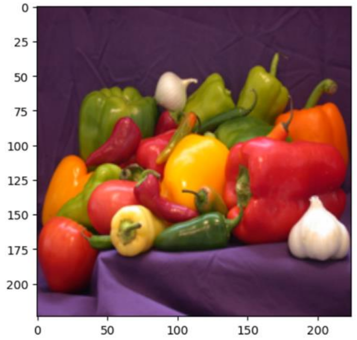




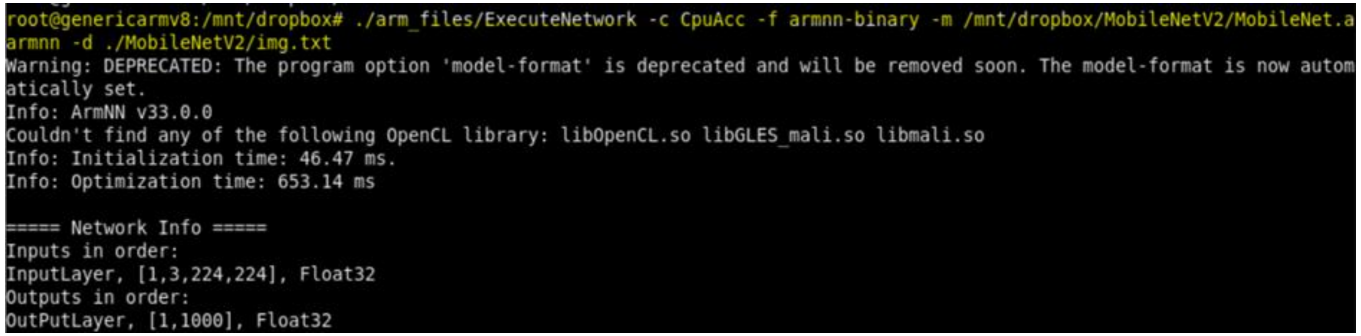
Using Hybrid Platform to study AI Architecture

- Input image

```
<matplotlib.image.AxesImage at 0x7ff25b12c220>
```

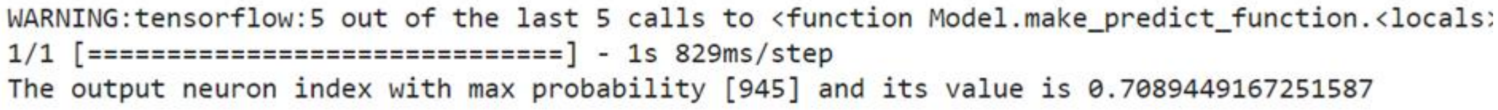


- Passing the image to MobilNetV2 returning that the 945th Neuron gets the maximum probability of 0.878826



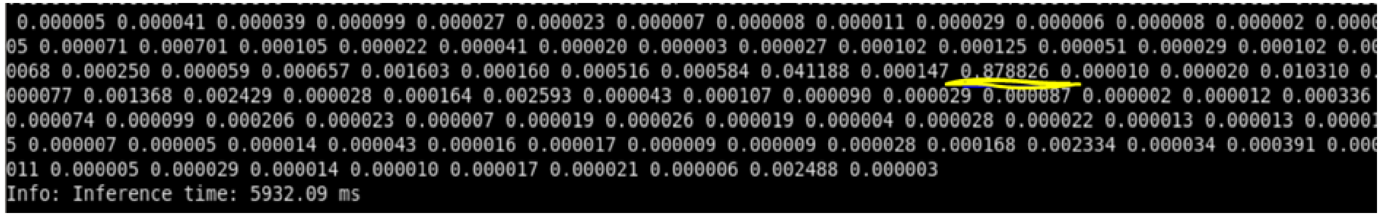
```
root@genericarmv8:/mnt/dropbox# ./arm_files/ExecuteNetwork -c CpuAcc -f armnn-binary -m /mnt/dropbox/MobileNetV2/MobileNet.a
armnn -d ./MobileNetV2/img.txt
Warning: DEPRECATED: The program option 'model-format' is deprecated and will be removed soon. The model-format is now autom
atically set.
Info: ArmNN v33.0.0
Couldn't find any of the following OpenCL library: libOpenCL.so libGLES_mali.so libmali.so
Info: Initialization time: 46.47 ms.
Info: Optimization time: 653.14 ms

===== Network Info =====
Inputs in order:
InputLayer, [1,3,224,224], Float32
Outputs in order:
OutPutLayer, [1,1000], Float32
```



```
WARNING:tensorflow:5 out of the last 5 calls to <function Model.make_predict_function.<locals>
1/1 [==============================] - 1s 829ms/step
The output neuron index with max probability [945] and its value is 0.7089449167251587
```

