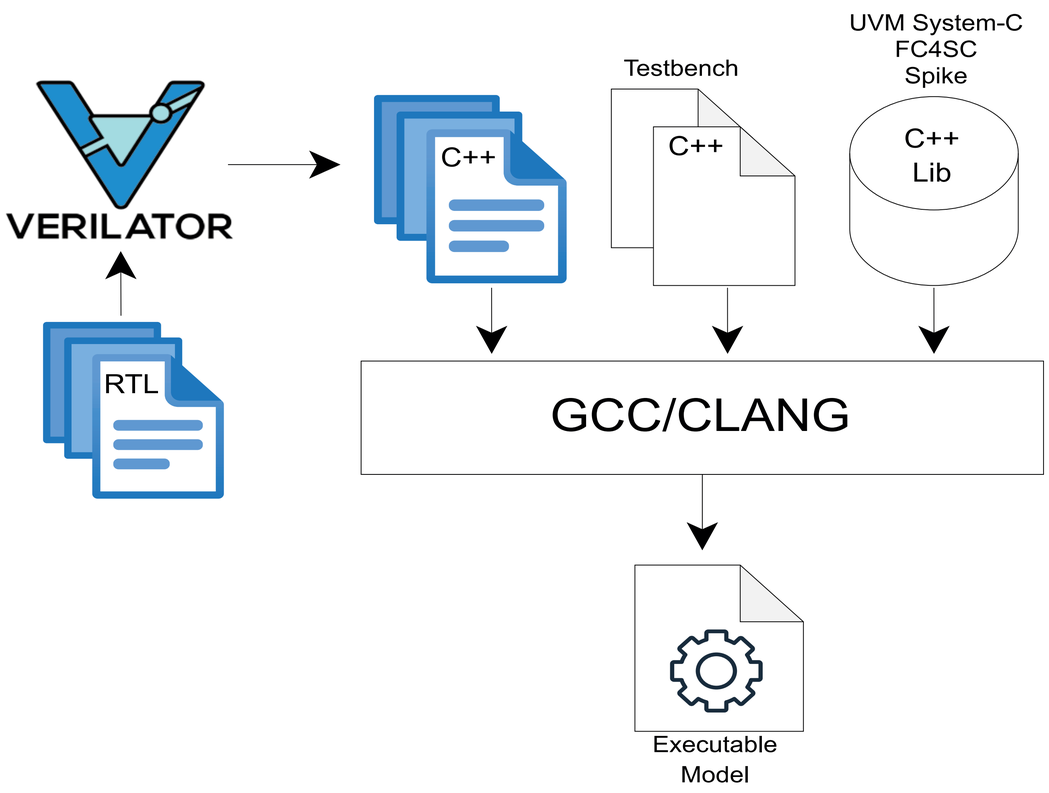
Open source verification environment for RISC-V

Josep Sans, Alberto Moreno, Àlex Torregrosa and Roger Espasa

Introduction

The verification environment is built around Verilator, and extended with different open source libraries to complete a fully UVM-compliant verification environment, which can be run in any platform, without requiring any license. This allows high parallelism, which (1) speeds up the execution of the regression set, and (2) enables usage of a resource-hungry coverage driven verification methodology. Furthermore, Verilator's code translation style from RTL to C++ also enables the use of fuzzing techniques for the DUT, which further helps increase overall coverage.

Verification environment



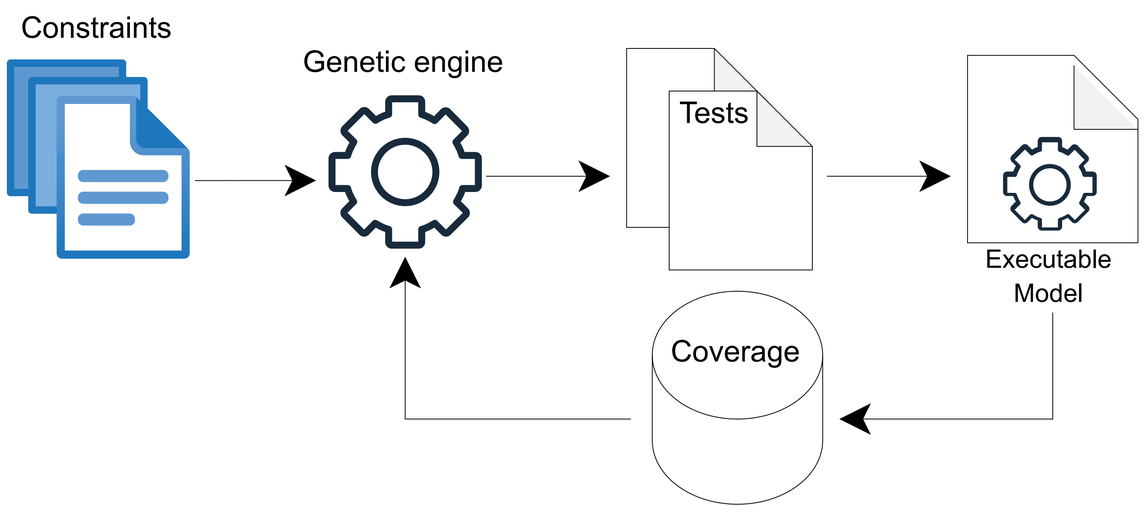
Verilator: an open-source tool that is able to transpile RTL code written in Verilog or SystemVerilog into a C++ model which can be instantiated in a wrapper along with the test bench code having as a result an executable containing the RTL functional model and its test bench.

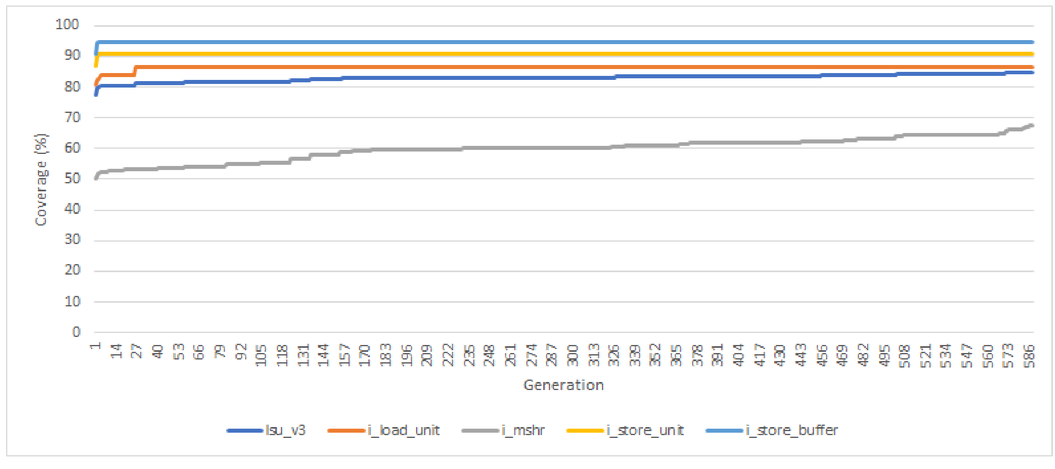
UVM-SystemC: the port of the UVM library into the System-C runtime enables the use of the same verification concepts already established and to reuse all the existing knowledge in order to build the test benches.

FC4SC: a library that provides support for functional coverage.

Spike: a RISC-V simulator that enables the creation of a cosimulation environment with the RISC-V core under test.

Open Flow 1: Generic coverage driven constrained random verification





Open Flow 2: Fuzzing coverage driven constrained random verification