Reducing Time to Point of Interest with Accelerated OS Boot

Frank Schirrmeister, Cadence®
Robert Kaye, ARM®
Agenda

- Design Challenge

- The Hybrid Emulation – Virtual Approach
  - Enabling technology in Palladium XP and Virtual System Platform
  - Fast Models from ARM

- ARM Example

- Industry Examples and Outlook
  - NVIDIA, CSR
  - Improvement considerations
Design Challenge
Example SoC and System

System on Chip (SOC)
An Example Project Timeline

Spec to GDSII: 49 - 83 wks
- 8-12 wks

Design & Integration & Verification: 35 – 63 wks
- 14 wks

Netlist to GDSII: 21 - 32 wks
- 11 - 17 wks

RTL-Becomes stable

Only small gate level changes and ECO's

Production

Post silicon Validation

Source: Cadence, IBS
Timeline for System Critical Bugs

- **SoC in System**
  - System Environment
  - **Time for critical bugs in System Environment to be removed**
  - Idea to spec
  - RTL
  - RTL Becomes stable
  - Bug rate
  - Only small
  - gate level changes
  - and ECO's
  - Production
  - Post silicon Validation

<table>
<thead>
<tr>
<th>Spec</th>
<th>RTL-Design &amp; IP Integration &amp; Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Qualification</td>
<td>Netlist to GDSII</td>
</tr>
<tr>
<td>Fab</td>
<td>Post Si</td>
</tr>
</tbody>
</table>

Source: Cadence, IBS
Software is Key to verification

Applications (Basic to Angry Birds)
Middleware (Graphics, Audio)
OS & Drivers (Linux, Android)
Bare metal SW
SoC in System
System on Chip
Sub-System
IP

- Idea to spec
- RTL becomes stable
- Bug rate
- Only small gate level changes and ECO’s
- Production
- Post silicon Validation
- Spec
- RTL-Design & IP Integration & Verification
- IP Qualification
- Netlist to GDSII
- Fab
- Post Si

More and more SW becoming requirement for tape out
May hold final tape out if bug too critical

Source: Cadence, IBS
Platforms - There is No ‘One-Fits-All’

- **SDK OS Sim**
  - Highest speed
  - Earliest in the flow
  - Ignore hardware

- **Virtual Platform**
  - Almost at speed
  - Less accurate (or slower)
  - Before RTL
  - Great to debug (but less detail)
  - Easy replication

- **RTL Simulation**
  - KHz range
  - Accurate
  - Excellent HW debug
  - Little SW execution

- **Acceleration Emulation**
  - MHz Range
  - RTL accurate
  - After RTL is available
  - Good to debug with full detail
  - Expensive to replicate

- **FPGA Prototype**
  - 10’s of MHz
  - RTL accurate
  - After stable RTL is available
  - OK to debug
  - More expensive than software to replicate

- **Prototyping Board**
  - Real time speed
  - Fully accurate
  - Post Silicon
  - Difficult to debug
  - Sometimes hard to replicate
Choosing the Right Engine

- Chip
- FPGA based Prototyping
- Acceleration Emulation
- RTL Simulation
- Virtual Prototyping

SW Development

HW/SW Validation

HW/SW Verification

HW Verification

SW Development

Early Software Development

Software Debug

System Speed

Hardware Accuracy

Hardware Debug & Turn-around-time

CONCEPT

Software

PRODUCT

TLM Sim

RTL Sim

A&E

FPGA

SW Development

Hardware

Software

Hardware

PRODUCT

+ - + - + - + - + - + - + - + - + - + - + -
HW/SW Concurrency Gap

Next Generation SW-Driven SoC Flow
- Continuous SW Development & Bringup
- Continuous System Validation
- HW Development & Verification

SW-Enhanced SoC Flow
- SW Dev On model
- SW Dev and Bringup On real HW design, Silicon
- System Validation
- HW Development & Verification

Traditional SoC then SW Flow
- HW Development & Verification
- SW Dev and Bringup on Silicon
- System Validation

Powered By
Platform Hybrids
Emulation + Virtual Platform + FPGA

Enabled By
Virtual Platform
FPGA Prototype
Emulation

Legend
- SW
- System
- HW

Tapeout | Silicon Samples | Product Ships
Early SW Execution on Palladium

TLM Virtual Platform – VSP
- Up to 100MHz
- Early Availability for SW Developers
- Advanced SW Debug
- Fast SW Turnaround Time

