

Shift Left with Synopsys Virtual Prototype

Using VDKs for Early Software Bring-up & Test of Power Management Software for AP SoC

Xiaowei Pan

March 23, 2017



CONFIDENTIAL INFORMATION

The following material is confidential information of Synopsys and is being disclosed to you pursuant to a non-disclosure agreement between you or your employer and Synopsys. The material being disclosed may only be used as permitted under such non-disclosure agreement.

IMPORTANT NOTICE

In the event information in this presentation reflects Synopsys' future plans, such plans are as of the date of this presentation and are subject to change. Synopsys is not obligated to develop the software with the features and functionality discussed in these materials. In any event, Synopsys' products may be offered and purchased only pursuant to an authorized quote and purchase order or a mutually agreed upon written contract.

Agenda

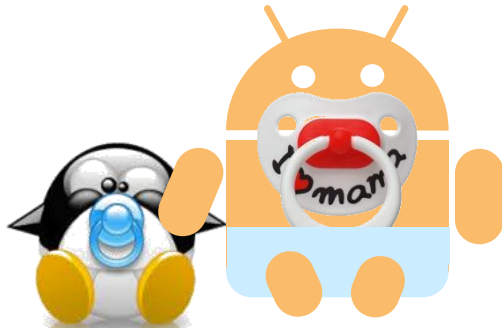
- From Reference Software to Production
- Shift Left – with Virtual Prototype & Virtualizer Development Kits
- Case Study – Power Management SW Bring-Up using VDK
- Q&A

From Reference Software to Production

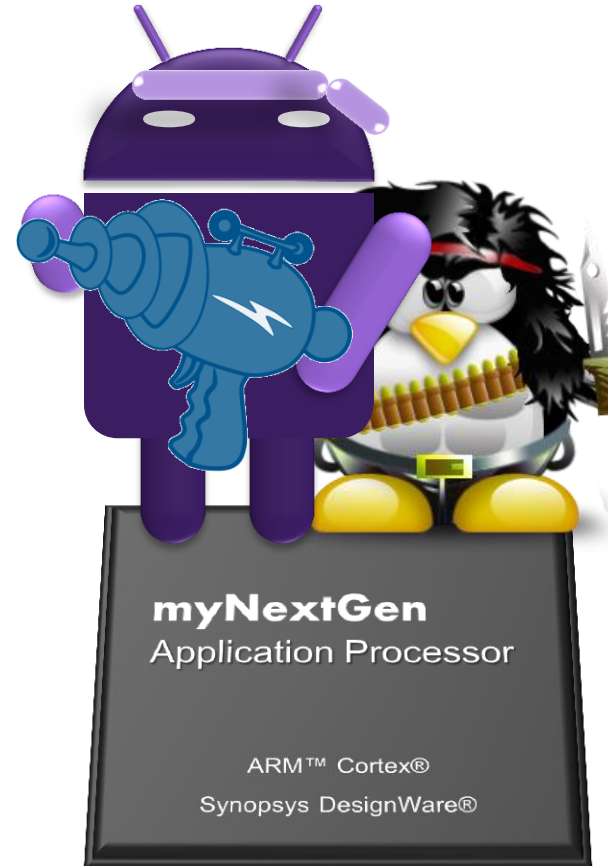
From Reference Software to Production

Still some miles to go

Open source
reference



Benchmark winning
production SW



SoC Software Bring-Up

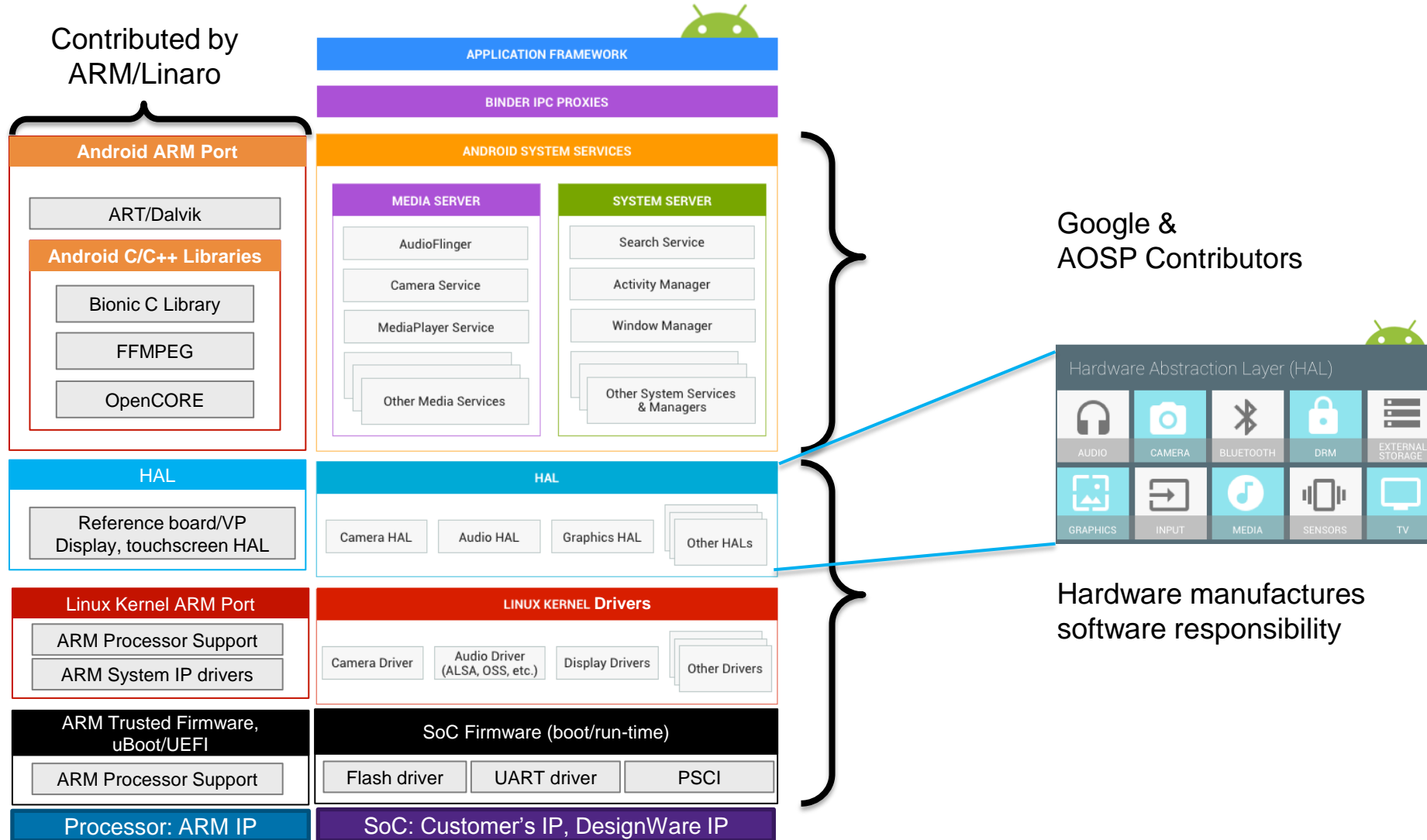
Software re-use is key, but not everything

- Android mobile hand-set software stack is open source
 - ARM firmware, uBoot/UEFI, Linux, Android
 - Enables fast bring up of a minimal Android
- Reference software is targeting reference platforms
 - Targets: VExpress TC2, Juno board, ARM foundation model
 - SoC specification is different
- Open source IP drivers of existing hardware platforms
 - Tailored and tested only for specific IP configurations
 - High risk of inefficiencies or defects for your IP configuration

Significant work to be done before software is production ready!

