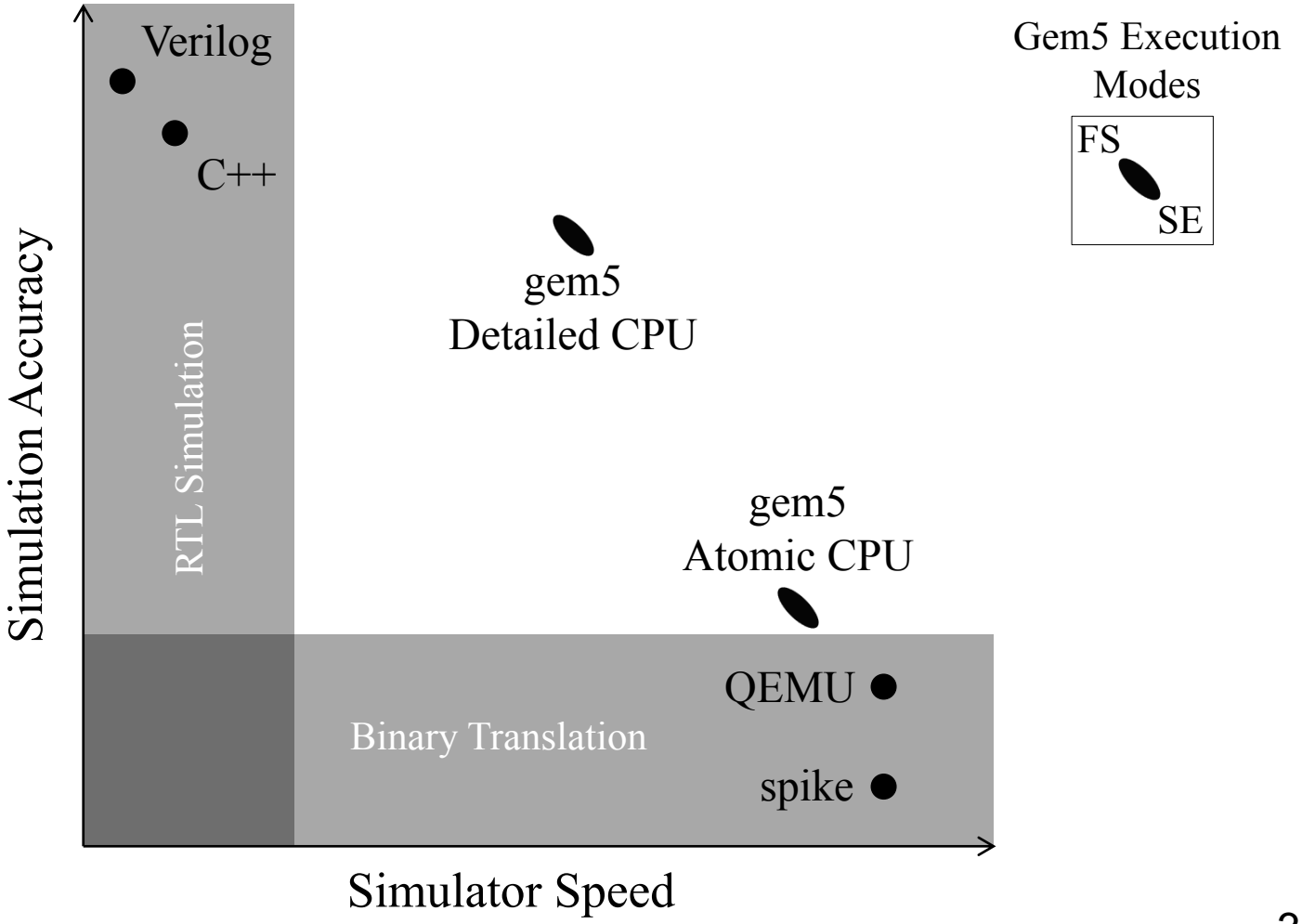
RISC5: Implementing the RISC-V ISA in gem5

Alec Roelke (ar4jc@virginia.edu)
Mircea R. Stan (mircea@virginia.edu)

Introduction

- Increasing complexities of designs increases simulation overhead
- Proprietary libraries and architectures impede research and collaboration
- Solution: gem5—fast, flexible, free
- Useful for RISC-V hardware development

RISC-V Simulation Platforms



Outline

- Implementation of RISC-V in gem5
- Comparison with Other RISC-V Simulators
- Performance and Validation
- Example Simulation Flow
- Future Work

