High Performance or Cycle Accuracy?

You can have both!

Bill Neifert, Carbon Design Systems
Rob Kaye, ARM
ATC-100
• Modelling 101 & Programmer’s View (PV) Models
• Cycle Accurate Models
• Bringing the two worlds together
• Results
• Summary
CHALLENGES IN MOBILE SOC DESIGN

A typical verification challenge for today ...
big.LITTLE™ processing heterogeneous multicore

- Increasing complexity
- Increasing SW content
  - Multiple OSs
  - Secure services
- Diversified development community
- Higher performance and lower power

Are we building the thing right?
Are we building the right thing?
ONE SIZE DOES NOT FIT ALL
DIFFERENT TASKS – DIFFERENT TOOLS

- **Cycle Accurate (CA)**
  - Validation View
    - HW Validation
    - Driver Development
    - HW/SW Co-Verification
  - 1-20 KIPS

- **Approximately Timed (AT)**
  - Performance View
    - Architecture Exploration
    - Performance Evaluation
    - Benchmarking
  - 50-200 KIPS

- **Loosely Timed (LT)**
  - Programmers View
    - Application Development
    - SW Profiling
    - Architecture Compliance
  - 50-200 MIPS
• High level modeling and virtual prototypes exploited to improve development productivity and reduce time to market.

• Models required early in product development lifecycles

• Models are utilized for:
  – Architecture & device modeling
  – Early software development
  – Early HW / SW co-validation
  – Device implementation compliance
  – Device validation support

• Models are used ...
  – Within ARM
  – Supplied to silicon partners
  – Supplied to Ecosystem partners

Accelerating SoC Time to Market ...

- High performance models suitable for SW community
- Available early for SW development
- Accurate to HW for consistency
- Easy to debug enabling improved productivity
- Easy to deploy to facilitate the developer community

Source: INTERNATIONAL TECHNOLOGY ROADMAP FOR SEMICONDUCTORS, 2010 update
LOOSELY TIMED MODELING
(A.K.A. PROGRAMMER’S VIEW)

- Architecture Envelope Model (AEM):
  - Executable version of the ARM Architecture Reference Manual ("ARMARM")
  - Catch architectural defects early

- Core LT Model
  - e.g. ARM® Cortex™-A15, Cortex-A7 processors
  - Early software development platform
  - Device implementation compliance: execute same validation suites as RTL
LOOSELY TIMED MODEL
HOW PERFORMANCE IS ACHIEVED

• Code Translation
  – Optimized translation of ARM instructions to X86 equivalents

• Direct Memory Access
  – Translated code is cached for fast memory access
AN EXAMPLE:
BIG.LITTLE SYSTEM BRING-UP TIMELINE

Virtual Platform
- 1st VP to ARM SW Dev
- Lead Partner Delivery
- General Release
- Bi-monthly incremental deliveries

Software Stack
- Lead Partner Delivery
- APM Demo
- Task Migration S/W Development on Virtual Platform

Android Port
- Linaro Delivery
- big.LITTLE™ Android Gingerbread
- Ice Cream Sandwich

Hardware
- Delivered
- Software Stack Running
- Performance Tuning

2011

2012
• 2.2GHz Laptop
• Cortex-A15x1 – 75 seconds (90 mips)
• Cortex-A15x4 + Cortex-A7x4 – 140 seconds
• 14 days continuous, >100T instructions

• Recommendation: High end workstation, 64-bit OS, large physical memory
- Complex designs need multiple verification and validation platforms
- Virtual platforms allow early HW / SW co-development and analysis
Fast, accurate Programmer’s View (LT) Models aligned with the ARM IP and tools roadmaps
- Enables development at all levels of the software stack prior to silicon availability

Models for newly available ARM processors
- Cortex-A15, Cortex-A7, Cortex-R5, Cortex-R7, big.LITTLE reference platforms
- Models of ARMv8 cores available to lead partners
- ARMv7 and ARMv8 architecture models for validation of processor IP

Export to SystemC and TLM-2.0

Comprehensive debug, trace and simulation control
- Debug with DS-5™ toolchain
• Highly accurate interconnect (& System IP) models
  – > 95% accuracy
  – Use with traffic generators for interconnect-centric exploration
• Accurate CPU models
  – > 80% accurate: deliver representative traffic to interconnect
  – > 95% accurate: Architecture validation & SW performance optimization
• Execution speed ~ 100’s KIPS
  – Execute significant workloads
• Modelling 101 & Programmer’s View (PV) Models
• Cycle Accurate Models
• Bringing the two worlds together
• Results
• Summary
CYCLE ACCURATE MODELS
WHAT ARE THEY REALLY?

