Brief introduction to the 5 levels of RISC-V processor verification

Kevin McDermott, Imperas Software

7 December 2021

Introduction

- RISC-V means many people are designing new processors, or modifying source of processors
- For RISC-V anybody can be ‘an architecture licensee’
- And every core needs verifying… (its not like buying in pre-verified IP)
- Many people are new to CPU DV for the first time
  - Traditionally done behind closed doors in commercial/proprietary companies

- This presentation aims to introduce the main approaches of CPU DV
- And discusses their pros and cons

Challenges in RISC-V CPU DV

- Feature selection and choices require serious consideration due to implications of every choice
  - Experienced architecture teams know the costs associated with every feature
    - Every addition dramatically increases (doubles ?) verification & compounds verification complexity
    - Costs of simple added feature can be huge – and unknown to inexperienced teams
    - Adds schedule, resources, quality costs == big risks…
- No off-the-shelf toolkit available for DV of processors
  - No EDA vendor has 'RISC-V CPU DV kit' product
  - There are in-house proprietary solutions in CPU developers… Intel, AMD, Arm, …
  - Building your own adds schedule, resources, quality costs – and risks
- Current SoC cost is 50% for HW DV (with CPUs bought in as proven IP)
  - Developing own CPU adds huge DV incremental schedule, resources, quality challenges

Agenda

- RISC-V CPU HW DV approaches
  1. “hello” test
  2. Simple check
  3. Trace-compare
  4. Data-path lockstep-compare
  5. Async lockstep-compare

Note that not all projects have the same requirements, schedule or verification needs – so each projects DV needs may / will differ

1: ‘Hello World’ DV

- “if I can get a program to run – then my DV is done… right?”
- “my DV challenge is sorted if I can get Linux to boot on my design…”

- Basically this level of DV is where developer feels if they can get their current compilation of their current program to run (through one path) - then their silicon design job is done

- This may be fine for test chips, research, academic, hobbyists, but NOT for products
- The approach is often due to lack of knowledge or interest in quality, …