Implementation of RISC-V in gem5

- Includes 64-bit base ISA + standard extensions (RV64G)
- Supports single-threaded execution in SE mode
- No support for privileged ISA yet

Integer and Multiply Instructions

- Most instructions based on MIPS
  - e.g. `fence` (RISC-V) : `sync` (MIPS)
- Internal behaviors mostly copied from MIPS and Alpha
- Nearly all instructions implemented
  - Only `eret` has no implementation
  - `fence` does not support ordering flags

Atomic Instructions

- *Release Consistency*: acquire an address to see changes after earlier release
- LR/SC and atomic RMW instructions
- Mark instructions with `ACQUIRE` or `RELEASE`
- RMW needs two memory operations—not supported with memory timing!
- Split into two micro-ops and add intermediate register

Floating-Point Instructions

- Development machine missing max magnitude round mode
- Verified against spike and a hardware design
  - Invalid computation results (NaN, ∞, 0)
  - Floating point exceptions

Simulation Features

Feature	gem5	Chisel	spike	QEMU
Binary Translation	✔		✔	✔
Checkpoints	✔			✔
Multicore Simulation	✔†	✔	✔	✔
Performance Statistics	✔	✔‡		
RTL Simulation		✔		
System Call Emulation	✔	✔§	✔§	

†Requires m5threads, which does not support RISC-V yet

‡Only with support from design and/or software

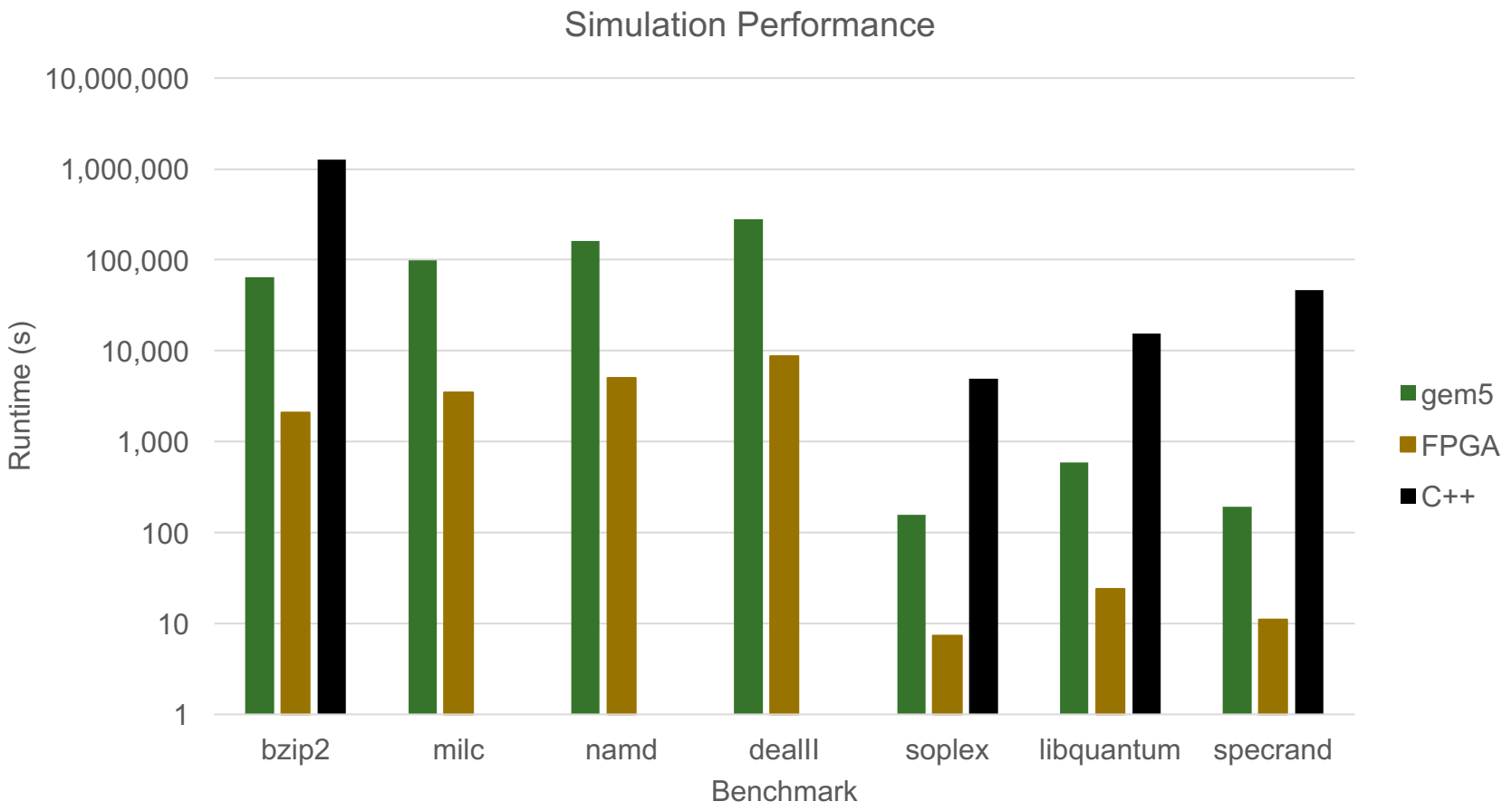
§System call emulation is supported via the RISC-V proxy kernel

