RISC-V Models for Architecture Analysis, Software Development and DV

Larry Lapides
Imperas Software Ltd.
Seattle RISC-V Meetup, 2 April 2020
“nobody designs a chip without simulation”, at Imperas we believe that:

“nobody should develop embedded software without simulation”

Imperas develops simulators, tools and debuggers, and models (especially processor models) to help embedded systems developers get their software running...
- and hardware developers get their designs correct

10+ years, self funded, profitable, UK based, team with much EDA (simulators, verification), processors, and embedded experience

- www.imperas.com
- www.OVPworld.org
Imperas and RISC-V

- DAC 2016 (Jun): We first learn about RISC-V – looks academic and fragmented
- RISC-V Workshop 2016 (Nov): 350 attendees from serious companies (Google is the host for the workshop), and the ISA looks to be converging
- Q1 2017: Imperas joins the RISC-V Foundation; we build our first RISC-V processor model
- Q3 2017: Imperas starts participating in the Compliance Working Group
- Q1 2018: Imperas introduces methodology for adding/optimizing custom instructions for RISC-V cores
- Q2 2018: First paying customer using Imperas RISC-V models for software development and design verification (DV)
- Q1 2019: First tape out of RISC-V SoC based on using Imperas model as DV reference model
- Q2 2019: Imperas starts collaborating with Google on DV flows with instruction stream generator
- Q2 2019: Imperas joins CHIPS Alliance and OpenHW organizations
- Q3 2019: Imperas provides first RISC-V model to fully support vector and bit manipulation extensions
- 2019: Imperas RISC-V related revenue now 4x Arm related revenue
RISC-V Customer Use Cases

- Architecture analysis, including (especially) custom instructions
- Software development, debug and test
- Processor and SoC verification
Before the Use Cases, Some Background Information on Processor Models and Virtual Platforms and Tools
Components of Open Virtual Platforms (OVP) Fast Processor Models

OVP models are open source and free

- All models have both C and SystemC/TLM2 native interfaces
- Available under the Apache 2.0 open source license
- Require an Imperas simulator license to run
- 1 simulator license is all that is needed for multi-core and many-core platforms

http://www.ovpworld.org/info_riscv

Main components

- Add memory-based components (e.g. loosely-coupled caches, devices)
- Create and manage resources (e.g. registers) shared between multiple processor instances
- Add CPU-aware information for VAP analysis tools
- Add processor independent I/O support
OVP Library of Fast Processor Models

- Existing Imperas Open Virtual Platforms (OVP) Fast Processor Models of ...
  - Generic or envelope models of RV32/64 IMAFDCEVB M/S/U privilege modes
    - Vector and bit manipulation instructions were added as soon as specs stabilized
    - Imperas OVP models were the first models to support these instruction extensions
  - Andes cores: A(X)25, N(X)25, N(X)25F, 27-series including NX27V, ...
  - SiFive cores: SiFive Series 2, Series 3 (e.g. E31), Series 5 (e.g. E51, U54), Series 7

- Custom instructions easily added by user or by Imperas
  - New instructions are added in side file so as not to perturb the verified model
    - Imperas tools work with the complete processor model, including the custom instructions
  - Custom instructions can be analyzed for effectiveness using instruction coverage, profiling tools
  - Timing estimation tools can be extended to custom instructions
  - Video demo: http://www.imperas.com/risc-v-custom-instruction-design-and-verification-flow-0

- Models are open source, distributed under the Apache 2.0 open source license
Open Virtual Platforms
Peripheral Models

• 100s of peripheral models available in the OVP Library
• All models are open source
  • Distributed under the Apache 2.0 open source license
• All models have both C and SystemC interfaces
• iGen productivity tool enables easy building of peripheral models
• Imperas debugger supports peripheral introspection, such as break points on peripheral registers
Extendable Platform Kits™ (EPKs™)

- EPKs are virtual platforms, with software running, to help users start quickly
- EPKs include
  - Individual models, binary and source
  - Platform model, binary and source
  - Software and/or OS running on platform
- Over 50 EPKs in library (Arm, MIPS, RISC-V, ...)