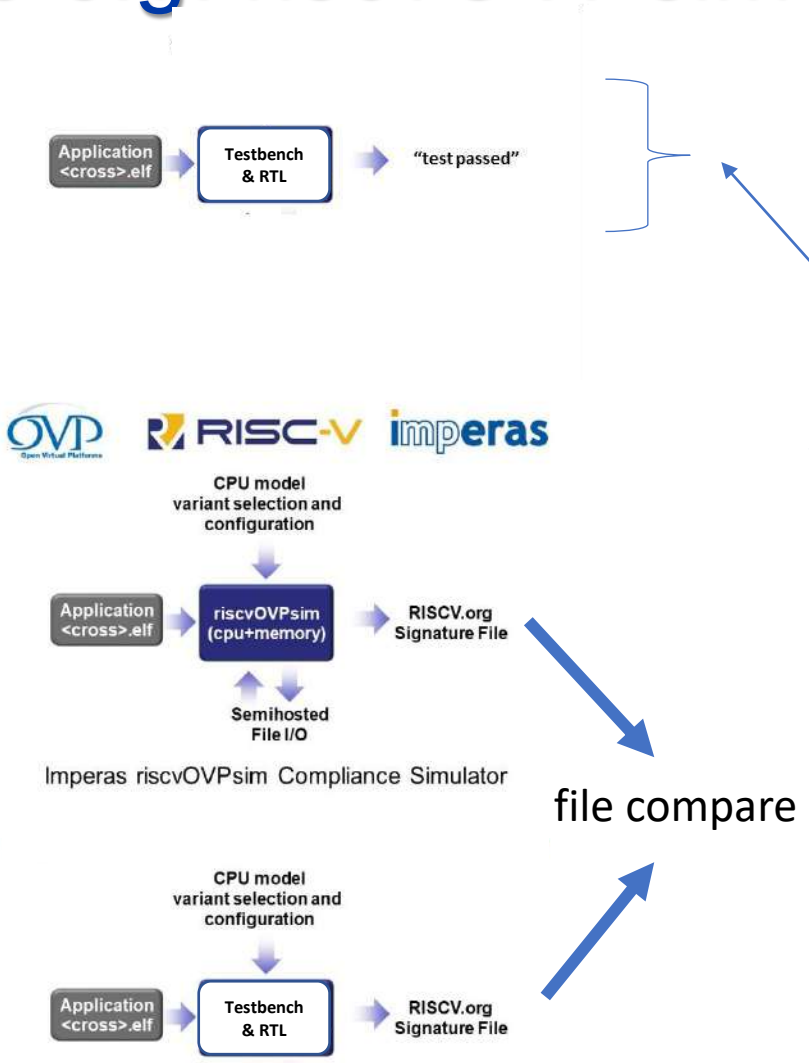
‘Hello World’ DV

- This is not DV!

Agenda

- RISC-V CPU HW DV approaches
  1. “hello” test
  2. Simple check
  3. Trace-compare
  4. Data-path lockstep-compare
  5. Async lockstep-compare

2: Simple check (use e.g. riscvOVPsim ISS from GitHub)



- Run RTL DUT in testbench
  - (no real testbench)
  - Just runs the program
- Either
  - Each test program checks its results = go/no go test
    - Prints message to log
    - or writes bit to memory
- Or, then run ISS, write signature file
  - Compare/diff file results (afterwards)
