Open-Source Has a Great Price, But Verification Adds the Real Value

Manny Wright
Senior Consulting Engineer
Imperas Software Ltd
San Jose, CA
mannyw@imperas.com
(408) 590-2913

Katherine Hsu
Imperas Software Ltd
Menlo Park, CA
USA

Aimee Sutton
Imperas Software Ltd
Ottawa, Ontario
Canada

Abstract

RISC-V has taken the processor segment by storm since this open standard ISA (Instruction Set Architecture) project began in 2010. Its popularity in research institutes and the academic community soon spread to commercial implementations across all markets from the tiniest embedded applications to the performance critical server farms with AI accelerators. RISC-V adopters are exploring processor design freedoms enabled by this open standard ISA. In turn, these design freedoms are also driving the interest in open-source hardware.

The OpenHW Group's RISC-V based processors are commercial adoption ready open-source cores. Free is a great price but the real value is in the verification investment by OpenHW to ensure these cores are of industrial quality and compliant to the RISC-V ISA standard. Functional verification of open-source RISC-V processor IP to industrial grade for commercial adoption is discussed in this paper.

What is RISC-V

RISC-V is the fifth generation of RISC (Reduced Instruction Set Computer) architecture instruction set developed by the University of California, Berkeley in 2010 as a research project. The RISC-V instruction set defines how the CPU operates by the software that runs on it. It is an abstract model of a computer. The hardware realization of the RISC-V Instruction Set Architecture (ISA) is a *processor* core that can be integrated into an SoC (System on Chip) for targeted application.

In 2011, the RISC-V ISA was released as an open standard. Ever since, it has garnered serious interest in academics, research institutes, as well as industrial projects because it is free and open for anyone to develop their own processor to run software. In 2016, the RISC-V Foundation – now renamed to RISC-V International [1] – was formed. This consortium of companies, universities, government research institutes and individual contributors controls, maintains and further develops the ISA and the ecosystem in collaboration with the community to meet today’s computational needs. The organization has grown to 2,000+ members from more than 70 countries over the span of just these few years.

The RISC-V ISA was developed to be modular, configurable, and extendable by users. This means that a designer can select, on top of the 32- or 64-bit base instructions, standard extensions for multiplication, atomic, compressed, floating point, bit manipulation and vector instruction set extensions, and more, as required to achieve the optimal processor configuration for their specific application. In other words, RISC-V is particularly well suited for the development of domain-specific processors.

Why and Where is RISC-V Being Adopted

RISC-V is the new ISA on the block. It does not have the history of the x86 or Arm ISAs, which is a good news-bad news story. The bad news is that RISC-V is unlikely to be used in the huge markets that x86 (compute) and Arm (mobile) dominate. The good news is that 1) RISC-V does not have to support the legacy software that these processors need to for their key markets, and 2) those ISAs do not dominate in new, still-evolving markets such as IoT, automotive and AI/ML. That good news, plus the flexibility of RISC-V (due to the significant configurability options within the RISC-V specification), enables domain specific processors that can have significant performance, power and area (PPA) advantages over more established approaches.

The RISC-V Ecosystem

As noted above, it is the RISC-V ISA that is the open standard. In the RISC-V community companies have been started, or have evolved from previous strategies, to provide licensed RISC-V processor IP. These include Andes, Codasip, and SiFive among about three dozen processor IP and systems companies. There are also some companies, organizations and universities providing RISC-V open-source processor IP.

The OpenHW Group [2] was started because its founding members wanted to pool resources to develop industrial quality open-source RISC-V processors. The starting point for the OpenHW implementations was the cores developed under the PULP project in Europe, headed up by ETH Zurich. The PULP designs had the largest set of users of any of the open-source RISC-V processors, however, at the time the OpenHW Group was founded, most of the use was by other academic and research institutions. Only a few firms had produced test chips as proof-of-concept experiments, not for commercial production by semiconductor and systems companies. For those industrial users to adopt an open-source design, that design would need to undergo comprehensive and robust design verification (DV). Simple instruction verification, including the sanity check of running some simple software such as the classic “Hello World” or booting an RTOS or Linux are all insufficient. While all of these steps are important milestones in the design process, they do not exercise the design sufficiently to constitute a full functional verification test plan.