- Not defined in SystemC TLM-2.0
- Requires protocol knowledge
- Can be impossible to fully validate if hand-created

Source: Google.com
• Ideally, all models are implementation-accurate
  – 100% confidence in developed system
  – Accurate hardware and software design decisions

• Multiple reasons why they’re not always used
  – RTL not available yet
    – Execution speed
    – Model creation/validation time
CARBONIZED ARM MODELS

- 100% accurate model of ARM IP compiled from ARM RTL
  - Instrumented for interactive debug, memory, pipeline and cache analysis
  - Mappings included to enable automatic interchange with ARM Fast Models

- Models for nearly all available ARM processors
  - Cortex-A15, Cortex-A7, big.LITTLE, Cortex-A9, Cortex-A8
  - Cortex-R5, Cortex-R4, Cortex-M4, Cortex-M3, Cortex-M0
  - ARM11™ MPCore™, ARM1176™, ARM1136™, ARM926™, ARM946™, ARM968™, ARM7TDMI-S™
    - Early availability programs on new processors

- Reference platforms and software packages included

- Comprehensive debug, trace and simulation control
  - Debug with DS-5 toolchain
• Modelling 101 & Programmer’s View (PV) Models
• Cycle Accurate Models
• Bringing the two worlds together
  • Results
  • Summary
WHY NOT APPROXIMATELY TIMED?
THE DANGER OF THE MIDDLE

- AT models are great as part of HLS process
- Not as well-suited for other design tasks
  - Can’t achieve the speeds needed for software
  - Doesn’t have the accuracy needed for architecture
  - Require large design/validation effort
• PV Models alone: insufficient detail for accurate analysis
• CA Models alone: unable to run large software workloads
• Option 1: dedicated models
• Option 2: re-use existing models
  – PV for performance
  – CA for detailed timing information
• Mixing PV and CA models can be very effective
  – Maintain core in PV
  – Use CA for model(s) under investigation
• CA when needed, dormant when not
• Any model can be CA but avoid processor/memory datapath if maximum performance is desired
• Good for firmware, not for architecture
• Utilize existing LT and CA models
• Use LT models to quickly execute to point of interest
  – Software breakpoint
  – Arbitrary point in time (requires sync step)
• Swap to CA models to continue execution
  – Debug hardware/software interaction
  – Gather performance analysis data
• Checkpoints are used to create swap points
• ALL models need to support checkpointing

Checkpoints are used to create swap points
ALL models need to support checkpointing

Checkpoint Data via CADI including side-effects
CUSTOM MODEL SWAP

Models presenting a compliant ARM ESL API interface may participate in swap

Target model needs to support writing to necessary registers via ARM ESL API (including side-effects!)

Only state elements are supported, not thread context

Need system management of sockets, debug connections, etc
IF IT’S NOT TESTED, IT’S BROKEN

- Extract checkpoints from LT run
- Compare end results to validate execution
- Intermediate comparisons may be inconsistent due to different execution model in actual processor
• Modelling 101 & Programmer’s View (PV) Models
• Cycle Accurate Models
• Bringing the two worlds together
• Results
• Summary
CUSTOMER EXAMPLE: PARTITIONED SPEED AND ACCURACY

- Majority of system executes in LT domain, GPU in CA
- GPU only active when being initialized or processing frames
- Boots Linux in <20 seconds, processes frames in <90 seconds
- Display 9 frames of graphics in virtual prototype while hardware prototype is still booting Linux
LEVERAGING SPEED AND ACCURACY DRIVER DEVELOPMENT

- Create checkpoint at each point of interest, each engineer debugs at correct time with 100% accuracy
- Able to boot OS and run to relevant checkpoint OS in 10-15 seconds leveraging speed of ARM Fast Models
- Each checkpoint is debugged independently with 100% accuracy
LEVERAGING SPEED AND ACCURACY
PERFORMANCE ANALYSIS

- Break long run into multiple run to execute in parallel and gather accurate data for cycle count, power analysis, etc.
- Results can be aggregated to deliver accurate results for much longer runs than normally possible with accurate models
DEPLOYMENT RESULTS

- Ability to swap Fast Model -> Carbonized model is in active use
- Execution time is unchanged
  - No effect on Fast Model runtime speed
  - CA models execute at regular speed after restore
  - Checkpoint creation/restore is <10 seconds
- Currently supported cores
  - Cortex-A15, Cortex-A7, Cortex-A9
• Modelling 101 & Programmer’s View (PV) Models
• Cycle Accurate Models
• Bringing the two worlds together
• Results
• Summary
• Swapping LT and CA models enables a single virtual prototype to be used by all design groups
  – 50-200 MIPS performance for software development
  – 100% accuracy for firmware, architecture and debug