  - This is the approach taken by RISCV International for their architectural validation ("compliance tests")

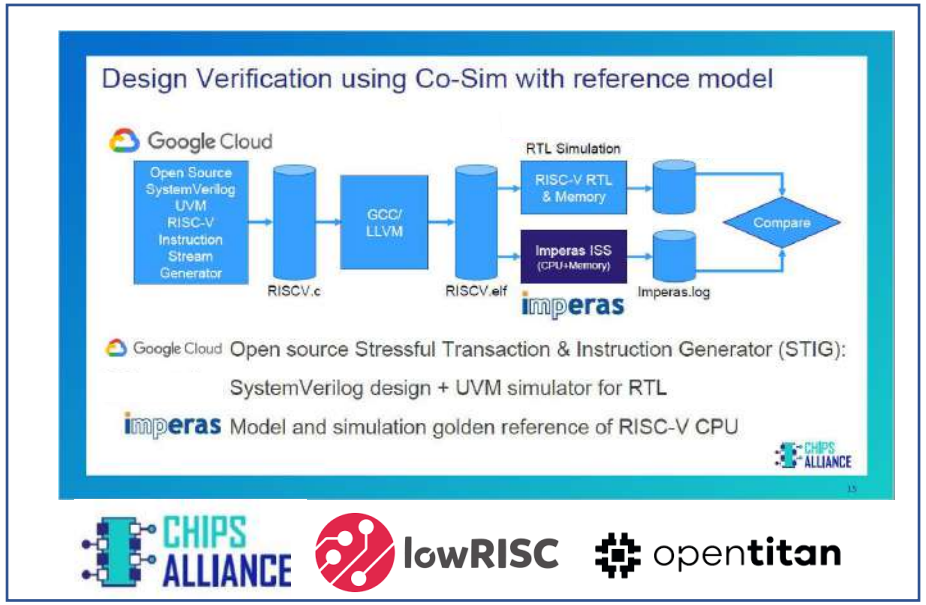
2: Simple check (use e.g. riscvOVPsim ISS from GitHub)

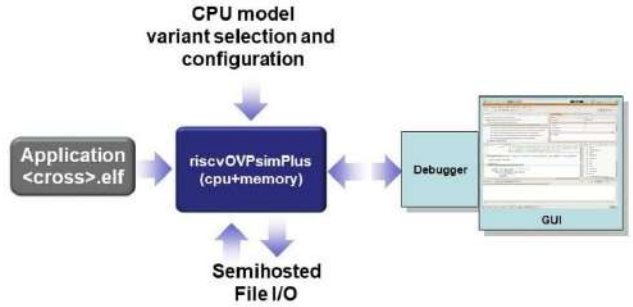
- Summary
  - Very simple, needs basic ISS, and tool chains
    - Free ISS: https://github.com/riscv-ovpsim
    - Free compiler: https://github.com/Imperas/riscv-toolchains
  - Basic bring up
  - Good for simple test runs
  - Basic functionality testing
  - Still need accurate, configurable, version selectable, complete, reference model
  - Not a robust DV solution