This need for comprehensive RISC-V processor DV is new to the semiconductor community. Previously with the standard single-sourced proprietary processor IP cores, users would license the IP confident that the processor IP vendor would deliver a pre-verified product. The users just needed to worry about integrating the processor IP into the rest of the SoC. (Historically, this gave rise to the verification IP market, with the first verification IP product ever released supporting verification of the integration of components with the Arm AMBA bus.) This is the same for the new RISC-V processor cores from the processor IP vendors: users only need to worry about the integration testing with the rest of the SoC. However, for open-source cores or new designs developed from the open standard specification, users have to do the complete RISC-V processor functional design verification work themselves. This shift in the verification responsibility, moving from a few specialist IP providers to every adopter that explores the new design freedoms of the open standard ISA of RISC-V, represents the biggest change in semiconductor verification since the adoption of SystemVerilog and UVM.

This is where other organizations in the RISC-V community come into the OpenHW story. The OpenHW Group, with the requirement for industrial quality open-source cores, and therefore comprehensive processor DV, needed collaboration not only from end users and developers of the cores, but also organizations that could support the DV efforts with tools and expertise. Imperas, with its expertise in processor models and DV, was one of the founding members of the OpenHW Group.

RISC-V Design Verification

The first project in OpenHW was the CV32E40P core [4] which is a small and efficient, 32-bit, in-order RISC-V core with a 4-stage pipeline that implements the RV32IM[F]C instruction set architecture, and the Xpulp custom extensions for achieving higher code density, performance, and energy efficiency. The team within the OpenHW verification task group responsible for this core included Silicon Labs, Imperas, EM Micro and OpenHW personnel.

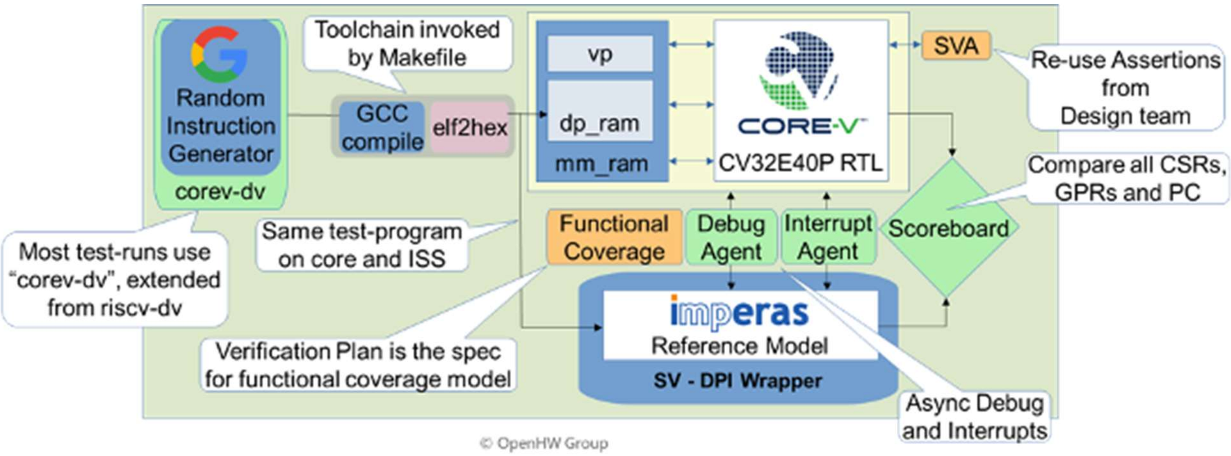
Random Instruction Generator
corev-dv
Toolchain invoked by Makefile
GCC compile
elf2hex
vp
dp_ram
mm_ram
CORE-V
CV32E40P RTL
SVA
Re-use Assertions from Design team
Compare all CSRs, GPRs and PC
Scoreboard
Functional Coverage
Debug Agent
Interrupt Agent
Same test-program on core and ISS
Most test-runs use "corev-dv", extended from riscv-dv
imperas
Reference Model
SV - DPI Wrapper
Verification Plan is the spec for functional coverage model
Async Debug and Interrupts
© OpenHW Group

Fig. 1. Initial verification flow and components for DV of the OpenHW CV32E40P

The first step in any significant verification project is to set out the goals and objectives in a verification plan. As is typical of open-source community projects all the documentation is available and a good reference for any new DV project can be found on GitHub at the OpenHW CORE-V VERIF area [3]. The initial verification methodology evolved to the flow and components shown in Fig. 1.

