Open-Source RISC-V Vector Test Suites: A Comparative Analysis

Fatima Saleem[1], Umer Imran[1], Quswar Abid[1] and Dr. Kamran Malik[1]

[1]10xEngineers, Pakistan

Abstract

Architecture-level verification of RISC-V Vector (RVV) cores presents a complex challenge due to the diverse behavior of vector instructions under various instruction parameters and core configurations. The vector instruction test suite must be capable of addressing all legitimate combinations of vector instructions under all the different configurations to accomplish comprehensive verification targets. In this paper, we present a comparative analysis of seven open-source test suites for vector ISA: Imperas, RIOS Labs, RISCV-Torture, RISCV-DV, FORCE-RISCV, Yang's Generator and Tenstorrent. We provide an objective evaluation of the attributes and coverage of these test suites using the RVV Specification v1.0 as a reference. Our study reveals that a combination of these test generators can provide a fairly comprehensive verification strategy for vector cores. On a stand alone basis, FORCE-RISCV covers the complete RVV v1.0 specification upto a VLEN of 4096. Similarly, Imperas is fairly exhaustive but limited to VLEN of 256.

Introduction

RISC-V is an open-source Instruction Set Architecture (ISA) that offers a wide range of instructions to enable the efficient processing of data. Among these instructions are vector instructions, which are used to process multiple streams of data in parallel. The RISC-V Vector [1] Architecture Compliance Testing (VACT) is crucial in ensuring that a specific vector hardware implementation conforms to the RVV specification and correctly, as well as completely, implements the vector instructions. This typically involves the use of test suites, which are collections of test cases designed to exercise different aspects of the vector ISA. The aim is to develop comprehensive RVV architecture tests that cover not only the full vector instructions but also the various configuration parameters defined in RVV like vector register length (VLEN), register multiplier (LMUL), element width (SEW), etc.

It is important to note that the vector test suite is significantly more extensive than the RISC-V scalar test suite. The vector test suite consists of numerous instructions which can be divided into seven categories defined in RVV Spec as listed in Table 1 (Column 1). To overcome the challenge [2] of verifying a RISC-V based vector core, comprehensive RVV architecture test suites are being developed by the open-source community. In this paper, we do a comparative analysis of all seven open-source RVV test suites to identify the capabilities strengths and weaknesses of each test suite.

Imperas Vector Test Suite[3] is a very comprehensive verification tool for RVV processors. Imperas has made available its RISC-V Reference Model (riscvOVPsim) and a version of its Vector Test Suite (VTS Community Edition). The VTS Community Edition offers a variety of pre-built test cases for the RISC-V Vector Extension. Imperas has only released its 32-bit vector test suite, which supports VLEN 128b and 256b only, resulting in reduced coverage of some of the vector instruction categories.

The RIOS Lab Vector Test Generator [4] is an RVV automated test generator created by the RIOS Lab in collaboration with the RISC-V Foundation for the RISC-V community. The RVV Sail Model, contributed by RIOS Lab, serves as the standard golden model for executing RVV instructions. By utilizing the configuration features of the RIOS Labs Vector Test Generator, users are able to select specific instructions and set configuration parameters to generate specific tests. This generator is fully compatible with the latest RISCOF infrastructure, which is widely used in RISC-V testing.

RISC-V Torture [5] is a Scala-based test generator that supports RVV v0.9 specification and generates tests by integrating predefined randomized test sequences. The tests are randomly generated, and at the end of the test program execution, verification occurs by comparing the register log with the output from the Spike ISA simulator. Antmicro used RISC-V Torture Tests to assess Renode's RISC-V implementation.