Agenda

- RISC-V CPU HW DV approaches
  1. “hello” test
  2. Simple check
  3. Trace-compare
  4. Data-path lockstep-compare
  5. Async lockstep-compare

3: Entry Level DV: post-sim trace-compare (use e.g. riscvOVPsimPlus ISS from OVPworld)





Imperas riscvOVPsimPlus Reference Simulator

- Process
  - use random generator to create tests
  - during simulation of ISS write trace log file
  - during simulation of RTL write trace log file
  - at the end of both runs run through compare program to see failures
- Free riscvOVPsimPlus Includes Trace and GDB interface
  - Free ISS: https://www.ovpworld.org/riscvOVPsimPlus

3: Entry Level DV: post-sim trace-compare (use e.g. riscvOVPsimPlus ISS from OVPworld)

Summary

- Compares files created after test runs
- Can be signature, logging, or instruction trace
- Usually the easiest method to implement (dependent on tracing formats)
  - Capture of program flow (monitor the PC)
  - Capture of program data (monitor the Registers, Memory)
- Potentially very large data files
- Potential for wasteful execution (if early failure)
- Will not work for on async events, control flow, or hardware real time effects, MP, OoO, multi-issue, …
- Not a robust DV solution for commercial cores
- Can engage with Imperas for licenses of reference models and optional development to add customer own instructions, CSR, behaviors

Agenda

- RISC-V CPU HW DV approaches
  1. “hello” test
  2. Simple check
  3. Trace-compare
  4. Data-path lockstep-compare
  5. Async lockstep-compare

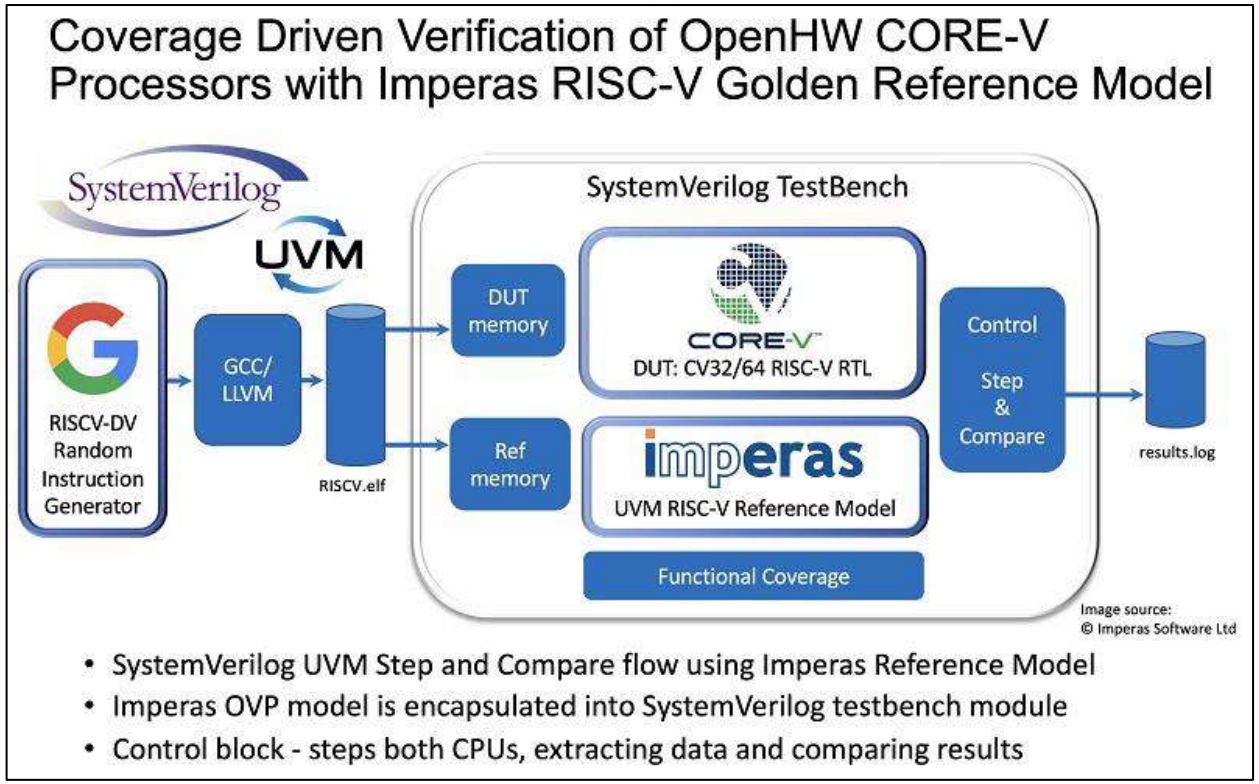
4: Imperas Industrial Quality Sync DV (data-path lockstep)