The key components of the SystemVerilog testbench are:

1) DUT – Device Under Test. This is the RTL of the processor core CV32E40P
2) ISG – Instruction Stream Generator COREV-DV
3) RISC-V verification Reference model from Imperas
4) Instruction comparator between the DUT and Reference Model (Scoreboard)
5) Functional coverage models
6) Additional test suites and directed tests

The key work of the testbench is to run the same instructions sequences on the DUT and compare with the Imperas RISC-V verification reference model. In addition to highlighting any differences the functional coverage monitors and records the progress towards the overall verification coverage plan. To fully test a processor implementation, it is necessary to test a variety of state conditions and dynamic events across the complete operation of the processor. The ISG is a random generator that provides a stream of valid instruction sequences to fully explore the operational space. The OpenHW DV flow uses a modified open-source ISG originally developed by a team at Google. Additional test suites also include the RISC-V International Architecture validation tests for the target specification extensions. While these tests are useful to ensure compatibility with the RISC-V software ecosystem they do not, in themselves, sufficiently test the processor functionality. Finally, for particular areas of concern, a set of directed tests are used to test corner-case scenarios of processor operation. It is important to note that part of the verification methodology defines the process to manage and correct any identified issues. In the event a bug is discovered the team does not simply identity and correct the issue, but must also add a directed test to ensure this scenario is covered during the future regression tests after the verification project is completed. In working practice, the verification team typically expands the scope of these directed tests to cover related and surrounding areas since it is common that an identified bug may well lead to other closely related issues.

For the CV32E40P the comparisons between the DUT and reference model can be a relatively simple process with a state-based comparison at the point of an instruction retirement. This is due to the simple pipeline structure of the design. The roadmap for more complex pipelines is discussed later in this paper.

The RISC-V verification reference model plays two important roles in the verification testbench. Firstly, in general the model needs to support all the valid configuration options of the open standard RISC-V ISA. This includes all standard ratified extensions and the associated implementation configuration options, in addition in the future, as these specifications evolve, it will also be necessary to accommodate previous revisions and versions as well. Plus, any custom extensions. In the case of the CV32E40P this included a combination of both standard RISC-V extensions and a number of custom PULP extensions, so these have also been added to the reference model. As explained above the main operation of the testbench is to run the RTL in a SystemVerilog simulation and using the same instruction sequences compare the operational behavior of the DUT and reference model. The second key feature of the reference model is to support the debug and analysis when an issue is identified. Given the cost and time investment for a full processor verification plan the overall efficiency of analyzing and resolving bugs is critical. The advantage of a single environment with a SystemVerilog simulation allows the DV engineering team to efficiently trace back to the behavior from the point of failure and identify the root cause of the design bug.

Coverage is fundamental to the overall verification plan and answers the key question, “are we done yet?”. In the testbench shown in Fig 1 coverage monitors record the scenarios that are achieved in the DUT. For simple processors the coverage can be achieved with only simple state related issues. However as will be explained later in the future roadmap plan, coverage measurement needs to also include more sophisticated state situations including asynchronous activities around interrupts and debug events.

CV32E40P DV Project Completion

This verification project was successful and uncovered over 40 RTL bugs and documentation clarification issues, and the CV32E40P IP has now been implemented in several commercial SoCs. However, it took a lot of work to build this bespoke DV environment, and it did not seem that this would be extendable and scalable to support the future roadmap with multiple new OpenHW core projects. As with many projects a successful completion is an opportunity to plan for the future ahead.

The Roadmap of Complexity

With the completion of the initial CV32E40P project a number of additional projects and sub-projects have started. Since RISC-V is based on an open standard ISA it is very easy to see opportunities to improve the hardware to better support the target application and software use-case. During the CV32E40P project a number of ideas and suggestions were considered that resulted in a roadmap of new cores some of which are listed in Fig 2. In addition to the extra work required to configure the verification environment, the roadmap also includes some advanced features such as RISC-V vector extensions, privileged mode, security features and advanced pipelines with out-of-order performance features.

