Optimizing Hardware/Software Development for Arm-Based Embedded Designs

David Zhang / Cadence

Zheng Zhang / Arm

#Arm Tech Symposia

Copyright © 2018 Arm, All rights reserved.
Agenda

Application challenges in ML/AI and 5G

Engines for system development and verification
  • Core engines and their key characteristics

Modeling and debug needs
  • Types of models and their characteristics
  • Debug solutions

Putting it all together
  • A couple of examples at different abstraction layers
Application Challenges in Machine Learning/Artificial Intelligence and 5G
The Internet of a “Trillion Things”

“SoftBank CEO Masayoshi Son sees a future with 1 trillion Internet of Things devices”

VB news

Image source: http://bit.ly/2lDtTEY, licensed for re-use with modification
5G

Unique challenges

Diverse requirements

- New devices (fog, mist)
- New applications (video, security)

Key Challenges

- Significant software content
- Big designs – emulation throughput
- Architectural analysis
- Physical and virtual interfaces
- Specific I/O – TCAM
- Interfaces to 5G testers, ecosystem
- Virtualization – Cloud
Machine Learning and Artificial Intelligence

Unique challenges

Diverse requirements

• Training – High throughput, big designs
• Inference – Flexibility
• Specific IP

Key Challenges

• Significant software content
• Big Designs – Emulation throughput, debug
• Physical and virtual interfaces
• Specific I/O – HBM
• Virtualization
Example Designs

Different complexities in things, hubs, servers, and network

**Generic Designs**

- **Software Stack**
  - Processor Subsystem
  - Customer’s Application-Specific Components
  - High-Speed, Wired Interface Peripherals
  - General-Purpose Peripherals
  - Low-Speed Peripherals

- **IP for CPU**
  - Arm Cortex®-A, M, R, GP, and interconnect
  - Customer: specific blocks
  - Complexity: Medium to high
  - Differentiation in hardware and software

- **Ultra-low power**
  - Software: Arm Mbed™ OS
  - Complexity: Small
  - Differentiation in implementation, AMS, eSW

**IoT Edge Node**

- **IoT lower-level SW stack**
  - Cortex-M
  - IoT Subsystem
  - Interconnect
  - Generic I/O

**Mobile**

- **Power**
  - Performance
  - Thermal
  - Complexity: High
  - Differentiation in software and power, accelerators (GPUs)

**Network / Infrastructure**

- **Performance, power**
  - Workload optimization
  - Complexity: High
  - Differentiation in software and specific IP

#Arm Tech Symposia

Copyright © 2018 Arm, All rights reserved.
Engines for System Development and Verification
Dynamic Engines in the Design Flow

There is no “one size 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 hardware debug
  - Little software execution

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

- **FPGA Prototype**
  - 10s 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

Image sources: Cadence
Formal Trends
Ease of use, capacity, and performance

#Arm Tech Symposia
Copyright © 2018 Arm, All rights reserved.
Simulation Trends

A new era of parallel simulation

- Interpreted simulators (e.g., Verilog-XL)
- Compiled simulators (e.g., Incisive®)

Performance and scalability via compilers
Performance and scalability via multi-core

Use cases

Hardware Trends

Emulation, FPGA, common compile flow - Congruency

**Palladium® Z1**
- Advanced Debug
- SoC acceleration, hardware/software
- Power and performance analysis

**Protium™ S1**
- Performance
- Software development
- Hardware/software regressions

**Congruency and common environment**

Optional for Debug
Providing Access as Early as Possible using Models ...

- **Ever-increasing verification** requirements driven by growing hardware and software **complexity**
- **Fast time to results** is essential to ensure projects can **meet schedules**
- **Right tools for the right job**: Combination of formal, simulation, emulation, and FPGA prototyping
Example: Cadence Verification Suite

Core engines, fabric, and flows

- Strong multi-core engines
- Flow-driven engine integrations
- Differentiated and comprehensive solutions
Complexity drives design at different abstraction levels

Can there be one executable specification?

- Complex HW/SW system
- During development, often not all components are available at the same level of abstraction
  - RTL
  - TLM
  - Hard IP
  - Actual silicon
- Variable-fidelity integration is required
Hardware/Software Debug Is One of the Biggest Challenges

Software, hardware, and system...

