

## ABSTRACT

Computer architecture simulators are widely used by computer architects to evaluate different design options. This work explores different x86 computer architecture simulators and quantifies the simulation inaccuracies. x86 is one of the oldest and widely used instruction set architectures (ISAs) used in desktops and servers. We selected *gem5*, *Sniper*, *MARSSx86* and *Zsim*, and configured them to model one of the state-of-the-art high-performance processors, Intel's *Haswell* microarchitecture. We compare simulators features and statistics, and quantify the experimental error of simulators for single- and multi-core runs compared to a real hardware platform.

## SELECTED x86 SIMULATORS

### Gem5 (v. Sep. 2015) [2]

- Full-system & application-level cycle-level simulator.
- Supports many ISAs (x86, ARM, SPARC, Alpha, MIPS).
- Supports various CPU models (non-pipeline, in order pipeline, out-of-order pipeline).
- Highly configurable.

### Sniper (v. 6.0)[3]

- Many-core application-level x86 simulator.
- Provides a balance between detailed cycle-level simulation and one-IPC (single issue pipeline model) simulation.
- 'Instruction window centric' core model added to the simulator to improve its accuracy.
- Supports in order and out-of-order pipeline models.

### MARSSx86 (v. 0.4)[4]

- Full-system cycle-level x86-64 simulator.
- Based on PTLsim and QEMU.
- Supports in order and out-of-order pipeline models.

### ZSim (v. Apr. 2016) [5]

- Parallel and scalable application-level x86-64 simulator.
- Supports in order and out-of-order pipeline models.
- Extensively uses dynamic binary translation and focuses on simulating detailed memory hierarchies.

Table 1: Feature Comparison of Selected Simulators

Feature	Gem5	Sniper	MARSSx86	Zsim
Platform / target Support	P++	P	P	P
Full System	✓	X	✓	X
Fast forwarding & cache warmup	✓	✓	✓	✓
Checkpointing	✓	X	✓	✓
Details of stats.	D++	D	D+	D+
Energy/power	E+	E	E	E
HMP support	M,G,S	S	S	S
GPU modeling	✓	X	X	X
In Order Pipeline	✓	✓	✓	✓
Community support	C++	C++	C++	C+

Note: [feature's 1<sup>st</sup> letter]++ is better than [feature's 1<sup>st</sup> letter] + which is better than [feature's 1<sup>st</sup> letter] which is better than [feature's 1<sup>st</sup> letter] -, S=Single-ISA, M=Multi-ISA, G=GPU

## METHODOLGY

- All simulators configured to model hardware configuration similar to Intel *Haswell*, Intel i7-4770 CPU with 3.4 GHz (see Table 2).
- SPEC-CPU2006 and a subset of MiBench embedded benchmark suites simulations for timing and performance results compared to real hardware runs.
- SPEC benchmarks executed for 500 million instructions chosen from a statistically relevant portion of the program, after a warming up period of 100 million instructions.
- IPC (instructions per cycle), branch misprediction and cache miss ratios measured on real hardware using hardware monitoring counter tools (PAPI).
- The same 64-bit binaries used for all simulators.

Table 2: Target Configurations

Parameter	Core i7-like
Pipeline model	Out of order
Fetch width	6 instructions per cycle
Decode width	4-7 fused mops
Decode queue	56 uops
Rename and issue widths	4 fused uops
Dispatch width	8 uops
Commit width	4 fused uops
Reservation station	60 entries
Reorder buffer	192 entries
Stages	19
L1 data cache	32KB, 8 way
L1 instruction cache	32KB, 8 way
L2 cache size	256 KB, 8 way
L3 cache size	8 MB, 16 way
L1, L2 and L3 cache latency	4,12 and 36 cycles
Branch predictor	Tournament
Branch misprediction penalty	14 cycles
BTB and RAS entries	4K and 16

## RESULTS

Note: *avg-E* : average absolute error

*avg-E-NO*: average absolute error with no outlier (more than 50% error)