Chisel: Bachrach et al., DAC 2012
spike: https://github.com/riscv/riscv-isa-sim
QEMU: Bellard, ATEC 2005

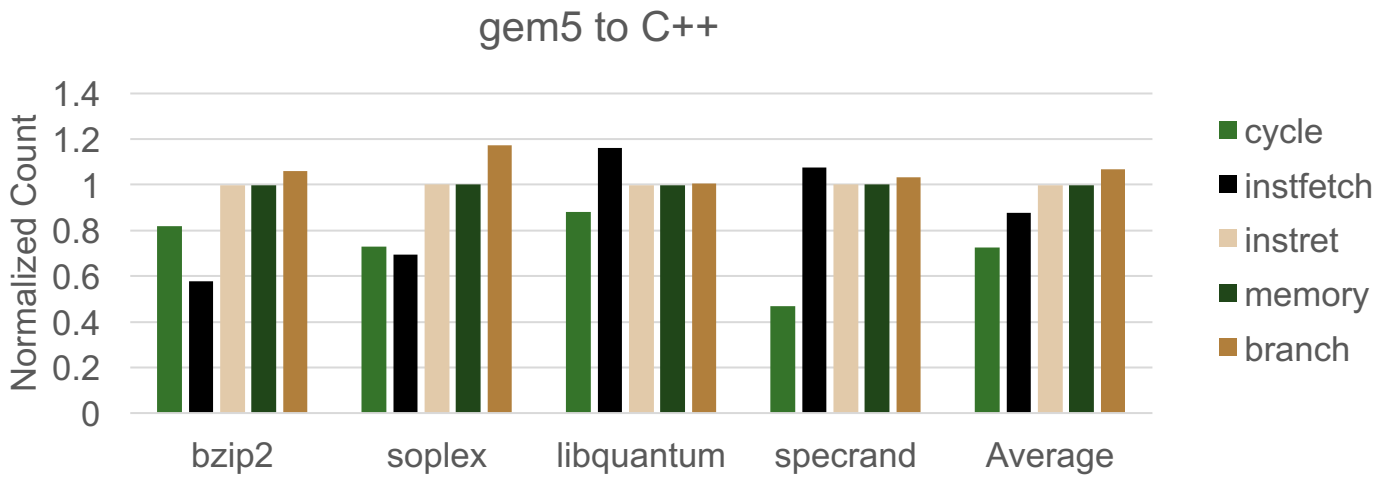
Performance and Validation

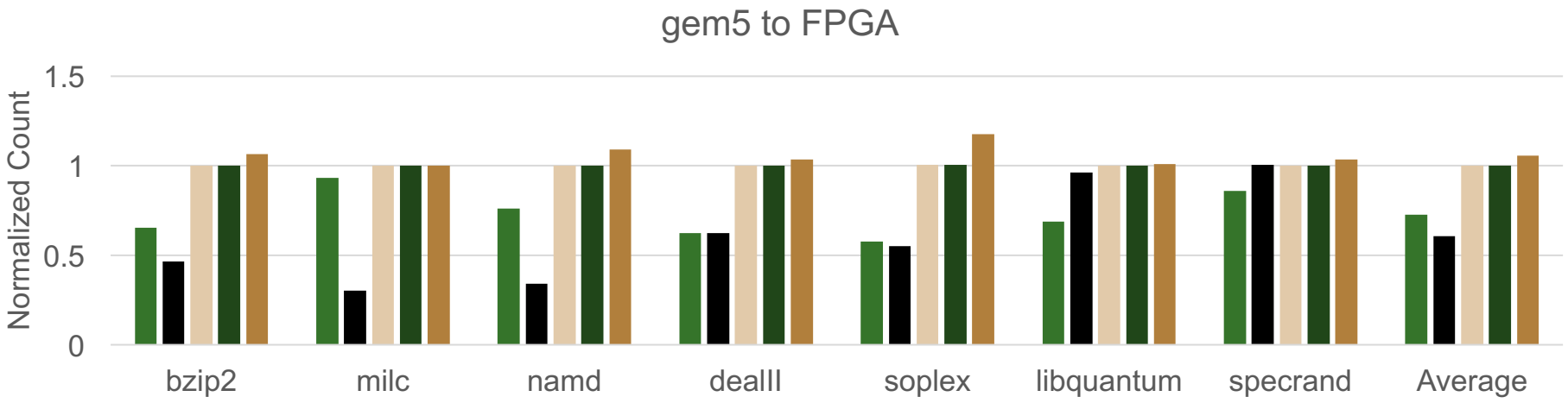
- Configure gem5 to resemble Rocket Chip
- Compare with FPGA and C++ simulator
- Run several SPEC CPU2006 benchmarks