Android System Architecture

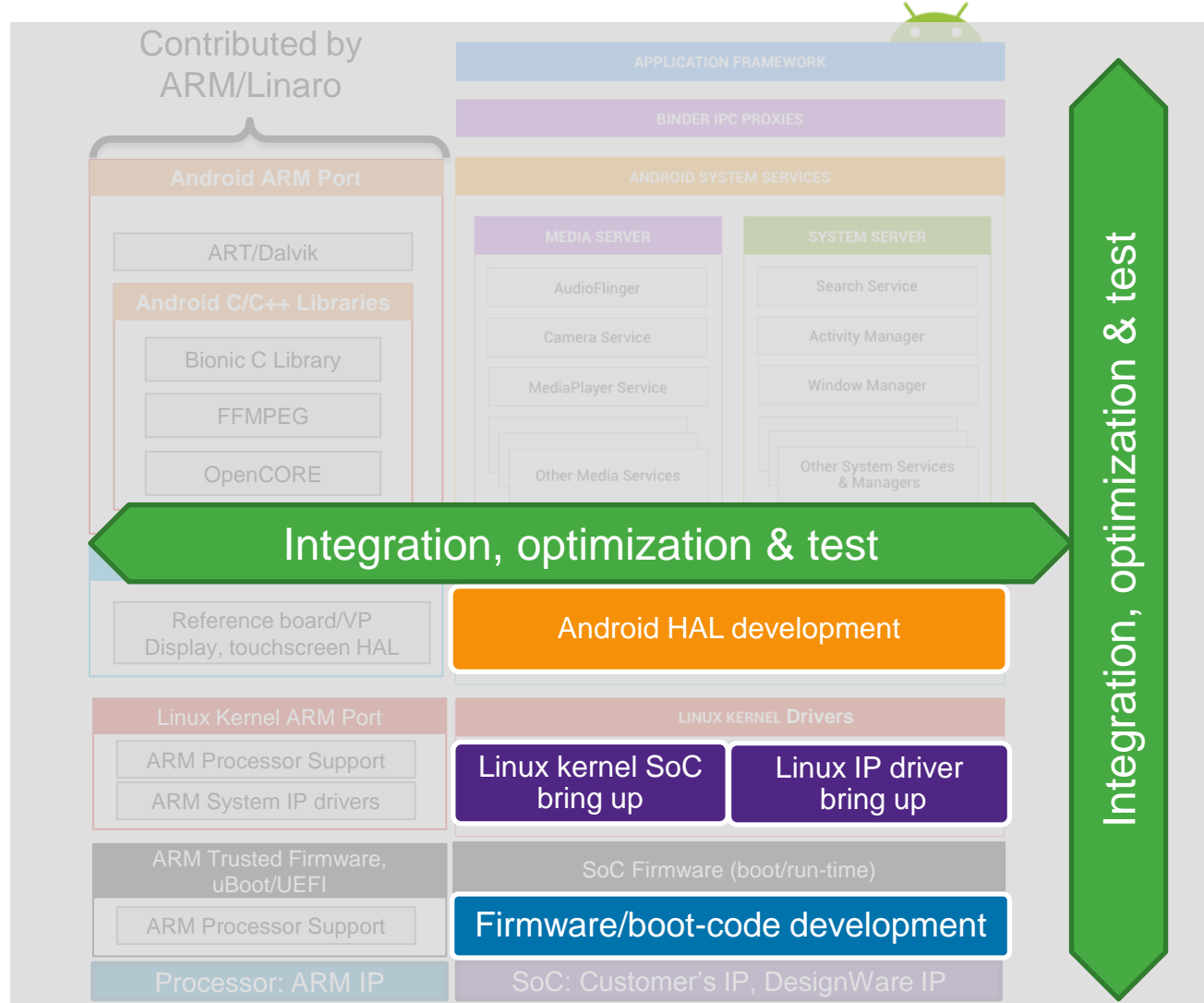
Layers, components and ownership



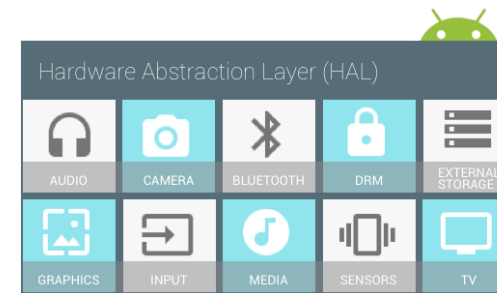
Sources: <https://source.android.com/devices/index.html>

Android System Architecture

Layers, components and ownership



Google & AOSP Contributors

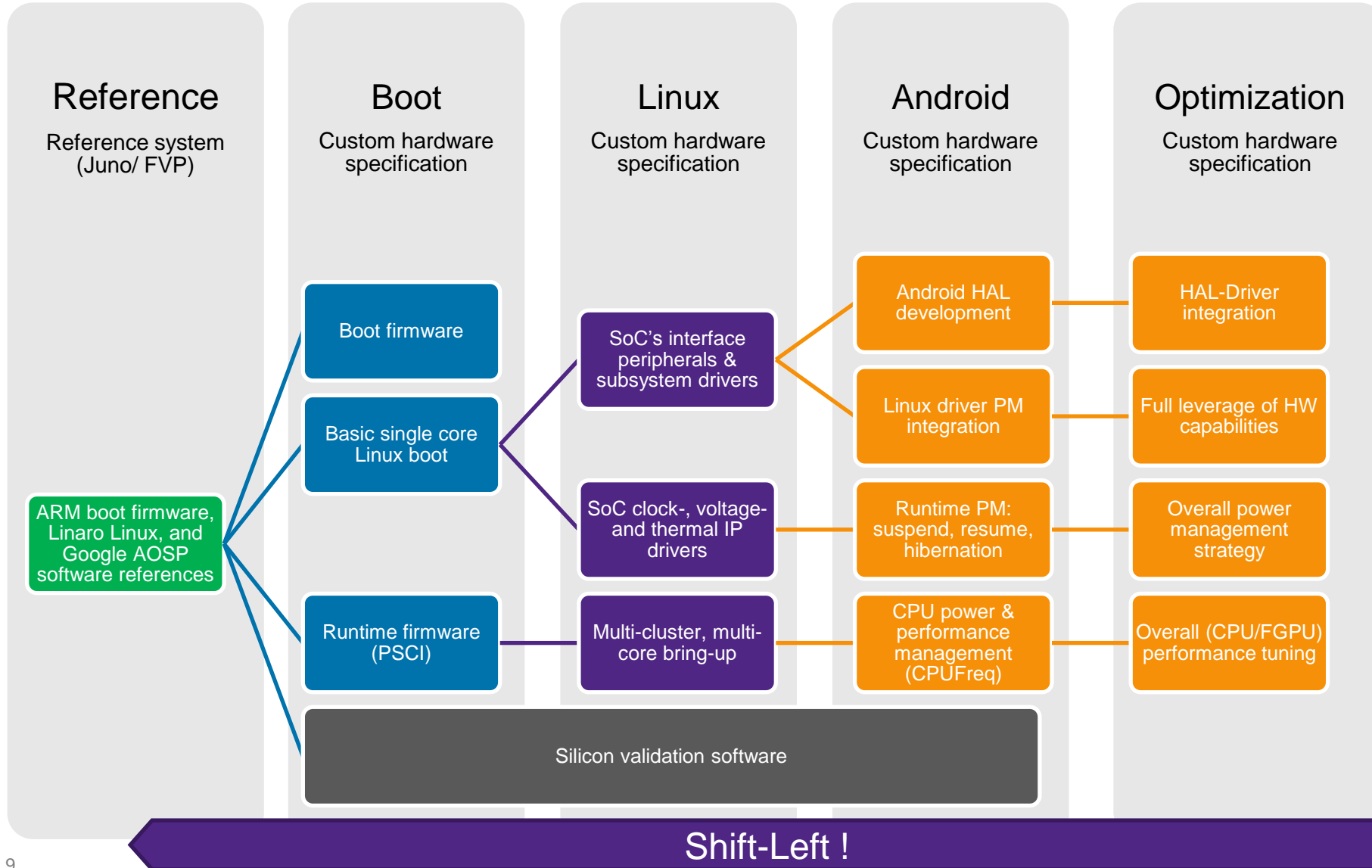


Hardware manufactures software responsibility

Sources: <https://source.android.com/devices/index.html>

Steps to Software Readiness

From open source reference to final software



McKinsey on Semiconductors

In fact, semiconductor companies of any size could realize great cost savings and productivity benefits by making virtual platforms an integral part of their SOC planning and design cycles. The

Shift Left

Virtual Prototype & Virtualizer Development Kits

Qualcomm Shifts Left with Virtualizer and VDKs

SNUG Silicon Valley 2016



Virtual Platforms for Large Scale Pre-Silicon Software Development

Avin Kannur (Presenter)
Staff Engineer
Qualcomm Technologies, Inc.