- Tandem lockstep run – both reference and DUT run together in lock step
- Not very complex to obtain, set up
- Compare PC, CSRs, GPRs, other internal state – instruction by instruction
- No requirement on data saving
- No requirement on known good results

- Will not work for async events and control flow , … – it is all about the data flow

- [OpenHW evolved into using Async – see later slides]

Example flow:



Initial OpenHW flow

4: Imperas Industrial Quality Sync DV (data-path lock-step)

Summary

- Instruction by instruction lockstep comparison (excludes async events)
  - Comparison of execution flow
  - Comparison of program data
  - Comparison of programmers and internal state
- Immediate comparison
  - Allows for debug introspection at point of failure – very powerful
  - Does not waste execution cycles after failure
- Will not work for async events, control flow, or hardware real time effects, …
- Not too hard to develop & set up (depends on RTL tracer features)
- Lock-Step / Compare is by far the best and most efficient approach
  - But does not address async events (see level #5)

- Need to engage with vendors such as Imperas for licenses of reference models and optional development to add customer own instructions, CSR, behaviors

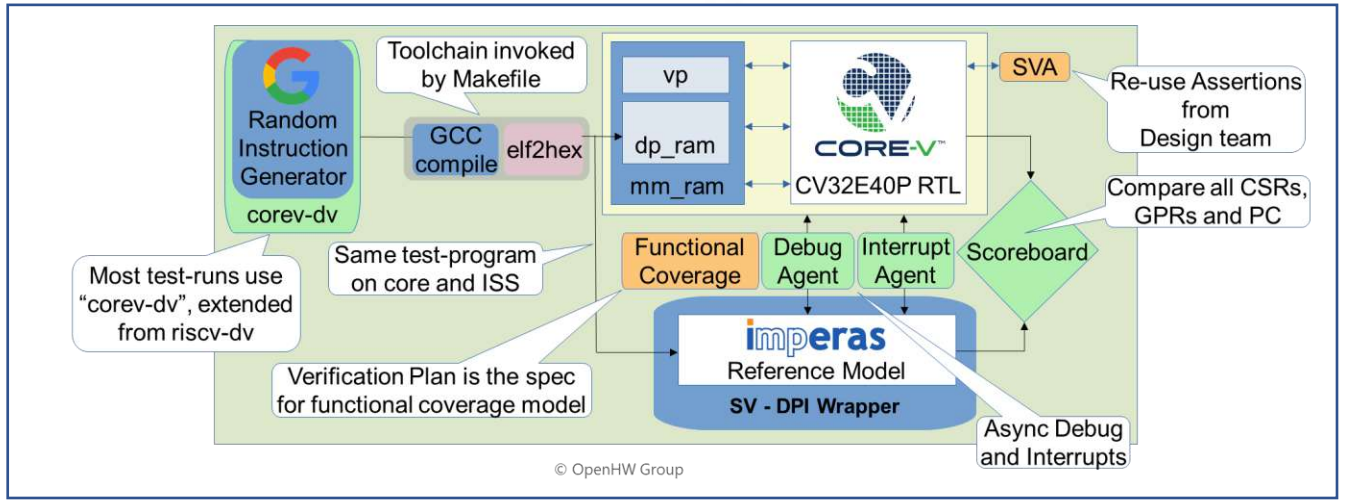
Agenda

- RISC-V CPU HW DV approaches
  1. “hello” test
  2. Simple check
  3. Trace-compare
  4. Data-path lockstep-compare
  5. Async lockstep-compare

5: Imperas Industrial Highest Quality Async DV (async lockstep)

- Builds on & extends Industrial Quality Sync DV
- Adds focus on async capabilities
- Depending on design this can include: OoO, MP, Debug mode, interrupts, multi-issue, …
  - Example SystemVerilog Components
    - tracer: Reports instructions for checking and register writebacks
    - step_and_compare: Manages the reference model and checks functionality
    - interrupt_assert: Properties for interrupt coverage/checking
    - debug_assert: Properties for debug coverage/checking
- Will be hard, complex, and expensive to get working
  - Challenge is extracting async info from micro-architecture RTL pipeline
    - See latest developments with RVVI and ImperasDV

Example flow:



Current CV32E40P OpenHW flow
(Imperas model encapsulated in SystemVerilog)

Imperas Industrial Highest Quality Async DV (async lockstep)

Summary

- Instruction by instruction lockstep comparison (**includes async events**)
  - Comparison of execution flow, of program data, of programmers and internal state
- Immediate comparison
  - Allows for debug introspection at point of failure – very powerful
  - Does not waste execution cycles after failure
- Includes focus on async events, control flow, or hardware real time effects
- Can be hard to develop & set up (depends on RTL tracer features and pipeline understanding)
  - See latest development for RVVI and ImperasDV
- Can be expensive in terms of time, resources, licenses and costs a lot per bug found
  - But the bugs are even more expensive if not found early enough…
- Lockstep / Compare is by far the best and most efficient approach (industry ‘gold standard’)

Agenda

- RISC-V CPU HW DV approaches
  1. “hello” test
  2. Simple check
  3. Trace-compare
  4. Data-path lockstep-compare
  5. Async lockstep-compare
  - Summary

