imperas

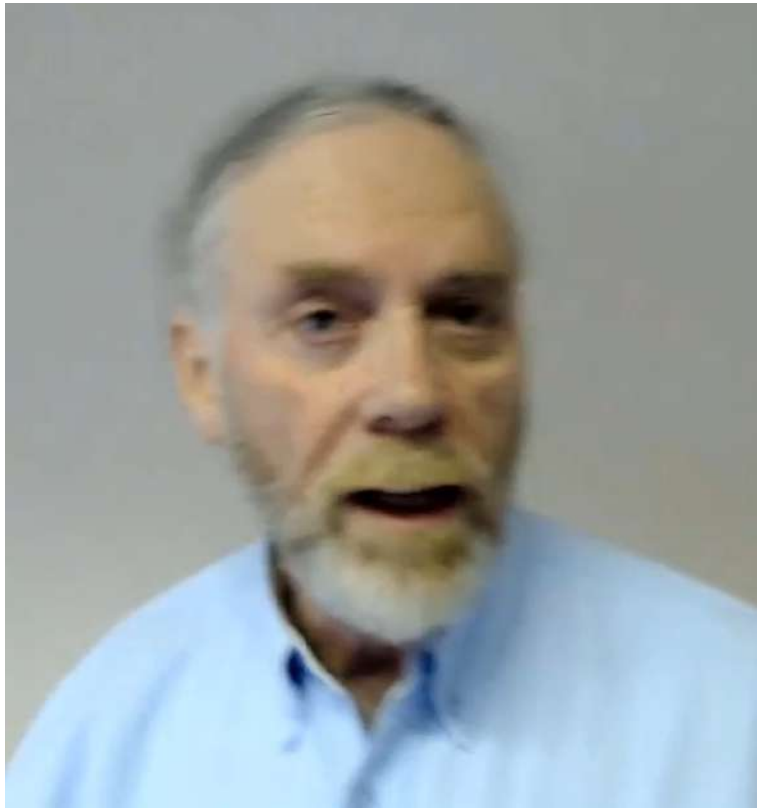
Open-Source RISC-V Cores with Industrial Strength Verification

RISC-V Summit

Lee Moore, Simon Davidmann

Imperas Software

6 December 2021

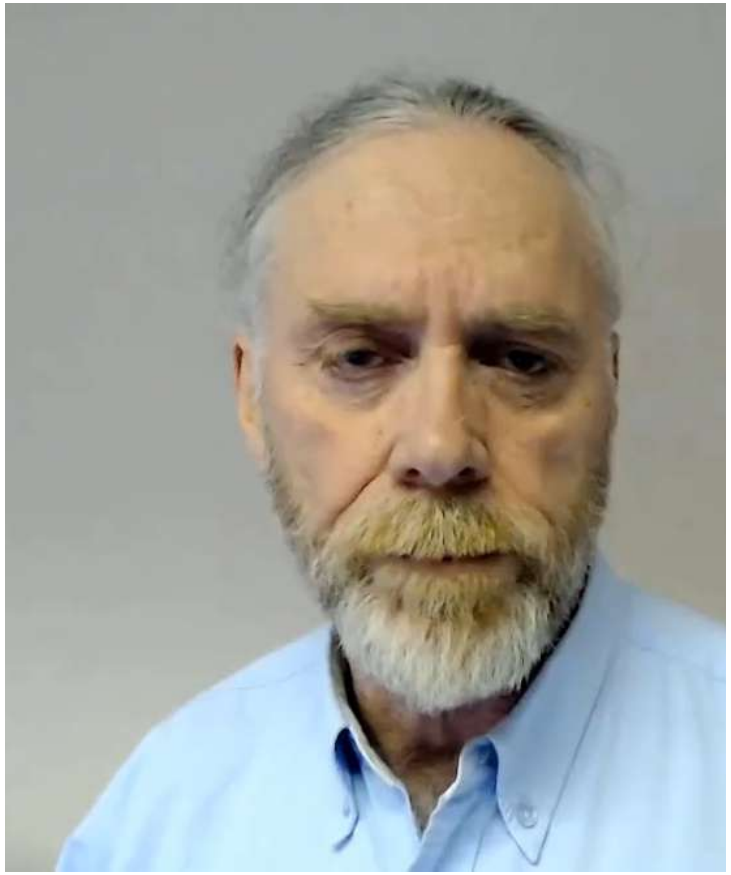


Agenda

- Background of OpenHW
- Evolution of ‘core-v-verif’ core verification environment
- Case Study demonstration
- Components of Industrial Strength DV
  - Adoption of standards
- Summary

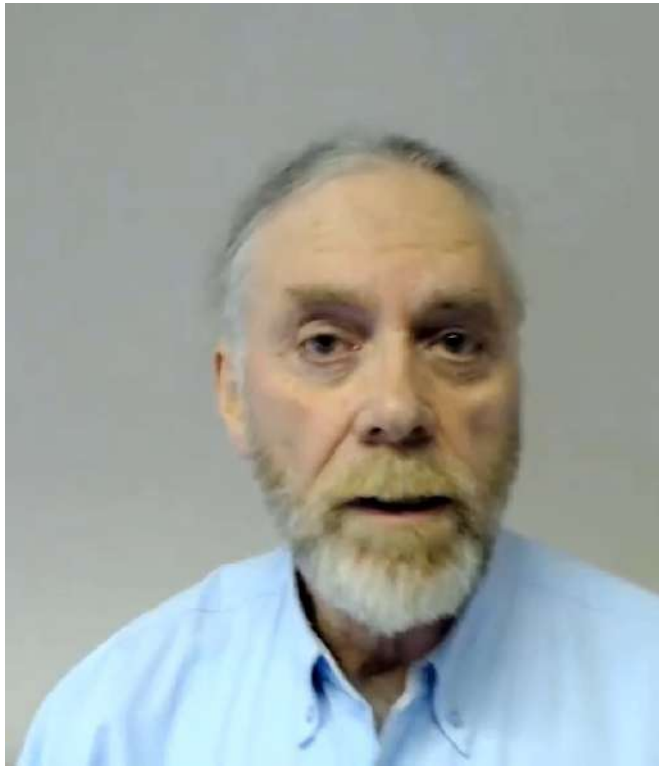
- Global, Non Profit based in Canada, Founded by Rick O'Connor, founding Executive Director of RISC-V Foundation
- Focus: develop open source cores with industrial quality verification
- Founded late 2019, 80+ member companies, 10+ partners
- Original cores evolved from the 'RI5CY' range developed at PULPplatform / ETH Zurich
- Currently 9 RISC-V cores under development
  - 32bit, 64bit, controller/application, bare metal/Linux, etc.

