Introduction to RISC-V CPU design verification

Kevin McDermott, Imperas Software
Agenda

• Welcome and agenda

• Introduction to RISC-V DV

• Imperas News Highlights for DVCon 2022

• Conclusions
Imperas at DVCon 2022

• Tutorial: ‘Introduction to the 5 levels of RISC-V Processor Verification’
  • Simon Davidmann and Lee Moore – Imperas Software
  • Monday February 28th 9:00-11:00am PST

• Presentation: ‘Introduction to RISC-V CPU design verification’
  • Kevin McDermott – Imperas Software
  • Tuesday March 1st 12:30-1:00 pm PST

• Presentation: ‘Imperas RISC-V Design Verification solutions’
  • Larry Lapides – Imperas Software
  • Tuesday March 1st 1:00-1:30pm PST
Introduction to RISC-V CPU design verification
Highlights for this talk

• RISC-V is changing the options that SoC designers have in their tool kits
• RISC-V means many teams are designing new processors, or modifying source of processors
• For RISC-V anybody can be ‘an architecture licensee’
• And every CPU needs verifying... in detail... (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 RISC-V CPU DV
• And discusses pros and cons of the different approaches
• Also it introduces the main components needed in any RISC-V processor DV environment
Introduction to Imperas
Involvement with RISC-V

• Imperas develops simulators, tools, debuggers, modeling technology, and models to help embedded systems developers get their software running...
• ...and hardware developers get their designs correct
• 14+ years, self funded, profitable, UK based, team with much EDA (simulators, verification), processors, and embedded experience
• Staff worked in Arm, MIPS, Imagination, Tensilica, Cadence, Synopsys, ....
• and in verification in EDA on development of Verilog, VCS, SystemVerilog, Verisity and their methodologies
• Started work with customers on RISC-V in 2017
• Contributed to RISC-V compliance since 2018, RISC-V DV since 2019
• Our RISC-V focus is CPU verification
• We provide configurable reference models, the fastest highest quality simulators, advanced development tools and the absolute best solution for RISC-V hardware design verification
• 20+ of the leading RISC-V CPU developers use and rely on Imperas solutions

www.imperas.com  www.OVPworld.org
Agenda

• Brief Introduction to RISC-V
• RISC-V CPU HW DV approaches
• Components of RISC-V CPU DV environment
Agenda

• Brief Introduction to RISC-V
• RISC-V CPU HW DV approaches
• Components of RISC-V CPU DV environment
RISC-V History

• RISC-V (pronounced "risk-five") is an open standard instruction set architecture (ISA) that began in 2010 and is based on established reduced instruction set computer (RISC) principles

• Unlike most other ISA designs, RISC-V is provided under open source licenses that do not require fees to use

• The project began in 2010 at the University of California, Berkeley, but now many current contributors are volunteers not affiliated with the university

• Unlike other academic designs which are typically optimized only for simplicity of exposition, the designers intended that the RISC-V instruction set be usable for practical computers
Industry innovation on RISC-V

- Hardware
  - RV32
  - RV64, multi-heart CPUs, vectors, bit manipulation, hypervisors, debug mode
- Software
  - RTOS
  - Firmware
- Software
  - Linux
  - Drivers
  - AI Compilers
- Hardware
  - IoT SoCs
  - Microcontrollers
- Software
  - Tests
  - Bare metal software
- Hardware
  - Proof of Concept SoCs
  - Minion processors for power management, communications, ...

Complexity

2010 - 2016
2017 - 2018
2019 - 2020
2021
Rapid RISC-V growth led by industrial

Semico Research predicts the market will consume 62.4 billion RISC-V CPU cores by 2025, a 146.2% CAGR 2018-2025. The industrial sector to lead with 16.7 billion cores.

Source: Semico Research Corp
Cloud and data center: Top providers like Amazon and Alibaba are designing their own chips.

Automotive: is transforming from autonomous vehicles to infotainment to safety, the whole vehicle relies on innovative electronics.

Industrial IoT: incorporating artificial intelligence in manufacturing and industrial processes.

Mobile and wireless: continue rapid evolution with each generation of hardware and increased capability.

Consumer and IoT devices: bring incredible innovation and volume with billions of connected devices in the next 5-10 years.

Memory: was largest semiconductor category by sales with $158 billion in 2018, and the fastest-growing.

