Verification of digital circuits using Java

Bachelor's project

19th of May 2025

Rasmus Wiuff

DTU

Outline

Introduction

- Chip design requires verification
- Verification most commonly done using UVM
- Commonly used frameworks: Chisel, SpinalHDL, pyuvm
- ABV and formal verification improves the verification step

Verification cycles reduced by 25-30%
Pre-silicon bug detection rates improved by 20%
Security vulnerability detection increased by 40%

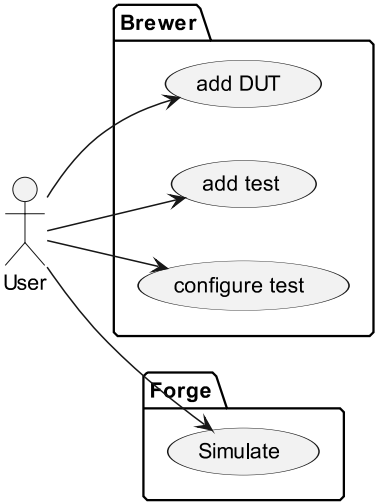
Problem specification

A chip verification framework written in Java, supporting SystemVerilog and core ideas from ABV, thus making it easy for designers to write their designs in SystemVerilog and use a well known language to implement assertion based tests.

Challenge	Success Criteria
Simulation driver	Launching and handling output from Verilator
Peek-poke-step	Basic verification tests
Assertions	SVA assertions
Test-translation	Translate tests into a testbench
Concurrency	Concurrent execution of the tests

Usecases

- Adding devices
- Adding tests
- Configuring tests
- Run simulations



Separation of responsibility

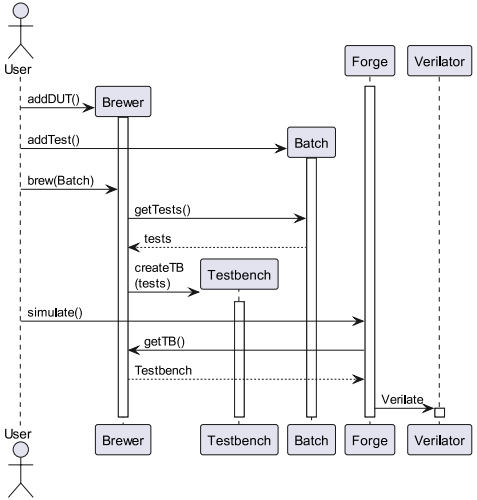
The Brewer

- Adding devices
- Handling test logic
- Preparing testbenches

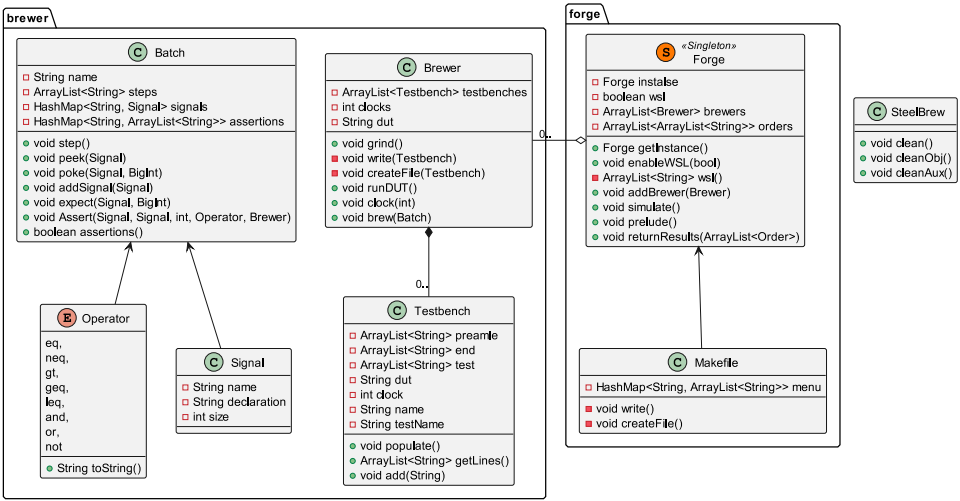
The Forge

- Define command arguments
- Launch Verilator
- Collect Verilator output
- Handle concurrency

The final workflow



The program structure



Defining tests

Running tests

Using the project

Further development

Conclusion

Q&A

Questions?