Maven Silicon's RISC-V Processor IP Verification Flow

Technical Paper

Author: *P R Sivakumar, Founder and CEO, Maven Silicon*
LinkedIn Profile: https://www.linkedin.com/in/sivapr/

1.Introduction

RISC-V is a general-purpose license-free open Instruction Set Architecture [ISA] with multiple extensions. It is an ISA separated into a small base integer ISA, usable as a base for customized accelerators and optional standard extensions to support general-purpose software development. RISC-V supports both 32-bit and 64-bit address space variants for applications, operating system kernels, and hardware implementations. So, it is suitable for all computing systems, from embedded microcontrollers to cloud servers.

To know more about RISC-V, refer : *D&R Article - Is your Career at Risk without RISC-V?*

In this open era of computing, RISC-V community members are ambitious to create various kinds of RISC processors using RISC-V open ISA. However, the risk of using RISC-V ISA is higher because the proven processor verification flow is still proprietary to established processor fabless IP companies and IDMs as an unrevealed secret. So, **how can we make the RISC-V verification flow open and empower the RISC-V community?**

If you have been wondering, 'How can I verify my RISC-V processor efficiently without risking TTM', then you should explore and adopt Maven Silicon's RISC-V verification flow for your processor IP verification, explained in this technical paper.

I have defined Maven Silicon's RISC-V verification flow using the correct by-construction approach. The approach is to build a pre-verified synthesizable RISC-V IP fundamental IP building blocks library and create any kind of multi-stage pipeline RISC-V processor using this library. Finally, the multi-stage pipeline RISC-V processor IP can be verified using Constrained Random Coverage Driven Verification [CRCDV] in Universal Verification Methodology [UVM] and FPGA prototyping.

Let me explain the verification strategy and walk you through Maven Silicon's RISC-V verification flow.

2.Verification Strategy

2.1 Block-Level Verification: Generate RISC-V IP Fundamental Blocks Library using formal verification

Verify all the RISC-V IP fundamental building blocks like ALU, Decoder, Program Counter, Registers, Instruction, and Data memories using Formal Verification. Design Engineers [DE] who do RTL coding of the RISC-V IP building blocks can embed the assertions [SVA/PSL] into the RTL modules. Verification Engineers [VE] will verify the RISC-V RTL IP blocks using the formal verification EDA tool. DEs can synthesize the verified RTL blocks and fix all synthesis-related issues.

Finally, VEs can further verify those synthesizable RTL blocks and create RISC-V IP Blocks Library.

2.2 IP-Level Verification: Verify RISC-V IP which is built using pre-verified RISC-V IP Blocks Library using Constrained Random Coverage Driven Verification [CRCDV]

DEs can realize any kind of multi-stage pipeline RISC-V processor using the pre-verified RISC-V IP fundamental blocks library. VEs will create the verification environment in UVM and verify the RISC-V multi-stage pipeline processor using CRCDV.

VEs will also create necessary reference models, interface assertions for protocol validation, and functional coverage models. Finally, VEs will sign off the IP level regression testing based on the coverage closure [Code + Functional Coverage].

The verified RISC-V processor IP can be verified further by booting OS using FPGA Prototyping if the IP implements a general-purpose processor that supports a standard Unix-like OS.

3. Maven Silicon's RISC-V Processor IP Verification Flow

As shown in figure1, Maven Silicon's RISC-V Verification flow implements the verification strategy explained above.

