An Introduction to RISC-V Processor Verification

Larry Lapides

14 March 2023

RISC-V

The RISC-V Disconnect

RISC-V Core User

Expects core quality to be the same as Arm

RISC-V Core Developer

Unlikely to have resources needed be able to develop all the technologies required to perform the same level of verification as Arm

Agenda

- Verification plan
- Methodology choices
- Key technical pieces
- Putting it together: results
- Summary

Agenda

- **Verification plan**
- Methodology choices
- Key technical pieces
- Putting it together: results
- Summary

Verification Plan

- The obvious starting point, but often neglected
- With processor DV 5x more difficult than SoC DV*, you have to have a plan
  - SoC development without processor: 1:1 DV engineers to designers
  - Processor development: 5:1 DV engineers to designers

- What are the metrics for success?
- What resources are available, both people and tools?
- What methodology/methodologies will be used?

* Harry Foster, 2022 verification report, presented at DVCon 2023

Agenda

- Verification plan
- **Methodology choices**
- Key technical pieces
- Putting it together: results
- Summary

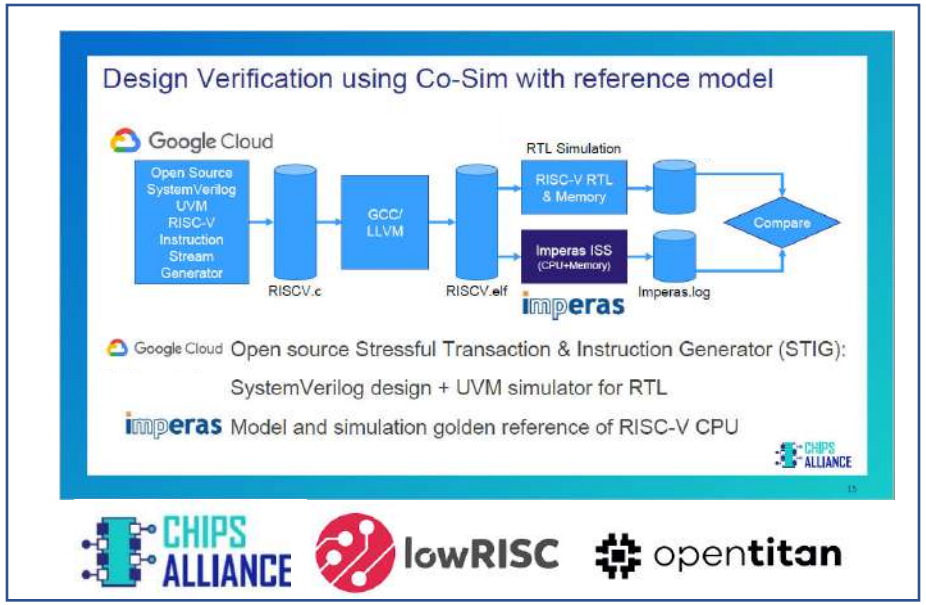
5 Levels of RISC-V Processor DV Methodology

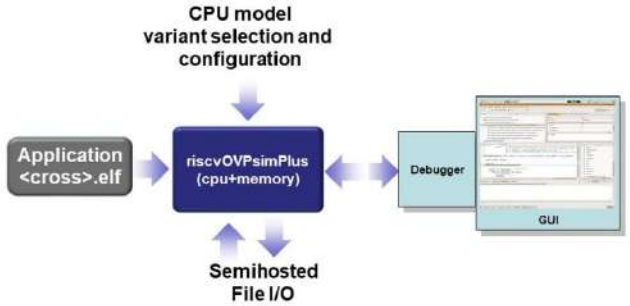
1) Hello World
2) Self-checking tests (e.g. Berkeley torture tests pre-2018)
3) Post-simulation trace log file compare
4) Synchronous step-and-compare
5) Asynchronous continuous compare

5 Levels of RISC-V Processor DV Methodology

1) Hello World
2) Self-checking tests (e.g. Berkeley torture tests pre-2018)
3) Post-simulation trace log file compare
4) Synchronous step-and-compare
5) Asynchronous continuous compare