Andes N25 (RV32)

- CoreGPIO
- CoreTimer
- CoreUART
- CorePLIC
- RAM
Imperas Environment

- Application Software & Operating System
- Virtual Platform
  - Peripheral
  - Memory
- OVP CPU
- JIT simulator engine
- SlipStreamer API
- Multiprocessor / Multicore Debugger
- Eclipse IDE

Verification, Analysis & Profiling (VAP) tools
- Trace
- Profile
- Code coverage
- Memory monitor
- Protocol checker
- Assertion checkers
- OS task tracing
- OS scheduler analysis
- Fault injection
- Function tracing
- Variable tracing
- ...
RISC-V Customer Use Cases

- Architecture analysis, including (especially) custom instructions
- Software development, debug and test
- Processor and SoC verification
How is Processor Performance Optimized?

• Move to multicore
• Optimize the pipeline
• Improve memory usage/latency
• Custom instructions for application/domain optimization (unique to RISC-V)
Flow to add new custom instructions

Characterize C Application
- Instruction Accurate Simulation
- Trace / Debug
- Timing Simulation
- Function Timing / Profiling

Develop New Custom Instructions
- Design Instructions
- Add to Application
- Add to Model
- Add Timing

Characterize New Instructions in Application
- Instruction Accurate Simulation
- Trace / Debug
- Timing Simulation
- Function Timing / Profiling

Release & Deploy
- Check RISC-V Compliance
- Use as reference for RTL Design Verification
- Use in Imperas/OVP Platforms, EPKs
  - Heterogeneous / Homogeneous
  - Multi-core, Many-core
- Imperas Multi-Processor Debug, VAP tools
- Port OS, RTOS (Linux, FreeRTOS...)
- Use in many simulation envs (inc. SystemC)
- Deliver to end users

Optimize & Document model
- Instruction Coverage
- Line Coverage
- Instruction Performance
- Generate PDF model doc
Flow to add new custom instructions

Characterize C Application

- Instruction Accurate Simulation
- Trace / Debug
- Timing Simulation
- Function Timing / Profiling
Instruction Accurate (IA) Simulation of C Application

- Cross compiled C application targeting RV32IM
  - Character stream encoder, with ChaCha20 encryption algorithm
- IA simulation
  - Imperas RISC-V ISS with configurable model of RISC-V specification selecting RV32IM
- Semihosting
  - Enables bare metal application to very simply access host I/O

➡️ runs fast
- Over 1 billion instructions a second (standard PC)
  - Linux and Windows supported host OS
Flow to add new custom instructions

Characterize C Application
- Instruction Accurate Simulation
- Trace / Debug
- Timing Simulation
- Function Timing / Profiling

Develop New Custom Instructions
- Design Instructions
- Add to Application
- Add to Model
- Add Timing

Characterize New Instructions in Application
- Instruction Accurate Simulation
- Trace / Debug
- Timing Simulation
- Function Timing / Profiling
**Cycle Approximate Simulation Including Custom Instructions**

- IA simulation + timing annotation + custom instructions
  - Includes timing estimation for RV32IM processor
  - Need to add timing estimation for new custom instructions
- Simulate using C code application with inline assembler of custom extensions
- IA simulator + timing tool + custom extension instruction library

- See estimated improvement in throughput of application on new processor
  - Was 16.59 secs without custom instructions
  - **Now 9.21 secs with custom instructions**
Function Profile of Application with Custom Instructions

• IA simulation + timing annotation + custom instructions with sampled profiling

➤ Shows where slowest function is
  • Now much faster...

➤ Shows benefits of using custom instructions
  ➤ processLine was 21.35% now 16.3%
Software Debug and Analysis Tools
Automatically Work With the Custom Instructions

New custom instructions, new additional state registers

New custom instructions in trace disassembly
RISC-V Customer Use Cases

- Architecture analysis, including (especially) custom instructions
- **Software development, debug and test**
- Processor and SoC verification
Security is Critical

Nagravision

- Application
  - Devices that protect streaming video
    - Attach to smart tv or set top box
    - Build end user device, software and SoC
  - IoT devices

- Imperas use model
  - Peripheral models for the virtual platform are built by Nagravision (proprietary models) or modified from the OVP Library (standard I/O, e.g. USB)
  - Use Imperas debugger for software debug and for driver-peripheral model software-hardware co-debug
  - Use SW Verification, Analysis and Profiling (VAP) tools such as OS-aware tools, code coverage, memory analysis, ...
  - High performance simulation is critical for Continuous Integration (CI) testing