- Developed quality DV testbench and flows: 'core-v-verif'
- Imperas partners providing 'golden reference' RISC-V models & DV technology



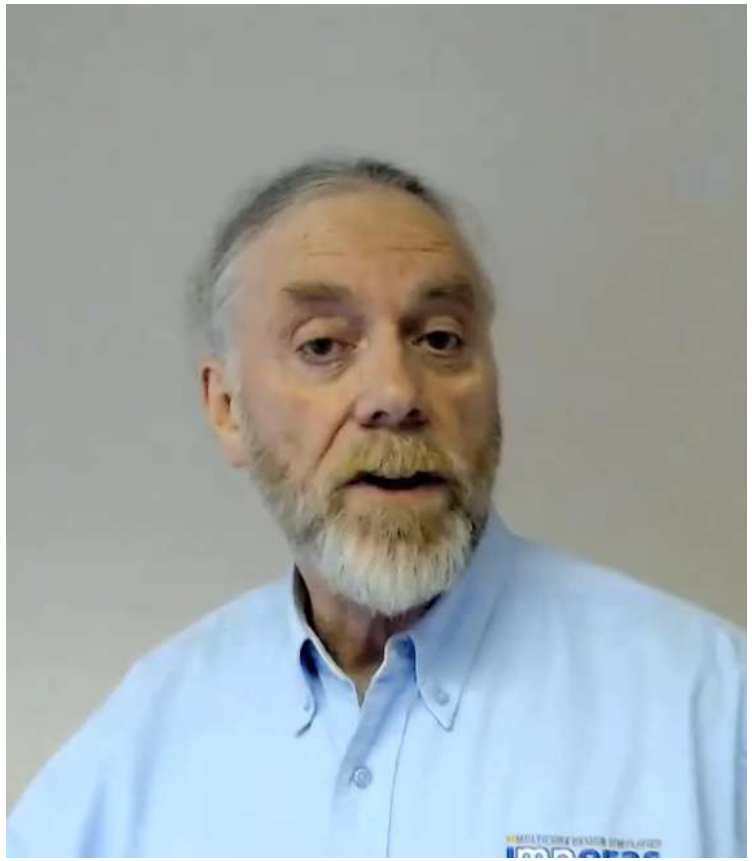
Agenda

- Background of OpenHW
- Evolution of ‘core-v-verif’ core verification environment
- Case Study demonstration
- Components of Industrial Strength DV
  - Adoption of standards
- Summary

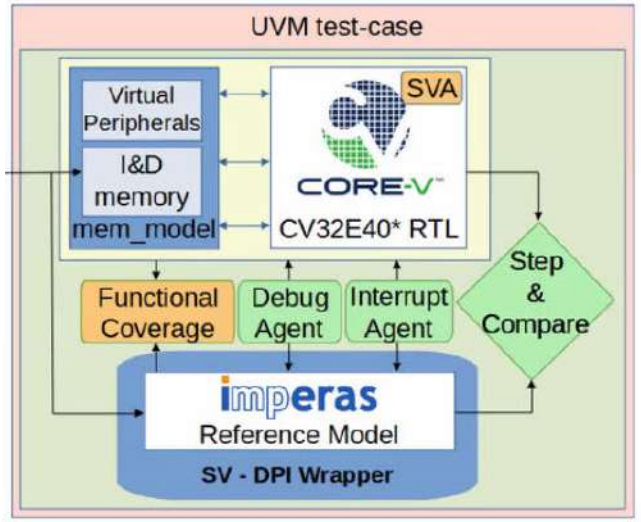


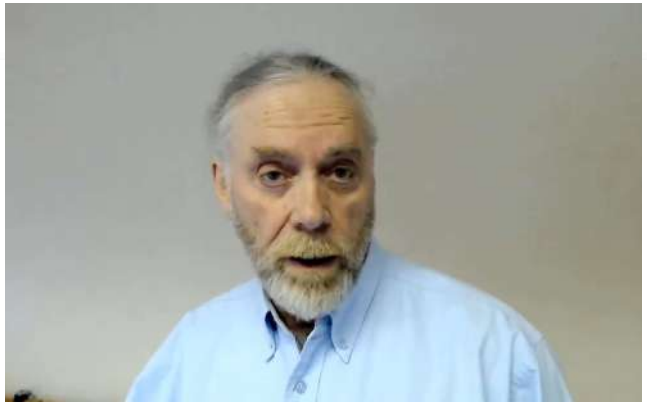
OpenHW ‘core-v-verif’

- Provides a robust, comprehensive simulation environment for the cores:
  - RISC-V 32bit: CV32E40P, CV32E40X, CV32E40S, CV32A6, …
  - RISC-V 64bit: CV64A6, …
- Freely available on github at openhwgroup/core-v-verif
- Industrial-strength verification
  - SystemVerilog UVM environment
  - Runs on any commercial SystemVerilog-compatible simulator
  - Complete code coverage
  - Well-defined comprehensive functional coverage
  - Open and complete verification plans for each core