- Label on the ImageNet dataset corresponding to 945th Neuron is “Bell Peppers”



```
 0.000005 0.000041 0.000039 0.000099 0.000027 0.000023 0.000007 0.000008 0.000011 0.000029 0.000006 0.000008 0.000002 0.0000
05 0.000071 0.000701 0.000105 0.000022 0.000041 0.000020 0.000003 0.000027 0.000102 0.000125 0.000051 0.000029 0.000102 0.00
0068 0.000250 0.000059 0.000657 0.001603 0.000160 0.000516 0.000584 0.041188 0.000147 0.878826 0.000010 0.000020 0.010310 0.
000077 0.001368 0.002429 0.000028 0.000164 0.002593 0.000043 0.000107 0.000090 0.000029 0.000087 0.000002 0.000012 0.000336
0.000074 0.000099 0.000206 0.000023 0.000007 0.000019 0.000026 0.000019 0.000004 0.000028 0.000022 0.000013 0.000013 0.00001
5 0.000007 0.000005 0.000014 0.000043 0.000016 0.000017 0.000009 0.000009 0.000028 0.000168 0.002334 0.000034 0.000391 0.000
011 0.000005 0.000029 0.000014 0.000010 0.000017 0.000021 0.000006 0.002488 0.000003
Info: Inference time: 5932.09 ms
```

https://deeplearning.cms.waikato.ac.nz/user-guide/class-maps/IMAGENET/

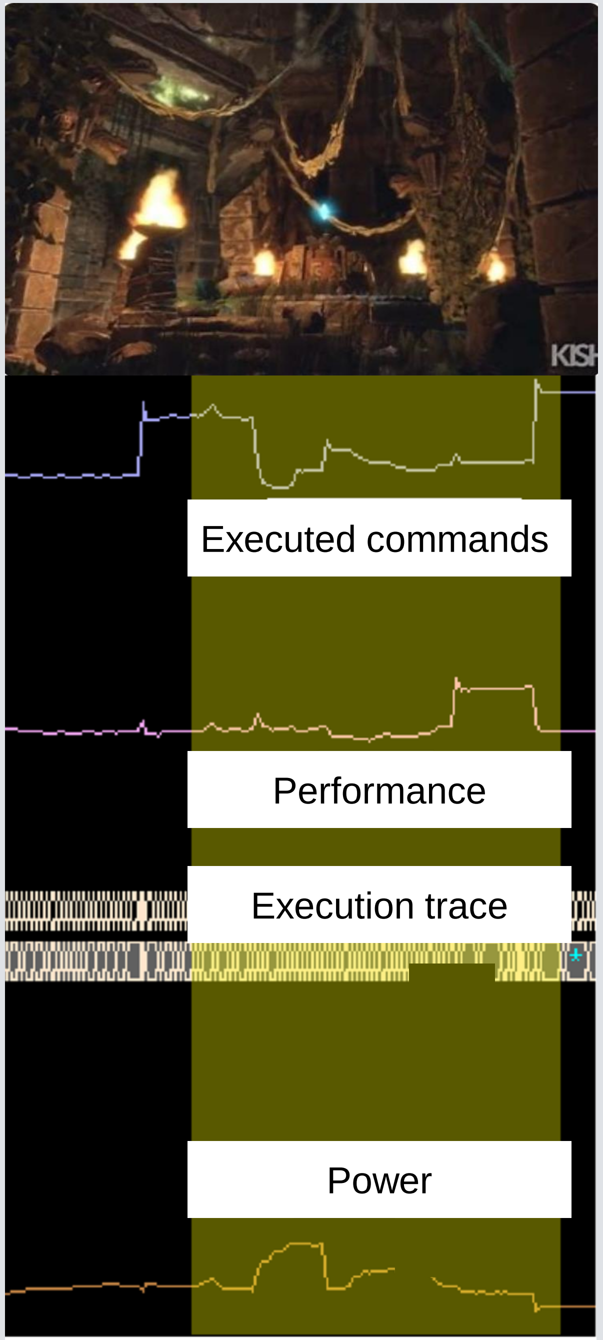
GPU Benchmark Case Study

Example: GFXBench Aztec Ruins

- Hybrid execution performance requirements
  - 1 minute of real time execution
  - Relevant observation interval starts 20 seconds into the benchmark

- Observations
  - High-power sometimes not correlated with high performance
  - High-performance sometimes not correlated with high power

- Root-cause examples
  - Power bugs, where certain GPU domains do not get disabled
  - Power optimizations causing high performance under lower power



Shift Focus to Accelerate System Success

- Hybrid well established for SW bring-up
- State-of-the-art Hybrid Emulation
  - A faster platform for HW prototyping
- Focus shifting to “Software-Defined Systems”
  - System = software + hardware
  - Bring-up and optimization of user applications

snug

THANK YOU

Our
Technology,
Your
Innovation™