RISC-V adoption spans industries and workloads.
More than 2,200 RISC-V Members across 70 Countries

- **102 Chip**
  - SoC, IP, FPGA

- **4 I/O**
  - Memory, network, storage

- **17 Services**
  - Fab, design services

- **42 Software**
  - Dev tools, firmware, OS

- **4 Systems**
  - ODM, OEM

- **13 Industry**
  - Cloud, mobile, HPC, ML, automotive

- **95 Research**
  - Universities, Labs, other alliances

- **1,900+ Individuals**
  - RISC-V engineers and advocates

RISC-V membership grew 133% in 2020. In 2021, RISC-V membership has already doubled.
RISC-V Innovation Roadmap

2010 - 2016
ISA Definition
RISC-V Foundation

2017
RV32 and RV64I
Base instructions: Integer, floating point, multiply and divide, atomic, and compact instructions
Priv modes, interrupts, exceptions, memory model, protection, and virtual memory

2018
Arch compatibility framework, Processor trace

2019
Zfinx
ZiHintPause
BitManip
Vector
RISC-V Profiles & Platforms
Crypto Scalar
Virtual Memory
Hypervisor & Advanced interrupt architecture
Cache mgmt ops
Code size reduction*
Trusted Execution Environment*
P (Packed SIMD)*

2020
RV32E and RV64E
64 bit and 128 bit addresses*
Vector Atomic and quad-widening*
Quad floating point in integer registers*
Crypto Vector*
Trusted Execution phase 2*
Jit pointer masking & I/O synch*
BitManip phase 2*
Cache management phase 2*
... and more

2021

Industry Adoption
Proliferation of RISC-V CPUs across performance and application spectrum
RISC-V dominant in universities
Strategic and growing adoption in HPC, automotive, transportation, cloud, industrial, communications, IoT, enterprise, consumer, and other applications

Technical Deliverables

* On track, subject to change
RISC-V == Freedom…

Freedoms enabled by RISC-V are a huge opportunity
RISC-V == Freedom…

Freedoms enabled by RISC-V are a huge opportunity

Freedoms enabled by RISC-V are a huge challenge for verification

the largest change in the industry since? …
Agenda

- Brief Introduction to RISC-V
- RISC-V CPU HW DV approaches
- Components of RISC-V CPU DV environment
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...

- As of 2021, No off-the-shelf toolkit/products 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

• Brief Introduction to RISC-V

• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)

• Components of RISC-V CPU DV environment
Agenda

• Brief Introduction to RISC-V

• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)

• Components of RISC-V CPU DV environment

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

• Brief Introduction to RISC-V
• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)
• Components of RISC-V CPU DV environment
"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

This approach is often due to lack of knowledge or interest in quality, ...
Agenda

• Brief Introduction to RISC-V
• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)
• Components of RISC-V CPU DV environment
#1: Simple (results) check (1b)
(use e.g. riscvOVPsim ISS from GitHub)

• Run RTL DUT in testbench
  - (no real testbench)
  - Just loads & runs the test 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 log or signature file
    • Compare/diff file results (afterwards)
    • This is the approach taken by RISCV
      International for their architectural validation
      ("compliance tests")
Agenda

- Brief Introduction to RISC-V
- RISC-V CPU HW DV approaches
  - #0 “hello world” test
  - #1 self checking tests (e.g. Berkeley torture tests pre2018)
  - #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  - #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  - #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  - #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)
- Components of RISC-V CPU DV environment
#2: Entry Level DV: post-sim trace-compare
(use e.g. riscvOVPsimPlus ISS from OVPworld)

- **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](https://www.ovpworld.org/riscvOVPsimPlus)

- **ISG:** riscv-dv from Google Cloud / Chips Alliance
  - Free ISG: [https://github.com/google/riscv-dv](https://github.com/google/riscv-dv)
Agenda

• Brief Introduction to RISC-V
• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)
• Components of RISC-V CPU DV environment
#3: Industrial Quality Sync DV (sync-lock-step-compare)

- 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 in test
- Will not work for async events and control flow, ... – it is all about the data flow
- [OpenHW evolved into using async – see later slides]
• Brief Introduction to RISC-V
• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)
• Components of RISC-V CPU DV environment
#4: Industrial Highest Quality Async DV (async-lock-step-compare)

- Builds on & extends Industrial Quality sync-lock-step-compare 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
- Typically 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:

2nd generation CV32E40P OpenHW flow (2H2020) (Imperas model encapsulated in SystemVerilog)
Agenda

• Brief Introduction to RISC-V
• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)
• Components of RISC-V CPU DV environment
• Brief Introduction to RISC-V

• RISC-V CPU HW DV approaches
  • #0 “hello world” test
  • #1 self checking tests (e.g. Berkeley torture tests pre2018)
  • #2 Post simulation trace log file compare (e.g. Google riscv-dv 2019)
  • #3 sync-lock-step-compare (e.g. CV32E40P in OpenHW 1H 2020)
  • #4 async-lock-step-compare (e.g. CV32E40P in OpenHW 2H H2020)
  • #5 test bench use of standards (RVVI) (e.g. CV32E40X/S in OpenHW 2021)

• Digression into why we need standards...

