How to Address RISC-V Compliance in the Era of OPEN ISA and Custom Instructions

RISC-V Summit, December 2018

Simon Davidmann, Lee Moore

© 2018, Imperas Software
Agenda – RISC-V Compliance

• Compliance for RISC-V is extremely important
• Imperas background
• RISC-V Foundation’s compliance suite
• Experiences when checking compliance on user designs
• Summary
Agenda – RISC-V Compliance

• Compliance for RISC-V is extremely important
• Imperas background
• RISC-V Foundation’s compliance suite
• Experiences when checking compliance on user designs
• Summary
Compliance for RISC-V is important

Q: What is meant by “compliance”?  
A: The device works within the envelope of the agreed specifications

Q: Is there an easy process or path to follow to develop methodologies/tools for compliance?  
A: NO – all established ISAs are single company controlled and those companies work extremely hard on proprietary solutions to ensure that all designs that go out their door work correctly – so RISC-V has to pioneer compliance collectively

Q: What happens if the RISC-V industry builds devices that are not complying with specifications?  
A: Users cannot assume that tools like C compilers, operating systems, and application software will be transferable across devices and work correctly
Non-Compliance

- Non-compliance makes it very hard or impossible to benefit from the developing RISC-V ecosystem
  - Won’t be able to use GCC, LLVM, GDB, etc...
- Fragmentation is the result of non-compliance and would be a disaster for the RISC-V industry as it leads to incompatible processors
  - Locked in to one CPU, software not portable across different chips / boards
- If you build a non-compliant processor or simulator or tool
  - => IT IS NOT RISC-V
Agenda – RISC-V Compliance

• Compliance for RISC-V is extremely important

• Imperas background
  • Why is Imperas interested in RISC-V & RISC-V compliance
  • Imperas related experience
  • How can Imperas tools help with compliance

• RISC-V Foundation’s compliance suite

• Experiences when checking compliance on user designs

• Summary
Imperas Focus

• “nobody designs a chip without simulation”, at Imperas we believe that:
  “nobody should develop embedded software without simulation”

• Imperas develops simulators, tools, debuggers, modelling technology, and 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/riscv
Imperas solutions

Staff have extensive experience building tools & have been involved with:
- CPU design, CPU modeling, HDL simulator, constrained random RTL verification, embedded systems, embedded software
- silicon IP, compliance suites
- development & process of creating 3 standards with OVI, Accellera, IEEE

Imperas virtual platform solutions
- Use open standard Open Virtual Platform APIs to build/extend open source models, platforms
- Library of 200+ CPU models
  - 12+ ISAs including: Arm, MIPS, ARC, Renesas, Power, Xilinx, Altera, RISC-V, + several proprietary architectures
  - includes 250+ behavioural models for building models of SoC, board, reference designs
  - including Ethernet, USB, CAN, ...
- Simulations can be single core, multi-core, heterogeneous, custom models
- Tools used by ARMv8 architecture licensees in mobile CPU development

Full tool suite for
- Processor development including custom instructions
- Peripheral & platform modelling
- Multi processor debug from script or GUI / Eclipse
- Verification, Analysis, Profiling, including OS aware tools
Imperas & RISC-V

- Jun 2016 – watched DAC ‘academic’ presentation
- Nov 2016 – attended RISC-V workshop and saw progress
- 2017Q1 – started feeling interest from customers
- Mar 2017 – joined RISC-V Foundation
  - Compliance, Formal, Vector, Marketing Task Groups
- Nov 2017 – released first OVP model of RISC-V
- Mar 2018 – released complete RV32/RV64 envelope model
  - all options of specifications (no hypervisor, vectors yet)
- Jun 2018 – released virtual platform of SiFive FU540 RV64GC booting SMP Linux
- Jun 2018 – Andes certify Imperas as reference RISC-V simulator
- 2018Q3 – Imperas customers developing RISC-V SoCs
- Sep 2018 – released first commercial tool flow for custom instruction extension development, testing, profiling, and verification
- Oct 2018 – released riscvOVPsim – free envelope simulator
Agenda – RISC-V Compliance