Image sources: Cadence, Arm
Modeling and Debug Needs
Parallelize Hardware and Software Development

“Better software, faster”

- Time savings of a year or more reported by partners
- Complete software stacks running on silicon within days of delivery
SoC Simulation Views

System validation view
- HW/SW co-verification

Component validation view
- HW validation
- Driver development
- HW/SW co-verification

Programmer’s view
- Software development
- Software profiling
- Software driven verification
- Architecture compliance

Cycle Accurate (CA)
1-20 KIPS

Loosely timed
50-200 MIPS
Arm Addresses the Virtual Model Landscape

- Arm Cycle Models
- Fast Models with Timing Annotation
- Hybrid with Emulation
- Pure Virtual Hybrid (FM + CM)
- Arm Fast Models

#Arm Tech Symposia

Copyright © 2018 Arm, All rights reserved.
The Value of Cycle-Accurate Simulation

- Choose the right IP to meet design requirements
- Easily understand the behaviour and performance of the IP
- Enable development of accurate firmware and drivers
- Debug complex hardware/software issues in advance of silicon

IP Selection & Optimization

- Easily try different IP configurations and software loads
- Generate benchmarking data for design decisions and customer wins
- Understand the impact of hardware and software optimizations
- Ensure that system performance requirements are met

System Architecture

Software Development

Performance Analysis
Arm Cycle Models are compiled directly from RTL

- Retain 100% of functionality
- Cycle Model Studio enables customers to compile their own RTL
- Arm uses Cycle Model Studio to provide Cycle Models of most Arm IP
- Visibility options to enable system debug by hardware or software designers
Arm Fast Models

Fast, functionally complete models of all ARM IP

- 10s to 100s of MIPS speed to enable OS boots and software development
- Accuracy to run even the lowest level unmodified binaries
- Timing annotation to aid system optimization

Earliest availability models

- Architecture models represent ISA
- Developed in conjunction with RTL teams

Unmatched functionality

- Caches
- TrustZone® technology
- Virtualization
- Crypto
- Multi-core/multi-cluster

SystemC TLM interfaces to ensure compatibility with all environments
Fast Models of Arm IP

Wide selection of models and solutions

**IP Models**

Complete portfolio of current and future Arm CPU cores
- Includes support for DynamIQ™ technology
Includes models of most Arm SystemIP
- CCI, CCN, GIC, SMMU, etc
Models available for graphics and video cores as well
- Graphics adapter enables OpenGL rendering

**Fixed Virtual Prototypes**

Wide variety of prebuilt, fixed platforms
Form the basis for Arm’s System Guidance offerings
Starting point for further customization or deployment
Model Availability

Total IP Project Duration
- Architecture Exploration
- Performance Model
- IP Design
- Alpha
- Beta
- LAC
- EAC
- Models

Cycle Model Design
- Cycle Model releases

Fast Model Design
- Fast Model releases

Swap & Play
Tools Created by Experts in the Architecture

Quality tools you can depend upon

- Over 25 years in the tools business
- Used in the development of billions of Arm-based devices on the market
- Developed alongside Arm architecture and system components

Global reach

- Around-the-clock support by 12 expert Arm service centres
- 35+ qualified tools distributors worldwide
Support for All Stages of Product Development

Virtual prototype
CADI virtual debug interface

Prototype and Silicon Bring-up
High performance debug and trace

Catalog chipsets
Low cost debug for standard parts

Linux/Android devices
Application debug and analysis

Higher engineering efficiency achieved through single IDE for all stages in the product development
Arm Development Tools for all Arm Cores

Compiler
Best-in-class, safety certified code compilation designed alongside Arm IP

Multi-Core Debugger
Family of debug tools for silicon bring-up and software development for complex SoCs

Virtual prototypes for architectural exploration and early software development

Streamline
Providing visibility of software performance running in Linux or baremetal
Arm Compiler 6 for Functional Safety

Stable branch qualified for use in functional safety applications such as automotive, industrial, and railway

- No further toolchain qualification effort required for standards in TÜV SÜD certificate
- Complete documentation package to support certificates against most safety standards

### Qualification Kit
- Safety manual
- Defect report
- Test report
- Dev process docs

### Extended Maintenance
- Five year commitment
- Technical support
- Critical defect fixes

### Functional Safety Certified
- TÜV SÜD certification
- ISO 26262 (ASIL D)
- IEC 61508
- EN 50128
- IEC 62304*
Putting It All Together
Early, High-Performance Software Execution on Palladium Technology

**ARM Fast Models**
- Up to 100MHz
- Early availability for software developers
- Advanced software debug
- Fast software turnaround time

**Cadence® Palladium® XPI/II or Z1**
- Up to 4MHz
- From early-RTL to full-SoC validation
- Advanced hardware debug
- Fast hardware turnaround time

**Hybrid Values**
- Boot complex OS at up to 50MHz
- Speed up software-driven tests 1-10X over emulation
- Early availability for software developers
- Advanced hardware + software debug
- Fast hardware and software turnaround time
Hybrid Virtual Prototype

Virtual prototypes are great for software development

Combine the performance of virtual prototypes with the accuracy of emulation

- Leverage existing RTL models
- Model components where PV is the inappropriate abstraction level
- Component/system performance analysis
- Software driven verification
- Simulation acceleration

Protocol Transactors (ARM AMBA®)

Signal Transactors (interrupts, reset, power, etc.)