“At NAGRA, we have adopted the Imperas virtual platform-based software development and test tools for our application and firmware teams. The simulation performance, and the tools for software analysis, have added significant value to our daily Agile Continuous Integration (CI) methodology. Our view is that software simulation is mandatory to reach metrics required for high quality secured products.”
RISC-V Customer Use Cases

• Architecture analysis, including (especially) custom instructions
• Software development, debug and test
• Processor and SoC verification
Compliance Is Not Verification

- Need to be clear what focus of testing is
  - Architecture
    - ISA Definition
  - Micro-Architecture
    - In-Order, Out-Of-Order, Simple-Scalar, Super-Scalar, Transactional Memory, Branch Predictors, ...
- These are very different
  - One is about ISA specification
  - Other is about details of a specific implementation
  - This is the difference between “Compliance” and Design Verification

- In the RISC-V Foundation, “Compliance” testing is checking the device works within the envelope of the agreed specification
  - i.e. “have you read and understood the specification”
  - Compliance testing is not a full hardware verification...
riscvOVPsim
Free, GitHub RISC-V ISS

- Industrial quality, free ISS / reference model for compliance testing
  - GitHub.com/riscv/riscv-compliance
  - GitHub.com/riscv/riscv-bitmanip
- Can also be used for test development, software development, design verification
- Implements full RISC-V envelope
  - Configurable for all features and version
- Includes full open source Apache 2.0 model
- Kept up to date for specification changes
  - Updated weekly for Vector and Bitmanip spec changes
  - Awaiting Hypervisor and DSP specs stability before developing
- Works ‘out of the box’ with full tracing, debug, and many other options

Imperas riscvOVPsim Compliance Simulator
Imperas Support for RISC-V DV

- riscvOVPsim:
  - Freely available ISS as part of the RISC-V compliance test suite
  - Everyone developing a RISC-V processor seems to be using this

- Custom instructions
  - Obviously custom instructions need verifying, but also the underlying processor needs to be re-verified
  - Imperas has an easy flow for adding custom instructions to the RISC-V reference model

- Google Instruction Stream Generator (ISG) technical support as a service
  - Everyone developing a RISC-V processor seems to be using this
  - Not easy to use for more complex processors
  - Also want to use functional coverage with the ISG; Imperas understands this part of the ISG flow

- Step/compare DV methodology
  - Trace and signature comparison flows are inefficient, and in some cases insufficient, for complex processor DV
  - Imperas has a UVM-compliant SystemVerilog encapsulation of RISC-V reference models
    - Reference models can be single- or multi-hart, have any combination of features including custom instructions and be composed of just the processor or be a processing subsystem

- Vector Test Suite
  - The RISC-V vector instructions are as complex as the rest of the RISC-V ISA combined
  - Imperas has a directed test suite that enables more comprehensive testing of vector instructions
Google: open source riscv-dv instruction stream generator
EDA Partners: SystemVerilog design + UVM simulator for RTL
  - working with Cadence, Mentor, Synopsys, and Metrics RTL simulators
Imperas: model and simulation golden reference of RISC-V CPU
Imperas: extending this flow to support step-and-compare DV methodology

Imperas have added Vector and Bitmanip extension instructions to the Functional Coverage (not yet publicly released)
Step and Compare Methodology

• Testbench loads .elf program into both memories, resets CPUs (RTL and OVP model)
• Steps CPUs (DUT and reference), extracting data, and comparing
• Advantages
  • Tests stop immediately upon failure – no wasted simulation cycles
  • There is no stored log file – test log data is dynamic
  • Supports indeterminate and asynchronous events (multi-hart processors and interrupts)
Imperas Users Benefit From Improved System (Software & Hardware) Quality, and Reduced Schedules & Cost

- OVP (Open Virtual Platforms) models of RISC-V processors
- OVP models of RISC-V based SoCs, including peripherals, memory, ...
- Custom instruction support
- Compliance checking knowledge (Imperas knows the RISC-V spec)
- Processor and SoC verification expertise and methodology
- Software development, debug and test tools
Thank You!

Larry Lapides
LarryL@imperas.com

For more information:

http://www.imperas.com/imperas-riscv-solutions
http://www.ovpworld.org/info_riscv