• Compliance for RISC-V is extremely important
• Imperas background
  • RISC-V Foundation’s compliance suite
    • Short history
    • Current status
• Experiences when checking compliance on user designs
• Summary
The RISC-V Foundation’s Compliance Task Group

• Jun 2017 group started
  • Compliance testing is a testing technique which is done to validate whether the system developed meets the prescribed standards or not. It is **not design verification** testing
    • compliance testing is looking for issues like missing registers, modes, instructions – not for bugs in RTL implementations...
  • The tests have to be written to ensure compliance/non-compliance is **observable in a test signature**. The signatures are published so that the user does not have to run a reference model and can compare the results of their target runs to the reference signature

• Jan 2018 initial rv32i test suite provided by Codasip
• Jun 2018 Imperas, Embecosm worked on GitHub, made repo public
• Oct 2018 Imperas improved test coverage, added new suites, ported 32bit riscv-tests
Compliance Suite

- It is ‘work in progress’

- Two components
  - Test suites
    - Each suite focuses on a feature set of the RISC-V envelope
    - Initial focus is instructions, user mode spec, e.g. rv32i, rv32im, rv32imc, rv64i, ...
    - Awaiting RISC-V platform specifications to subset privilege spec, before starting privilege suites
  - Framework
    - Make, bash, and scripts
      - Encapsulate compiler tools, linkers, simulators, and targets (Devices Under Test)
    - Includes simulator: as example target, and to generate reference signatures
    - Run: Select suite and target
      - Runs each test, target produces signatures, compares to saved golden reference signature

- Available: www.github.com/riscv/riscv-compliance
### Compliance Suite Status (rv32i)

### Notes on rv32i test suite:
- 54 hand coded directed tests (average 150 instructions each)
  - [Link to test suite](https://github.com/riscv/riscv-compliance/tree/master/riscv-test-suite/rv32i/src)

### Notes on Decode Coverage:
- Decode Coverage: observe changes on all bits of legal decodes
- Decode Coverage data from the Imperas Fault Simulation Coverage tool
  - ran 478,390 simulations in 308 secs
Coverage Metrics for Compliance Tests

• Coverage metrics used in RTL design verification are **not applicable** as they are often functional and connected to specific microarchitecture (RTL)

• Imperas’ Code Coverage Tool provides coverage of model source, which is useful to see how much of model is exercised by tests but **does not** show how much of specification is covered

• Imperas’ Fault Simulation Coverage tool provides *Instruction Decode Coverage*

  - explores the decodes of the instructions and mutates legal bits
  - detects that there is a test that stimulates & observes each bit
  - tool is very fast, runs in parallel, provides other metrics including data coverage
  - very useful for users as **provides coverage of custom instructions**
Test Suite Status


Currently there are twelve test suites checked into this repository.

If you are looking to check compliance of RV32I in user mode then run the suites: RV32I and RV32UI.

Test suites status:

Pretty Solid:

- RV32I (originally developed by Codasip, updated significantly by Imperas to improve coverage)
  - 54 focused tests, using the correct style/macros, excellent coverage of most instructions
  - no coverage of fence, scall, sbreak, pseudo and csr instructions
- RV32IM (developed by Imperas)
  - 7 focused tests, using the correct style/macros, excellent coverage
- RV32IMC (developed by Imperas)
  - 24 focused tests, using the correct style/macros
- RV64I (developed by Imperas)
  - 8 focused tests, using the correct style/macros
- RV64IM (developed by Imperas)
  - 3 focused tests, using the correct style/macros

Work in progress (user mode):

- RV32UI (from github/riscv-tests with poor coverage, Ported by Imperas)
Challenges for writing tests for RISC-V and compliance suite developers

- How to write the tests
  - Need to white box walk through to examine detailed operation
- How to explore when things don’t go as planned
- Requirements on a simulator to assist with test development
  - Needs full implementation of specifications
  - Needs full configurability to all options of specification
    - Envelopes of legal subset choices
  - Full trace to see all resources affected by all instructions
  - Need access to comprehensive debug
  - Has to be able to be encapsulated, and controlled in external environment
  - ...

© Imperas Software, Ltd.  December 18
riscvOVPsim simulator

- Instruction Accurate simulator using high performance Just-in-Time Code Morphing that executes RISC-V binaries and runs on Linux/Windows PC
  - Runs very fast, 1,000 MIPS
- Industrial quality for use in test development, software development, compliance testing and design verification
  - Includes capabilities to perform RISC-V compliance signature generation
- Maintained and supported by Imperas Software (www.imperas.com)
- Simulator restricted to single processor model and fully populated memory
  - Loads cross compiled ELF binaries
  - Uses semi-hosting to provide easy access to host I/O
  - Single executable (includes model, memory, platform, simulator, tools)
- Includes Apache 2.0 Open Source model of complete RISC-V ISA envelope model of 2.2, 2.3, 1.10, 1.11 revisions of the RISC-V Foundation specifications (not hypervisor/vectors are not stable)
- **FREE. No license keys or license management**

© Imperas Software, Ltd.
December 18
Issues to be sorted by RISC-V Foundation’s Compliance Working Group

- Detailed specifications needed
  - the structure of the Compliance Suite Framework (and how used)
  - what content of test should be (how written)
  - list of all the tests needed in each test suite (what is covered of specification)

- Formal process for RISC-V Foundation to allow user to claim their processor is a RISC-V processor
  - i.e. process to explore and record ‘is it a RISC-V compliant design?’
  - the ‘rubber stamp: RISC-V inside’
Users of the Test Suites

• Very hard to really know usage of the tests suites from GitHub
  • 46 unique cloners (download, use)
  • 23 forks of repo (download, use & make changes)

• From questions on GitHub & Task Group & emails these people are (public) users:
  • Tools/Service:
    • Imperas
    • Embecosm
    • Bluespec
    • Galois
    • Valtrix
    • XtremeEDA Corporation
  • Processors:
    • Andes
    • Microsemi
    • ETH (PULP)
    • Incore (Shakti)
    • Codasip
    • Syntacore
  • Groups:
    • Formal TG with Forvis

• Plus all the RISC-V Summit 2018 soft-core competitors...
• If you are a user – let us know - send me an email: simond@imperas.com
Example of Design Verification Use-case: Using riscvOVPsim with Valtrix STING Instruction Generator

- STING is a design verification tool developed by Valtrix Technologies for RISC-V based CPU and SoC implementations
- STING generates architecturally correct random, directed and control graph flow based instruction sequences for testing the functionality and architectural compliance of the target implementation
- STING implements a checking mechanism where the results from running a test on DUT and a functionally accurate ISA simulator are compared
- Imperas’ riscvOVPsim is used in the checking flow for its superior ISA modelling, verification & configuration capabilities
Agenda – RISC-V Compliance

- Compliance for RISC-V is extremely important
- Imperas background
- RISC-V Foundation’s compliance suite

  Compliance Experience
  - Imperas experience of examining compliance on various designs
  - Examples of Fails...
  - Recommendations

- Summary
Imperas: Exploring designs

Note: these notes are Imperas exploring designs brought into Imperas, not a discussion of what our customers are doing with our tools & test suites

- Many candidate DUTs (Device-Under-Test) = customers, partners, & for fun...
  - proprietary RTL, open source RTL
  - FPGA
  - Silicon
  - Simulators

- Process – goal is to load .elf file, run, write signature, compare with golden reference

- Develop encapsulation of DUT
  - so can run public GitHub Compliance Suite – and – Imperas internal compliance test suites

- Different levels of complexity and challenge
  - ISA Simulators relatively easy (as can read/write files)
  - RTL simulators a little harder, 2 approaches
    - Memory read/write
    - GDBserver – control as if hardware, inject, extract data
  - Hardware (FPGA, silicon)
    - Pod, JTAG, GDBserver type approaches to get data in/out
Imperas: Internal test suites

• Over 10+ years, Imperas has learnt how to test processors/models
  • Expertise in 12+ ISAs: x86, Arm, MIPS, ARC, Renesas, Power, Xilinx, Altera, RISC-V, ...
  • Developed testing approach with sophisticated regression test environment
    • Tests: directed white box, litmus, random, constrained random, benchmark, OS, ...