Rajiv Narayan (Co-Presenter)
Principal Engineer/Mgr
Qualcomm Technologies, Inc.

March 31, 2016
Santa Clara Convention Center



[Download](#) presentation

Virtual Platform Impact

How is it adding value to the Organization?

- Chip bring-up after Silicon on Dock (SOD) in matter of weeks as compared to months earlier
- Product SW deliveries pulled in the order of weeks to months
- Enables validation of HW with a stable SW during design development phase – catching HW bugs before Tape Out
- Time to identify a SW bug in VP v/s Silicon
- Prediction of KPIs for next revision of chipset and/or derivatives



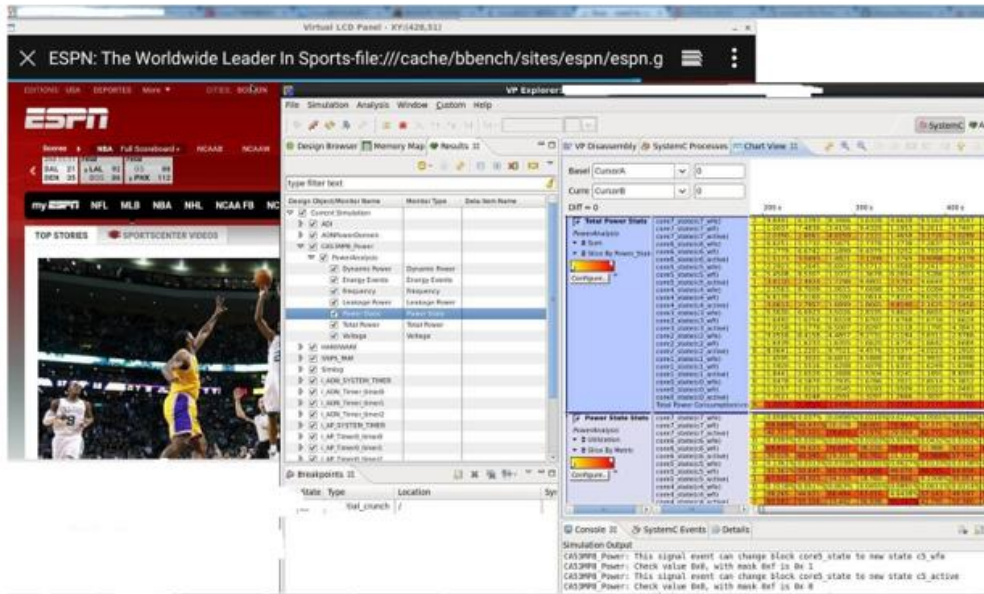
SNUG 2016

8

Early SW Bring-up & Power Analysis with Virtual Prototype

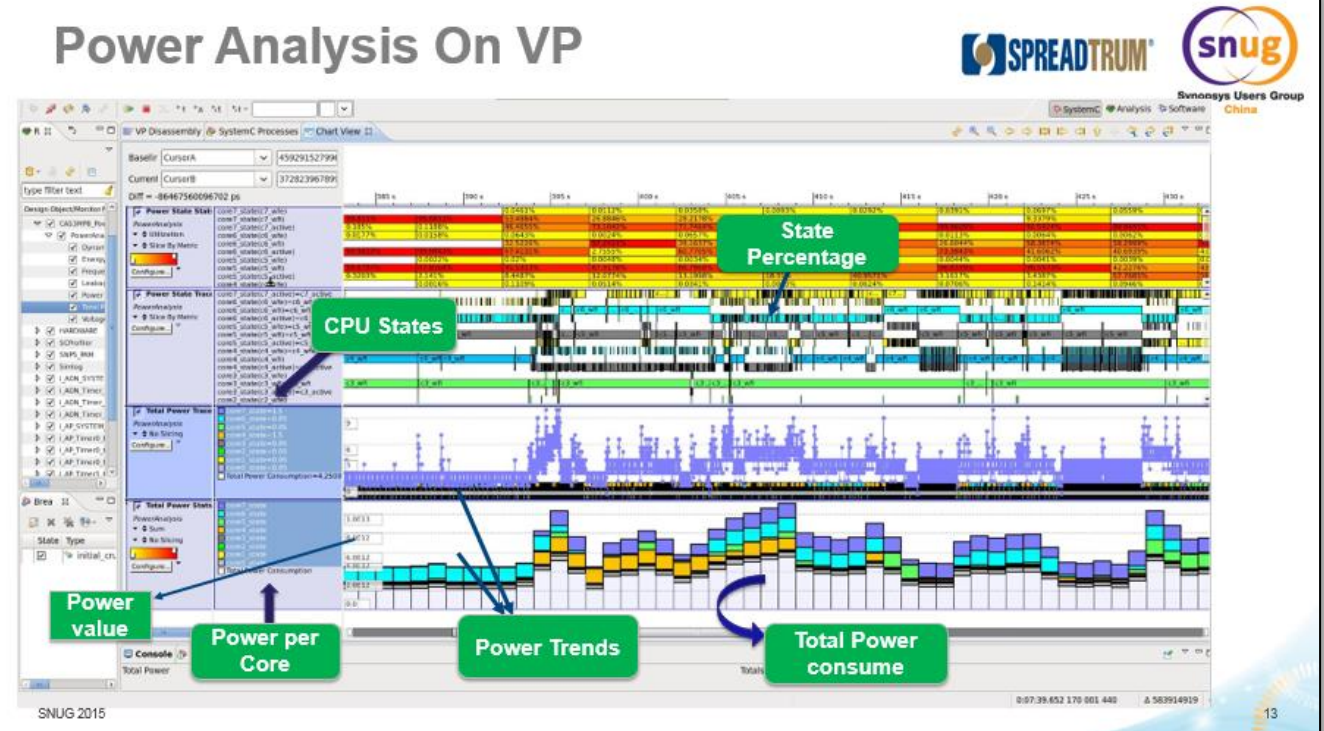
SNUG Shanghai 2016

Android



SNUG 2015

Power Analysis On VP

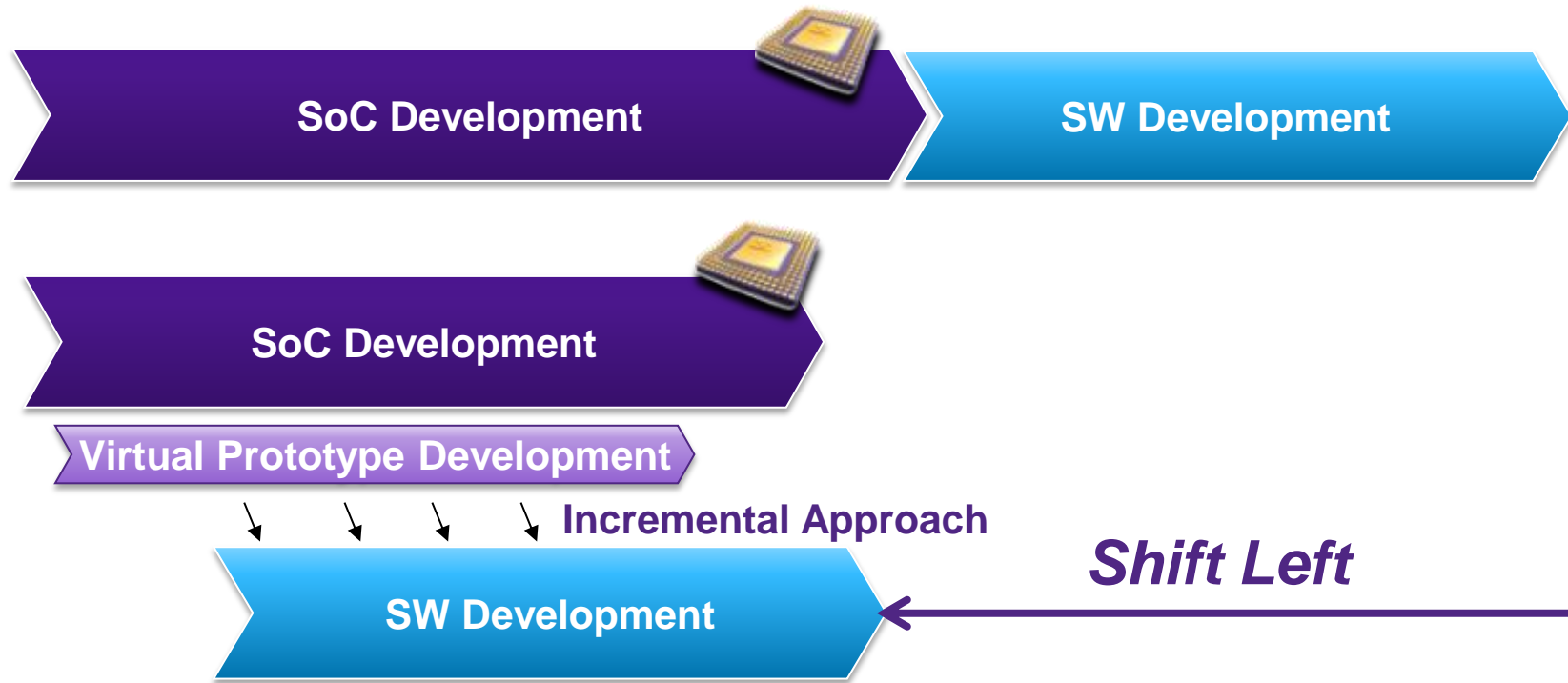


SNUG 2015

13

Why do Virtual Prototyping?

Challenges of the Sequential Design Flow



- Break Dependencies on RTL Availability (by using Transaction Level Models)
- Agile Software Development in Lock Step with Virtual Prototype Development

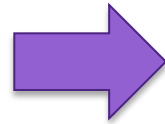
What are Virtualizer™ Development Kits?