Agenda

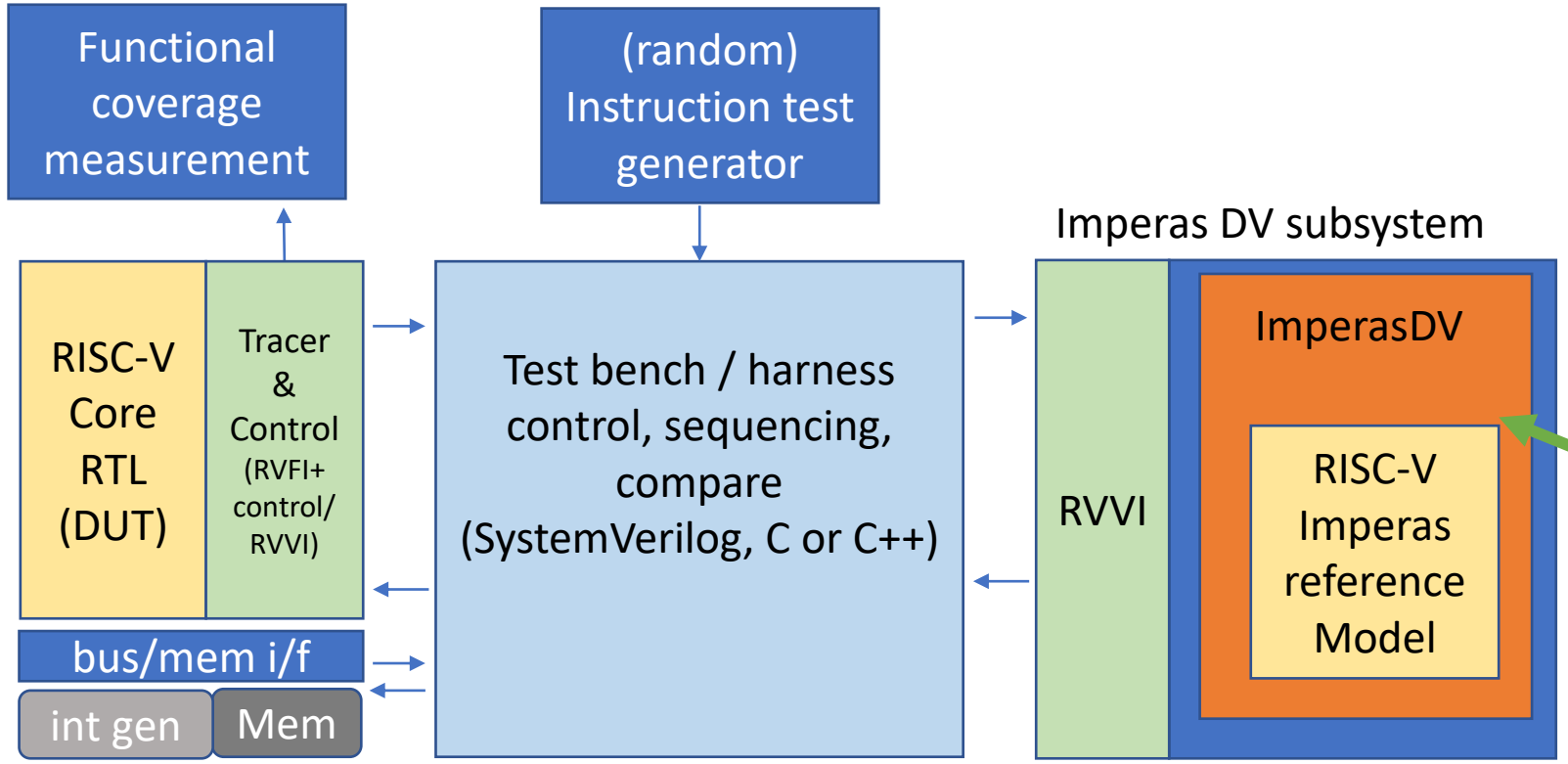
- RISC-V CPU HW DV approaches
- Imperas RISC-V HW DV verification

Introducing ImperasDV

Agenda

- RISC-V CPU HW DV approaches
- Imperas RISC-V HW DV verification
  - Reference model encapsulation

Main blocks in Imperas RISC-V CPU DV



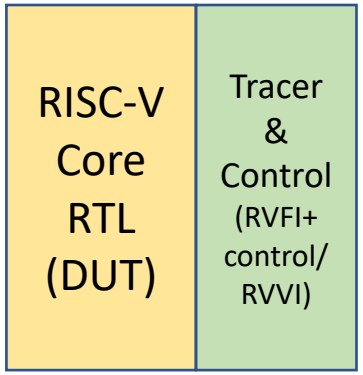
5 components of RISC-V CPU DV

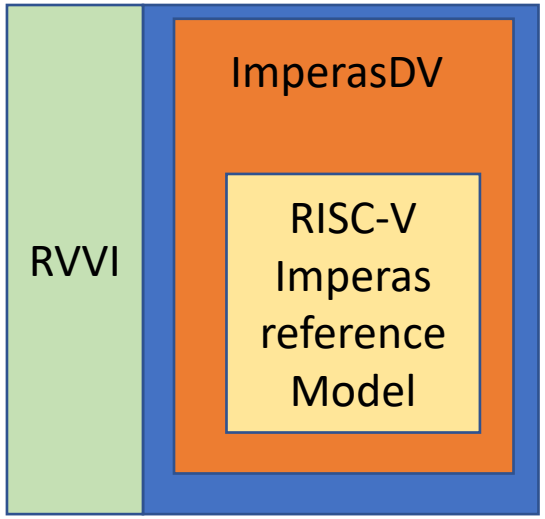
- DUT subsystem with 'tracer'
- (random) instruction test generator
- Functional coverage measurement
- Test bench / harness
- Imperas DV subsystem