Memory Transactors (shared memory)

Virtual Prototype

ARM Cortex-A57 Model

ARM Cortex-A53 Model

GIC Model

SMMU Model

CCI/CCN Model

Peripheral model

Peripheral model

Debug and Simulation Control (CADI)

Model Trace Interface (MTI)

Virtual I/O

DS-5 Debug and Trace

I/O Accesses

Trace and Debug
Example Hybrid Implementation

With Virtual Platform + Palladium Turnaround Time and Debug
- Run Unmodified, Complex SW such as Android Boot, OGI, Tests at FPGA Speeds
- Add Smart Memory: Patented TLM / RTL Coherence

Fast Processor Model
- A53 x 4
- A57 x 2

SW Integrator
- Timers
- UARTs
- eMMC

Reconfigurable Interconnect (memory mapped)

SW Integrator RTL I/F (ARM TRM standard signals)
- Clocked, Resets
- ACE
- Interrupts

CPU Sub-system RTL I/F
- GIC400
- UARTs
- Timers

Peripheral Fabric
- Memory Controller
- Boot CPU
- Scratch RAM
- CSI
- DSI
- Ethernet Display
- USR2
- USB3
- SATA

Customer Design
- Use Most Accurate Design Model: RTL on Palladium Platform
- Connect Using Proven Technology, Methodology
- Add a Minimal Virtual Model of Your CPU Sub-system
Hybrid Solution – Best of Both Worlds

Software executes on Virtual System Platform (VSP) based on Arm Fast Models

Orders of magnitude faster than emulator

Early software development in parallel with RTL development

Efficient usage of emulation capacity

RTL for rest of SoC on Emulator

*Software Integrator* provides cross-domain memory coherency
NVIDIA: Palladium – Fast Models Hybrid Use Case

Traditional Emulation SW Validation

- Speed
  - Design runs at Palladium
  - SW test targets a few orders of magnitude faster
  - Linux > 1 billion
  - Android > 20x
  - Windows RT 5x
- OS boot would take several hours
- Functionality requires:
  - Reduce the cost of
  - Not a test platform
  - Standard drivers

Cadence Palladium Hybrid Solution

- Fast access to Palladium emulated memory
- Ability to add standard simulation models
- Full-chip environment SW verification
- Boot OSes, run real SW
- Linux kernel boot
- Palladium only = 2 mins
- Hybrid = 40 - 50 mins
- Android
  - Palladium only = 2 mins
  - Hybrid = 75 - 90 mins
- Windows
  - Palladium only = 2 mins
  - Hybrid = 100 - 110 mins

Performance Results

- SW Validation Results
  - Reduce 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
Cycle Model Integration with Cadence Interconnect Workbench

- No RTL access is required in order to perform analysis
- Same performance results as RTL without need to look at cycle level
- Direct integration
Summary: Arm Models and Cadence Verification

ARM Fast Models
- Palladium Hybrid for early OS and software bring-up

ARM Processor and Interconnect IP
- Perspec™ software-driven use case verification with ARMv8 library
- Palladium Dynamic Power Analysis profiling real traffic
- Indigo™ synchronized eSW debug on ARM RTL CPU, Protium w/ ARM DS-5 Debug
- SoC Workbench packaged ARMv8 IP and VIP for fast SoC integration
- Interconnect Workbench for SoC verification and performance analysis
- VIP Portfolio with ARM AMBA® ACE support on Xcelium, JasperGold, and Palladium solutions

ARM TARMAC Trace
- ARM DSTREAM + DSS

ARM Socrates™ tool
- ARM CoreLink™ Interconnect IP

ARM CoreLink™ Interconnect IP
- ARM CoreLink Interconnect IP

Cadence/ARMv8 System Verification
- Middleware
  - Operating Systems (OS)
  - Drivers
  - Firmware / HAL

- Verification Fabric
  - Formal & Static
  - Simulation
  - Emulation
  - FPGA Prototype

- VIP Portfolio
  - vManager
  - vManager Metrics
  - Indigo™ Debug
  - Perspec™ SW-Driven Test

- Application-Specific Components
- SoC Interconnect Fabric
- CPU Subsystem
  - A15
  - L2 cache
- Low-speed peripheral subsystem
- Low-speed Peripherals
- General-Purpose Peripherals
- High-Speed, Wired Interface Peripherals

Customer’s Application-Specific Components

Arm Subsystem (Mobile, Server)
- Arm Fast Models
- ARM Processor IP
- ARM TARMAC Trace
- ARM Socrates™ tool
- ARM CoreLink™ Interconnect IP
- ARM CoreLink™ Interconnect IP

#Arm Tech Symposia
Copyright © 2018 Arm, All rights reserved.
Trademark and copyright statement
The trademarks featured in this presentation are registered and/or unregistered trademarks of Cadence and Arm (or their subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

Copyright © 2018