Modernization of DiFuzzRTL Hardware Fuzzer

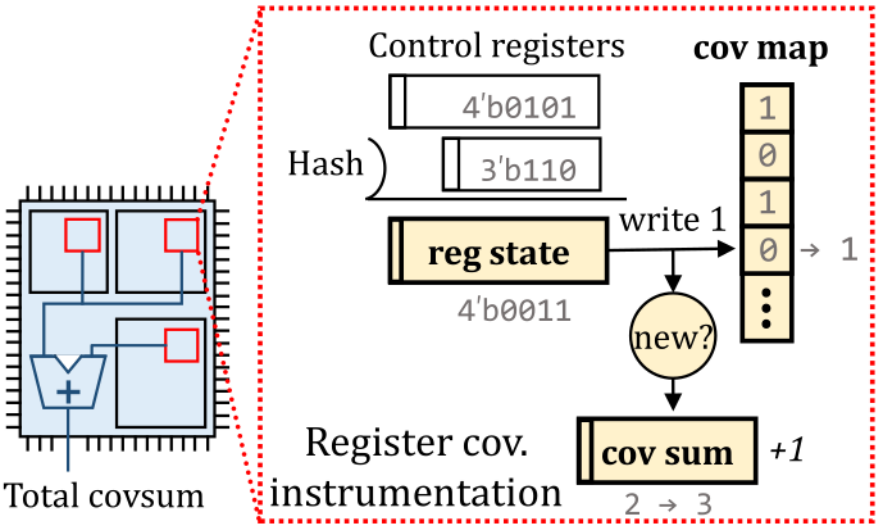


Figure 1: Coverage Instrumentation in DiFuzzRTL [2]

Fuzzing consists of repeatedly feeding a system with random or weirdly formatted input until it crashes or exhibits abnormal behavior. An example is DiFuzzRTL [2]. takes a FIRRTL design of a RISC-V CPU as input, instruments it, runs it with random input and checks the result for bugs against an RISC-V simulator. Despite its success, DiFuzzRTL is no longer actively maintained. Some of its dependencies are old and (very likely) contain bugs. As a result, research that builds on top gets more and more unsteady. The goal of this thesis is to rewrite DiFuzzRTL's instrumentation mechanism inside of the modern CIRCT compiler and test it against other fuzzers in Encarsia [1]. This thesis has a strong emphasis on practical work. The student is expected to get familiar with the concepts of hardware fuzzing, the DiFuzzRTL instrumentation (written in Scala), and the CIRCT-compiler and its dialects.

[1] Matej Bölcskei et al. "Encarsia: Evaluating {CPU} Fuzzers via Automatic Bug Injection". In: *34th USENIX Security Symposium (USENIX Security 25)*. 2025, pp. 2577–2593.

[2] Jaewon Hur et al. "Difuzzrtl: Differential fuzz testing to find cpu bugs". In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 1286–1303.

Practicalities

Promotor:	Bart Preneel
Daily supervisor:	Jakob Heirwegh jakob.heirwegh@esat.kuleuven.be office B01.24
Nature of the work:	25% literature, 75% applied work
Number of students:	1