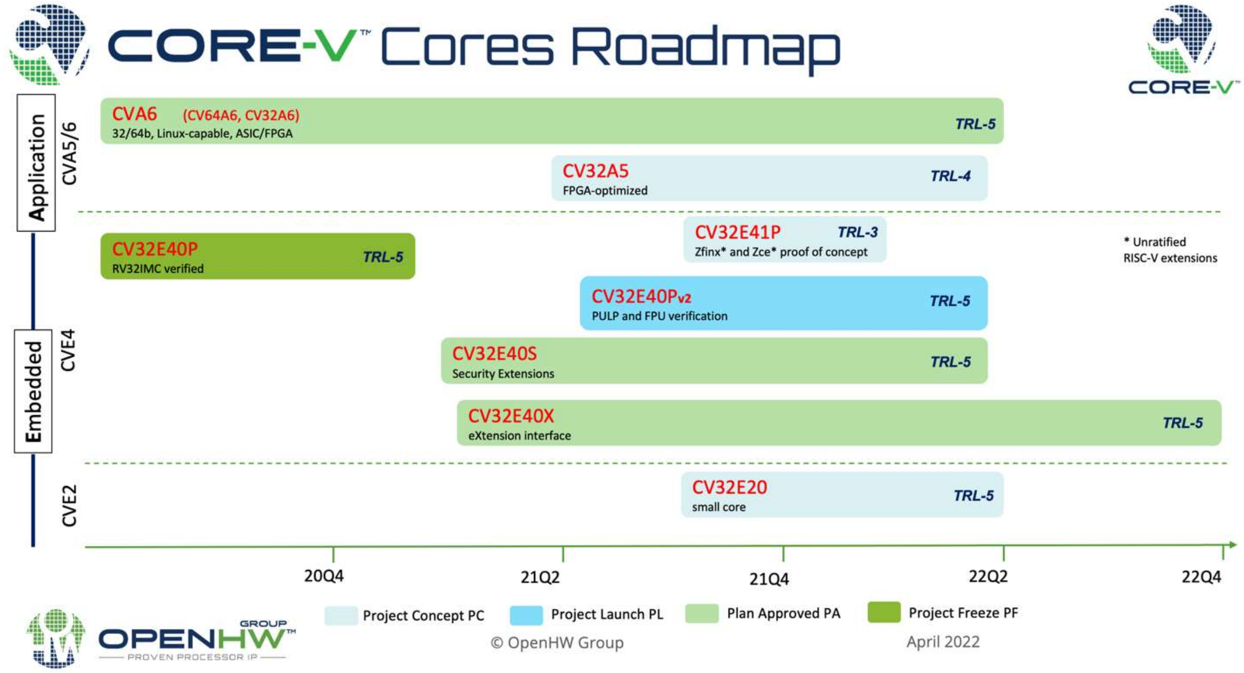

Fig. 2. OpenHW core roadmap

RVVI – One Standard to Unite the RISC-V Verification Ecosystem

With the completion of the verification project for CV32E40P it was clear that a number of the manual steps to integrate the testbench components would be a barrier to the future roadmap plans. However, since all RISC-V designs share some similarities as they are all based on the open standard ISA specification, the testbench could be set up as a universal component. But in reality, as with many technical projects, it’s not the components that are universal but the standards that allow them to be connected efficiently. This was the central concept behind the development of the RISC-V Verification Interface, RVVI [5]. RVVI is an open standard available on GitHub that defines the key interfaces between the DUT and reference model with the necessary features to support the comparison, debug and coverage aspects for RISC-V processor verification shown in Fig 3. In addition, as this is not limited to any one core project or company, it can be seen as a universal solution that others can build on. RVVI already supports all the ratified RISC-V specifications and tracks the latest draft extensions as they stabilize.

RVVI supports the complete open standard RISC-V ISA specification and all of the optional configuration features. In addition, it supports complex implementations including privileged mode, multi-hart, multi-thread and out-of-order pipelines. RVVI also provides support for comprehensive checking and coverage analysis of asynchronous events including interrupts and debug requests.

