Why custom instructions?

- You already know your application inside out and know there are critical loops that could be improved
  - Allow you to run at a lower frequency
- You’re migrating from other architectures and are missing a key instruction
- Additional functionality to simplify overall design (i.e., replacing MCU + DSP with customised MCU)
How to start customising

- Models let you explore quickly
  - Much faster to develop than RTL
  - Better profiling information available
  - Easier to debug software
RISC-V Model Requirements

- Model the ISA, including all versions of the ratified spec, and stable unratified extensions
- Model other behavioral components, e.g. interrupt controllers
- Easily update and configure the model(s) for the next project
- User-extendable for custom instructions, registers, ...
- Model actual processor IP, e.g. Andes, SiFive, Codasip, MIPS, OpenHW, Mi-V, ...
- Well-defined test process including coverage metrics
- Interface to other simulators, e.g. SystemVerilog, SystemC, Imperas virtual platform simulators
- Interface to software debug tools, e.g. GDB/Eclipse, Imperas MPD
- Interface to software analysis tools including access to processor internal state, etc.
- Interface to architecture exploration tools including extensibility to timing estimation

- Most RISC-V ISSs can meet one or two of these requirements
- Imperas models and simulators were built to satisfy these requirements, and matured through usage on non-RISC-V ISAs over the last 12+ years
Imperas is the Processor Model

- Imperas provides full RISC-V Specification envelope model
- Industrial quality model /simulator of RISC-V processors for use in verification, software development and architecture exploration
- Complete, fully functional, configurable model / simulator
  - All 32bit and 64bit features of ratified User and Privilege RISC-V specs
  - Vector extension, versions 0.7.1, 0.8, 0.9, 1.0
  - Bit Manipulation extension, versions 0.91, 0.92, 0.93, 1.0.0
  - Hypervisor version 0.6.1
  - K-Crypto Scalar version 0.7.1, 1.0.0
  - Debug versions 0.13.2, 0.14, 1.0.0

“The donation of a robust, commercial-quality simulator – riscvOVPsim – will enable our customers to adopt RISC-V even faster. This is the level of close industry collaboration that will drive the successful adoption of RISC-V.”

Yunsup Lee, co-founder and CTO, SiFive
Imperas Model Extensibility

Imperas develops and maintains base model
- Base model implements RISC-V specification in full
- Fully user configurable to select which ISA extensions
- Fully user configurable to select which version of each ISA extension
  - Updated very regularly as ISA extension specification versions change
- Fully user configurable for all RISC-V specification options
  - e.g. implemented optional CSRs, read only or read/write bits etc...

Imperas provides methodology to easily extend base model
- Templates to add new instructions
- Code fragment for adding functionality
- 100+ page user guide/reference manual with many examples
  - Includes example extended processor model

Imperas OVP model is architected for easy extension & maintenance

© 2023 Imperas Software Ltd.
Flow to Add New Custom Instructions

Custom instructions are added to optimize a specific application or set of applications within a domain.
Therefore, start by characterizing the application to be optimized.
Then add the custom instructions, evaluate, and iterate.

Characterize C Application

- Simulation
- Trace / Debug
- Function Profiling
Simulation of C Application

• Example: Character stream encoder, with ChaCha20 encryption algorithm (adding ChaCha20 specific custom instructions)
• Cross compiled C application targeting RV32IM
• Instruction Accurate (IA) simulation
  • Imperas simulator with configurable model of RISC-V specification selecting RV32IM
  
  ➢ Runs fast
  • Over 1 billion instructions per second (standard PC)
Function Profile C Application

- Same C application
- Sampled profiling with call stack analysis

➢ Shows proportion of time spent in each application function

- 21.35% spent in `processLine`
Flow to Add New Custom Instructions

Characterize C Application
- Simulation
- Trace / Debug
- Function Profiling

Develop New Custom Instructions
- Design Instructions
- Add to Application
- Add to Model
Add Custom Instructions to Model and Re-Simulate

- Use standard Open Virtual Platforms (OVP) instruction modeling APIs to add new instructions (and optional state) as **new extension library**
- Compile and link model extension library
- Simulate standard model extended with new library

➢ Instruction count and simulated time have been reduced
Function Profile Application Using Custom Instructions

- IA simulation + custom instructions with sampled profiling
  - Shows where slowest function is
    - Now much faster...
  - Shows benefits of using custom instructions
    - processLine was 21.35% now 14.71%
Flow to Add New Custom Instructions

Characterize C Application
- Simulation
- Trace / Debug
- Function Profiling

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

Characterize New Instructions in Application
- Simulation
- Trace / Debug
- Function Profiling
Flow to Add New Custom Instructions

Characterize C Application
- Simulation
- Trace / Debug
- Function Profiling

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

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

Optimize & Document model
- Instruction Coverage
- Line Coverage
- Generate model doc pdf
Flow to Add New Custom Instructions

Characterize C Application
- Simulation
- Trace / Debug
- Function Profiling

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

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

Release & Deploy
- Check RISC-V Compliance
- Use as reference for RTL Design Verification (DV)
- Use in Imperas/OVP Platforms, EPKs
  - Heterogeneous / Homogeneous
  - Multi-hart, multi-processor
- Imperas MultiProcessor Debug, VAP tools
- Port OS, RTOS (Linux, FreeRTOS, Zephyr, ...)
- Use in SystemC simulation environment
- Deliver to end users

Optimize & Document model
- Instruction Coverage
- Line Coverage
- Generate model doc pdf
Key technologies/differentiators:

- OVP Fast Processor Models
  - Most models
  - Highest quality
- Simulator engine
  - Highest performance
  - SlipStreamer API for non-intrusive tools
- Tools
  - MPD for platform-centric debug
  - VAP tools for comprehensive software analysis
Software Debug and Analysis Tools Automatically Work With the Custom Instructions

New custom instructions, new additional state registers

New custom instructions in trace disassembly
Thank you

Jon Taylor
JonT@imperas.com