AccUnit: Accelerating Unit Level Verification for RISC-V Processors Using FPGA

Chenang Zhu[1,2 *], Weidong Li[3 *], Yungang Bao[1,2] and Kan Shi[1,2]

[1] State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] ShanghaiTech University

Background

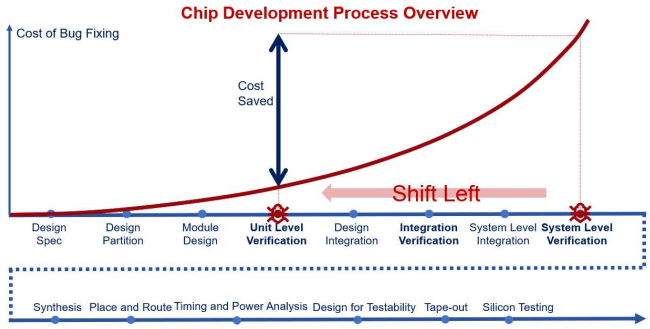

Figure 1: Cost of Bug Fixes Throughout the Chip Development Process

Chip verification is important and difficult

- It takes up to 70% of the entire chip design cycle.
- As chip development progresses, the cost of fixing bugs increase exponentially.

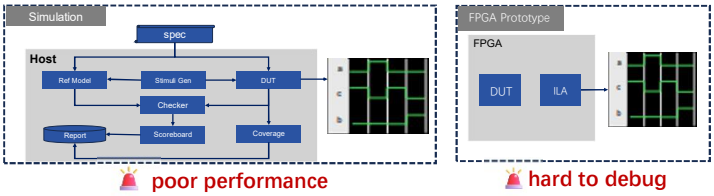

Figure 2: Traditional Verification Approaches

- Challenge1: Software simulation suffers from poor performance.
  Solution: Leverage FPGA acceleration
- Challenge2: Limited verification infrastructure on FPGAs, like coverage.
  Solution: Implement synthesizable coverage
- Challenge3: The potential of FPGA has yet to be fully unlocked.
  Solution: Exploit parallelism
- Challenge4: Verification depends heavily on manual effort.
  Solution: Auto Adaption & Auto-construction

The Overall Architecture of AccUnit

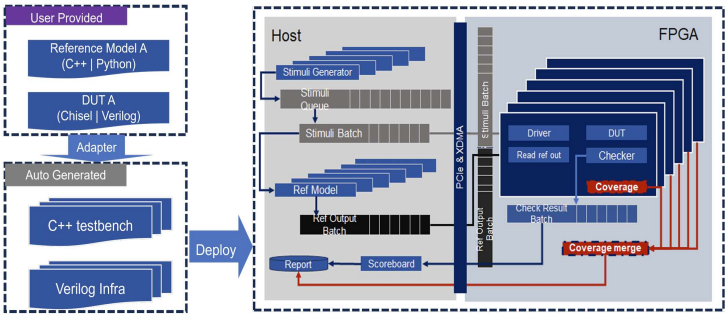

Figure 3: Overall Architecture of AccUnit

AccUnit is a tool flow for unit-level verification of RISC-V processors

AccUnit performs online checking between DUT and reference model

- DUTs and checkers run on FPGA.
- Stimuli generators and reference model run on Host.

Main Features

Parallel Verification

- Runs multiple $< \mathrm{DUT}, \mathrm{Ref} >$ model pairs simultaneously.
- Leverages both FPGA parallelism and host CPU multi-core capabilities.
- Dynamically adjusts stimuli generator thread counts based on the relative speeds of the generator and reference model.

Synthesizable Coverpoints

- Provides real-time coverage data for comprehensive DUT verification.
- Supports line, toggle, and ready-valid coverage.
- Easily configurable during compile time.

Automatic Deployment

- Adapts drivers and monitors for DUTs and their reference models automatically.
- Duplicates stimulus generators and <DUT, reference model> pairs for parallel verification automatically.
- Inserts coverpoints into DUTs automatically.
- Integrates all components into testbench automatically.

Evaluation Results

Experiment Platform

- **FPGA Board :**Virtex UltraScale+ VCU128
- **Host Server**: AMD Ryzen 5950x 16-core processors

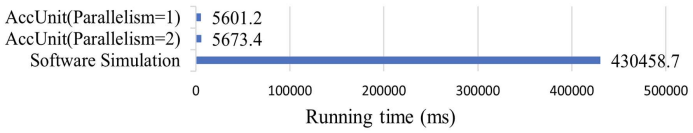

Figure 4: Performance Comparison between AccUnit and Software Simulation (1,000,000 iterations)

- AccUnit achieves up to a **75×** performance improvement compared to software simulation using Verilator.
- Double throughput when running 2 $< \mathrm{DUT}, \mathrm{Ref} >$ pairs in parallel.

Table 1: Resource Utilization of Coverage Instrumentation

Module	Cov Count	LUTs	Registers	LUTs with Cov	Registers with Cov
RAS	73	3043 (0.23%)	2007 (0.08%)	3206 (0.25%)	2072 (0.08%)
FTB	629	3167 (0.24%)	2621 (0.10%)	3697 (0.28%)	3175 (0.12%)
Vector	7078	152727 (11.72%)	28143 (1.08%)	168721 (12.94%)	35184 (1.35%)

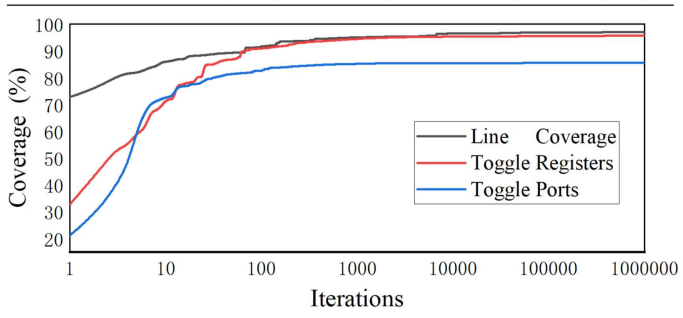

Figure 5: Coverage of Vector Module Collected by AccUnit

- AccUnit enables real-time coverage collection while maintaining acceptable hardware resource overhead.

[*] Equal Contribution
Weidong Li finishes this work during internship at Institute of Computing Technology, Chinese Academy of Sciences.