3) Post-Simulation Trace Log File Compare (Entry Level DV)

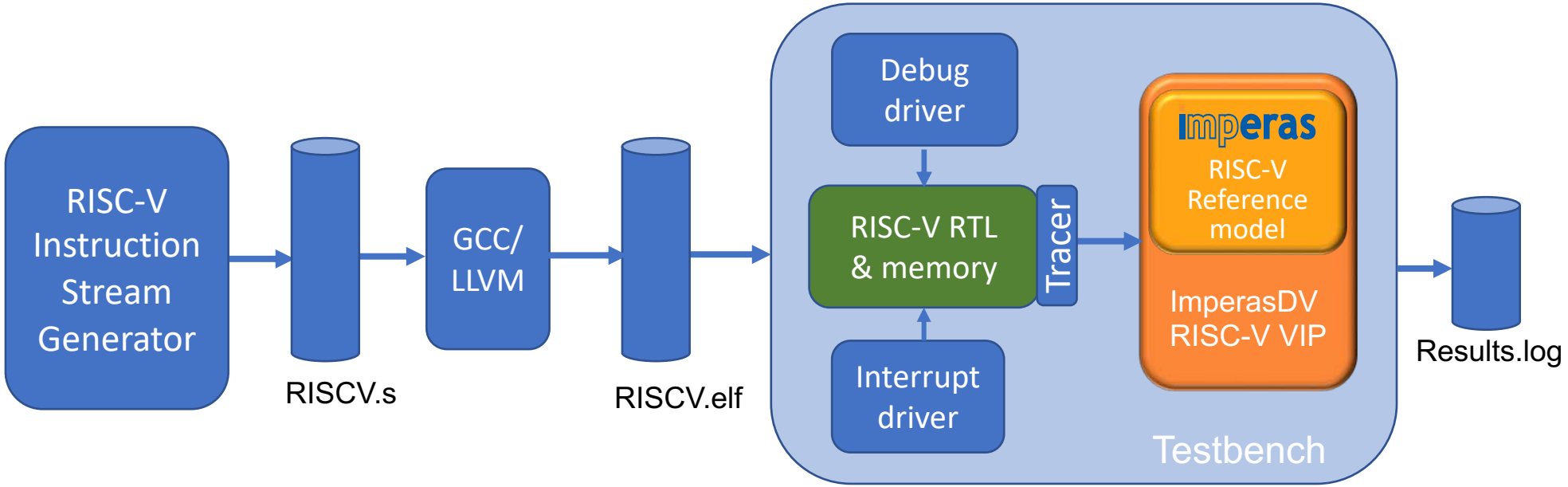




Imperas riscvOVPsimPlus Reference Simulator

- Process
  - use random generator (ISG) 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 logs through compare program to see differences / failures
- ISS: riscvOVPsimPlus includes Trace and GDB interface
  - Free ISS: https://www.ovpworld.org/riscvOVPsimPlus
- ISG: riscv-dv from Google Cloud / Chips Alliance
  - Free ISG: https://github.com/google/riscv-dv

5) Async Continuous Compare
(Highest Quality DV Methodology)