• Instructions are easy, the **issues are always with the ‘privilege’ modes**

• Imperas customers can license access to Imperas internal tests (inc. RISC-V suite)

• Built & productized tools to assist
  • Configurable simulator
    • so very easy to get models into specific correct states (hard with virtual memory and multi-processor)
    • control & visibility for 3rd party execution validation (RTL, simulator, ...)
  • Model code coverage (source)
  • Instruction coverage (functionality coverage)
  • Instruction decode coverage (test coverage)
  • Instruction profiling (simulation speed)
  • **Methodology for user custom instructions** as well as standard instructions
What did we find…

• Many candidate DUTs (Device-Under-Test) = customers, partners, & for fun…
  • proprietary RTL, open source RTL
  • FPGA
  • Silicon
  • Simulators
What did we find…

• Many candidate DUTs (Device-Under-Test) = customers, partners, & for fun…
  • proprietary RTL, open source RTL
  • FPGA
  • Silicon
  • Simulators
• Missing registers
• Missing instructions
• Floating point mode change issues
• PMP implementation issues
• ...

Page 25
What do we recommend for Compliance checking

Note: this is for compliance checking, not design verification...

• Clone www.GitHub.com/riscv/riscv-compliance

• Encapsulate your DUT target (in your cloned local version of test suite)
  • Use as starting point one of the example targets that are being added

• Get the initial rv32i suite running (and comparing to golden references)
  • add others as they are available and as appropriate

• Run & check whenever you modify your design (add to your regression suite)

• Update your clone/fork and run & check when there are changes to www.GitHub.com/riscv/riscv-compliance
Agenda – RISC-V Compliance

• Compliance for RISC-V is extremely important
• Imperas background
• RISC-V Foundation’s compliance suite
• Compliance Experience
  • Imperas experience of examining compliance on various designs
  • Examples of Fails...
  • Recommendations
• Summary
Summary

• Compliance – it’s a work in progress
  • Initial suites & framework can be used NOW
    • Users are telling us the tests are finding issues for them
  • Ongoing work on RV32 A, F, D user mode tests
  • Need to start compliance suite tests of privilege mode & platforms
  • New ISA extensions (e.g. Vector, BitManip, DSP, …) will need to add own suites

• Understanding & adoption has started

• Need to consider compliance as fundamental part of methodology
  • Continuously...

© Imperas Software, Ltd.
December 18
How can Imperas help a RISC-V developer

- FREE full envelope model and reference simulator – riscvOVPsim
  - Commercial tool implementing full RISC-V spec
    - Controllable to be precise design choices
  - As used in Compliance Working Group
    - (and potentially BitManip, Vector groups)
  - Open Source (Apache 2.0) OVP model
- Models of commercial IP/Silicon
  - MicroSemi CoreRISCV, MiVRV32IMA
  - SiFive E31, E51, U54, U54MC, ...
- Example extendable virtual platform kits as source
  - Microsemi, Andes, SiFive, ...
  - Running FreeRTOS, Linux, SMP-Linux, ...
- Full professional tools:
  - simulator, multicore debugger, Advanced Verification, Analysis, Profiling tools
- Methodology and tools to add, verify, and debug custom instructions
Thanks

• More Information:
  • www.imperas.com/riscv
  • www.OVPworld.org/riscv
  • www.github.com/riscv/riscv-compliance
  • simond@imperas.com
  • moore@imperas.com