Emulation – Palladium® XPI/II
- Up to 4MHz
- From early-RTL to full-SoC Validation
- Advanced HW Debug
- Fast HW Turnaround Time

Hybrid Solution with SW Integrator
- Boot Complex OS at 48MHz
- Speed UP SW-Driven tests 1-10X over emulation
- Early Availability for SW Developers
- Advanced HW + SW Debug
- Fast HW and SW Turnaround Time
Palladium/VSP Hybrid Solution

Architected for SW Performance
- High-speed virtual platform
- Asynchronous HW/SW Execution with Interrupt driven sync
- High-Speed Multi-Domain Memory Coherency

Designed to integrate HW and SW flows
- Does not require changes to HW or SW stacks
- Virtual connections into SW Engineer’s environments
- Seamless hybrid execution for both HW and SW users

Proven Methodology, Unique Expertise
- Cross-platform and design integration expertise
- Exclusive hybrid methodology delivers performance and repeatability
- Proven during successful application to SW-rich SoCs
Hybrid Example

Execute SW at 100MHz
With standard or custom processor models

Plug and Play Integration with RTL
SoC-specific transactors and RTL I/F

Validate SoC + OS at 5-10 MHz on PXP
High-performance memory coherency

Shorten SoC Debug
System Messages
HW / SW Debuggers

Component Color Key

<table>
<thead>
<tr>
<th>Component</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>RTL</td>
</tr>
<tr>
<td>RTL</td>
<td></td>
</tr>
<tr>
<td>TLM</td>
<td></td>
</tr>
<tr>
<td>Mem I/F</td>
<td></td>
</tr>
</tbody>
</table>

Palladium® XP
## Hybrid Performance
Compared to an All-RTL in Emulation Configuration

<table>
<thead>
<tr>
<th>Metric</th>
<th>All RTL in Palladium*</th>
<th>Hybrid**</th>
<th>Increase</th>
<th>Effective SW exec. speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux boot (minutes)</td>
<td>30</td>
<td>0.5</td>
<td>60X</td>
<td>48MHz</td>
</tr>
<tr>
<td>Android boot (minutes)</td>
<td>900</td>
<td>15</td>
<td>60X</td>
<td>48MHz</td>
</tr>
<tr>
<td>Windows RT boot (min.)</td>
<td>1800</td>
<td>30</td>
<td>60X</td>
<td>48MHz</td>
</tr>
<tr>
<td>512x512 2D test (min)***</td>
<td>30</td>
<td>2</td>
<td>15X</td>
<td>11MHz</td>
</tr>
<tr>
<td># Emulation gates used</td>
<td>70 Million</td>
<td>40 Million</td>
<td>0.6X</td>
<td></td>
</tr>
</tbody>
</table>

* 70 million gate application processor, all blocks in Palladium®
** Virtualized CPU sub-system with register model of L1 & L2 caches. All other SoC blocks in Palladium.
*** Includes Linux boot, data preparation, image processing by HW engine and result checking. 1.3 million memory transactions. All boot numbers are full production images. Linux includes all drivers. Android and Win RT with SW rendering

### Target Application
- Large, compute intensive SoCs

### Accuracy
- SW: Delivers programmers-view accuracy
- HW: Full accuracy except for timing between virtual CPU and SoC fabric
- Memory: in fast mode, memory transactions are performed back-door. Thus, hybrid models not recommended for power or performance estimation

### Target Users:
- HW-Dependent SW engineers,
- System validation engineers
Fast Models from ARM
SoC Simulation Views

- **Programmer’s View**
  - SW Development
  - SW Profiling
  - Architecture Compliance

- **System Performance View**
  - High-level performance analysis

- **Component Performance View**
  - Architecture exploration
  - Performance analysis
  - Benchmarking

- **System Validation View**
  - HW/SW Co-Verification

- **Component Validation View**
  - HW Validation
  - Driver Development
  - HW/SW Co-Verification

- **Simulation speed**
  - Fast
  - 50-200 MIPS
  - Approximately Timed (AT)