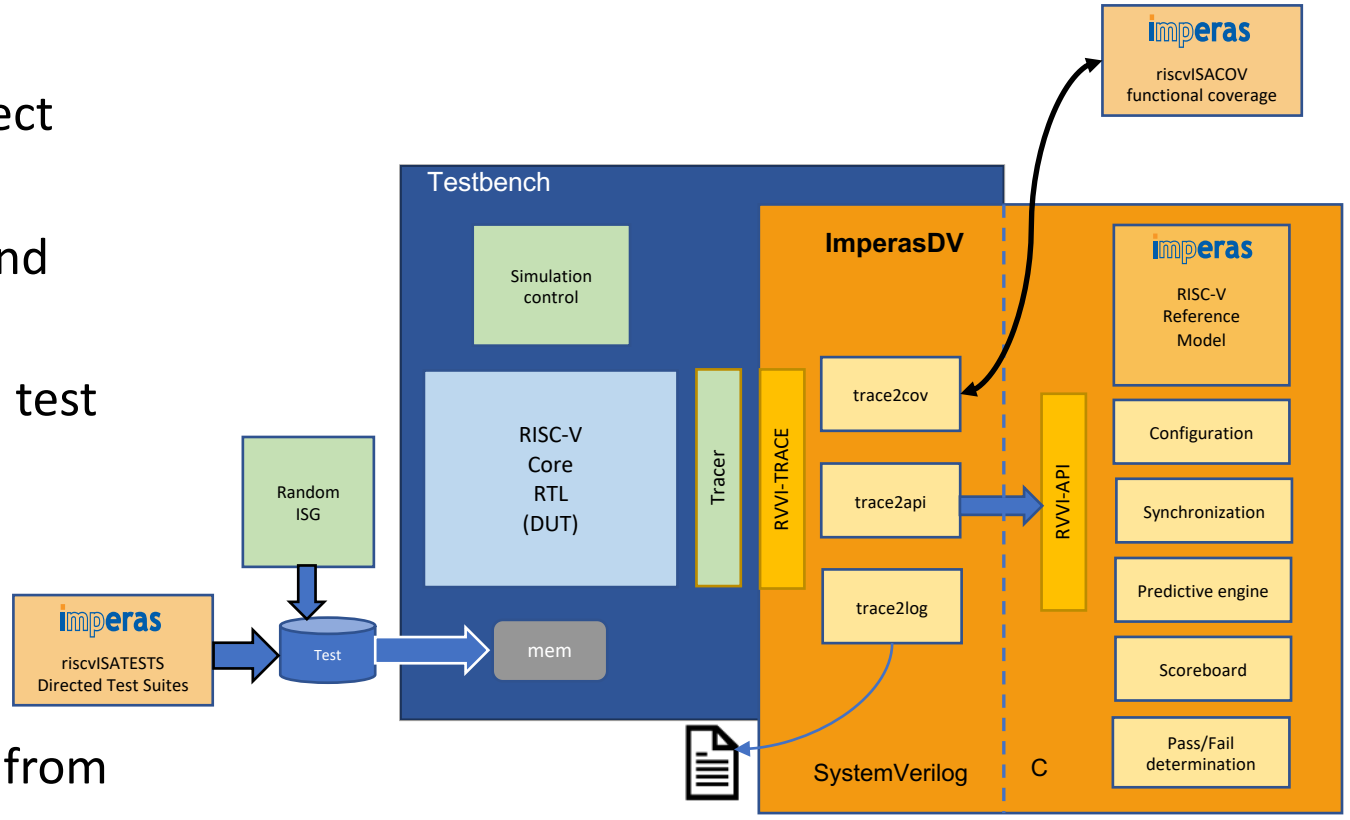
- Asynchronous events are driven into the DUT
- Tracer informs reference model about async events
- Verification IP handles async event predictive engine, scoreboarding, comparison, pass/fail
- ***Asynchronous continuous compare methodology is needed to support features such as interrupts, privilege modes, Debug mode, multi-hart, multi-issue and OoO pipeline, …***

Agenda

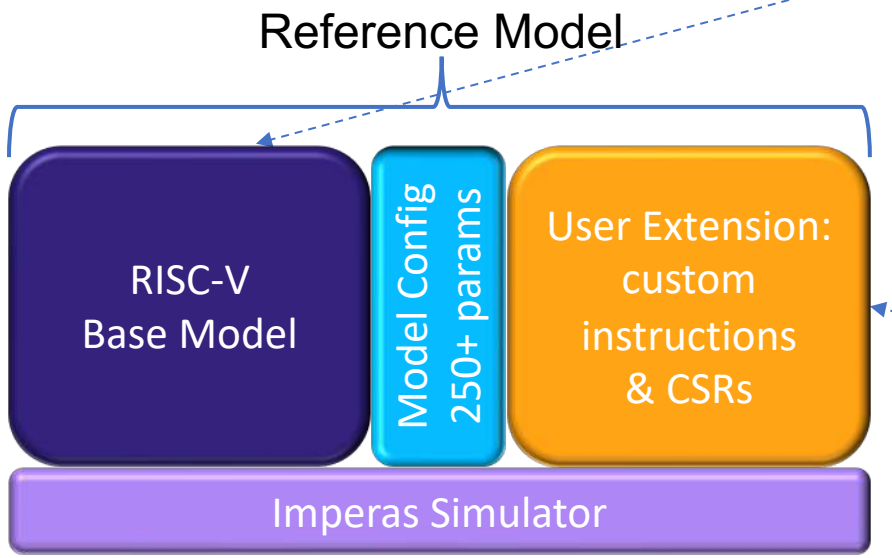
- Verification plan
- Methodology choices
- **Key technical pieces**
- Putting it together: results
- Summary

