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

myNextGen
Application Processor

ARM™ Cortex®
Synopsys DesignWare®
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

Contributed by ARM/Linaro

Android ARM Port
- ART/Dalvik

Android C/C++ Libraries
- Bionic C Library
- FFMPEG
- OpenCORE

SoC Firmware (boot/run-time)
- Flash driver
- UART driver
- PSCI

Hardware manufactures software responsibility

Google & AOSP Contributors

Hardware Abstraction Layer (HAL)

Sources: https://source.android.com/devices/index.html
## Android System Architecture

**Layers, components and ownership**

### Contributed by ARM/Linaro

- **SoC Firmware (boot/run-time)**
  - Flash driver
  - UART driver
  - PSCI

- **Linux Kernel ARM Port**
  - ARM Processor Support
  - ARM System IP drivers

- **Android C/C++ Libraries**
  - Bionic C Library
  - FFMPEG
  - OpenCORE

- **Android ARM Port**
  - ART/Dalvik

### Android ARM Port

- **ARM Processor Support**
- **ARM System IP drivers**

### ARM Trusted Firmware, uBoot/UEFI

### ARM Processor Support

### Processor: ARM IP

### Integration, optimization & test

- **Android HAL development**
- **Linux kernel SoC bring up**
- **Linux IP driver bring up**
- **Firmware/boot-code development**

### Integration, optimization & test

**Sources:** [https://source.android.com/devices/index.html](https://source.android.com/devices/index.html)

**Hardware Abstraction Layer (HAL)**

- Audio
- Camera
- Bluetooth
- DRM
- Display
- Input
- Media
- Sensors
- TV

**Google & AOSP Contributors**

**Hardware manufactures software responsibility**
Steps to Software Readiness
From open source reference to final software

Reference
Reference system (Juno/ FVP)

Boot
Custom hardware specification

Linux
Custom hardware specification

Android
Custom hardware specification

Optimization
Custom hardware specification

- ARM boot firmware, Linaro Linux, and Google AOSP software references
- Boot firmware
- Basic single core Linux boot
- Runtime firmware (PSCI)
- SoC’s interface peripherals & subsystem drivers
- SoC clock-, voltage-and thermal IP drivers
- Multi-cluster, multi-core bring-up
- CPU power & performance management (CPUFreq)
- HAL-Driven integration
- Full leverage of HW capabilities
- Overall power management strategy
- Overall (CPU/FGPU) performance tuning

Silicon validation software

Shift-Left!
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
Qualcomm Shifts Left with Virtualizer and VDKs

SNUG Silicon Valley 2016

Virtual Platforms for Large Scale Pre-Silicon Software Development

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

March 31, 2016
Santa Clara Convention Center

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

Download presentation
Early SW Bring-up & Power Analysis with Virtual Prototype

SNUG Shanghai 2016
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)

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!

Watch & modify each component’s registers and even interrupt signals!
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 …)

Virtualizer Studio

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

Synopsys Confidential Information
Efficient Solution for VDK Use
Synopsys Virtualizer™ Prototyping Solution

Embeded SW Debuggers

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

Synopsys Virtualizer
Custom VDKs
- Clocks
- Power
- Custom IP
- AudioSS
- ImagingSS
- ModemSS

Synopsys VDK Builder
Hybrid IPKs
- USB
- eMMC
- UFS
- Ethernet
- PCIe
- SATA
- MIPI-DSI
- MIPI-CSI
- HDMI

ARM FVPs & Fast Models
ARM Cortex CPU software bring up
- Cortex®-A73 & Cortex-A53 MPCore for ARMv8 big.LITTLE™
- Base System Architecture SoC

IP VDKS
- Custom IP

© 2017 Synopsys, Inc.
Broader Hardware & Software Scope

Mobile Application Processor SoC

Main Device Software Stack

On-SoC subsystem software

Off-SoC chip software

- Cortex A Cluster
- Cortex M Cluster
- Sensor Hub
- Clock/Reset Subsystem
- GPU
- Mem Ctrl
- LPDDR
- UFS
- DSI
- USB
- Ethernet
- PCIe
- eMMC
- HDMI
- SPI
- SCI
- UART
- I2C
- Digital Signal Processing Core
- Controller Core
- Power Management IC
- Modem
- Modem Firmware
- RTOS
- GPU firmware
- PRCM Firmware
- Sensor Firmware
- RTOS
Incremental VDKs ➔ Immediate Impact

Steps to software readiness

Start immediately with reference Virtualizer Development Kits (VDK)

- Reference Virtual Prototype
  - Out-of-the-box start

Extend with readily available components

- Extend with custom models
  - Customized Virtual Prototype
    - System controller model(s)
    - Flash memory controller IP model(s)
    - SoC memory map
    - Power Management IC & Thermal sensors
    - Clock Management Units
    - CPU multi-cluster

Validation & optimize software in context of the final RTL/hardware

- Physical Prototype
  - Hybrid prototype (SoC virtual, selected IP on FPGA)
  - Hybrid emulation (CPU virtual)
  - Full SoC Emulation
  - Silicon

- Validation & optimize software in context of the final RTL/hardware
  - Flash memory controller IP model(s)
  - SoC memory map
  - Power Management IC & Thermal sensors
  - Clock Management Units
  - CPU multi-cluster
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*

- Inject temperature stimuli for testing
- Develop & test thermal manager
- Identify un-necessary power consumers
- Reproduce/ simulate fault scenarios due to illegal clock & voltage programming
- Explore DVFS strategies (governors)
- Estimate & optimize CPU run management

**Power manager software**

- Sensors
- Temperature
- Energy consumption
- Component utilization
- Clock frequencies & voltages (PMIC)
- Power gating/DVFS
- PMIC & Clock
- Runtime PM
- DVFS
- PSCI
- Explore DVFS strategies (governors)
- Estimate & optimize CPU run management

© 2017 Synopsys, Inc.
Tiny Case Study: SoC Power Management

USB subsystem with your specific PMIC and Clock Controller

- DesignWare I2C host
- VDK for ARM & DesignWare (Base)
- Clock Controller
- Voltage Controller (PMIC)
- USB CORE
- USB PHY
- Interrupt
- Clk
- VDD
- I2C Bus
- Soc Bus
Case Study: Normal OS & Driver Operation
Booting and using USB for a file storage gadget
Case Study: Driver Faults from Power Bug

Booting and using USB for a file storage gadget

• Unpowered USB core
• Unpowered USB PHY

Abort exception!

No response!
Case Study: Root Cause Analysis with VDK

Using a VDK, there is more to see!

- USB PHY is powered on Ok
- USB CORE in Power Down State! Why?
- PMIC drives voltages: V4 and V6
- PMIC controlled by SW: OK
- Debug message: Exception fault!
- VDD connectivity: USB CORE connected to V3 and not V4!
- 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

- **Frequencies**
  - A73 active: 1.05V/2.1Ghz
  - A53 active: 1.0V/1.2GHz
  - A53 off: 0.9V/800Mhz

- **Voltages**
  - A73 active: 1.05V
  - A53 active: 1.0V
  - A53 off: 0.9V

- **Leakage Energy**
  - A73 idle: 800Mhz

- **Dynamic Energy**
  - A73 idle: 5.05V/2.1Ghz

- **Power State**
  - A53 idle: 1.0V/1.2GHz

- **State Utilization**
  - A73 active
  - A73 idle
  - A53 on
  - A53 off
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