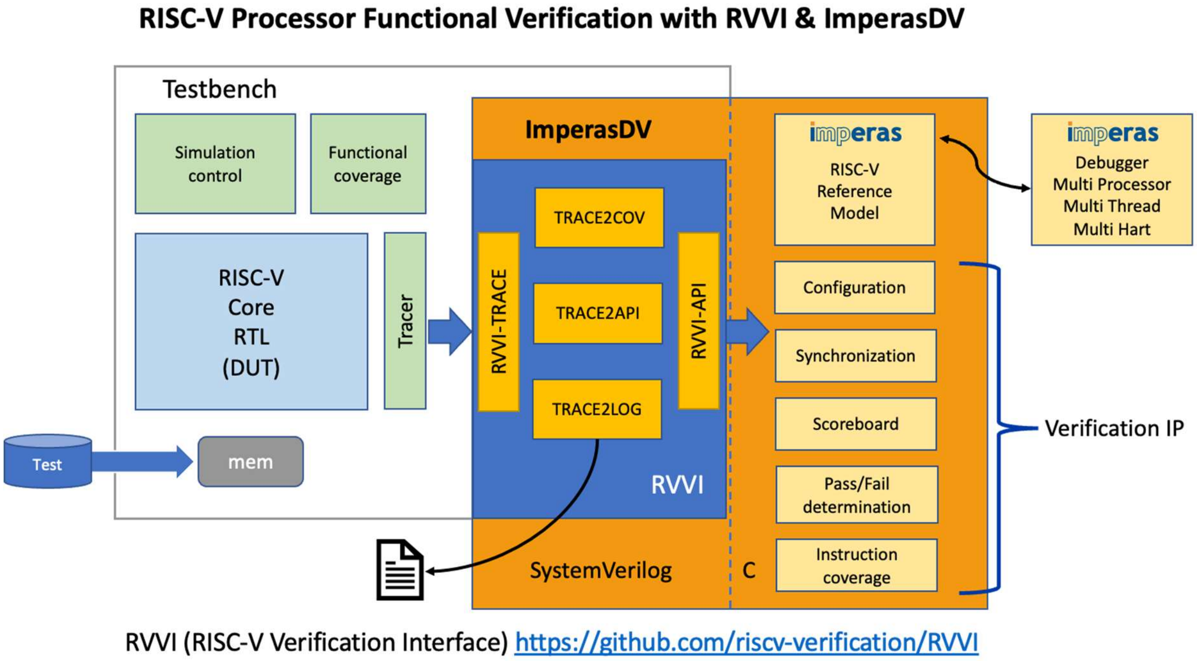

Fig. 3. Overview of the RISC-V Verification Interface (RVVI)

Conclusion

The open standard specification of RISC-V offers the potential of domain specific processors to address new challenges in optimized designs. This also marks the end of one-size-fits-all as the barriers are lowered for all SoC teams to explore the advantages of a custom processor. With this new flexibility also comes the verification requirements. The pioneering work of The OpenHW Group has achieved high quality open-source cores with industrial strength verification and compatibility with modern design and EDA flows.

Open-source hardware is now ready for commercial adoption. The key advantage of open-source cores, apart from the price, is the ability to tune and adapt the core IP to a new design requirement. In offering open standard flows and testbenches based on a reference methodology such as RVVI, developers can leverage the high-quality deliverables and verification standards as they build the next generation of optimized processors and custom devices.

The RISC-V design flexibility enables unprecedented innovation but a robust design verification is paramount to its success. The RISC-V Verification Ecosystem is ready to support its adopters to explore the new design freedoms RISC-V offers based on free and open standards such as RVVI.

Footnotes:

1. RISC-V International https://riscv.org
2. The OpenHW Group https://www.openhwgroup.org
3. OpenHW verification task group CORE-V Verif https://github.com/openhwgroup/core-v-verif
4. OpenHW CV32E40P https://github.com/openhwgroup/cv32e40p
5. RVVI (RISC-V Verification Interface) https://github.com/riscv-verification/RVVI