RISC-V Processor DV Components

- Reference model needed for comparison of correct behavior
- Verification IP provides ease of use, saves time and resources
- RVVI standard provides communication between test bench and reference model subsystem
- Functional coverage modules
- Test stimuli: potentially both directed and constrained random
- DUT with “Tracer” module for extraction of data from the RTL
  - Tracer is not hardware trace implementation
  - Tracer is monitor in RTL for DV purposes



Reference Model Needs to Support the Spec, Enable Custom Instructions, Be High Quality



Imperas develops and maintains Base Model

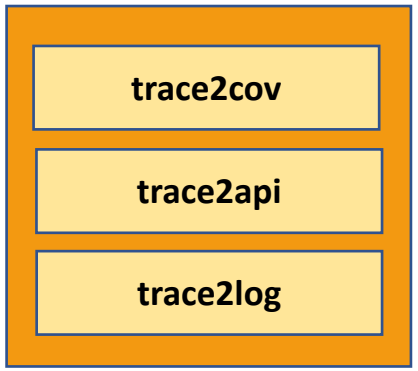
- Base Model implements RISC-V specification in full
- Fully user configurable to select which ISA extensions
- Fully user configurable to select which version of each ISA extension
- Base Model built using Test Driven Development methodology
- Built using public APIs matured over 15 different ISAs

Imperas provides methodology to easily extend base model

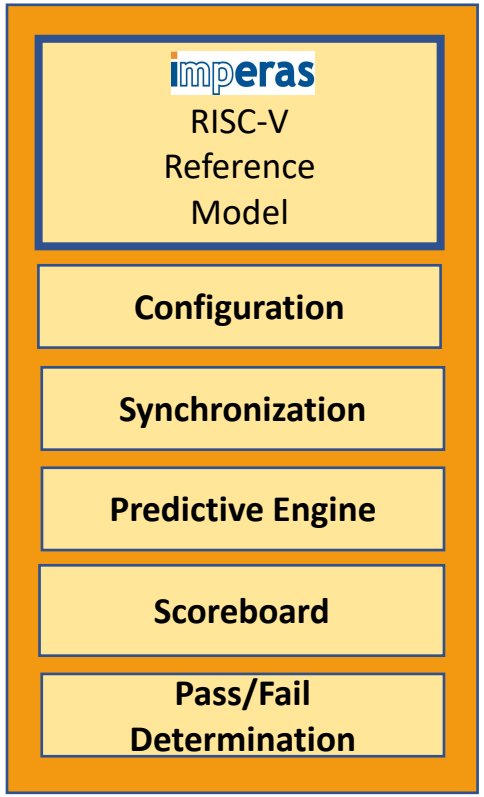
- Custom instructions added using same APIs as in Base Model
- Separate source files and no duplication to ensure easy maintenance
- 100+ page user guide/reference manual with many examples
- User extension source can be proprietary (Apache 2.0 open source license)

- ***RISC-V Base Model is used in all Imperas RISC-V processor models***
- ***RISC-V Base Model is used by > 150 organizations***