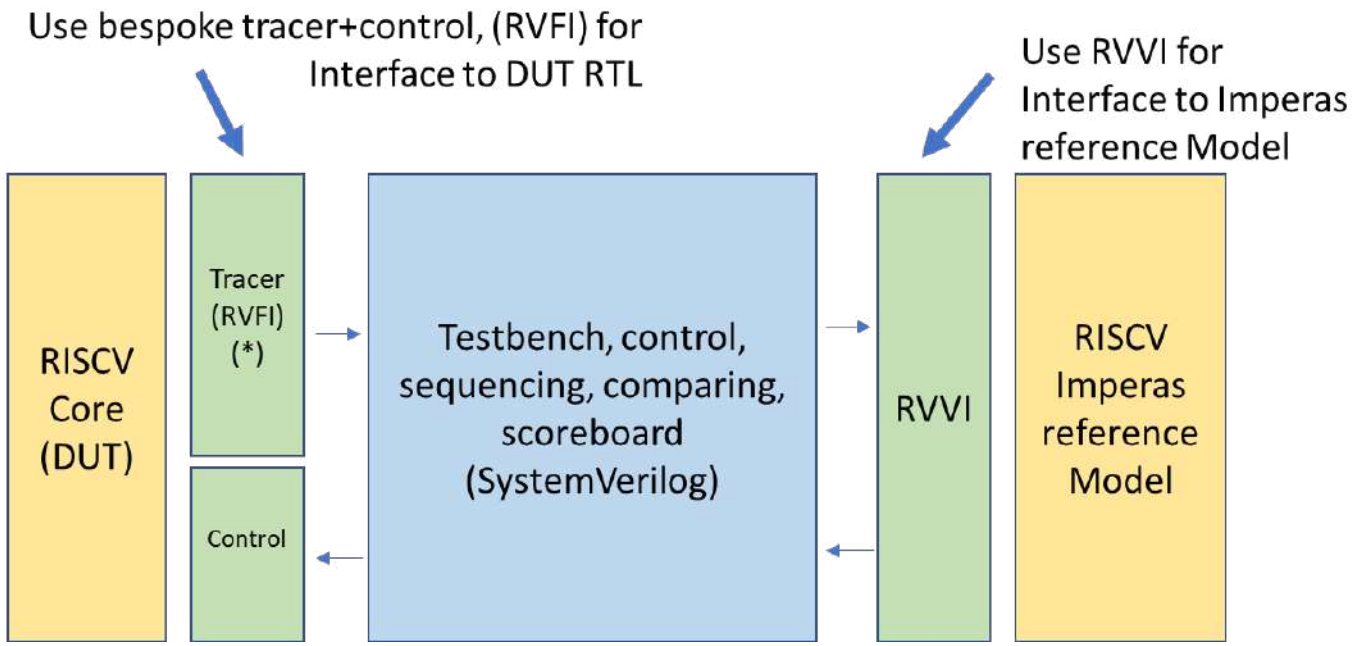
‘core-v-verif’ in 2020





- Runs core RTL and Imperas reference model co-sim
- Includes sync and async lock-step-compare
- For interrupts and debug includes
  - random stimulus
  - properties and assertions
  - functional coverage
- Uses directed and constrained random tests
- Encapsulation of Imperas OVP reference model in SystemVerilog
- Challenge was support for many different cores

‘core-v-verif’ in 2021 – evolution to using standards



(*) RVFI code is used as base for the DUT tracer/streaming i/f - has extensions as required

RVVI is new developing open standard (RISC-V Verification Interface) (more on later slides)

Agenda

- Background of OpenHW
- Evolution of ‘core-v-verif’ core verification environment
- Case Study demonstration
- Components of Industrial Strength DV
  - Adoption of standards
- Summary