Also Known as VDKs

- Software Development Kits that use a Virtual Prototype as a target
- VDK's are fully functional models of the system executing target code (SW / FW)



Development board



Early Availability

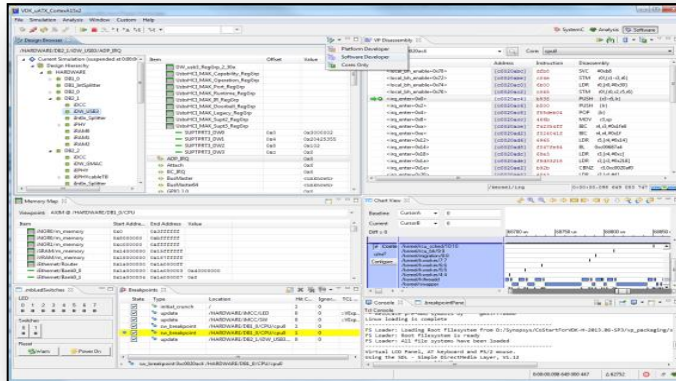
Easier Deployment

Better SW Development Productivity

- Visibility
- Control and repeatability
- Fault Injection support
- Scriptable

Software Developer's Extended View

System wide debug



Memories

Registers

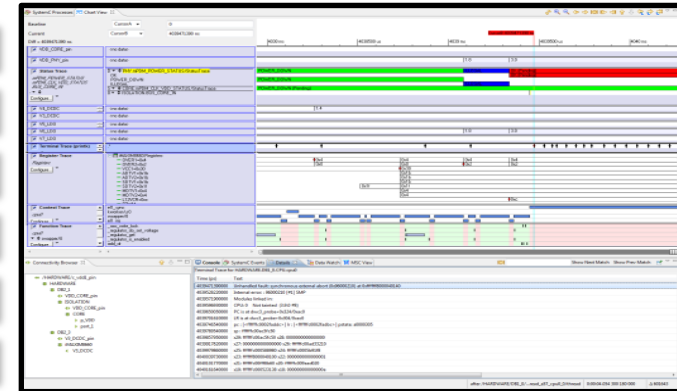
Signals

States

Disassembly

Log messages

System wide trace



Python/TCL command based control and Inspection APIs

A Virtual Prototype is not a black box!

Efficient Solution for VDK Creation

Synopsys Virtualizer™ Prototyping Solution

VDK Creation - Lowest Creation Cost

Large model library

Efficient Authoring tools

Extendable Reference VDKs

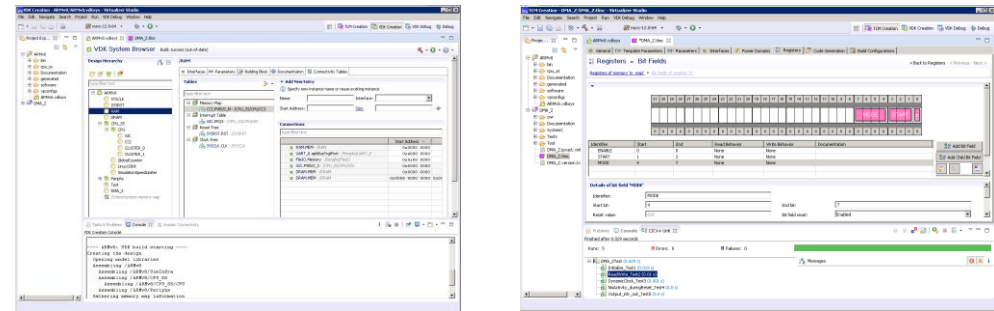
Standards-based (SystemC)



Synopsys DesignWare (USB, PCIe, UFS, GMAC ...)

Software Stack

Virtualizer Studio



ARM v8 Base; ARC HS; Automotive MCU/ADAS,...



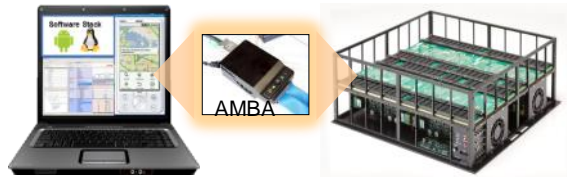
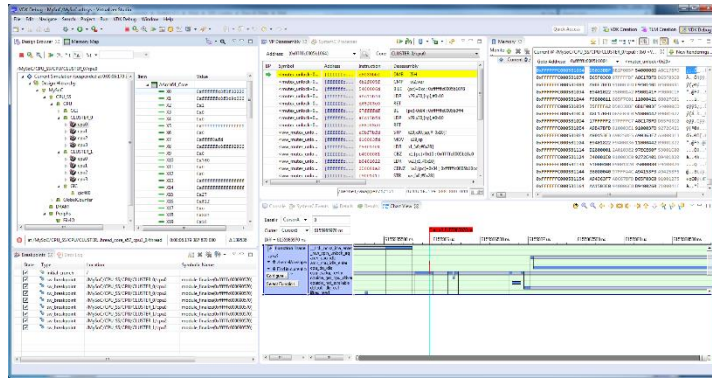
Efficient Solution for VDK Use