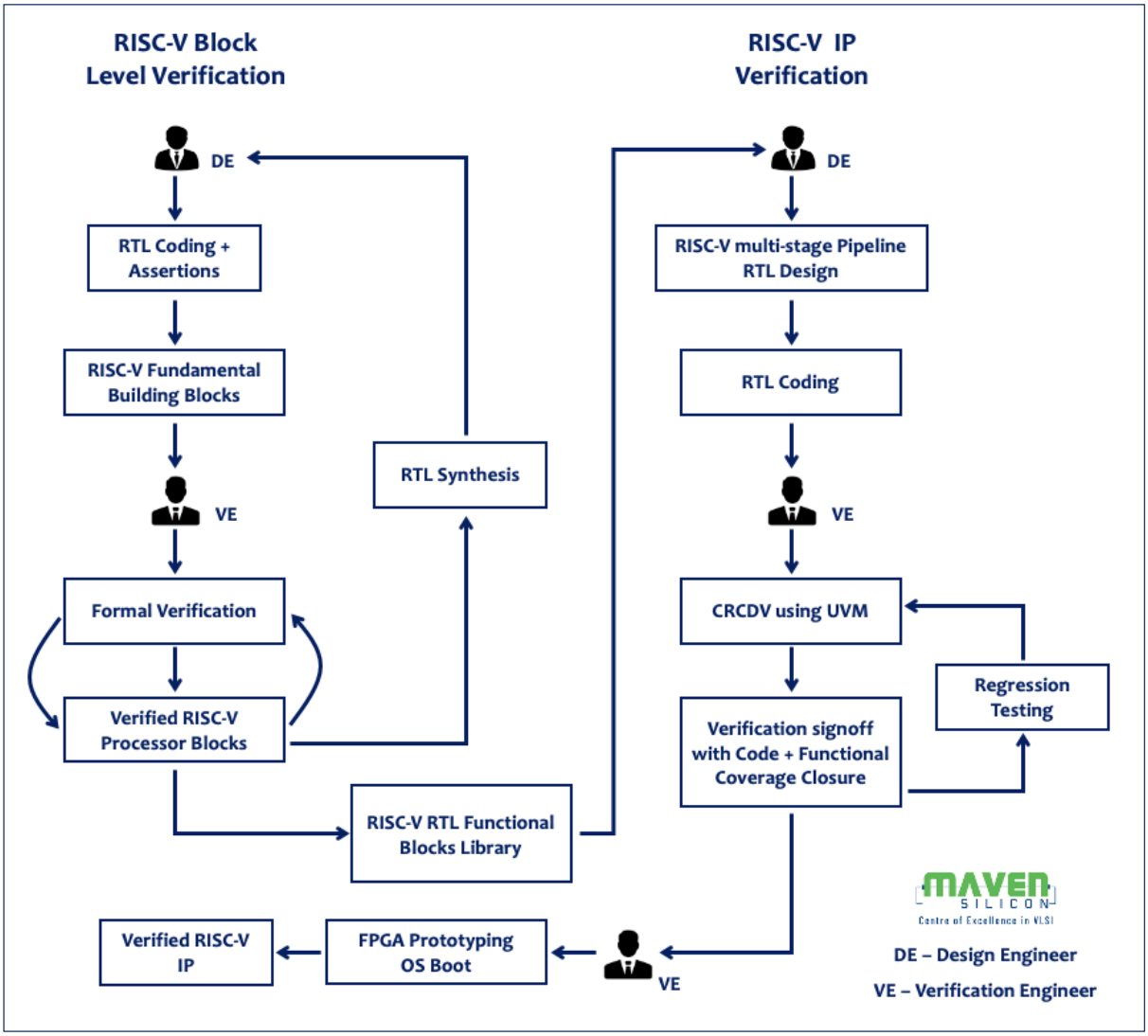

Figure1: Maven Silicon's RISC-V IP Verification Flow

4.RISC-V IP Verification using UVM

IP-level VEs can create the verification environment using Universal Verification Methodology, as shown in figure 2.

As our Maven Silicon's RISC-V IP RTL design uses an AHB interface, we have modeled the instruction and data memories as AHB slave UVM agents. The RISC-V processor reference model was modeled as AHB master UVM agent, and the complete UVM environment with all the testbench components scoreboard, subscribers with coverage models, reset, interrupt, and RAL UVM agents was validated by connecting RISC-V AHB agents back-to-back, especially to verify the dataflow and coverage generation. Once the verification environment became stable, one of the reference models was replaced by the RTL. UVM RAL was extensively used to sample the RISC-V IP registers and memories for the data comparison in the scoreboard.