Encapsulation of Imperas reference model

NOTE: ImperasDV can be used with SystemVerilog, C, C++, Verilator

Evolving RVVI: RISC-V Verification Interface (3 components, public open standard) [driven by RISC-V DV usage]

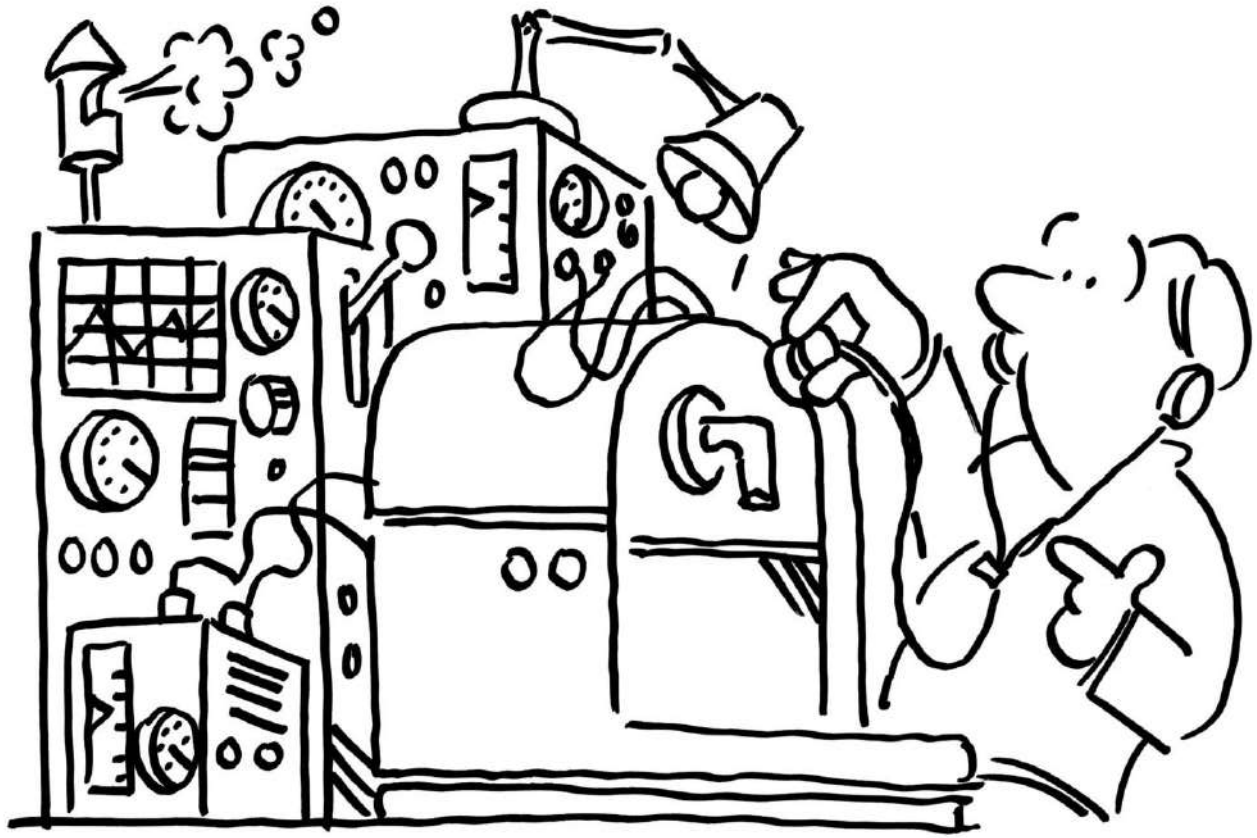




- https://github.com/riscv-verification/RVVI
- RVVI-VLG
  - 4 SystemVerilog Interfaces
    - RVVI_state
    - RVVI_control
    - RVVI_io (Interrupts, Debug)
    - RVVI_bus -(Data, Instruction Bus)
- RVVI-API
  - C/C++
  - SystemVerilog
- RVVI-VPI
  - Virtual Peripheral Interfaces
    - timers, interrupts, debug, random, printer/uart, …
  - Verilog and C macros & examples