Imperas is the Reference

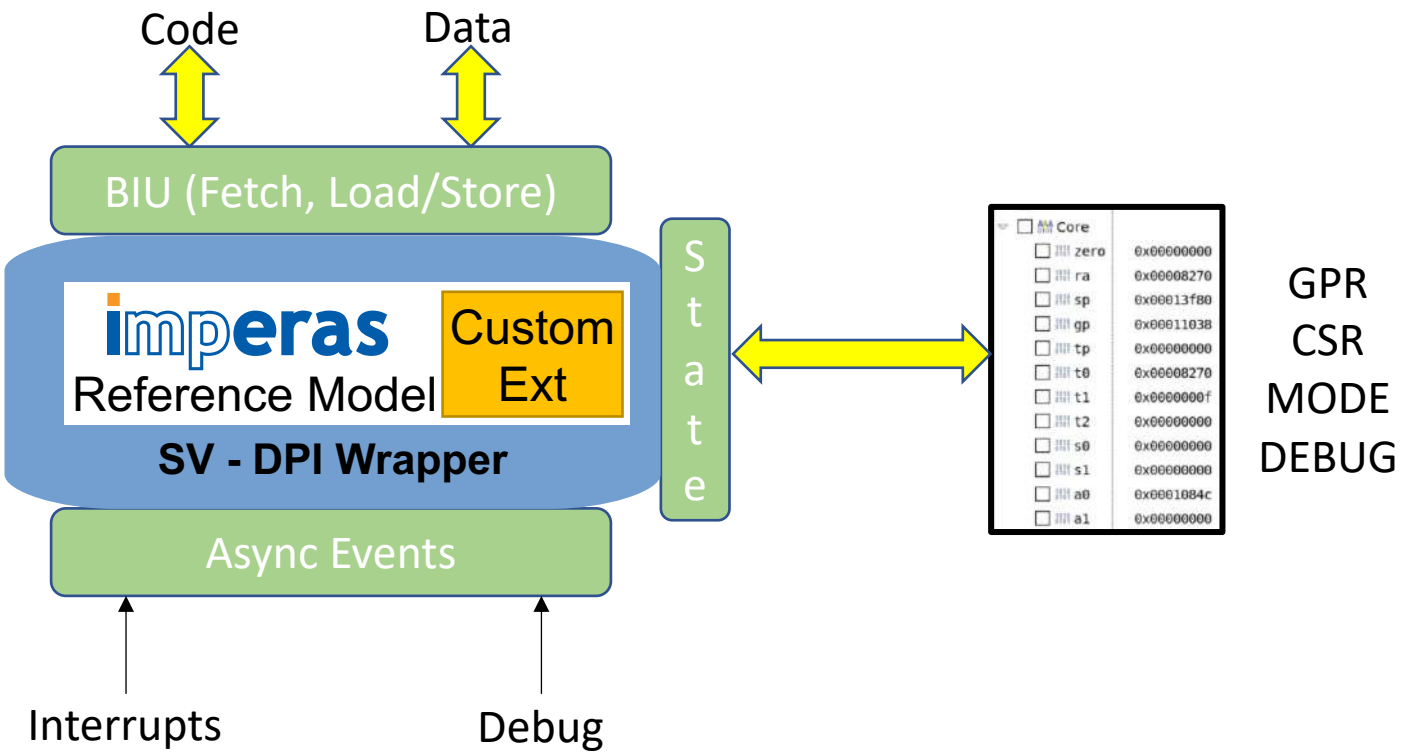
RISC-V Reference Model & Simulator

http://www.imperas.com/riscv

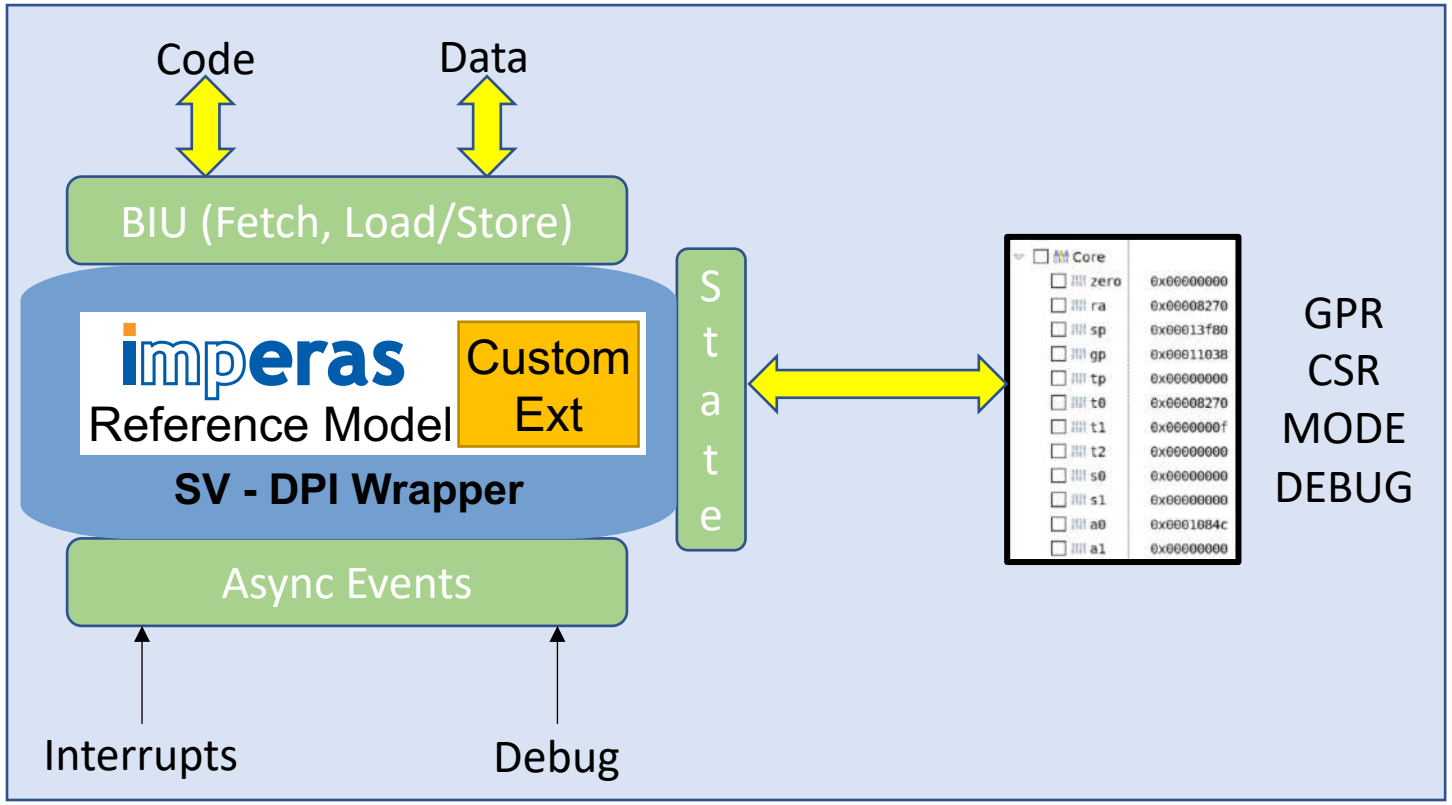
- Imperas provides full RISC-V Specification envelope model
- Industrial quality model /simulator of RISC-V processors for use in compliance, verification and test development
- Complete, fully functional, configurable model / simulator
  - All 32bit and 64bit features of ratified User and Privilege 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

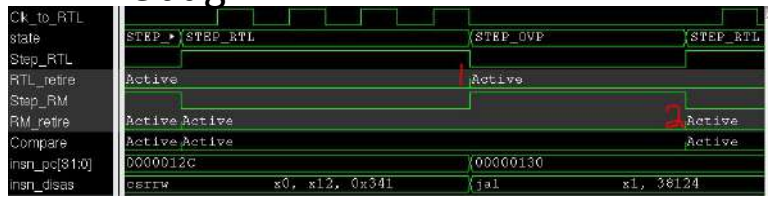
Reference Model Encapsulation



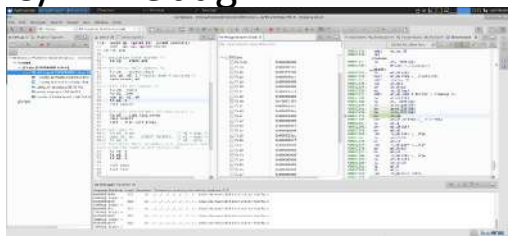
Reference Model Debug / Analysis Capabilities