RISC-V DV [6] s an open-source UVM-based random instruction generator that generates tests compliant with RVV v0.9. It provides excellent randomness and performance. Andes and Google collaborated to support RVV v0.9 in RISC-V DV. Their upcoming plans involve including support for RVV v1.0 and adding hooks to collect vector extension test cover-

Table 1: *RVV v1.0 instruction coverage by various tests suites*

RVV Categories (Total Instructions)	Tests Suites						
	Imperas	RIOS Labs	RISCV-Torture	RISCV-DV	FORCE-RISCV	Yang's Generator	Tenstorrent
Configuration (3)	3	3	2	2	3	3	3
Loads and Stores (310)	45	141	202	258	310	46	25
Integer Arithmetic (139)	137	133	139	136	139	139	60
Fixed-Point Arithmetic (32)	32	32	32	32	32	32	0
Floating-Point (91)	91	66	89	75	91	88	0
Reduction Operations (16)	16	15	16	16	16	16	0
Mask Operations (15)	15	12	15	15	15	13	5
Permutation (21)	21	17	20	18	21	19	5

Table 2: *Comparative Analysis of different RVV test suites*

Test Suite	Test Suite Attributes					
	Spec Version	Suite Type	Checking Method	Coverage Tool	Supported ISS	Integrated Core
Imperas	1.0	Tests	Signature based	riscvOVPsim	riscvOVPsim	NSITEXE DFP
RIOS Labs	1.0	ATG	Self-Checking	RISCV-ISAC	Sail	-NA-
RISCV-Torture	0.9	RTG	Signature based	-NA-	Spike	Renode
RISCV-DV	0.9	RTG	Signature based	-NA-	Spike	Andes NX27V
FORCE-RISCV	1.0	RTG	Signature based	riscvOVPsim	Handcar	-NA-
Yang's Generator	1.0	ATG	Self-Checking	-NA-	Spike	ARA
Tenstorrent	1.0	Tests	Self-Checking	-NA-	Whisper	Ocelot

age.

FORCE-RISCV [7] is an Instruction Sequence Generator (ISG) designed to generate tests for the design verification of RISCV processors. It utilizes randomization to select instructions, registers, addresses, and data for the tests and can produce valid test sequences with minimal user input. Additionally, FORCE-RISCV includes a set of advanced APIs, giving the user a high level of control over the instruction generation process. Although it supports the complete RVV v1.0 specification, there is no use case available demonstrating its integration with any RVV implementation to ensure its correctness.

Yang's Vector Test Generator [8] is an Automatic Test Generator (ATG) developed by Yang Liu, a developer from PLCT lab. It verifies the RISC-V Vector implementation. The test generator is written in GoLang and employs GNU Make to automate the test generation flow. The generated tests are fully self-checking, and any verification environment for a core utilizing the riscv-tests can easily use them.

Tenstorrent Vector Test Suite [9] comprises a collection of binary test files that were produced by their internal test generator. This generator has the ability to create a comprehensive range of tests with numerous repetitions, effectively covering the entire architectural verification space for vector instructions.

References

[1] "RISCV Vector Spec v1.0". In: (Sep 2021). URL: `https://github.com/riscv/riscv-v-spec`.

[2] *The challenge of RISC-V Compliance*. URL: `https://semiengineering.com/toward-risc-v-compliance/`.

[3] "Imperas Tests". In: (Feb 2023). URL: `https://github.com/riscv-ovpsim/imperas-riscv-tests`.

[4] "RIOS Lab ATG". In: (Dec 2022). URL: `https://github.com/xiwang-x/rvv-atg`.

[5] "RISCV-Torture". In: (Dec 2020). URL: `https://github.com/Lampro-Mellon/riscv-torture`.

[6] "RISCV-DV". In: (Feb 2021). URL: `https://github.com/chipsalliance/riscv-dv`.

[7] "FORCE-RISCV". In: (Jan 2023). URL: `https://github.com/openhwgroup/force-riscv`.

[8] "Yang's Generator". In: (Mar 2023). URL: `https://github.com/ksco/riscv-vector-tests`.

[9] "Tenstorrent Tests". In: (Feb 2023). URL: `https://github.com/tenstorrent/riscv_arch_tests`.