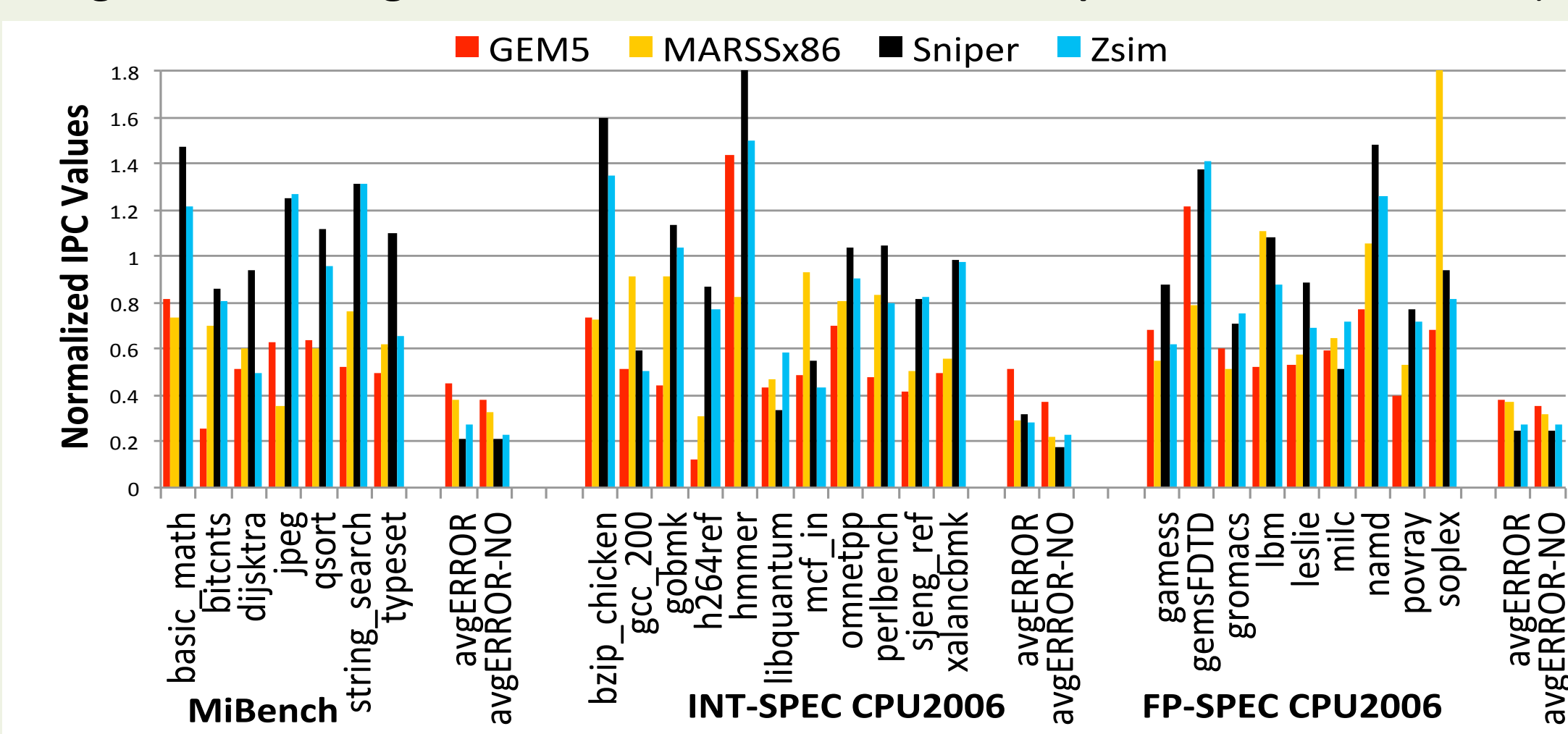


Figure 1: Normalized IPC values for single core runs

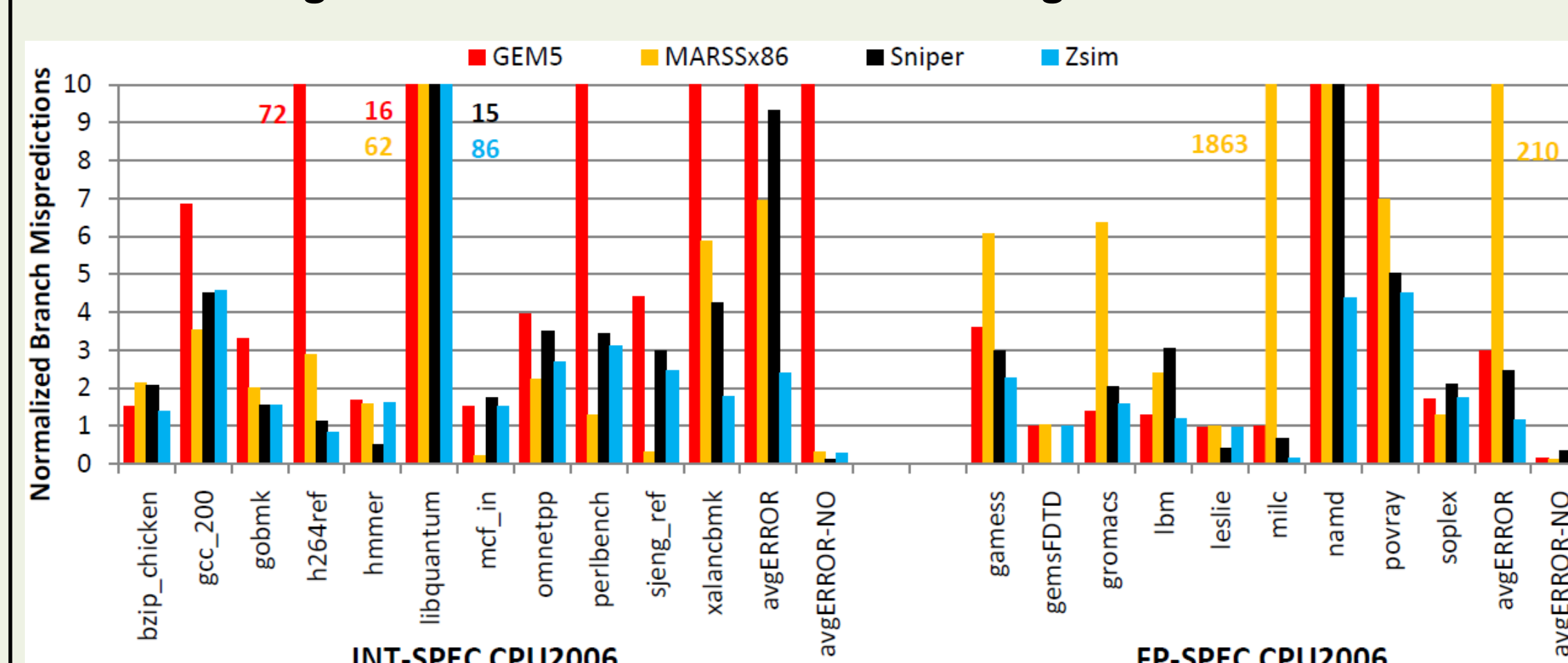