Verification IP



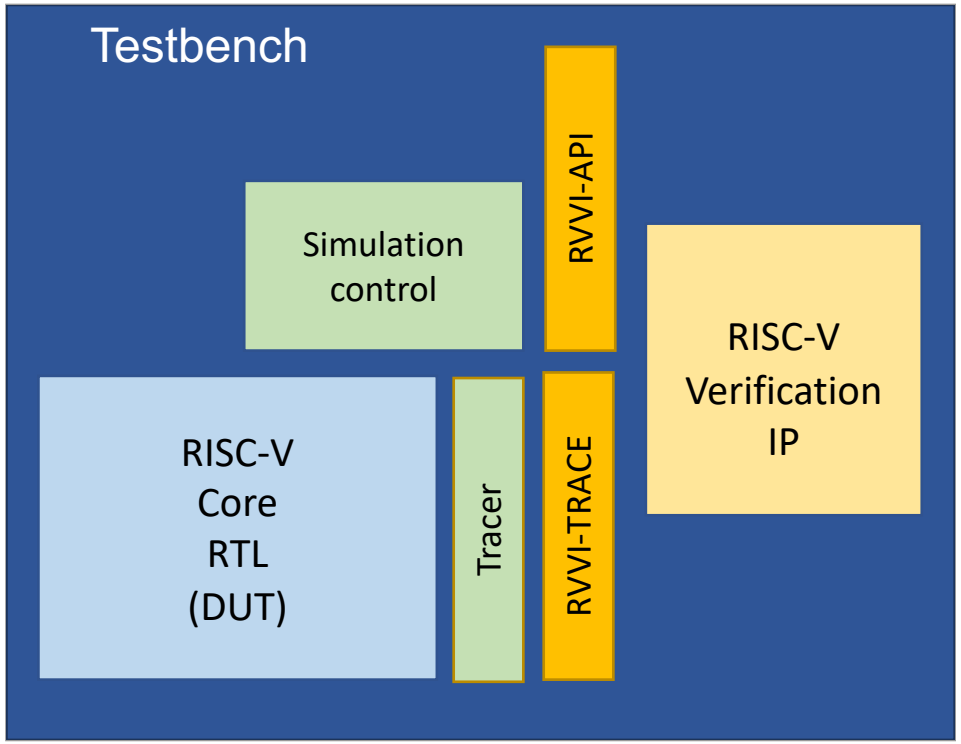
- Data prep for functional coverage
- Data movement from SystemVerilog to C verification IP
- Logging data



- Reference model encapsulation
- Includes DUT reference state storage
- Includes synchronization technology
  - Can run sync, async, interrupts, debug, multi-hart
- Predictive engine is key for asynchronous event DV
- Includes comparison technology
  - Comparisons are done on *DUT/Reference Model processor events*; enables DV of multi-issue and OoO pipeline processors

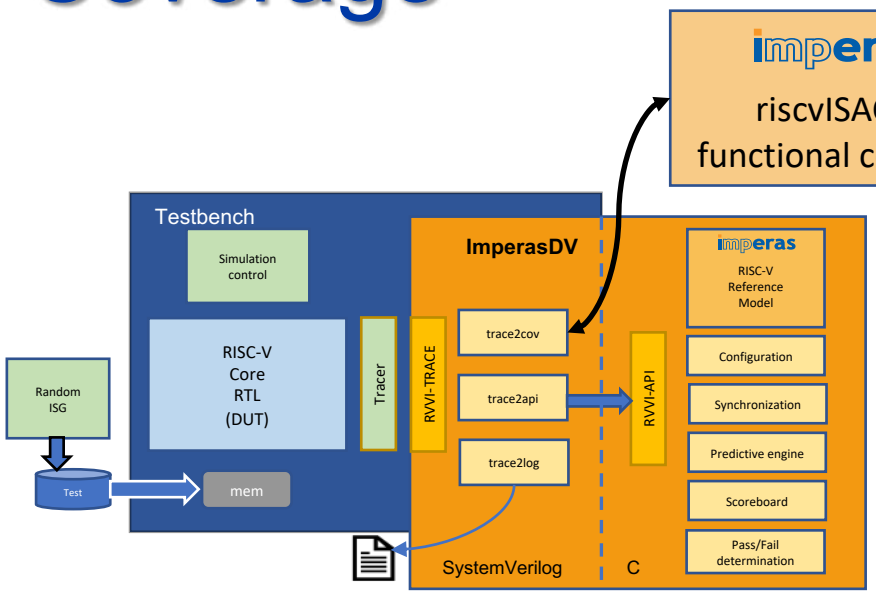
Open Standard RISC-V Verification Interface: RVVI