Key component is Reference Model

- RISC-V is highly configurable & extendable
  - 160… Questions ?
- So it can get a little …. complicated



Imperas is the Reference

imperas

imperas

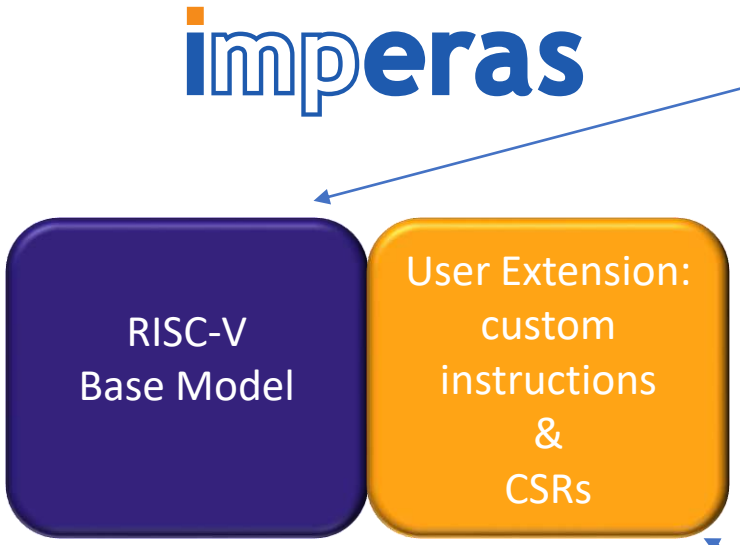
RISC-V Reference Model & Simulator

http://www.imperas.com/riscv

- Imperas provides full RISC-V Specification envelope model
- Industrial quality model /simulator of RISC-V processors for use in compliance, verification and test development
- 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
- Model source included under Apache 2.0 open source license
- Used as reference by :
  - Mellanox/Nvidia, Seagate, NSITEXE/Denso, Google Cloud, Chips Alliance, lowRISC, OpenHW Group, Andes, Valtrix, SiFive, Codasip, MIPS, Nagra/Kudelski, Silicon Labs, RISC-V Compliance Working Group, …

Imperas is used as RISC-V Golden Reference Model

Imperas Model extensibility



- Separate source files and no duplication to ensure easy maintenance
- Imperas or user can develop the extension
- User extension source can be proprietary

Imperas develops and maintains base model

- Base model implements RISC-V specification in full
- Fully configurable to select which ISA extensions
- Fully configurable to select which version of each ISA extension
  - Updated very regularly as ISA extension specification versions change
- Fully 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 model is architected for easy extension & maintenance

Agenda

- RISC-V CPU HW DV approaches
- Imperas RISC-V HW DV verification
- Summary

Summary

- RISC-V processor DV needs lock-step-compare to be of high quality
  - Lock-step is the only way to verify asynchronous behaviors
- Need standards like RVVI to allow component reuse to be efficient
  - For have several different cores, or evolving generations

- **ImperasDV** provides high quality processor verification for adoption within the established SoC Design Verification (DV) flows based on UVM and SystemVerilog.

- Imperas is used as key technology in terms of reference model and DV
  - All you need for high quality, cost-effective RISC-V processor DV… come talk to us
- Imperas: used as a reference by :
  - Mellanox/Nvidia, Seagate, NSITEXE/Denso, Google Cloud, Chips Alliance, lowRISC, OpenHW Group, Andes, Valtrix, SiFive, Codasip, MIPS, Nagra/Kudelski, Silicon Labs, RISC-V Compliance Working Group, …

Thank you

info@imperas.com

www.imperas.com

www.OVPworld.org

For more information on ImperasDV stop by our RISC-V Booth or visit

www.imperas.com/ImperasDV