Figure 2: Normalized branch mispredictions for single core runs

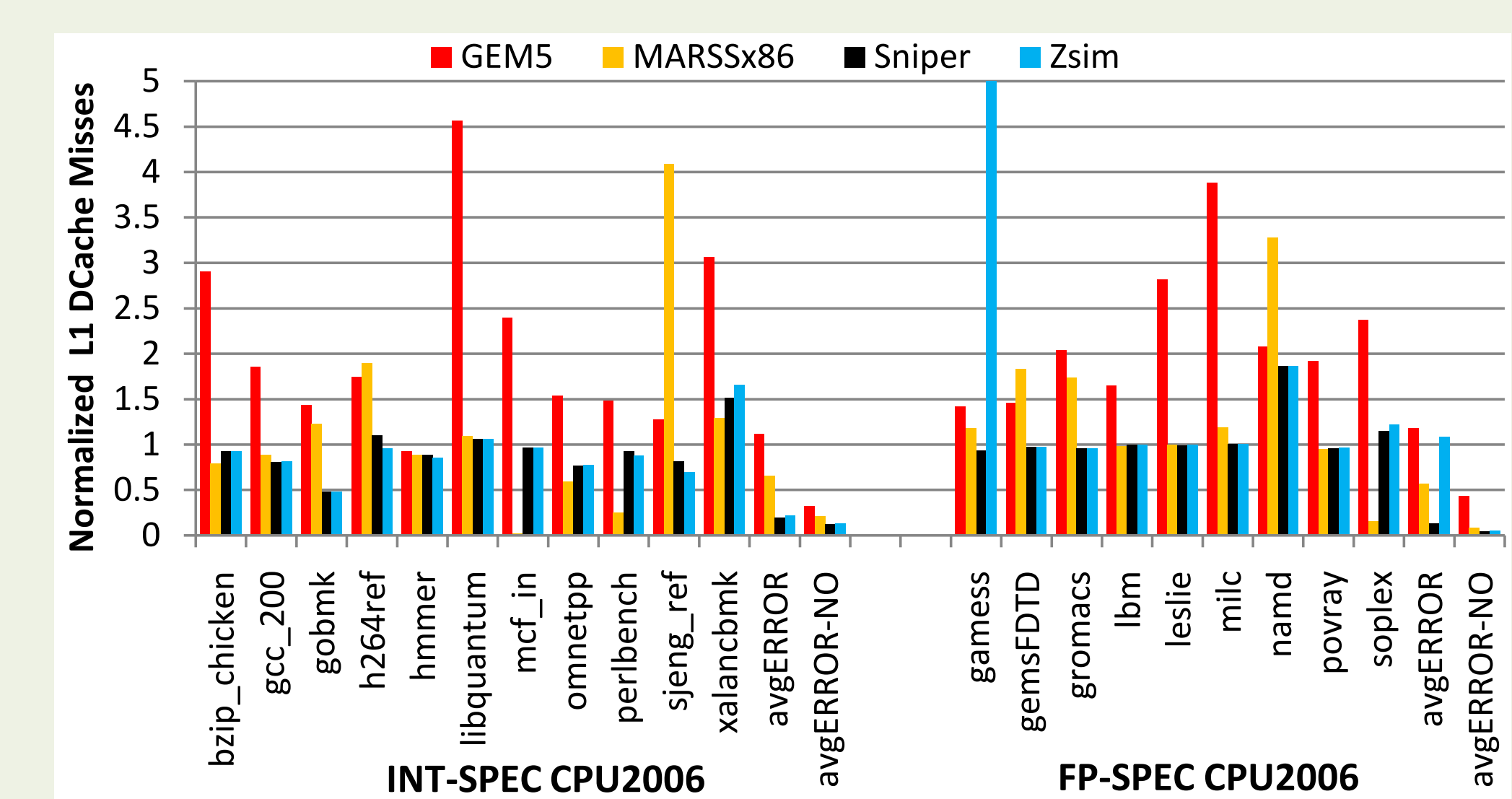


Figure 3: Normalized L1-d cache misses for single core runs

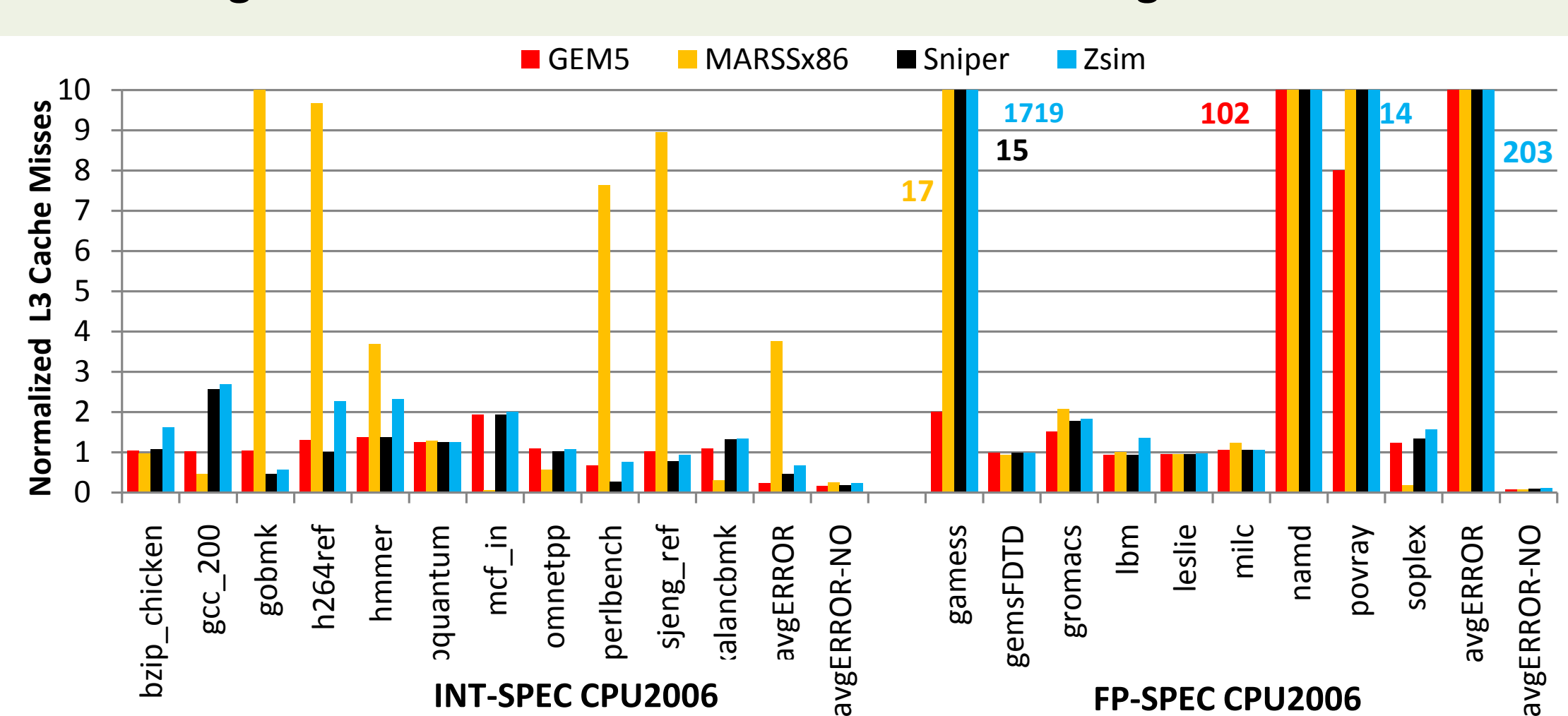


Figure 4: Normalized L3 cache misses for single core runs

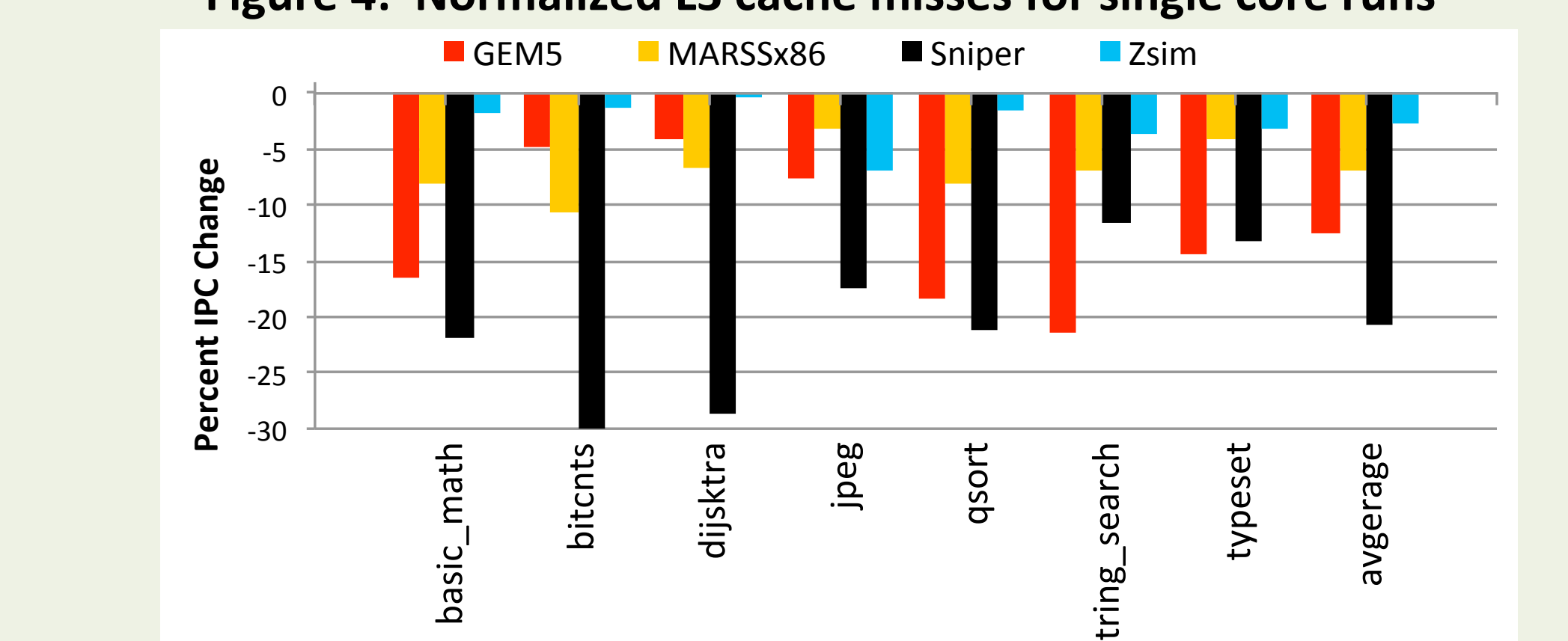


Figure 5: Percent change in IPC after halving the width of pipeline stages

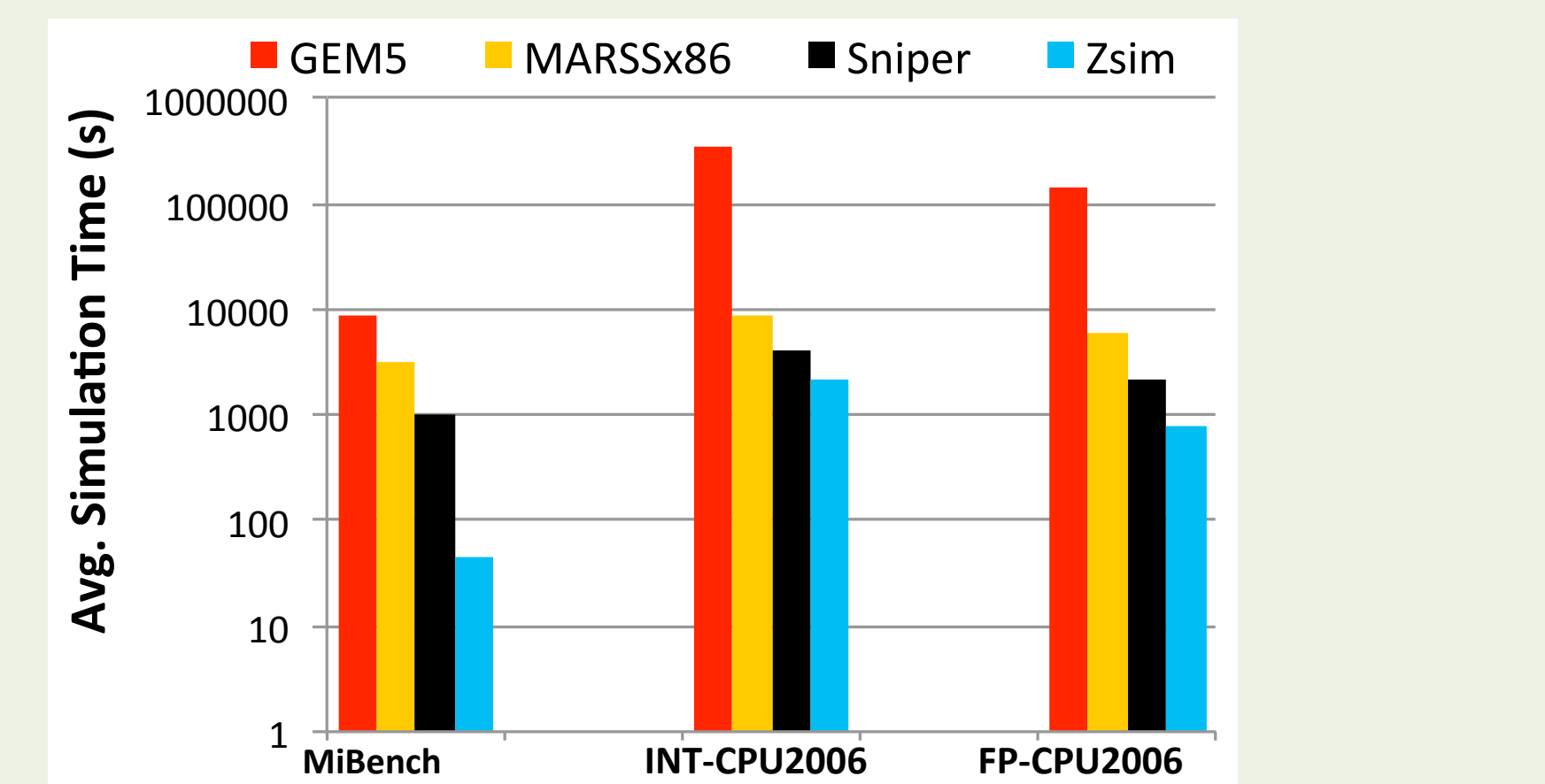


Figure 6: Average simulation time for all simulators (seconds)

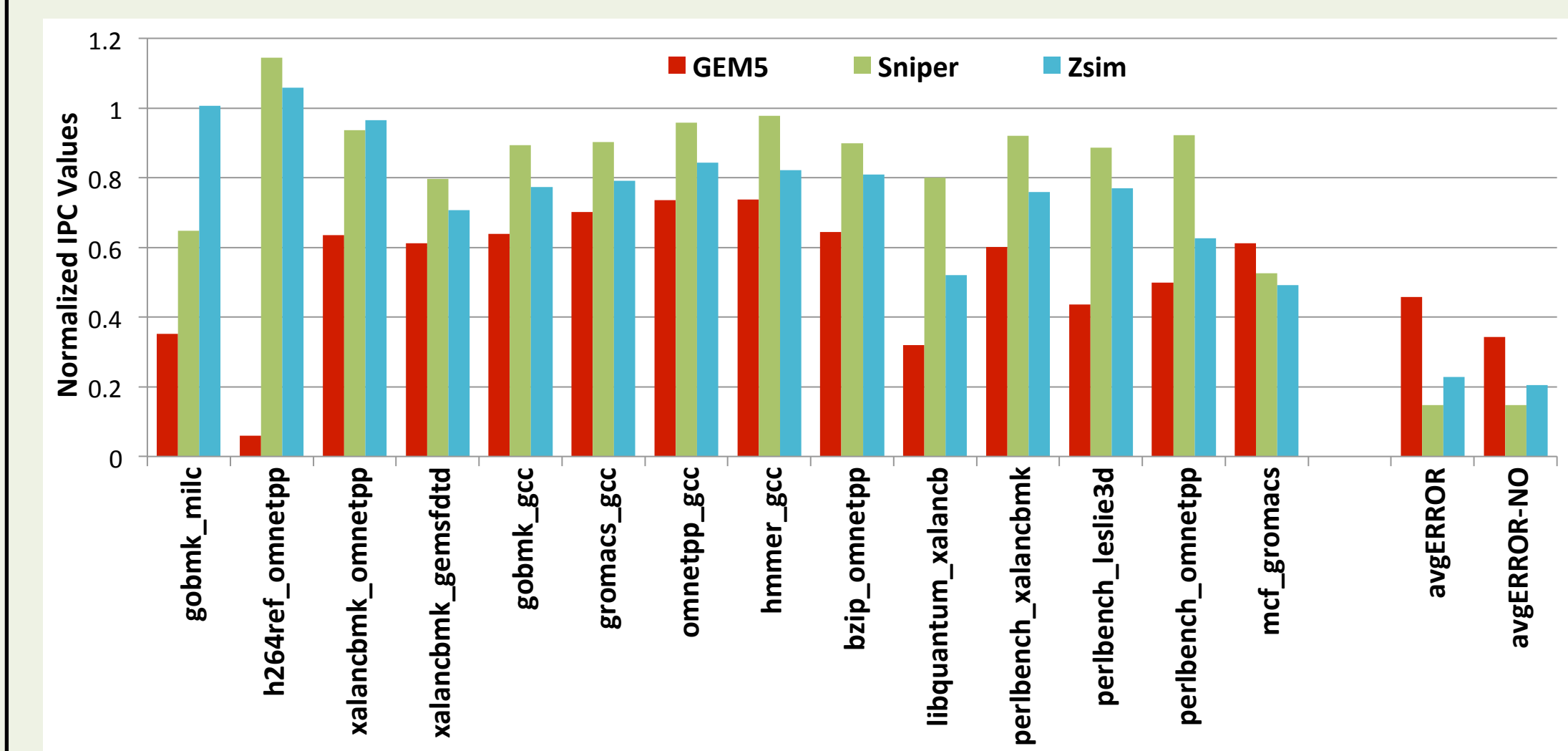


Figure 7: Normalized IPC values for dual-core runs for SPEC CPU

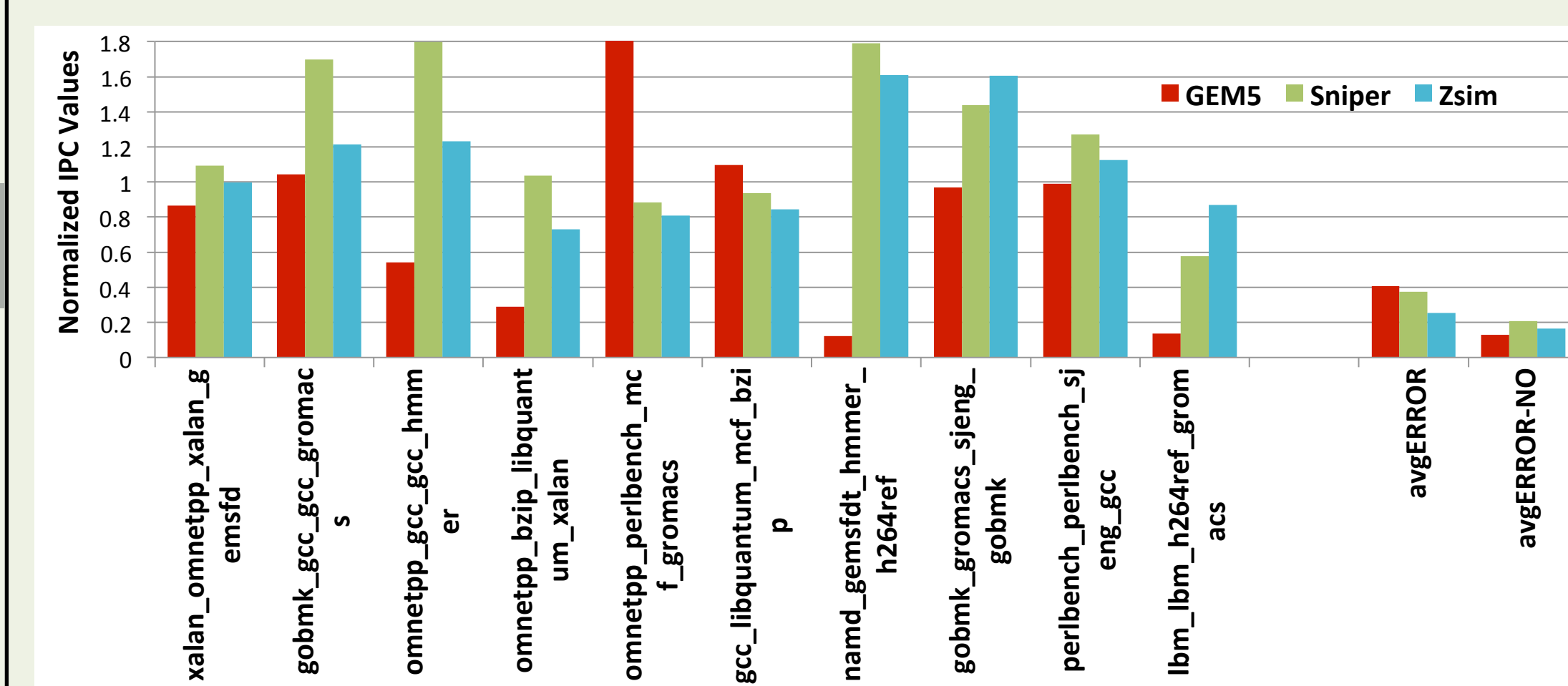


Figure 8: Normalized IPC values for quad-core runs for SPEC CPU

➤ Sniper has the least experimental error for all types of workloads.

➤ Zsim is the fastest simulator.

➤ Sniper and Zsim show similar experimental error for dual core and quad core runs (less than that on gem5).

➤ Observed sources of inaccuracies in the simulators:

- Simulation abstraction.
- High number of committed branch instruction and high experimental error in branch misprediction rate.
- High experimental error in cache misses.
- Lack of support of fused microoperations (uops), and uop cache of *Haswell* (significantly reduces the effective pipeline depth in case of u-op cache hit).
- Inaccurate decoding of instruction to uops.

## CONCLUSIONS

This study emphasizes on the importance of validating simulators and aims to help the community to point out sources of inaccuracies in simulators that can be modified later in future work. The obtained experimental results indicate that x86 simulators, which have been validated for Intel recent x86 architectures show less error compared to the ones that are not validated and calibrated for such targets. Errors due to abstraction and lack of details