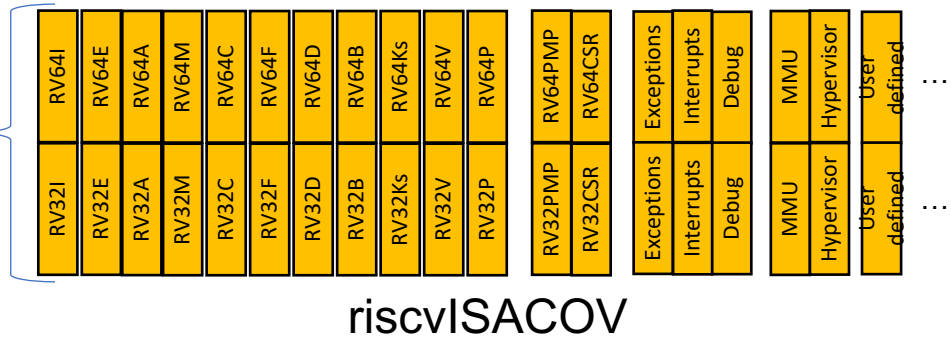
- Work has evolved over 3 years through collaboration with community
- Standardize communication between testbench and RISC-V VIP
- **RVVI-TRACE**: signal level interface to RISC-V VIP
  - SystemVerilog interface
  - Defines information to be extracted by RTL Tracer
- **RVVI-API**: function level interface to RISC-V VIP
  - C language APIs to support Verification IP
  - State information, stepping, compares, …

