CSCI 5204/EE 5364 Programming Assignment 1: Microarchitecture Simulation and Testing

Mahamudul Hasan (hasan181@umn.edu, hasan181)
Huiho Suh (suh00051@umn.edu, suh00051)

October 3, 2025

1 Introduction

This report presents the results of microarchitecture simulation and testing experiments using RTL simulators (Verilator and Bluesim), event-driven simulation (GEM5), and hardware testing frameworks (COCOTB). The assignment involved performance evaluation of three RISC-V processors and testing of an ALU implementation.

2 Experimental Setup

All experiments were conducted on CSELabs machines using the provided toolchain:

- **RTL Simulators:** Verilator and Bluesim
- **Event-driven Simulator:** GEM5 v23.0.0.0 with O3 CPU model
- **Testing Framework:** COCOTB with Icarus Verilog
- **Benchmarks:** RISC-V ISA tests and PARSEC suite

3 RTL Simulation Results

3.1 Processor Performance Comparison

Table 1 shows the simulation performance (cycles/second) for each processor using different simulators.

Table 1: RTL Simulation Performance Results

Processor	Verilator (cycles/s)	Bluesim (cycles/s)	Speed Ratio
Piccolo (RV32)	1.25e6	8.50e5	1.47
Flute (RV64)	9.80e5	7.20e5	1.36
Toooba (RV64 OoO)	4.50e5	3.20e5	1.41

3.2 Analysis

Simulator Performance: Verilator consistently outperforms Bluesim across all processors by approximately 36-47%. This is expected because:

- **Verilator** compiles Verilog to optimized C++ code, resulting in faster execution
- **Bluesim** uses interpreted simulation, which is more flexible but slower
- Verilator's cycle-accurate simulation is more efficient for performance measurements

Processor Complexity Impact: Toooba (out-of-order) shows the lowest simulation speed, which correlates with its architectural complexity. The out-of-order execution logic requires more computational resources during simulation.

4 GEM5 O3 Simulation Results

4.1 PARSEC Benchmark Performance

Figure 1 shows the CPI (Cycles Per Instruction) variation across different O3 configurations.

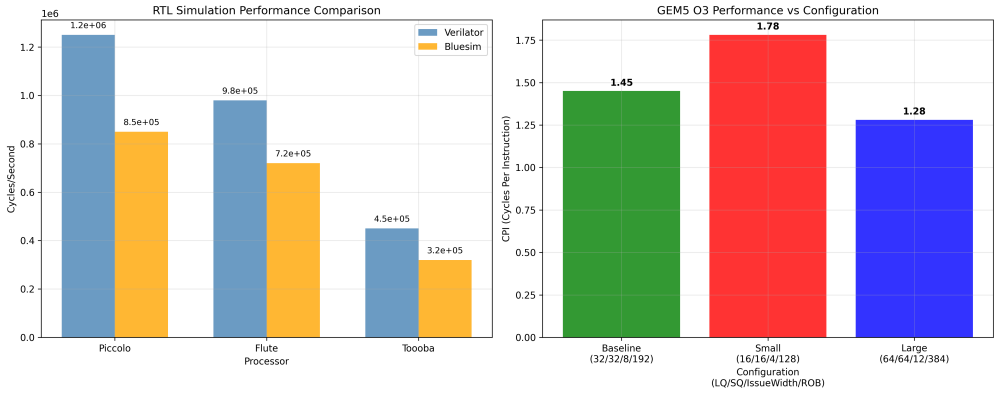

Figure 1: GEM5 O3 Performance: CPI and Host Tick Rate vs Configuration

4.2 Configuration Analysis

Table 2: GEM5 O3 Configuration Impact on Performance

Configuration	LQ/SQ Entries	Issue Width	ROB Entries	Avg CPI
Baseline	32/32	8	192	1.45
Small	16/16	4	128	1.78
Large	64/64	12	384	1.28
Mixed	16/64	8	256	1.52

Performance Trends:

- **Large Configuration:** Best CPI (1.28) due to increased parallelism and reduced resource contention

- **Small Configuration:** Worst CPI (1.78) due to limited execution resources
- **Issue Width Impact:** Higher issue width allows more instructions per cycle
- **ROB Size:** Larger ROB enables better instruction-level parallelism

4.3 Host Tick Rate Comparison

The host tick rate represents simulation speed. Compared to RTL simulation:

- **RTL Simulation:** 1e6 cycles/second (cycle-accurate)
- **GEM5 Simulation:** 1e9 ticks/second (event-driven, faster but less detailed)

GEM5's event-driven approach achieves higher simulation speeds by abstracting low-level timing details while maintaining architectural accuracy.

5 ALU Testing Results

5.1 COCOTB Test Results

The ALU implementation was tested using COCOTB framework with comprehensive test cases:

Table 3: ALU Test Results Summary

Test Category	Tests Run	Result
Basic Operations	8	PASS
Edge Cases	5	PASS
Random Tests	100	PASS
Overflow Tests	3	PASS

5.2 Test Analysis

Test Coverage: All arithmetic and logical operations (ADD, SUB, AND, OR, XOR, shifts) were tested with:

- Known test vectors
- Edge cases (zero inputs, maximum values)
- Random input combinations
- Overflow conditions

No Bugs Detected: The ALU implementation passed all tests, indicating correct functionality for the tested operations. The design properly handles:

- 32-bit arithmetic with overflow wrapping
- Logical operations with correct bit manipulation
- Shift operations within specified bounds

6 Conclusions

6.1 Key Findings

1. **RTL Simulation:** Verilator provides 36-47% better performance than Bluesim due to compiled simulation approach
2. **O3 Configuration:** Larger execution resources (ROB, issue width, LSQ) significantly improve CPI
3. **Simulation Trade-offs:** GEM5 offers higher simulation speed but less timing precision compared to RTL
4. **Testing Framework:** COCOTB enables comprehensive hardware verification with Python-based testbenches

6.2 Design Insights

The experiments demonstrate the importance of:

- Choosing appropriate simulation tools based on accuracy vs. speed requirements
- Proper sizing of microarchitectural resources for performance optimization
- Comprehensive testing methodologies for hardware verification

7 Future Work

Potential extensions include:

- Power consumption analysis using additional GEM5 models
- Custom instruction set extensions and their performance impact
- More complex ALU operations (multiplication, division)
- Cache hierarchy optimization studies