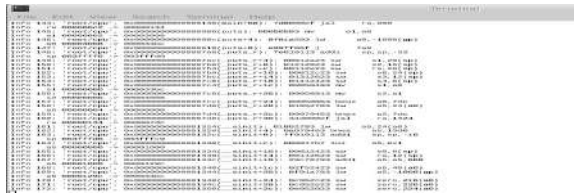
RTL Debug



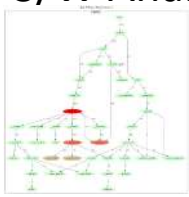
S/W Debug



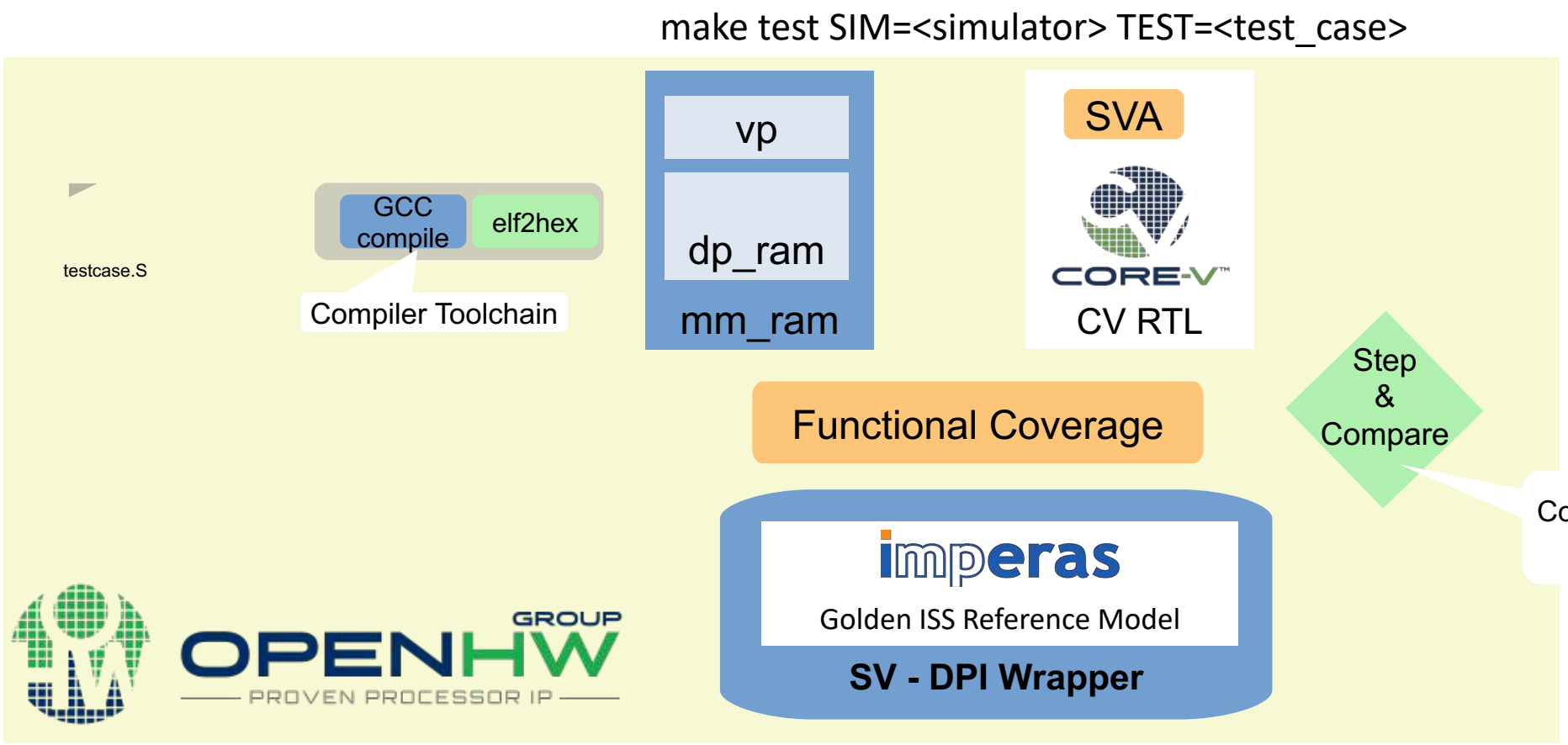
S/W Trace

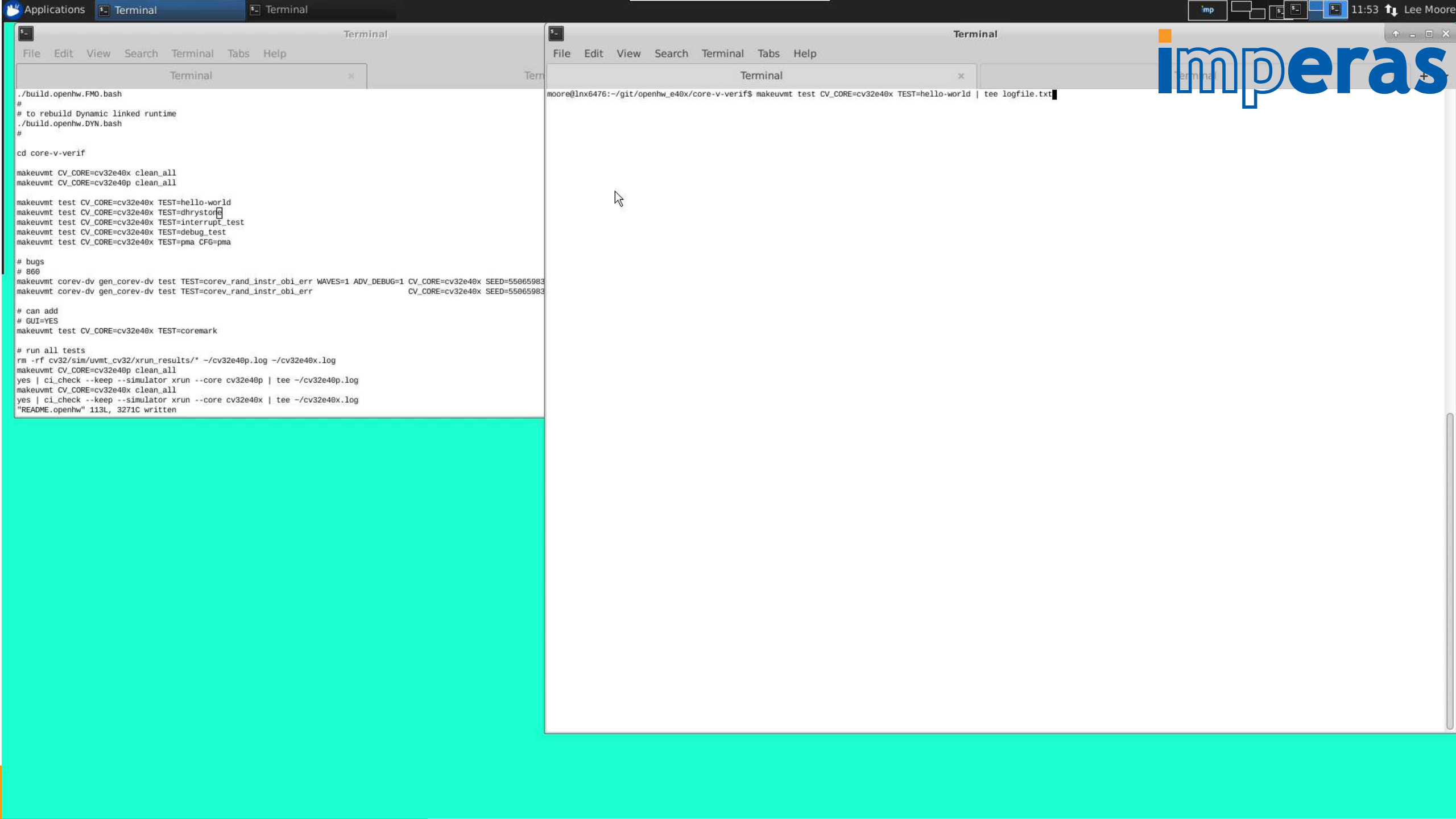


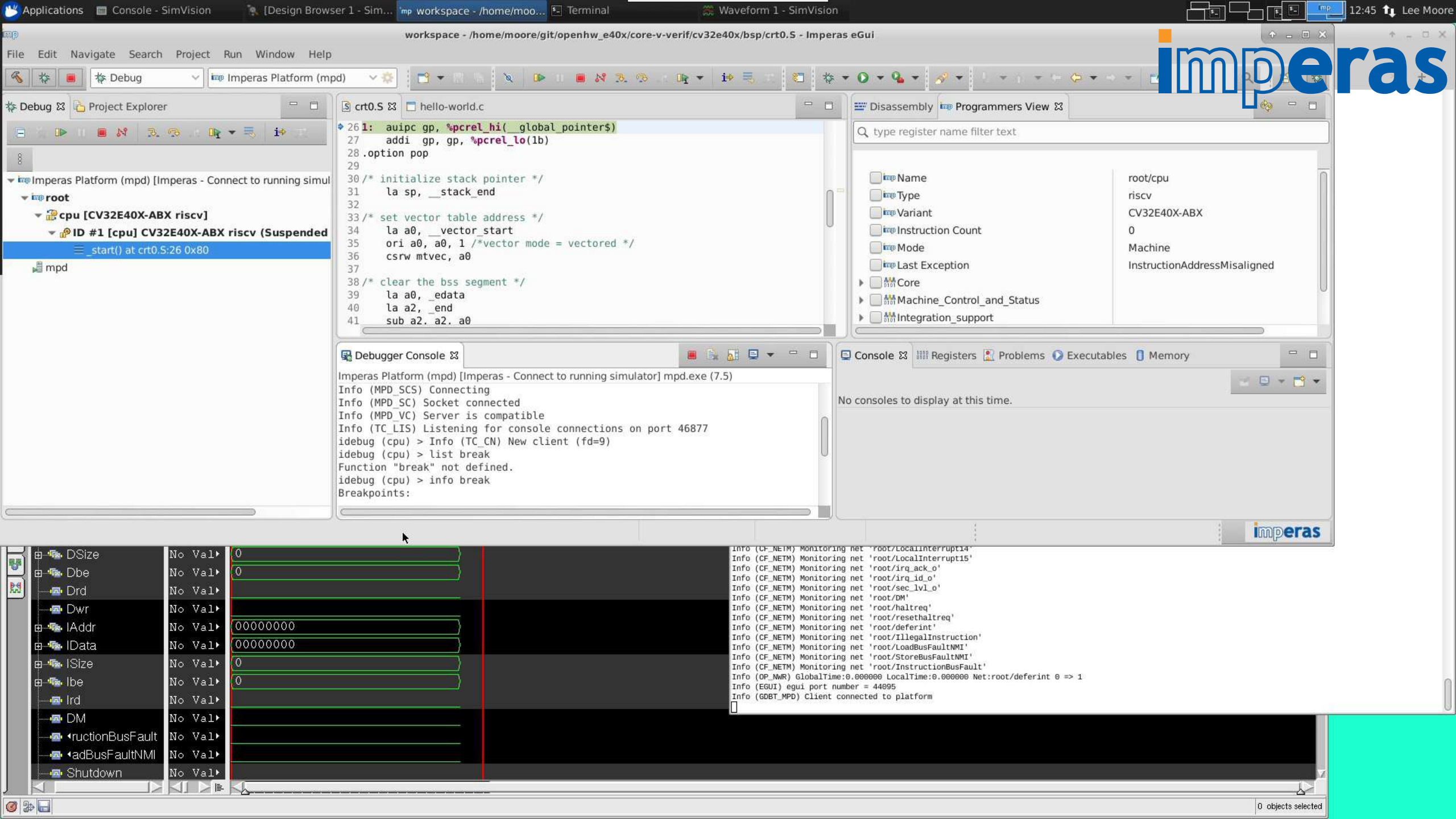
S/W Analysis



OpenHW cv32e40p Step/Compare UVM Testbench




Applications
Terminal
Terminal
11:53
Lee Moore
Terminal
File Edit View Search Terminal Tabs Help
Terminal
./build.openhw.FMO.bash
#
to rebuild Dynamic linked runtime
./build.openhw.DYN.bash
#
cd core-v-verif
makeuvmt CV_CORE=cv32e40x clean_all
makeuvmt CV_CORE=cv32e40p clean_all
makeuvmt test CV_CORE=cv32e40x TEST=hello-world
makeuvmt test CV_CORE=cv32e40x TEST=dhrystone
makeuvmt test CV_CORE=cv32e40x TEST=interrupt_test
makeuvmt test CV_CORE=cv32e40x TEST=debug_test
makeuvmt test CV_CORE=cv32e40x TEST=pma CFG=pma
bugs
860
makeuvmt corev-dv gen_corev-dv test TEST=corev_rand_instr_obi_err WAVES=1 ADV_DEBUG=1 CV_CORE=cv32e40x SEED=55065983
makeuvmt corev-dv gen_corev-dv test TEST=corev_rand_instr_obi_err CV_CORE=cv32e40x SEED=55065983
can add
GUI=YES
makeuvmt test CV_CORE=cv32e40x TEST=coremark
run all tests
rm -rf cv32/sim/uvmt_cv32/xrun_results/* ~/cv32e40p.log ~/cv32e40x.log
makeuvmt CV_CORE=cv32e40p clean_all
yes | ci_check --keep --simulator xrun --core cv32e40p | tee ~/cv32e40p.log
makeuvmt CV_CORE=cv32e40x clean_all
yes | ci_check --keep --simulator xrun --core cv32e40x | tee ~/cv32e40x.log
"README.openhw" 113L, 3271C written
Terminal
File Edit View Search Terminal Tabs Help
Terminal
moore@lnx6476:~/git/openhw_e40x/core-v-verif$ makeuvmt test CV_CORE=cv32e40x TEST=hello-world | tee logfile.txt
imperas