- **Simulation speed**
  - Slow
  - 1-20 KIPS
  - Cycle Accurate (CA)

- **Abstraction level**
  - Detailed
  - 1-20 KIPS
  - Abstract
Model Requirements for SW Development

- **Simulation Speed**
  - Model must be fast enough for software developers

- **Model Fidelity**
  - Models must be complete and faithful to the target implementation

- **Solution Flexibility**
  - Enable broad range of applications with varied requirements
Fixed Virtual Platforms

- **Foundation Platform for ARMv8**
  - Simple, entry level platform for Linux application developers
  - Debug via GDB server

- **“ARM® Versatile™ Express” (VE) FVP**
  - Versatile Express memory map
  - Debugger and Trace API

- **“Base” FVP**
  - VE + power management, system control

- **Software alignment with ARM, Linaro**

---

**Fixed Virtual Platform**

<table>
<thead>
<tr>
<th>ARM® Cortex®-A57 Fast Model</th>
<th>ARM® Cortex®-A53 Fast Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIC Fast Model</td>
<td>SMMU Fast Model</td>
</tr>
<tr>
<td>CCI/CCN Fast Model</td>
<td>Peripheral model</td>
</tr>
<tr>
<td>Peripheral model</td>
<td>Peripheral model</td>
</tr>
</tbody>
</table>

**Virtual I/O**

I/O Accesses
Virtual Platforms Based on Fast Models

- **ARM Fast Models**
  - De-facto standard for virtual prototyping of ARM-based SoCs
  - Fast, Flexible, Fidelity
  - Accurate software representation of the CPUs and fabric IP
  - Open APIs to software debuggers and EDA tool
Hybrid Virtual Platforms

- ARM Fast Models
  - Open APIs to software debuggers and EDA tools
  - Standardized TLM 2.0 bridges for AMBA
  - Mix and match models at different levels of abstraction
  - Concurrent debug of hardware and software
Use Case 1: No PV Model Available

- Graphics & Video difficult to model in PV
- ARM® Mali™ cores on Emulator
- Compute Sub-System on ARM Fast Models
Use Case 2: Component Analysis

- Analyse traffic in CCI-400
- CCI-400 on emulator
- Upstream components in ARM Fast Models
- Downstream in Fast Models or stubbed
Use Case 3: Verification speed-up

- Use Fast Processor Models to speed simulation
- Complete System in Emulator other than CPU
Typical Virtual Platform Flow

- Start with example Fast Model project or FVP
  - Sufficient for OS boot and initial software development

- Develop virtual platform features to support software program
  - Incrementally build out platform capabilities
  - Easy deployment as new features completed

- Export Fast Model subsystem to SystemC to extend platform and integrate with commercial EDA/ESL tools
  - e.g. VSP, Palladium

- Majority of software features and integration completed ahead of silicon
  - Fast platform execution – software teams efficient
  - Advanced debug and visibility capabilities

- Final optimisation and tuning against silicon
  - Power/performance tuning often the final stages of the project
ARM Example
Hybrid Palladium / VSP System

- ARM SysBench RTL instance as provided
- Able to boot Linux and run GPU benchmarks on Palladium
Hybrid Palladium / VSP System (AQUA)

- Virtual-only system
- Able to boot Linux image to prompt
- No specialized IP or DUT
- May comprise virtualized components such as virtual Ethernet, SD card, USB
- Full visibility and debuggability
Hybrid Palladium / VSP System

Hybridization of Sysbench RTL instance
- Wrappers for CPU and GIC
- Replace DDR memory
Hybrid Palladium / VSP System

- Full hybrid system
  - AMBA® ACE bridge connects virtual core to RTL
  - Interrupts are forwarded to virtual GIC
  - Smart memory provides coherent memory between RTL and VSP

- Via mapping on the virtual side, Peripherals and IP in the RTL can be shadowed
What was Measured?