Synopsys Virtualizer™ Prototyping Solution

Embedded SW Debuggers



VDK Debugger



VDK Use - Highest Productivity

Integration with SW Engineers Tools

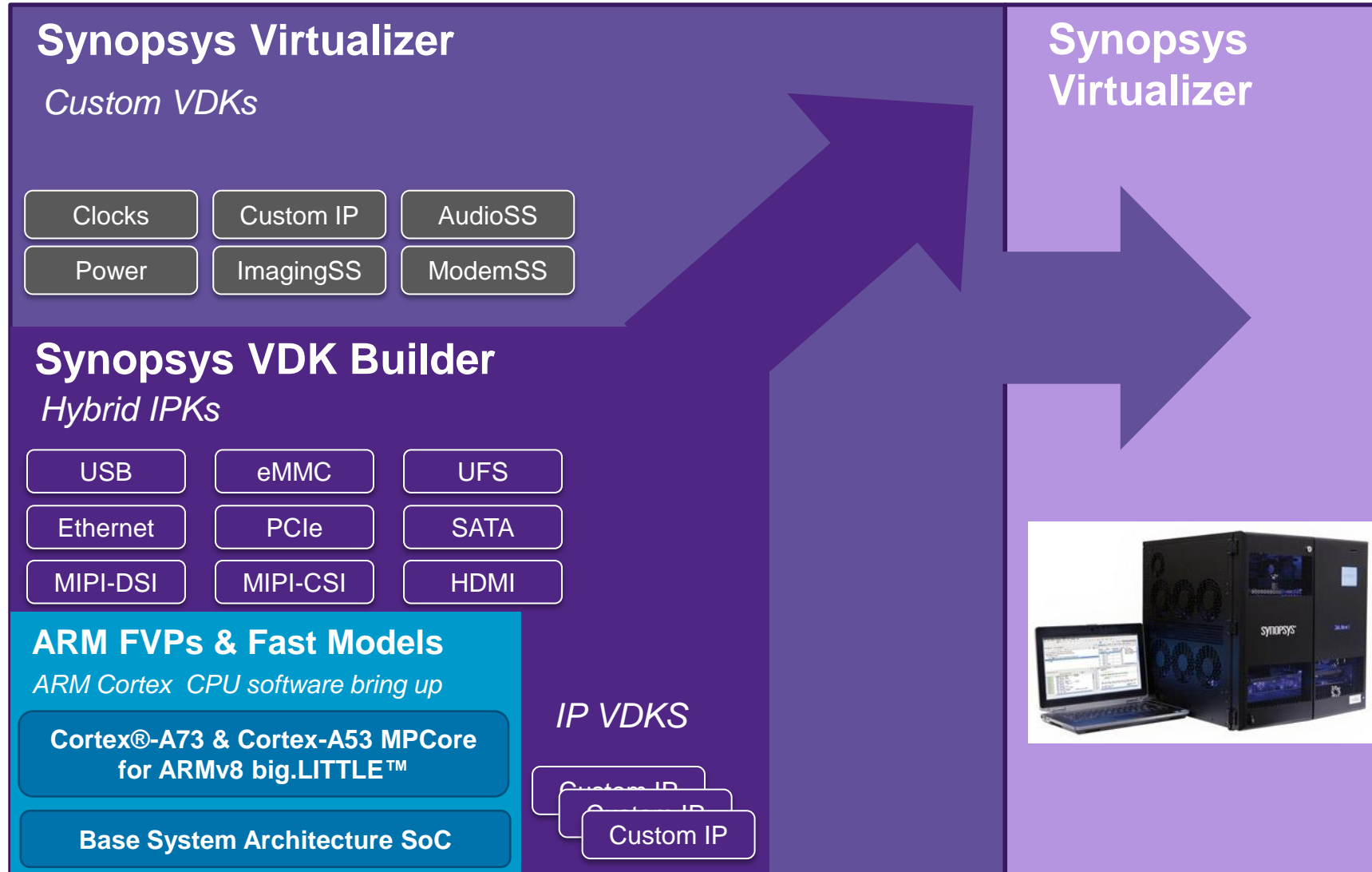
Full Platform Debug
Tracing and correlation of HW/SW execution

Support for Extended Use-Case

- Virtual- and Real-world IO
- Hybrid emulation and prototyping ...

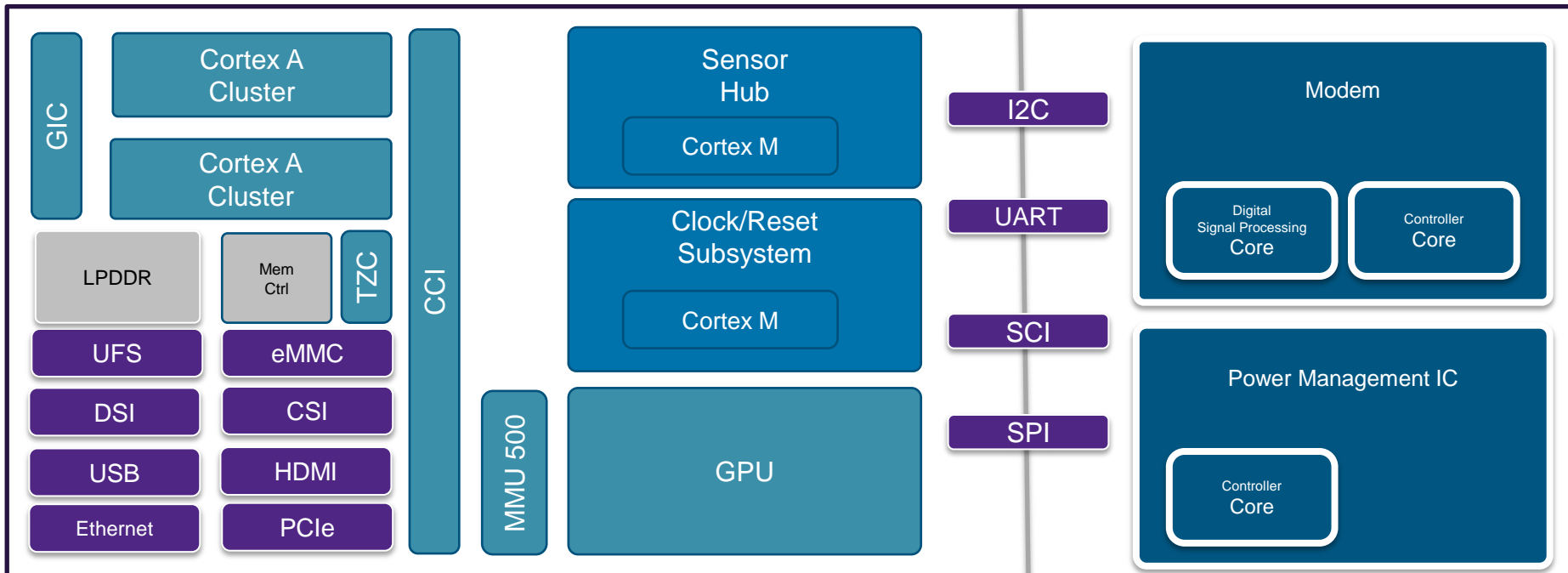
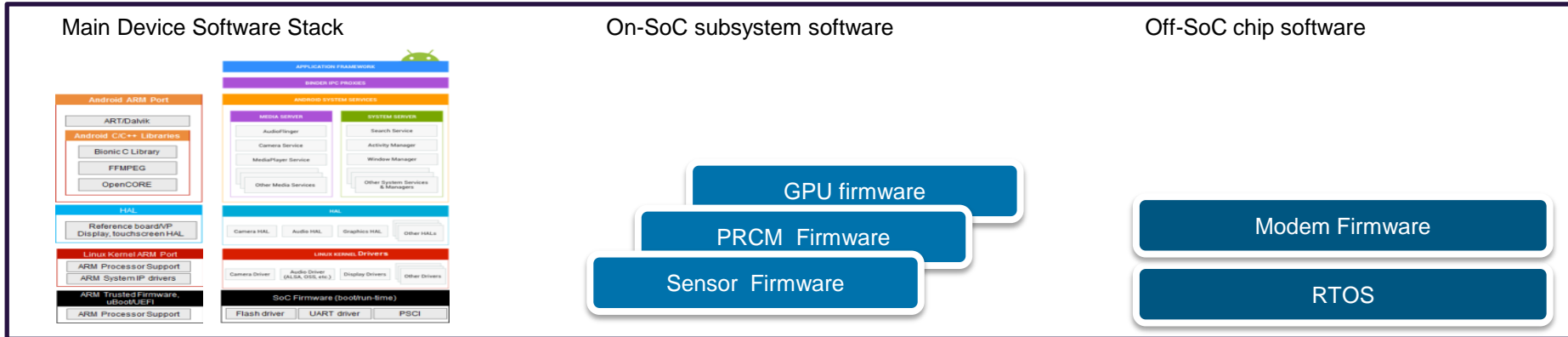
SoC SW development beyond the CPU

