Finding More Bugs in Your RISC-V CPUs with DiffTest and XFUZZ

Yinan Xu (xuyinan@ict.ac.cn)
SKLP, Institute of Computing Technology, Chinese Academy of Sciences

Abstract

- Verification is important for RISC-V.
- We propose an open-source, automated verification framework for RISC-V processors, including support for co-simulation (DiffTest) and coverage-guided fuzzing (XFUZZ).
- It has identified hundreds of bugs in well-known, open-source CPUs like rocket-chip, Spike, and XiangShan.

An Open-Source Toolchain for RISC-V Verification

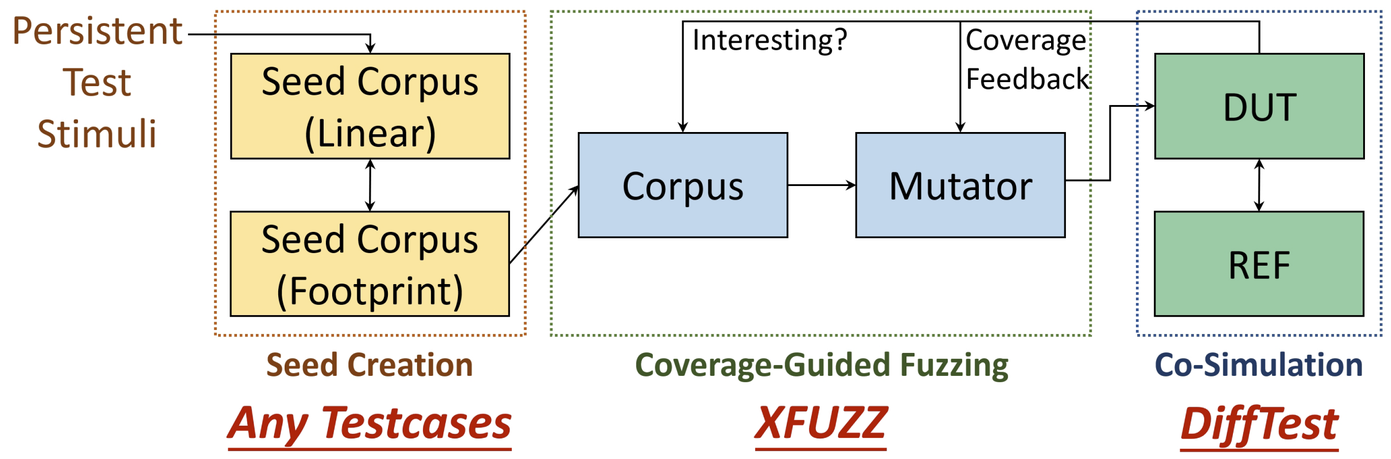
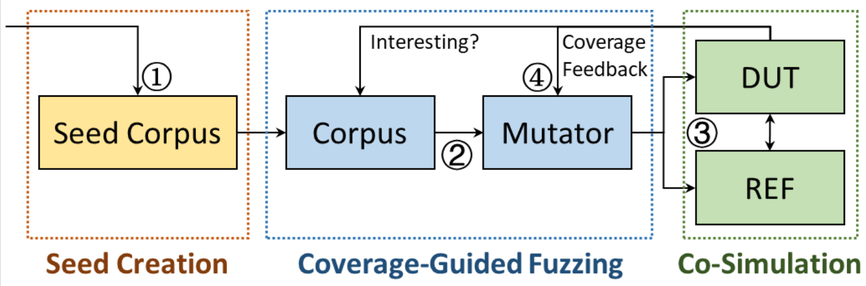


Figure: System overview of DiffTest and XFUZZ.

Background



CGF Workflow. Given ① *the seed corpus*, the fuzzer ② *mutates the corpus* and ③ *sends mutated inputs* to the design-under-test (DUT) for execution. Mutations are ④ *guided by DUT coverage feedback*.

Co-Simulation. During execution, the DUT behaviors are compared against the behaviors of a golden reference model (REF). If a mutated input leads to a mismatch, a bug is potentially detected and then reported.

DiffTest: Co-Simulation with Behavioral Nondeterminisim

- **DRAV Methodology**
  - Dynamically refines the golden reference to ignore permissible microarchitectural variations.
  - Reduces false positives by adapting "diff rules" at runtime.
- **Information Probes**
  - Instrument DUT to extract register states & memory traces.
  - Provides structured, replayable probes across simulators.
- **Standard Co-Simulation APIs**
  - Unified interfaces for Verilator, VCS, Palladium, and FPGA.
  - "Plug-and-play" support for new DUTs or reference models.

PathFuzz: Fuzzing Execution Paths via Footprint Memory

- **Seed Creation**: with large-scale real-world programs as seeds.
- **Coverage-Guided Fuzzing**: customized hardware coverage instrumentation, including structural and functional metrics.
- **Co-Simulation**: on-demand memory synchronization between the DUT and the REF, supported by DiffTest.

Why Traditional DV Falls Short

To build open-source verification infrastructures for RISC-V

- Challenge: RISC-V cores are getting more complex; hand-crafted directed tests miss many corner cases.
- Risk: Undetected functional bugs surface late (tape-out or silicon), at huge cost.
- Goal: An open-source, automated, coverage-guided verification framework that minimizes manual effort while maximizing verification state space exploration.

Evaluation: Finding Bugs in Open-Source Projects

- 95.3% toggle coverage for Rocket in less than 10 hours.
- Uncovering 4 long-standing bugs in Rocket and Spike, where the oldest persists for nearly 10 years.
- Reporting hundreds of bugs in NutShell and XiangShan through over thousands of hours of nightly regression testing.

Conclusion

- **Open-sourced** at https://github.com/OpenXiangShan and **deployed** on various open-source RISC-V CPUs!
- Automated DV Pipeline: Seamless integration of DiffTest + XFUZZ.
- Real-World Impact: 100s of bugs in open-source RISC-V CPUs.
- Contributions & new DUT integrations welcome!

RISC-V®

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES