

Validation of the gem5 Simulator for x86 Architectures

AYAZ AKRAM* (UNIVERSITY OF CALIFORNIA, DAVIS) AND LINA SAWALHA
(WESTERN MICHIGAN UNIVERSITY)

* FINISHED THIS WORK WHILE AT WMU



gem5 and HPC Simulation

Features suitable for HPC

Heterogeneous

Full System Support

Multicore / Multi-system simulation



Examples of HPC Simulation

AMD Exascale

ARM HPC Simulation

The Need of Simulator Validation

Simulation Errors

Modeling

Abstraction

Specification

Why validation is important?

State of gem5 validation

ARM validation efforts

Unaware of any x86 work

Validation Methodology

Benchmarks used

- Microbenchmarks

Target

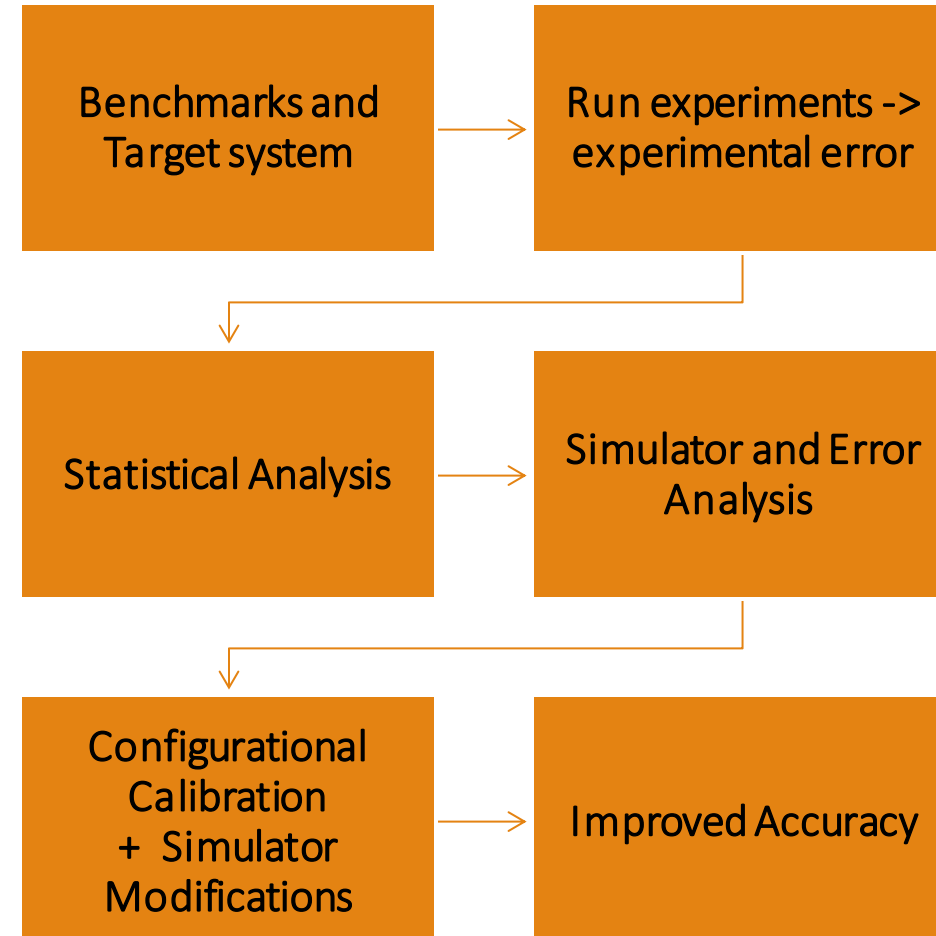
- Intel Haswell like

Statistical Analysis

- Correlation study

Modifications

- Configurational calibration
- Simulator changes



Microbenchmarks [1]

Control Benchmarks

- Hard to predict branches, random branches, indirect jumps, inflight branches

complex, random, switch, small

Dependency Benchmarks

- Dependency chains of various lengths

dep1, dep2, ... dep5, dep6

Execution Benchmarks

- Independent arithmetic operations

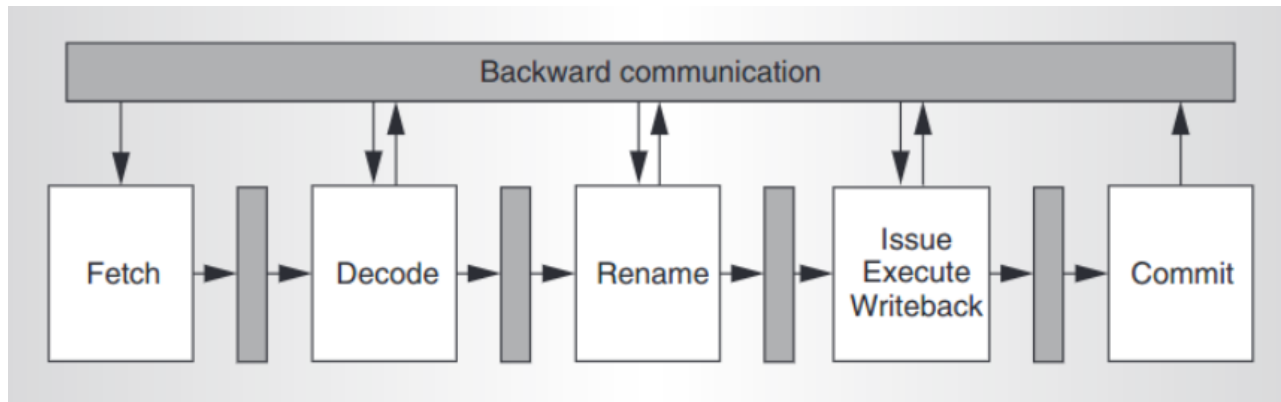
Int add, mul, and fp add, mul, div

Memory Benchmarks

- Dependent/independent load/stores

load dep, load ind, store ind

Target Configurations (Intel Haswell)



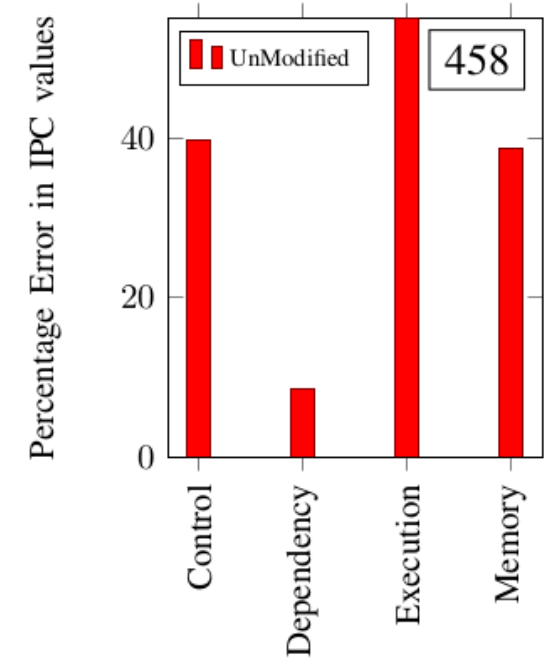
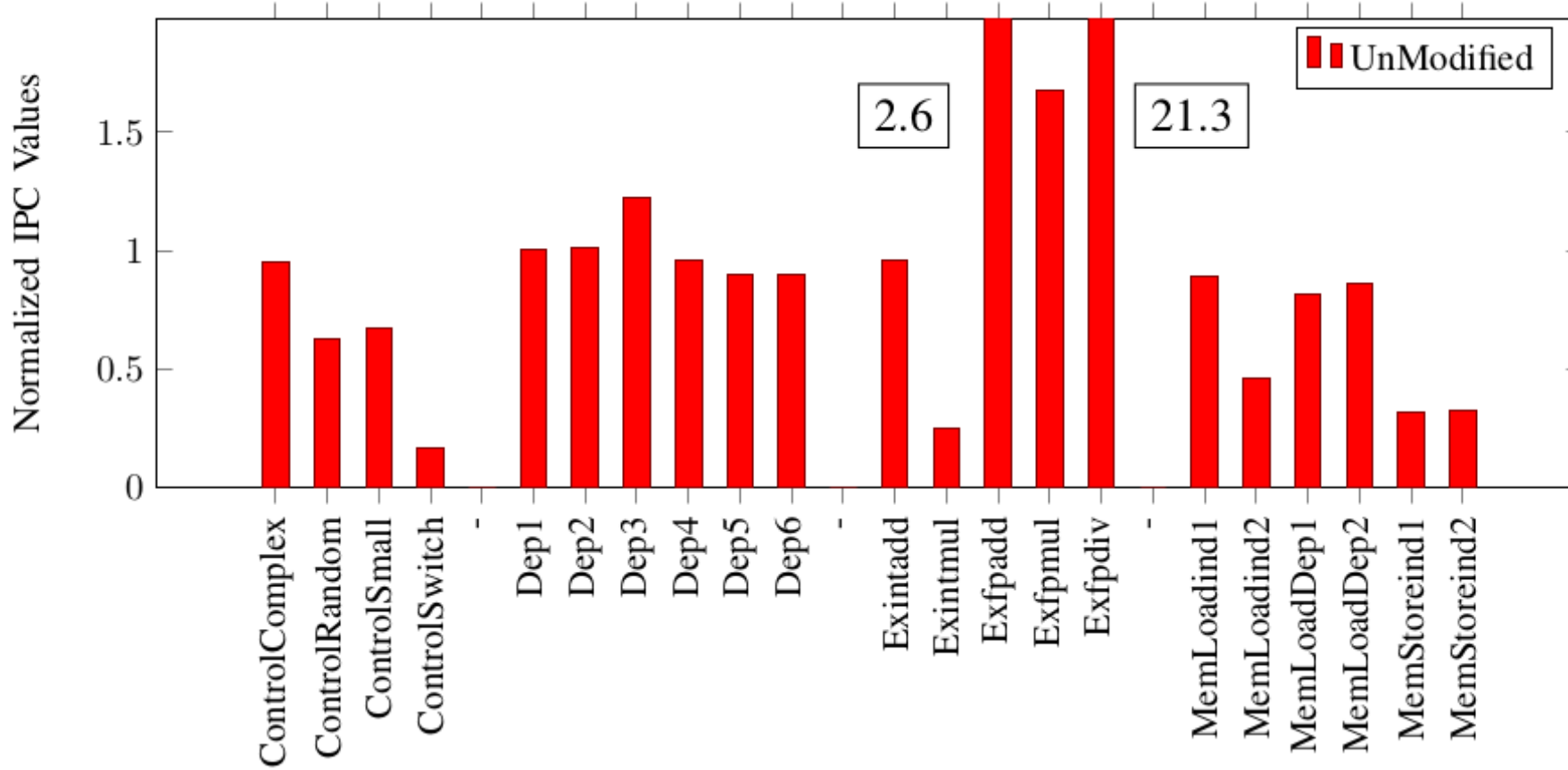
gem5 O3CPU pipeline

Parameter	Core i7 Like
Pipeline	Out of Order
Fetch width	6 instructions per cycle
Decode width	4-7 fused μ -ops
Decode queue	56 μ -ops
Rename and issue widths	4 fused μ -ops
Dispatch width	8 μ -ops
Commit width	4 fused μ -ops per cycle
Reservation station	60 entries
Reorder buffer	192 entries
Number of stages	19
L1 data cache	32KB, 8 way
L1 instruction cache	32KB, 8 way
L2 cache size	256KB, 8 way
L3 cache size	8 MB, 16 way
Cache line size	64 Bytes
L1 cache latency	4 cycles
L2 cache latency	12 cycles
L3 cache latency	36 cycles
Integer latency	1 cycle
Floating point latency	5 cycles
Packed latency	5 cycles
Mul/div latency	10 cycles
Branch predictor	Hybrid
Branch misprediction penalty	14 cycles

Results and Analysis

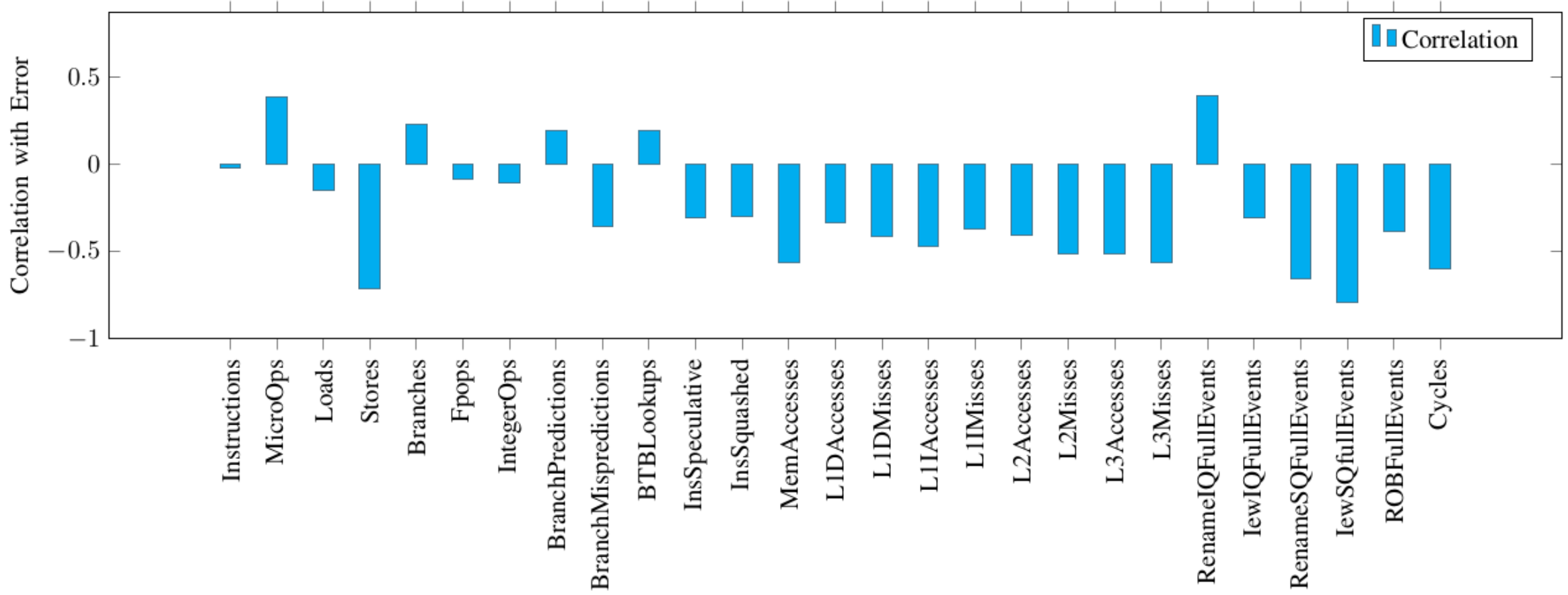
Observed Inaccuracies

High inaccuracy in some cases

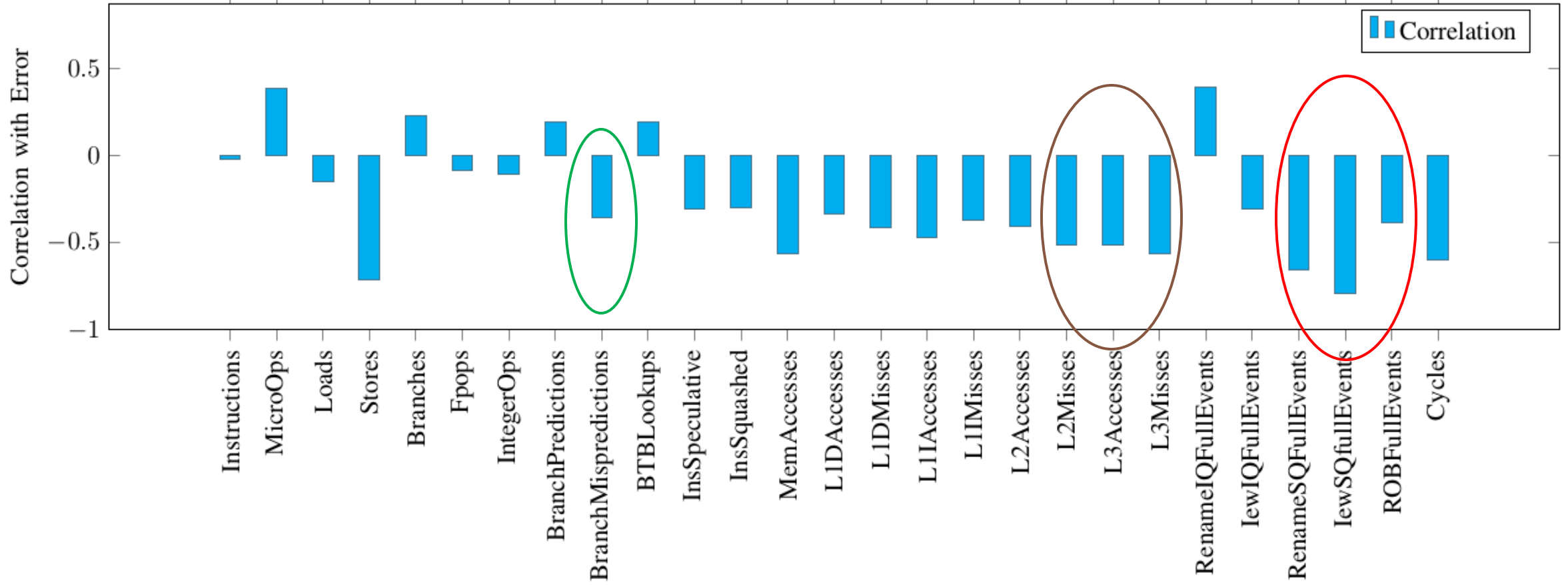


Average Error

Correlation Analysis



High negative correlation with percentage error



Examples of Configurational Calibration

IssueToExecute Delay Parameter

- Setting beyond one inhibits back-to-back execution of instructions
- Fix: control_small's inaccuracy 33.6% --> 0.4%

Fetch unit's throughput

- Blocking requests to I-cache
- Fixing I-cache hit latency to 1
- Improved accuracy for control_switch by 3x

Examples of Simulator Modifications

x86 Instruction to micro-op decoding

- Micro-op / instruction ratios --> very high
- Relied on ZSim, Sniper and instruction set manuals
- Achieved 5% of what is observed on real hardware and other sources
- Improved accuracy for many benchmarks
 - 10.5% --> 5.1% for dep5
 - 6.5% --> 1.5% for ex_int_add

Instruction Labels

- Misclassification of FP mul/div operations as FP add
- Fix improves the accuracy significantly

Examples of Simulator Modifications

Memory_load_ind2 (independent loads from 32KB array)