in the simulators do not necessarily always imply inaccurate simulation tools, as thoroughly validated simulators can still achieve acceptable relative performance. In future, we plan to discover more sources on inaccuracies of simulators and potentially fix them.

## REFERENCES

- [1] A. Akram and L. Sawalha, "x86 Computer Architecture Simulators: A Comparative Study," in *IEEE ICCD*, pp. 638-645, Oct. 2016.
- [2] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, et al., "The gem5 Simulator," *SIGARCH Comp. Arch. News*, vol. 39, pp. 1-7, 2011.
- [3] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the Level of Abstraction for Scalable and Accurate Parallel Multi-Core Simulation," in *ACM Int. Conf. for High Perf. Comp., Net., Storage and Analysis*, pp. 1-12, 2011.
- [4] A. Patel, F. Afram, and K. Ghose, "MARSS-x86: A QEMU-Based Micro-Architectural and Systems Simulator for x86 Multicore Processors," in *DATE* p. 29-30, 2011.
- [5] D. Sanchez and C. Kozyrakis, "ZSim: Fast and Accurate Microarchitectural Simulation of Thousand-Core Systems," in *ISCA*, vol. 41, pp. 475-486, 2013.
- [6] T. E. Carlson, W. Heirman, S. Eyerhan, I. Hur, and L. Eeckhout, "An evaluation of high-level mechanistic core models," *ACM TACO*, vol. 11, no. 3, p. 28, 2014.
- [7] <http://zsim.csail.mit.edu/tutorial/slides/validation.pdf>