Applications
Console - SimVision
[Design Browser 1 - Sim...
workspace - /home/moo...
Terminal
Waveform 1 - SimVision
12:45
Lee Moore
workspace - /home/moore/git/openhw_e40x/core-v-verif/cv32e40x/bsp/crt0.S - Imperas eGui
File Edit Navigate Search Project Run Window Help
Debug
Imperas Platform (mpd)
Debug
Project Explorer
Imperas Platform (mpd) [Imperas - Connect to running simul
root
cpu [CV32E40X-ABX riscv]
ID #1 [cpu] CV32E40X-ABX riscv (Suspended
_start() at crt0.S:26 0x80
mpd
crt0.S
hello-world.c
26 1: auipc gp, %pcrel_hi(__global_pointer$)
27 addi gp, gp, %pcrel_lo(1b)
28 .option pop
29
30 /* initialize stack pointer */
31 la sp, __stack_end
32
33 /* set vector table address */
34 la a0, __vector_start
35 ori a0, a0, 1 /*vector mode = vectored */
36 csrw mtvec, a0
37
38 /* clear the bss segment */
39 la a0, _edata
40 la a2, _end
41 sub a2, a2, a0
Disassembly
Programmers View
type register name filter text
Name root/cpu
Type riscv
Variant CV32E40X-ABX
Instruction Count 0
Mode Machine
Last Exception InstructionAddressMisaligned
Core
Machine_Control_and_Status
Integration_support
Debugger Console
Imperas Platform (mpd) [Imperas - Connect to running simulator] mpd.exe (7.5)
Info (MPD_SCS) Connecting
Info (MPD_SC) Socket connected
Info (MPD_VC) Server is compatible
Info (TC_LIS) Listening for console connections on port 46877
idebug (cpu) > Info (TC_CN) New client (fd=9)
idebug (cpu) > list break
Function "break" not defined.
idebug (cpu) > info break
Breakpoints:
Console
Registers
Problems
Executables
Memory
No consoles to display at this time.
imperas
DSize No Val 0
Dbe No Val 0
Drd No Val
Dwr No Val
IAddr No Val 00000000
IData No Val 00000000
ISize No Val 0
Ibe No Val 0
Ird No Val
DM No Val
ructionBusFault No Val
adBusFaultNMI No Val
Shutdown No Val
Info (CF_NETM) Monitoring net 'root/LocalInterrupt14'
Info (CF_NETM) Monitoring net 'root/LocalInterrupt15'
Info (CF_NETM) Monitoring net 'root/irq_ack_o'
Info (CF_NETM) Monitoring net 'root/irq_id_o'
Info (CF_NETM) Monitoring net 'root/sec_lvl_o'
Info (CF_NETM) Monitoring net 'root/DM'
Info (CF_NETM) Monitoring net 'root/haltreq'
Info (CF_NETM) Monitoring net 'root/resethaltreq'
Info (CF_NETM) Monitoring net 'root/deferint'
Info (CF_NETM) Monitoring net 'root/IllegalInstruction'
Info (CF_NETM) Monitoring net 'root/LoadBusFaultNMI'
Info (CF_NETM) Monitoring net 'root/StoreBusFaultNMI'
Info (CF_NETM) Monitoring net 'root/InstructionBusFault'
Info (OP_NWR) GlobalTime:0.000000 LocalTime:0.000000 Net:root/deferint 0 => 1
Info (EGUI) egui port number = 44095
Info (GDBT_MPD) Client connected to platform
0 objects selected

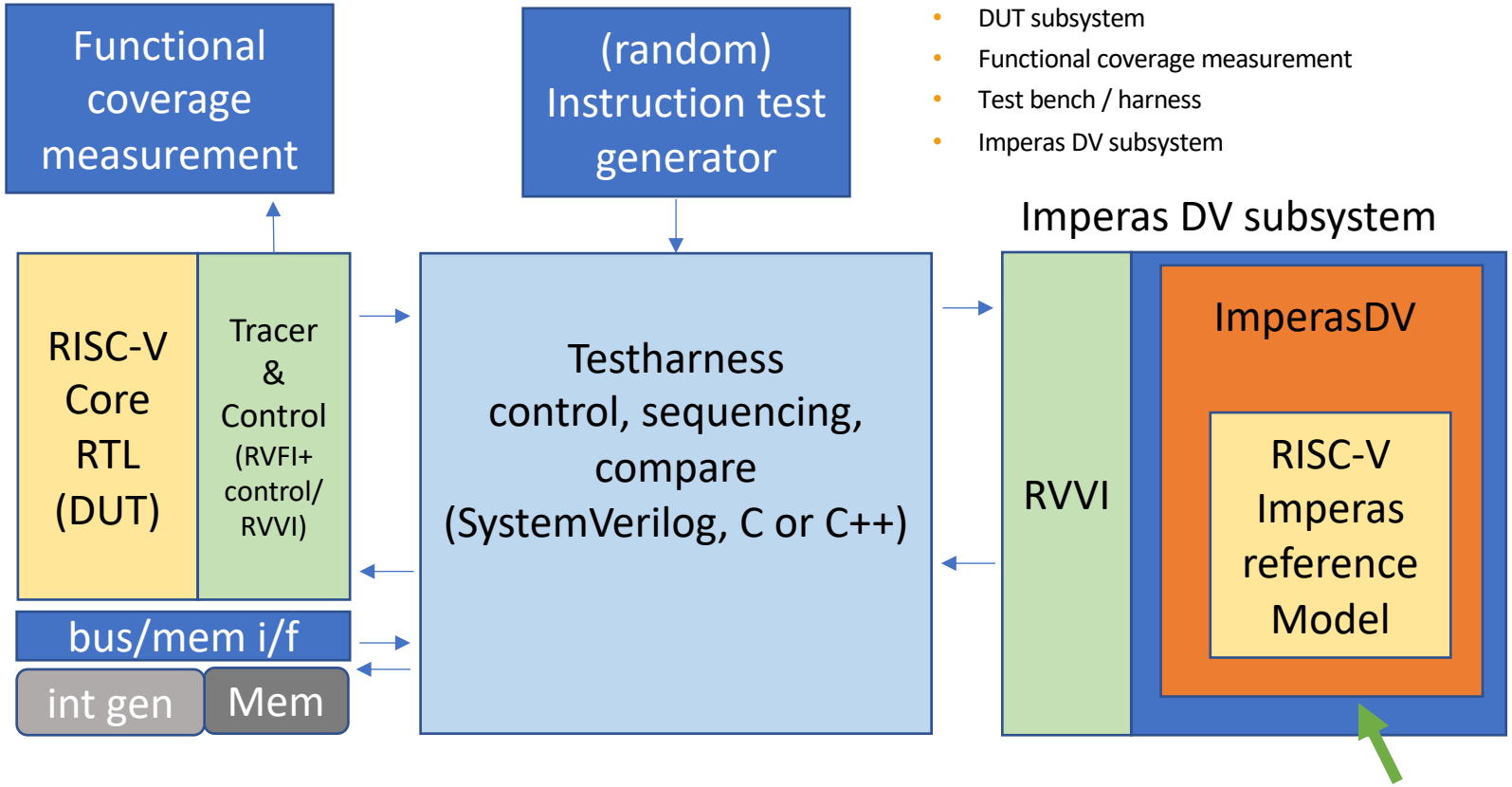
Agenda

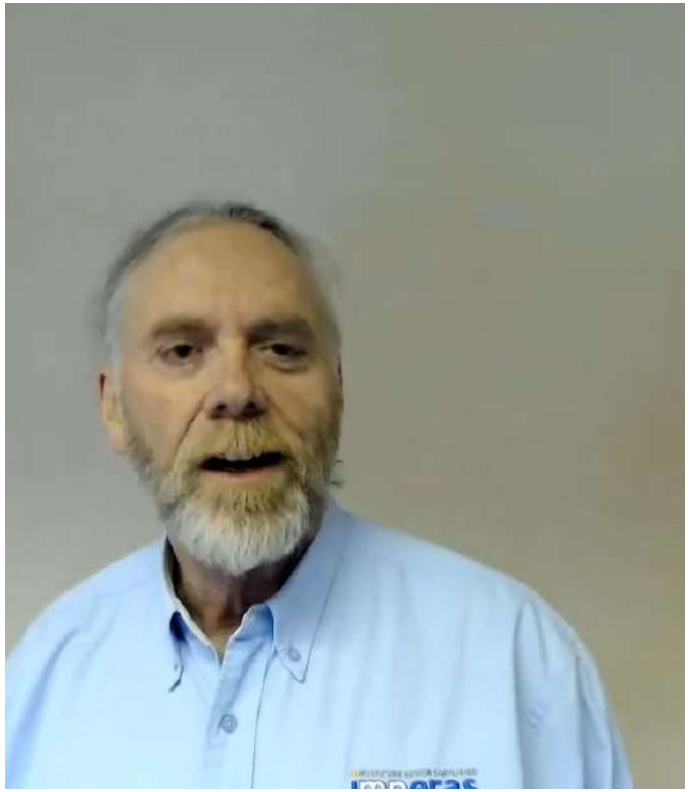
- Background of OpenHW
- Evolution of ‘core-v-verif’ core verification environment
- Case Study demonstration
- Components of Industrial Strength DV
  - Adoption of standards
- Summary

Components of Industrial Strength DV

5 components of RISCV CPU DV

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

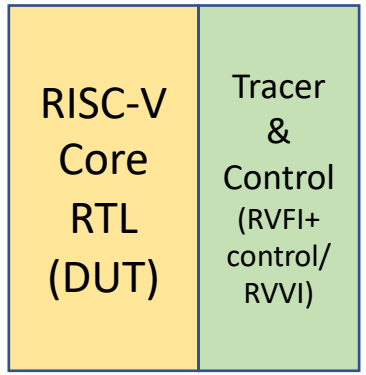


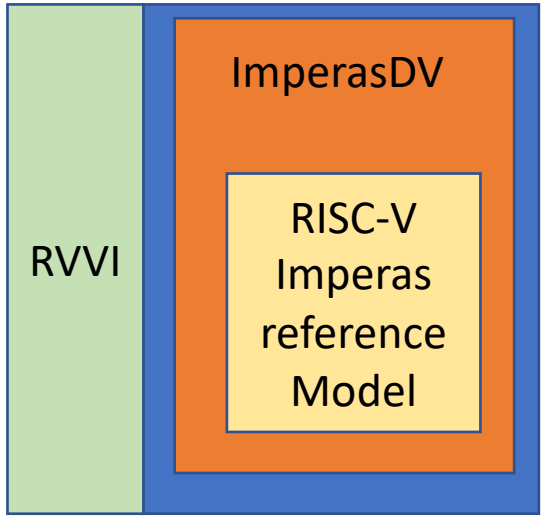


NOTE: ImperasDV can be used with SystemVerilog, C, C++, Verilator

Encapsulation of Imperas reference model

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





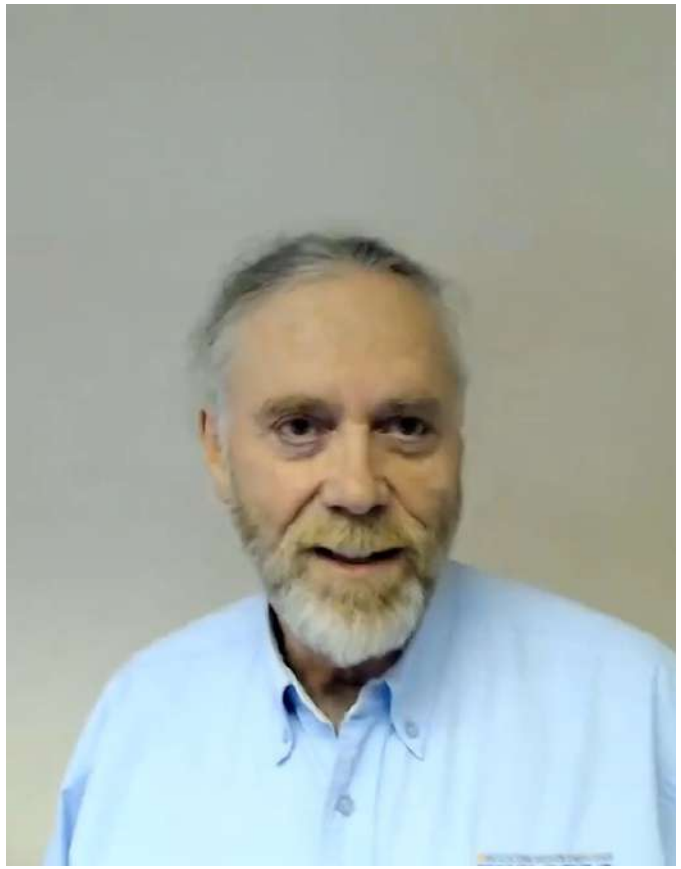
- https://github.com/riscv-verification/RVVI
- RVVI-VLG
  - 4 SystemVerilog Interfaces
    - RVVI_state
    - RVVI_control
    - RVVI_io (Interrupts, Debug)
    - RVVI_bus -(Data, Instruction Bus)
- RVVI-API
  - C/C++
  - SystemVerilog
- RVVI-VPI (work-in-progress)
  - Virtual Peripheral Interfaces
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

Agenda

- Background of OpenHW
- Evolution of ‘core-v-verif’ core verification environment
- Case Study demonstration
- Components of Industrial Strength DV
  - Adoption of standards
- 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
- Openhw core-v-verif is a high quality test bench
  - Open source means you can clone and use it if you modify / extend the OpenHW cores
  - And also you can make use of it or its components in your test benches for other cores
- 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/riscv

www.OVPworld.org/riscv