• Components of RISC-V CPU DV environment
Challenges moving forward – the need for standards

• There are many different components needed:
  • DUT & its encapsulation
    • ‘tracer’ information
    • Control
  • Reference model & its encapsulation
    • Configuration
    • Comparisons
    • Synchronization
    • Asynchronous operations
    • Control
  • Functional coverage measurement & assertions
  • Test bench
    • Configuration
    • Overall control
    • Scoreboarding
    • Reporting / Logging
  • Tests (directed or generated)
    • Program linker scripts and binary file reader
  
• And each component has different interfaces and requirements on the interfaces

2nd generation CV32E40P OpenHW flow (2H2020)
Challenges moving forward – the need for standards

- There are many different components needed:
  - DUT & its encapsulation
    - ‘tracer’ information
    - Control
  - Reference model & its encapsulation
    - Configuration
    - Comparisons
    - Synchronization
    - Asynchronous operations
    - Control
  - Functional coverage measurement & assertions
  - Test bench
    - Configuration
    - Overall control
    - Scoreboarding
    - Reporting / Logging
  - Tests (directed or generated)
    - Program linker scripts and binary file reader
  - And each component has different interfaces and requirements on the interfaces
Evolving to use developing standards (RVVI)

- Focus is on developing standard interfaces between components
  - Allows reuse
  - Allows development of independent VIP
- Two main components to consider
  - DUT
  - Reference model

Use bespoke tracer+control, (RVVI or proprietary) for Interface to DUT RTL

Use RVVI for interface to reference Model

For a full overview of RVVI see the DVCon 2022 Tutorial
Key component is Reference Model

- RISC-V is highly configurable & extendable
  - 200... Questions ?

- So it can get a little .... complicated
Example reference model: Imperas

- 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
    - Unprivileged versions 2.2, 20191213
    - Privilege versions 1.10, 1.11, 1.12
  - 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, 1.0.0
  - K-Crypto Scalar version 0.7.1, 1.0.0
  - Debug versions 0.13.2, 0.14, 1.0.0
  - P DSP/SIMD versions 0.5.2, 0.9.6
  - Zicbom, Zicbop, Zicboz, Zmmul, Zfh, Zfinx, Zce
  - Svnapot, Svpbmt, Svinval, Smstaten, Smepmp, ...
- 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, ...

http://www.imperas.com/riscv

Imperas is used as RISC-V Golden Reference Model
Imperas Model extensibility

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 fragments 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

- Separate source files and no duplication to ensure easy maintenance
- Imperas or user can develop the extension
- User extension source can be proprietary
Imperas News at DVCon 2022

Imperas announces RISC-V Physical Memory Protection (PMP) Architectural Validation test suite for high quality security applications

Imperas RISC-V Verification Solutions
https://www.imperas.com/imperasdv

The full press release is available at the Imperas virtual booth at DVCon 2022
Imperas unifies new RISC-V verification ecosystem with RVVI

New open standard RISC-V Verification Interface (RVVI)

RISC-V Verification Interface
https://github.com/riscv-verification/RVVI

The full press release is available at the Imperas virtual booth at DVCon 2022

For a full overview of RVVI see the DVCon 2022 Tutorial
Conclusions

• The open standard ISA of RISC-V offers many design freedoms
  • Many standard extensions and configuration options plus custom instructions
• The key verification requirements are to detect discrepancies with efficient debug
• The open standard RVVI offers a framework for verification reuse with support for both open-source and commercial tools
  • RISC-V Verification Interface
  • https://github.com/riscv-verification/RVVI
• Lockstep / Compare is by far the best and most efficient approach (industry ‘gold standard’)
  • https://www.imperas.com/imperasdv

• For more detail see the full tutorial
  • ‘Introduction to the 5 levels of RISC-V Processor Verification’
  • DVCon 2022 Recording from Monday February 28th 9:00-11:00am PST
Thank You!

KevinM@imperas.com

www.imperas.com

www.imperas.com/ImperasDV

www.OVPworld.org
• Tutorial: ‘Introduction to the 5 levels of RISC-V Processor Verification’
  • Simon Davidmann and Lee Moore – Imperas Software
  • Monday February 28th 9:00-11:00am PST

• Presentation: ‘Introduction to RISC-V CPU design verification’
  • Kevin McDermott – Imperas Software
  • Tuesday March 1st 12:30-1:00 pm PST

• Presentation: ‘Imperas RISC-V Design Verification solutions’
  • Larry Lapides – Imperas Software
  • Tuesday March 1st 1:00-1:30pm PST