Software Binary Compatible with the ARM Base Platform



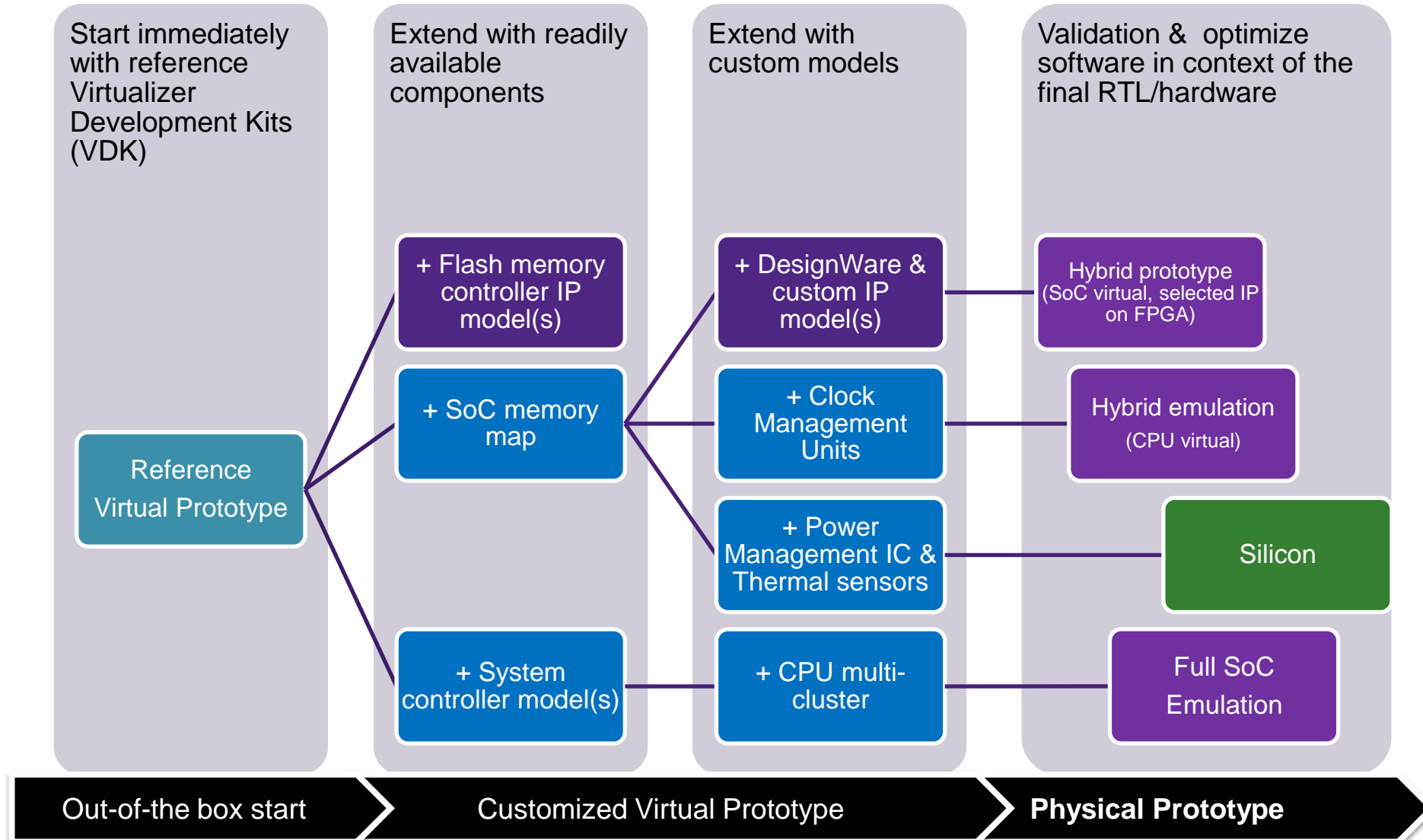
Broader Hardware & Software Scope

Mobile Application Processor SoC



Incremental VDKs → Immediate Impact

Steps to software readiness



Power Management SW Bring-Up Using VDKs

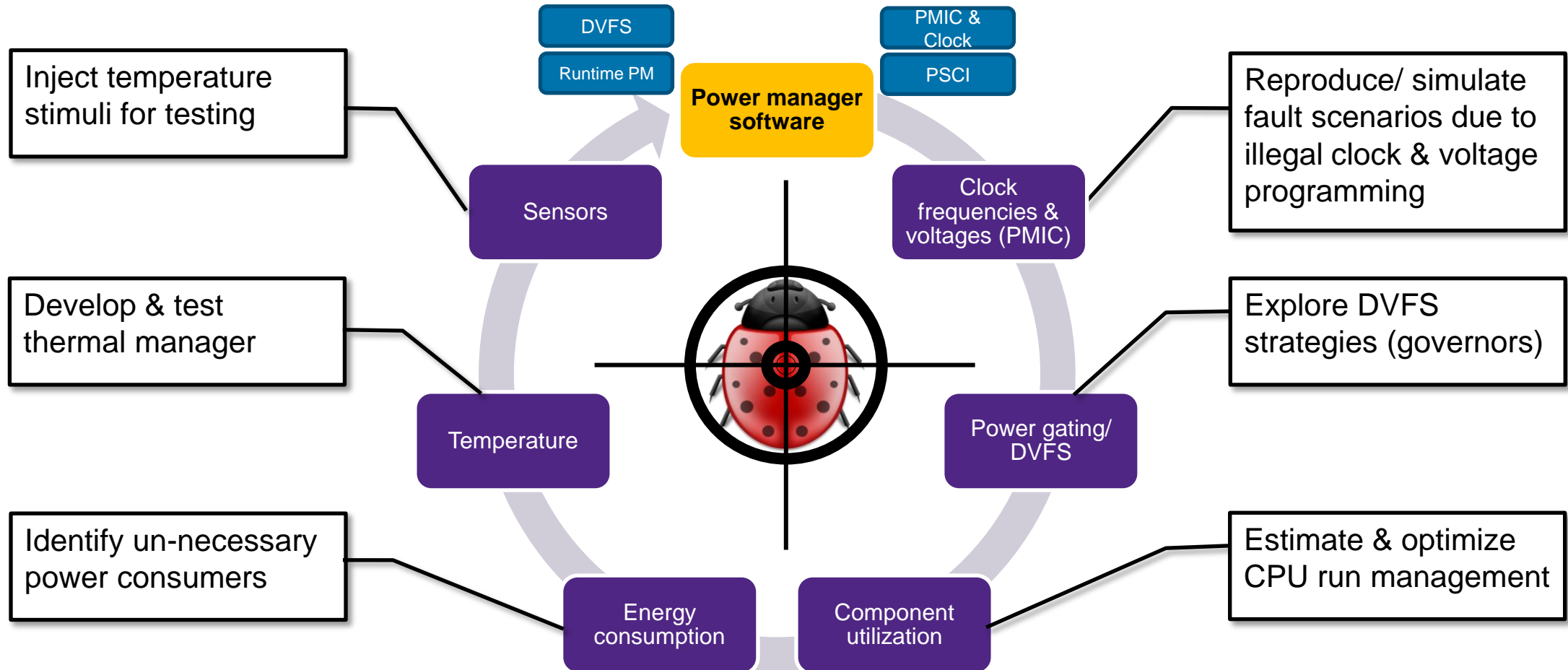
Case Study

- Accurate Implementation and Verification for Low power control
 - Driver functionality was confirmed to be same as target specifications
- Connection between Linux and power management system
 - Successfully developed whole functionality before final HW RTL or silicon availability
- Debug Power management system firmware