Performance Results



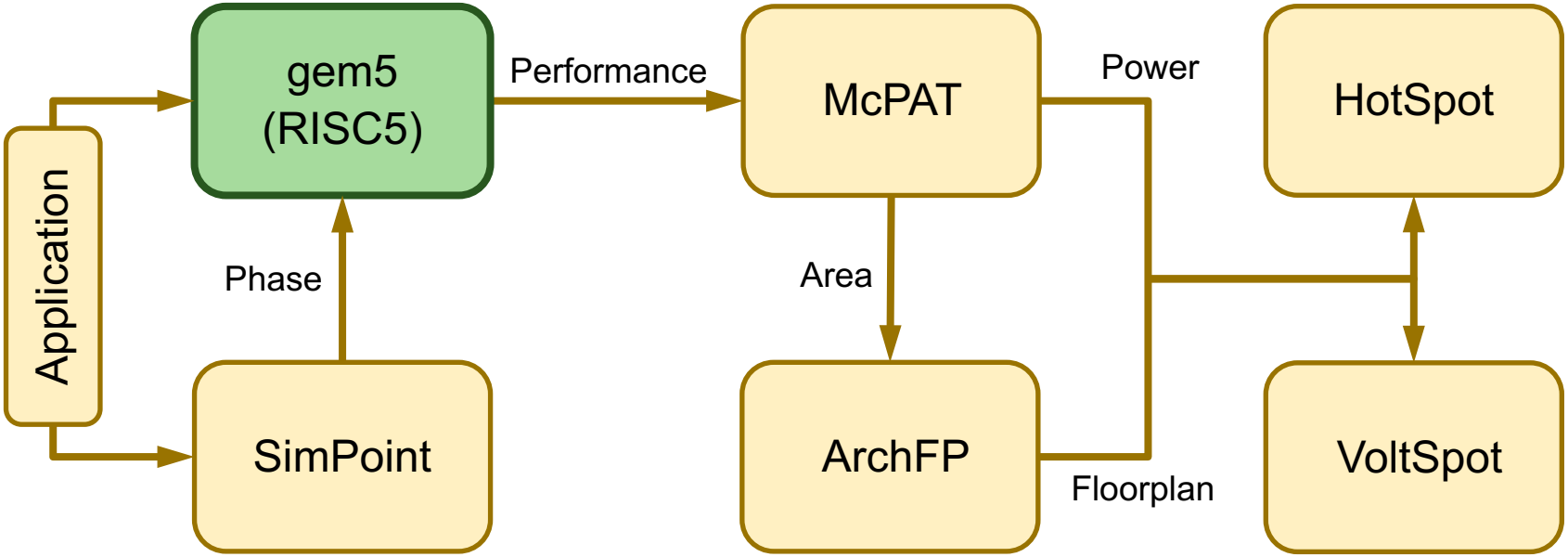
Validation Results





hplp

Example Flow

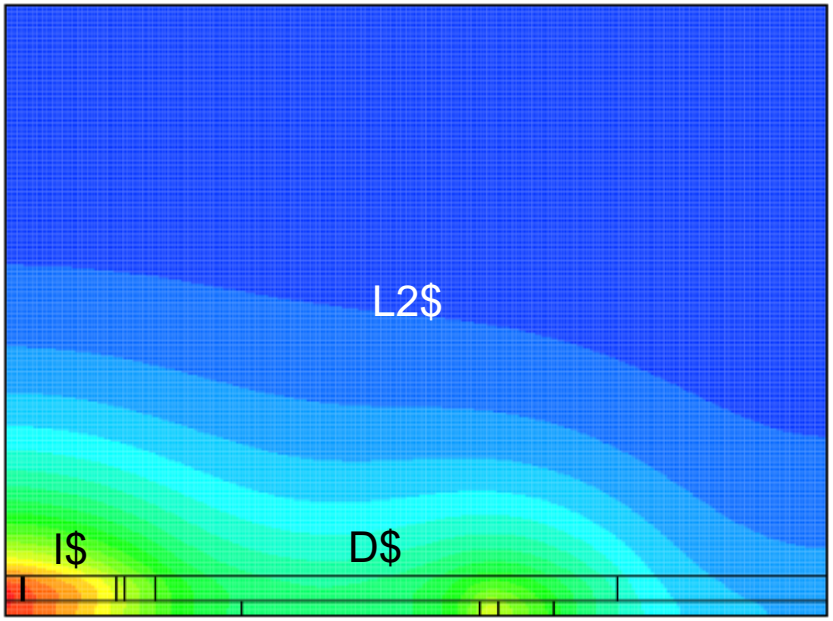


- Simulate a section of the *libquantum* SPEC CPU2006 benchmark
- Exclude VoltSpot due to cycle-by-cycle input requirement

SimPoint: Sherwood et al., ASPLOS 2002 ArchFP: Faust et al., VLSI-SoC 2012 VoltSpot: Zhang et al., ISCA 2014
McPAT: Li et al., MICRO 2009 HotSpot: Huang et al., TVLSI vol. 14 no. 5

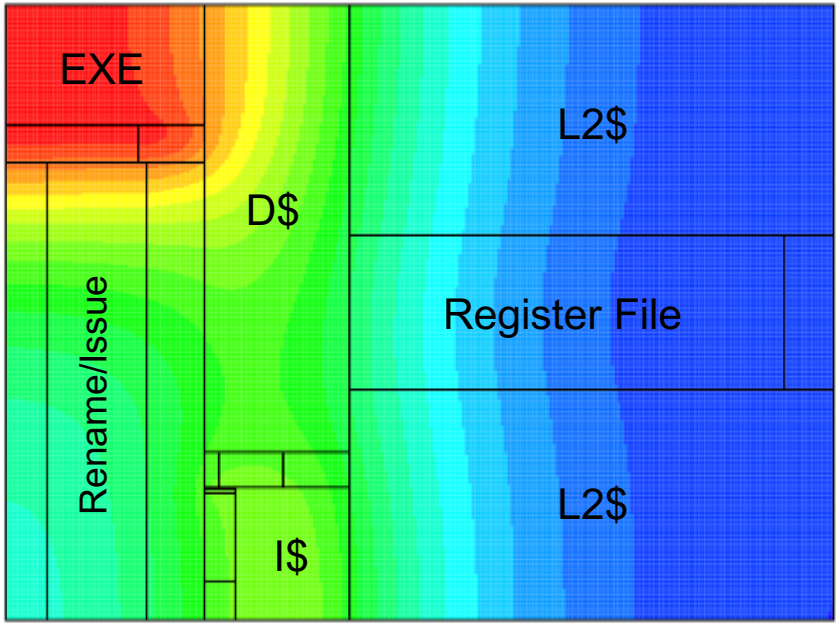
Example Results

Rocket Chip



Area: 4.175 mm^2
Power: 1.0215 W

BOOM



Area: 1.367 mm^2
Power: 5.1114 W

Process Size: 45 nm
Frequency: 1.5 GHz
V_{DD}: 1.0 V

hplp

Rocket Chip: https://github.com/freechipsproject/rocket-chip
BOOM: Celio et al., UCB/EECS-2015-167

Future Work

- Full support for all of gem5's features
  - Multithreaded workloads in SE mode
  - Full-system mode
  - Correct minor differences (e.g. floating-point rounding)
- Improve modeling accuracy
- Explore ISA effects on power/performance

Acknowledgments

- Pradip Bose, Schuyler Eldridge, and the rest of the IBM VELVET Team
- Members of the HPLP research group
- The gem5 community

Conclusion

- Implemented RISC-V in gem5
- Compared with and validated against Chisel C++ simulation and FPGA
- Executed an example simulation flow
- Significant work left until full support
- RISC5 is available as part of the main gem5 release at www.gem5.org

Version Information

Component	Version
gem5	`ffc29f2d9a5a`†
User-level ISA	2.1
Privileged ISA	N/A
Rocket Chip	`73e9508`
Chisel	`b18e98b`

†Refers to a Mercurial changeset hash; gem5 has since moved to git