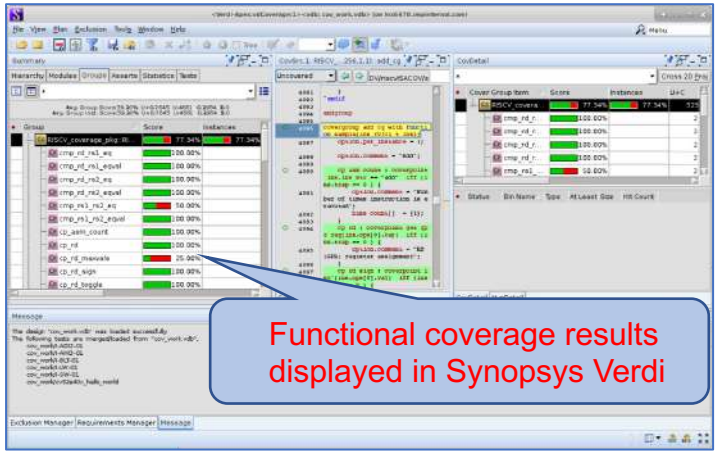


https://github.com/riscv-verification/RVVI

riscvISACOV Is Automatically Generated SystemVerilog Functional Coverage





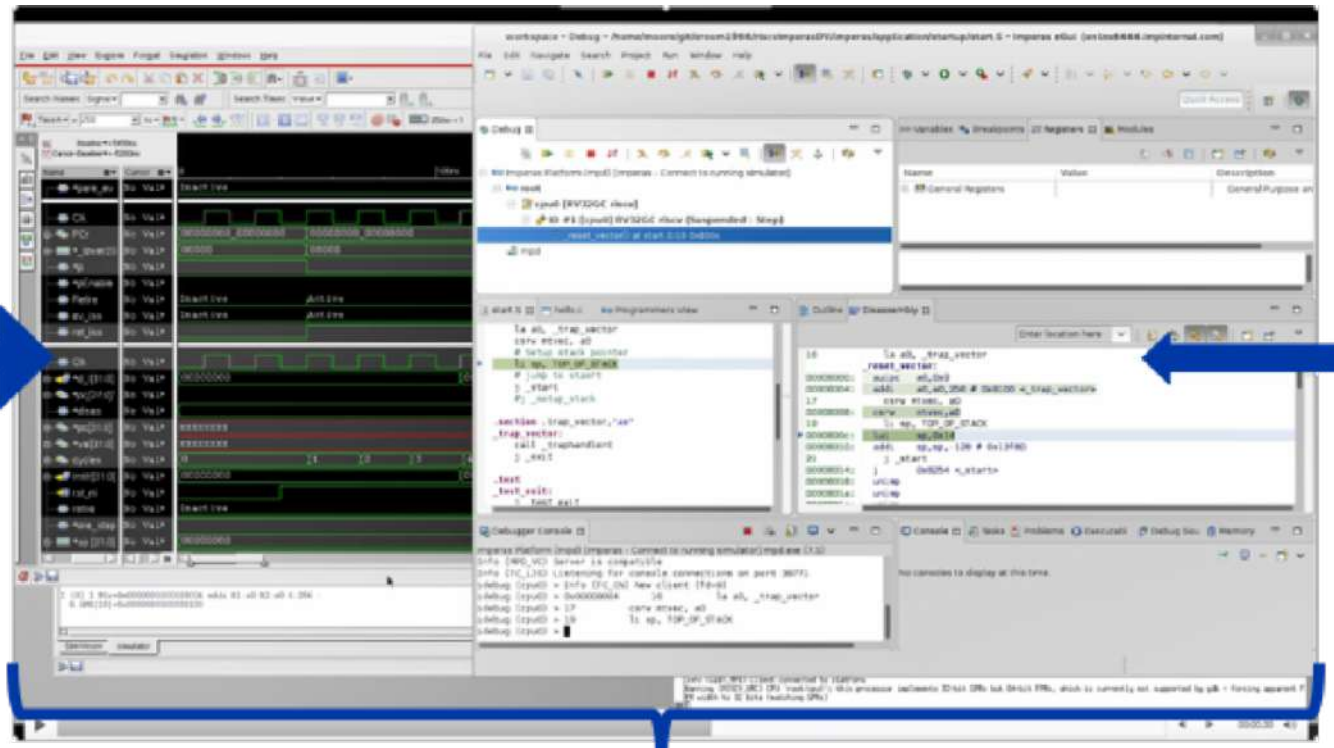


- Building functional coverage is a lot of work
  - 10-40 lines of SystemVerilog for each instruction => minimum of 10K lines of code
  - Then need crosses, etc.
- Imperas riscvISACOV provides functional coverage modules for the basic RISC-V intructions
- Imperas tools can automatically generate functional coverage code for custom instructions

Test Stimuli

- Instruction Stream Generator (ISG) and/or directed tests
- ISG generates test programs using constrained random approach
  - Most often obtain the ISG:
    - Commercial such as Valtrix STING
    - Open source such as Google riscv-dv
  - Require toolchains like GCC, LLVM for assemblers, linkers
  - Require functional coverage so that you know what you've tested!
- Directed tests
  - Free Imperas architectural validation test suites (50+), including RV32/64 I, M, C, F, D, B, K, V, P
    - https://github.com/riscv-ovpsim/imperas-riscv-tests
  - Imperas commercial directed test suites for vector extension, protected memory components
  - RISC-V Intl compliance tests