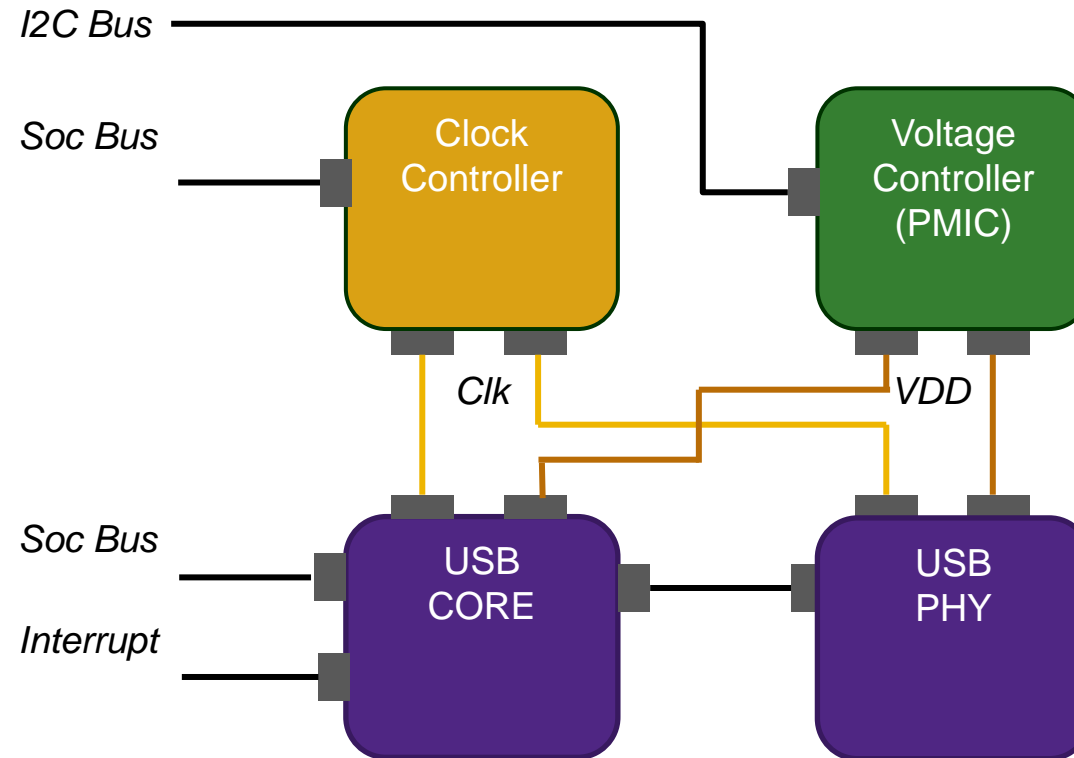
Software Power Management Bring-Up

Use-cases



Tiny Case Study: SoC Power Management

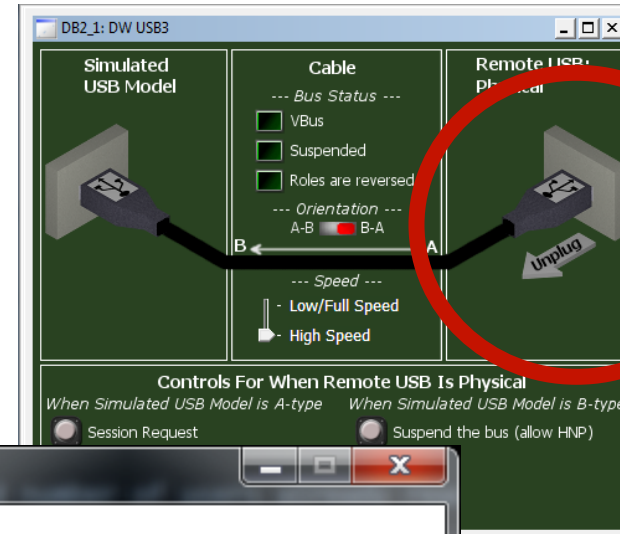
USB subsystem with your specific PMIC and Clock Controller



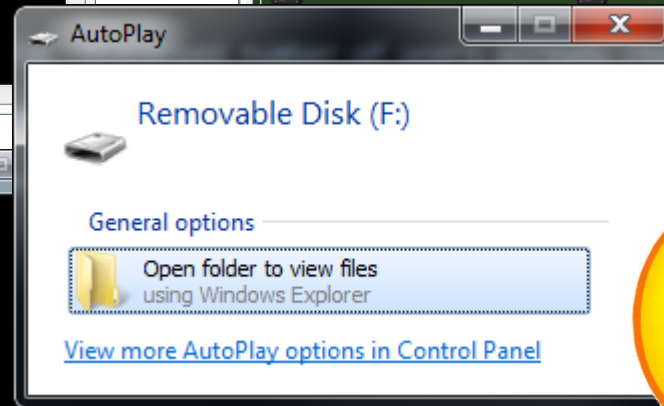
Case Study: Normal OS & Driver Operation

Booting and using USB for a file storage gadget

```
VDK_uATX_CortexA57x1_usb_pm - Linux Console (HARDWARE:UART0)
SLUB: Genslabs=11, Hwalign=64, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
Hierarchical RCU implementation.
RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=1.
NR_IRQS:64 nr_irqs:64 0
GIC CPU mask not found - kernel will fail to boot.
GIC CPU mask not found - kernel will fail to boot.
Architected local timer running at 100.00MHz (phys).
Console: colour dummy device 80x25
Calibrating delay loop (skipped), value calculated using timer frequency.. 200.00 BogoMIPS (lpj=1000000)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 256
hw perfevents: enabled with arm/armv8-pmu3 PMU driver, 7 counters available
Brought up 1 CPU
SMP: Total of 1 processors activated (200.00 BogoMIPS).
devtmpfs: initialized
atomic64 test passed
regulator-dummy: no parameters
NET: Registered protocol family 16
vds0: 2 pages (1 code, 1 data) at base fffffffc0005d4000
hw-breakpoint: found 6 breakpoint and 4 watchpoint registers.
Serial: AMBA PL011 UART driver
1c090000.uart: ttyAMA0 at MMIO 0x1c090000 (irq = 37) is a PL011 rev1
console [ttyAMA0] enabled
1c0a0000.uart: ttyAMA1 at MMIO 0x1c0a0000 (irq = 38) is a PL011 rev1
1c0b0000.uart: ttyAMA2 at MMIO 0x1c0b0000 (irq = 39) is a PL011 rev1
1c0c0000.uart: ttyAMA3 at MMIO 0x1c0c0000 (irq = 40) is a PL011 rev1
```



```
VDK_uATX_CortexA57x1_usb_pm - Virtual AT Keyboard, PS/2 Mouse/Touchscreen and LCD Panel
Welcome to Synopsys VDK For ARM Cortex v8 Processors!
udk-armv8 login: root
# start_dwusb3_filestorage
Starting FileStorage USB Gadget (on /mnt/gadgetfs) ...
Done
#
```



Case Study: Driver Faults from Power Bug

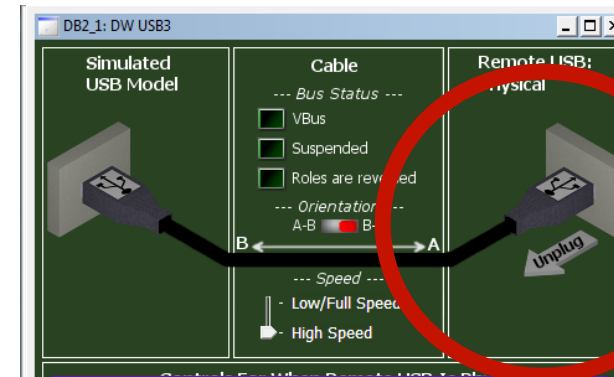
Booting and using USB for a file storage gadget