- Wall clock time for:
  - Linux boot
  - Driver loading & setup
  - Benchmark execution for 1st frame
  - Benchmark execution for all frames

Platform: Palladium only, Palladium/VSP Hybrid (6 domains)
- Measured with frame dumping active and inactive
Results: 50X speedup in Linux Boot
Industry Examples
PALLADIUM/VSP ARM V8 TEGRA HYBRID PROJECT #2

- Pre-silicon Android Validation
- Open GL Graphics Testing

Vikramjeet Singh
Sr System SW Manager
Mobile Devices
Pre-silicon goals for SW

- Co-develop and Co-verify pre-silicon HW designs

- High quality SW on silicon arrival
  - No silly bugs
  - Focus on performance and power optimizations
  - Cut down time to production

- Bring-up production software stack
  - Android, OpenGL...
  - Eliminate interface bugs
  - Early visibility into “eco-system” readiness
NVIDIA Validation with Palladium/VSP Hybrid

- Directed HW Arch Tests
- Directed Kernel Tests
- Kernel Use cases
- NVIDIA OGL Tests
- OS boot end to end
- OS use cases
**OS Validation**

- **Staged 64b’ development**
  - 64b’ Kernel with 32b’ User space
  - Fix interface issues (no silly bugs)
  - Demo capabilities to partners

- **Boot Times**
  - Kernel = 2 mins
  - Android = 90 mins
  - 10x faster than PD
SW Validation Results

- Eliminated reliance on other pre-silicon platforms

- SW problems found prior to Silicon return
  - SW race conditions
  - Code completeness
  - 64b’ <> 32b’ interface bugs

- After silicon return
  - Contributed to smoother bring-up
  - SW Ready to demo product at SOL
  - Less bugs resulted in focused effort to tune perf/w
Using Palladium-VSP Hybrid to Accelerate SW Development

Moshe Berkovich
June-2014
Challenges in pre-silicon SW development

• Complex SoCs need massive SW development
• Meeting TTM relies on stable working SW
• HW/SW Co-sim is critical for verification

SW development requirements:
1. High frequency platform
2. Debug capabilities
   - JTAG/UART interfaces for SW debug
   - Full waveform visibility
   - Activity logs (like ARM’s tarmac)
3. Fast bring-up
4. Quick turnaround cycle
5. Accuracy
Hybrid Platform Exploration

Minimize DDR access penalty

TLM units often accesses by CPU
Results

- Runtime comparisons
  - Palladium compiled at 1.5Mhz, CAKE 1X

**HYBRID IS EXCELLENT FOR CPU-CENTRIC DESIGNS**
Performance Results

- Boot OSes, run real world applications and benchmarks
  - Linux kernel boot
    - Palladium only = 45 mins
    - Hybrid = 2 mins
  - Android
    - Palladium only = Hours
    - Hybrid = 40 - 50 mins
  - Windows
    - Palladium only = Days
    - Hybrid = 75 - 90 mins

OS Validation

- Staged 64b development
  - 64b Kernel with 32b User space
  - Fix interface issues (no silly bugs)
  - Demo capabilities to partners

- Boot Times
  - Kernel = 2 mins
  - 40x faster than PD
  - Android = 90 mins
  - 10x faster than PD

SW Validation Results

- Eliminated reliance on other pre-silicon platforms

- SW problems found prior to Silicon return
  - SW race conditions
  - Memory management bugs
  - Code completeness

- After silicon return
  - Contributed to smoother bring-up
  - SW Ready to demo product at SOL
  - Less bugs resulted in focused effort to tune for power and perf

Results

- Runtime comparisons
  - Palladium compiled at 1.5Mhz, CAKE 1X

**HYBRID IS EXCELLENT FOR CPU-CENTRIC DESIGNS**
Android & AnTuTu on PXP

- Multiple customers booting Android on PXP
  - Brought up OS before tapeout
  - Validated SW with Full SoC, including GPU rendered desktop
  - Ran test applications

- Observed Android boot times
  - In-Circuit configuration: 13 hours
  - Hybrid configuration: 1 hour

- Several customers have run AnTuTu on PXP
  - Characterized SOC performance
  - Optimized SW stack
  - AnTuTu run time: 24 hours in ICE configuration
Outlook
Outlook: Enhancements

- Further Virtualization
  - PCI-e for Hybrid configurations

- Hybrid Assembly
  - Hybrid Model Library
  - Graphical User Interface

- Smart Memory
  - User configuration
  - Cache Coherency

- Embedded Software Debug

- Record/Playback at Virtual/RTL boundary
  - To support replay on standalone virtual platform
  - Swap between RTL and Virtual