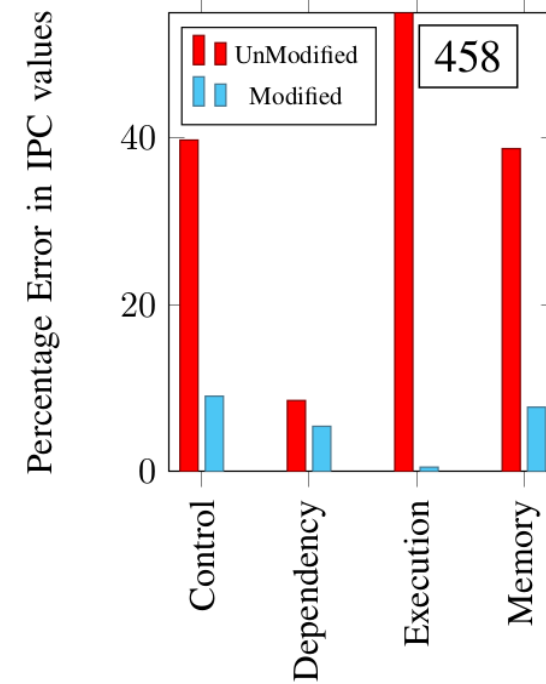
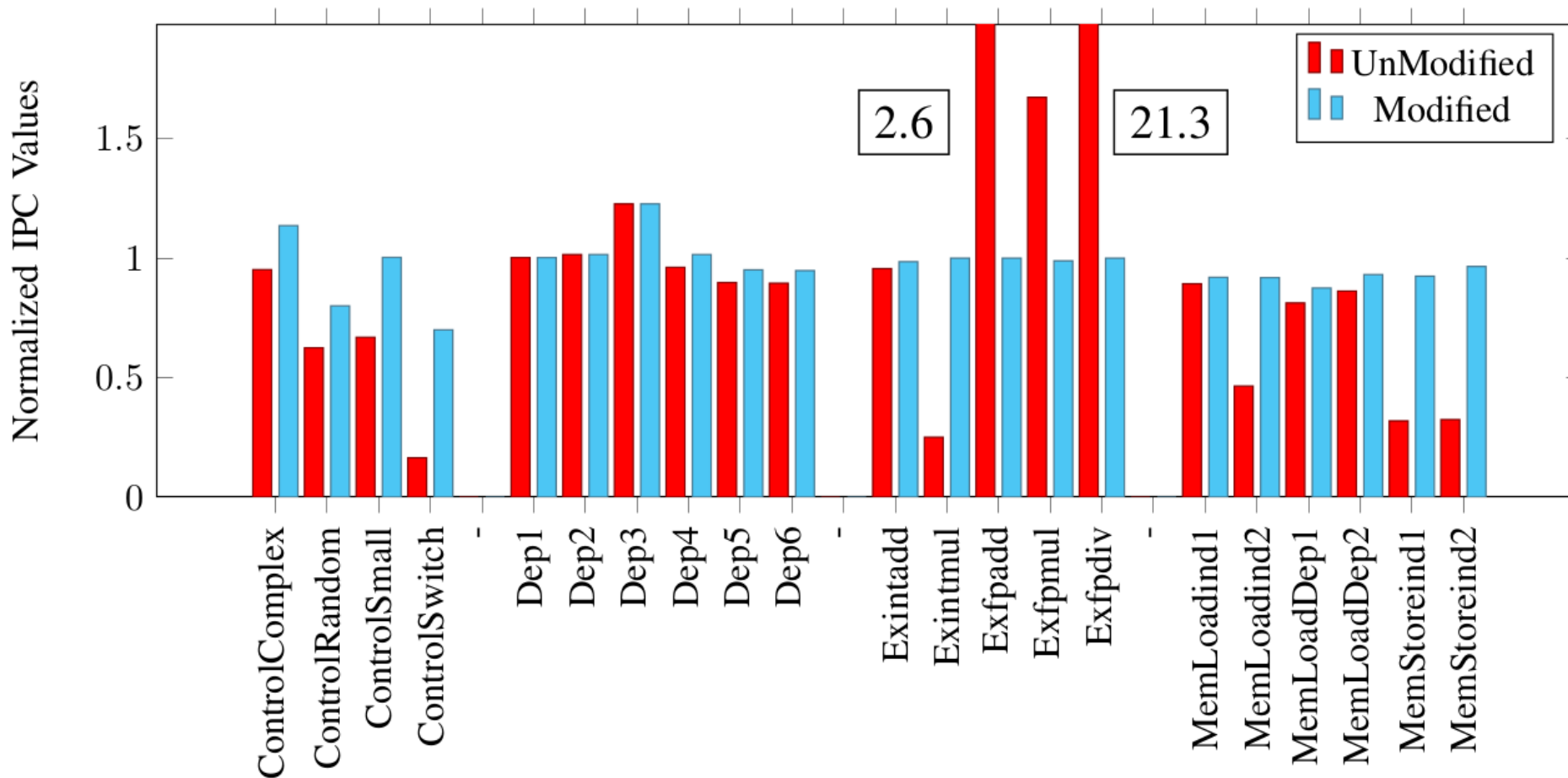
Mis-predicted to be taken on
loop termination

As a result, these accesses kick
out useful blocks

```
for (i = 0; i < repetitions; i++) {  
    for (jump = 0; jump <= size - 32; jump += 32) {  
  
        count += ptr_vector[jump + 0].value;  
        count += ptr_vector[jump + 1].value;  
  
        .....  
        count += ptr_vector[jump + 30].value;  
        count += ptr_vector[jump + 31].value;  
  
    }  
}
```

Improved Accuracy

Absolute Mean Error rate < 6%



Takeaways

- Importance of validation of simulators
- Improved accuracy of gem5's x86 O3CPU model for a given target
- Future work: studies with realistic HPC benchmarks and targets