Interactive Co-Debug is Enabled by ImperasDV Flow

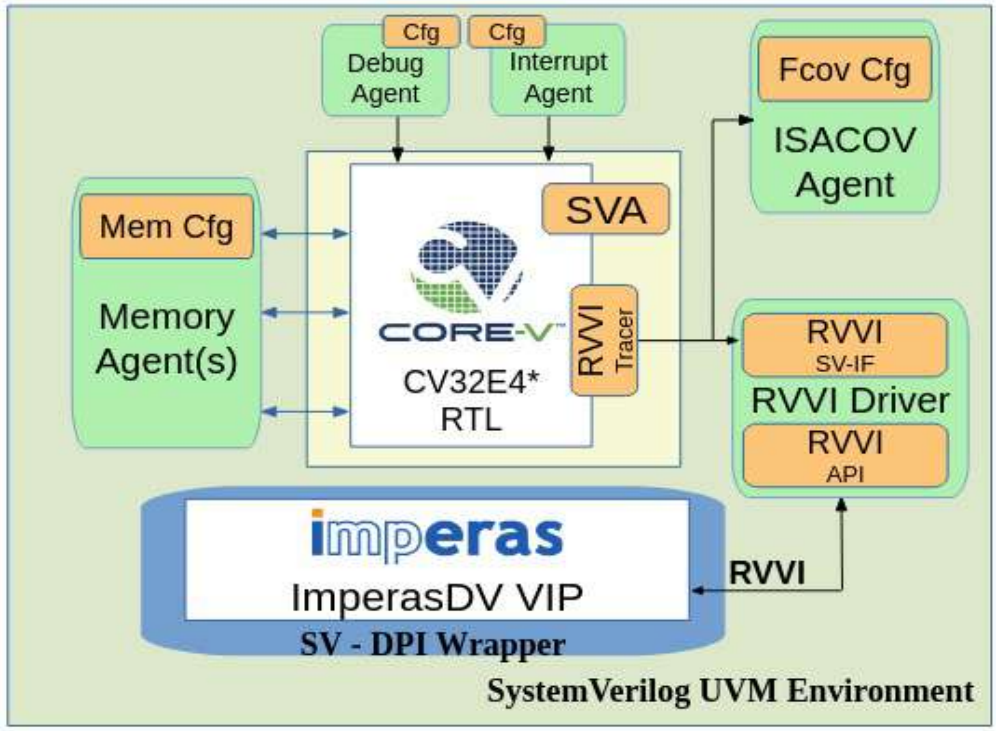


Agenda

- Verification plan
- Methodology choices
- Key technical pieces
- **Putting it together: results**
- Summary

RISC-V Processor DV Results

- OpenHW Core-V-Verif
  - OpenHW users now on 3rd generation of Core-V-Verif DV flow
  - CV32E40P successfully taped out
  - CV32E40Pv2, CV32E40S, CV32E40X, CV32E20 using this flow today; all expected to complete DV this year
- Other successful DV projects using Imperas include Codasip, MIPS, Nagra, NSITEXE, Nvidia Networking, …



Agenda

- Verification plan
- Methodology choices
- Key technical pieces
- Putting it together: results
- **Summary**

Imperas RISC-V Customers and Partners

Most RISC-V processor projects use Imperas

Users

- Marvell Semiconductor
- Nvidia Networking (Mellanox)
- NXP
- Silicon Labs
- Seagate
- Nagravision
- Dolphin Design
- lowRISC (Ibex)
- EM Micro US
- Top 10 semiconductor company with embedded, GPU use cases
- Top-tier systems company (AI application)
- Largest automotive ADAS/AI company
- Startup building accelerator based on multiprocessor RV64
- Japanese government projects “TRASIO” and “RVSPF”
- Numerous universities around the world
- 100+ organizations using free riscvOVPsimPlus

Processor IP Partners

- RISC-V Intl
- Andes
- Cobham Gaisler
- Codasip
- Imagination
- Intel FPGA
- MIPS
- Microchip
- NSITEXE
- OpenHW Group (Imperas is chair of the verification task group)
- SiFive
- Ventana

Tool Partners

- Breker
- Cadence (Palladium integration)
- Google (open source ISG)
- Synopsys
- Valtrix (test generation tools)

riscvOVPsimPlus / riscvISATESTS: Commercial Users

imperas

riscvOVPsimPlus / riscvISATESTS: University & Research Lab Users

imperas

Downloaders from OVPworld of riscvOVPsimPlus / riscvISATESTS (21-feb-2023)

ImperasDV Solves the RISC-V Disconnect

- Verification is the key to the success of the RISC-V processor development projects
- RISC-V processor DV methodology has been evolved by Imperas, together with customers and partners
- Tool costs will be small compared with the engineering resource costs
- Successful, experienced management teams are aware of this and executing on this basis
- ImperasDV solutions provide products to accelerate building a robust DV environment

Thank you

Larry Lapides

LarryL@imperas.com