To understand how this verification environment works, refer to this demo video:
Maven Silicon RISC-V UVM Verification Environment

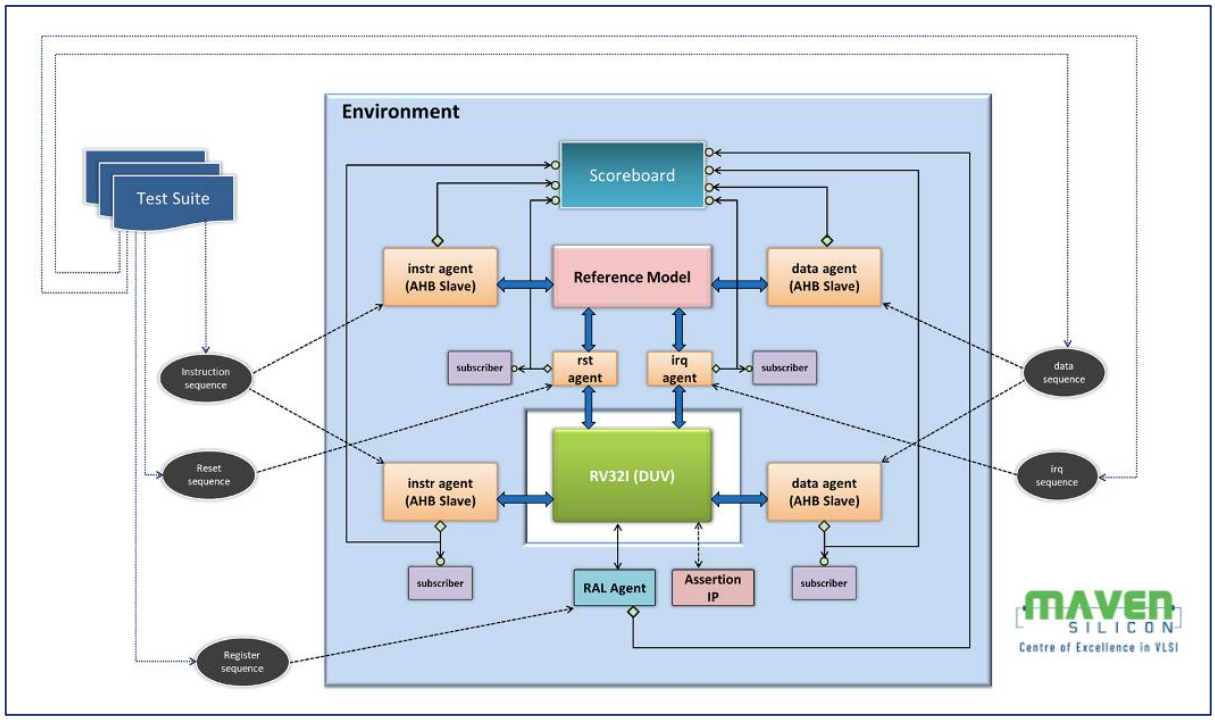

Figure 2: Maven Silicon's RISC-V IP UVM Verification Environment

To know further about various verification methodologies like formal verification, IP, Sub-system and SoC verification flow, CRCDV using code and functional coverage, refer:
D&R Article: SoC Verification Flow and Methodologies

One can also consider integrating Google's Instruction Stream Generator for the stimulus generation, and open source Instruction Set Simulator [ISS] like Spike as a reference model into their UVM environment and efficiently do exhaustive verification.

5. Conclusion

Efficient, high-quality RISC-V IP verification can be realized only through the **effective combination of various verification methodologies** like formal verification, CRCDV using UVM and OS booting using FPGA prototyping, and **reusabilities** like reusing pre-verified RISC-V Blocks library and scalable IP level UVM testbenches. As RISC-V is an open ISA, we can create the reusable RISC-V fundamental blocks pre-verified library and contribute to RISC-V International as an open-source RISC-V library. Using this pre-verified library, the RISC-V community members can create any kind of multi-stage pipeline RISC-V processor as they prefer and verify their RISC-V processor as per the flow explained in this technical paper. ***Isn't it the more efficient way of verifying your RISC-V processor without risking your TTM?***

About Maven Silicon

Maven Silicon is a trusted VLSI Training partner that helps organizations worldwide build and scale their VLSI teams. We provide outcome-based VLSI training with our variety of learning tracks i.e. RTL Design, ASIC Verification, DFT, Physical Design, RISC-V, and ARM etc. delivered through our cloud-based customized training solutions. To know more about us, *visit our website*.