- Unpowered USB core

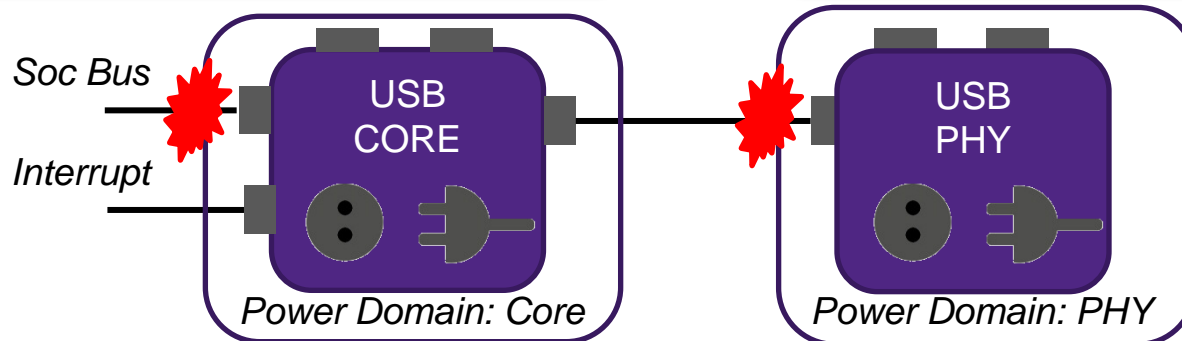
- Unpowered USB PHY

```
VDK_uATX_CortexA57x1_usb_pm - Linux Console (HARDWARE:UART0)
Unhandled fault: synchronous external abort (0x9600210) at 0xfffff8000040140
Internal error: : 9600210 [#1] SMP
Modules linked in:
CPU: 0 Not tainted (3.9.0 #4)
PC is at dwc3_probe+0x274/0xa50
LR is at dwc3_probe+0x254/0xa50
pc : [<fffffc0002fad2c>] lr : [<fffffc0002fad0c>] pstate: a000305
sp : fffffc00ac5fc50
x29: fffffc00ac5fc50 x28: 0000000000000000
x27: 0000000000000000 x26: fffffc00ad33210
x25: fffffc000588980 x24: fffffc0005b91f8
x23: fffffc000040100 x22: 0000000000000001
x21: fffffc00ff8b60 x20: fffffc009aad020
x19: fffffc000523078 x18: 000000000000000e
x17: 0000000000000007 x16: 0000000000000001
x15: 0000000000000007 x14: 0000000000003a3
x13: fffffc00ad9a000 x12: 000000000000300
```

Abort exception!



No response!



Case Study: Root Cause Analysis with VDK

Using a VDK, there is more to see!

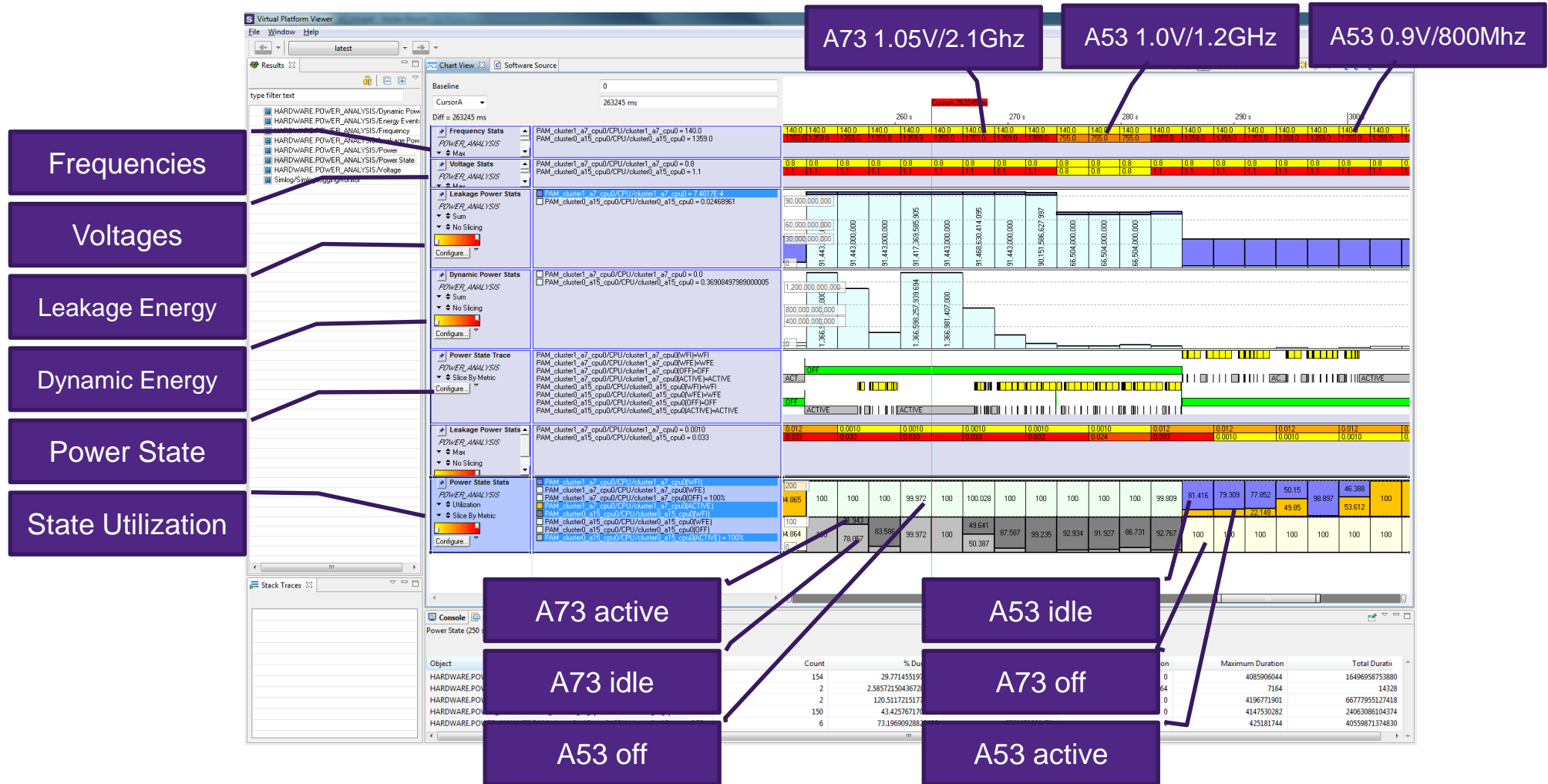
The screenshot displays the VDK interface with several key components and callouts:

- SystemC Processes:** Shows a hierarchy of hardware components including DB2.1, VDD_CORE_pin, ISOLATION, CORE, VDD, p_VDD, port_1, DB2.3, DCDC_pin, iMAX10550, and V3_DCDC.
- Status Trace:** Shows traces for PHY.mPDM_POWER_STATUS/StatusTrace, mPDM_POWER_STATUS, mPDM_CLK_VDD_STATUS, BUS_CORE_IN, CORE.mPDM_CLK_VDD_STATUS/StatusTrace, and ISOLATION.BUS_CORE_IN. A callout states: "USB PHY is powered on Ok".
- Terminal Trace:** Shows a console output with a callout: "Debug message: Exception fault!". The console text includes:

```
Time (ps) Text
4039471390000 Unhandled fault: synchronous external abort (0x96000210) at 0xffff8000040140
4039528220000 Internal error: 96000210 [#1] SMP
4039571900000 Modules linked in:
4039596930000 CPU: 0 Not tainted (3.9.0 #8)
4039650050000 PC is at dwc3_probe+0x324/0xac0
4039701610000 LR is at dwc3_probe+0x304/0xac0
4039746540000 pc : [<ffffffc0002fadbc>] lr : [<ffffffc0002fadbc>] pstate: a0000305
4039789540000 sp : fffffffc00ac5fc50
```
- Power State:** A callout asks: "USB CORE in Power Down State! Why?". The trace shows "POWER_DOWN" and "POWER_DOWN (Pending)" events.
- Voltages:** A callout states: "PMIC drives voltages: V4 and V6".
- Access:** A callout states: "At that time: Access to USB CORE".
- Control:** A callout states: "PMIC controlled by SW: OK".
- Connectivity:** A callout states: "VDD connectivity: USB CORE connected to V3 and not V4!".
- Correction:** A callout states: "Need to correct USB driver to driver V3 regulator!".

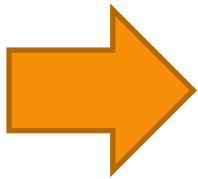
Dynamic Power Analysis: DVFS Support for ARM Cortex

Frequency, Power, Performance and Workload analysis based on VDK



Agenda

- From Reference Software to Production
- Shift Left – with Virtual Prototype & Virtualizer Development Kits
- Case Study – Power Management SW Bring-Up using VDK
- Q&A



Thank You

