A Brief Analysis of the SPEC CPU2000 Benchmarks on the Intel® Itanium® 2 Processor

James McCormick, HP

Allan Knies, Intel

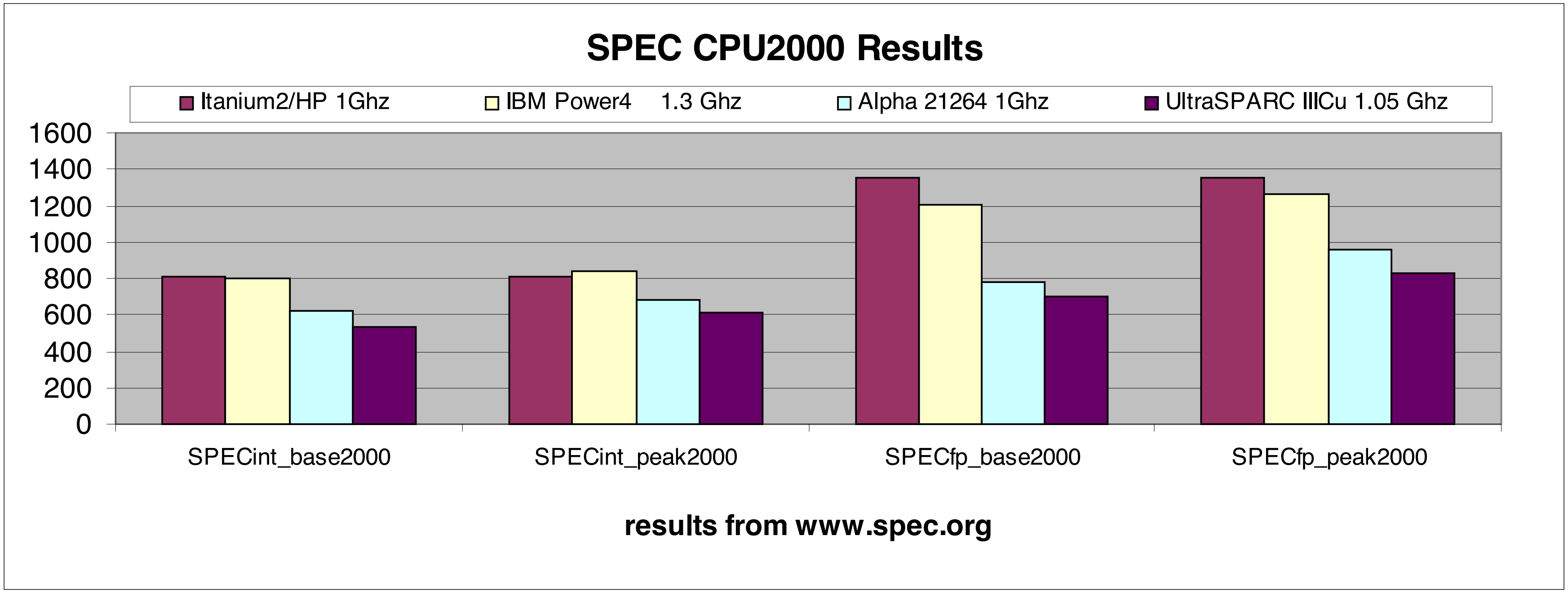
Agenda

This is a data-oriented presentation, not research

Agenda

- Brief performance summary
- Comparison of how HP and Intel compilers use Itanium® architecture features and compare to best RISC.
- Analysis of microarchitectural features of the Intel® Itanium® 2 processor and how it affects SPEC CPU2000 performance

Overall Performance



SPEC{int/fp}_base2000	810/1356	(best 0.18u)
Linpack 1000:	3.5 Gflops	(best overall)
TPC-C (SQL/4P):	78K tpmC	(best 4P number)
SPECweb_SSL:	1520 connections	(best of class)

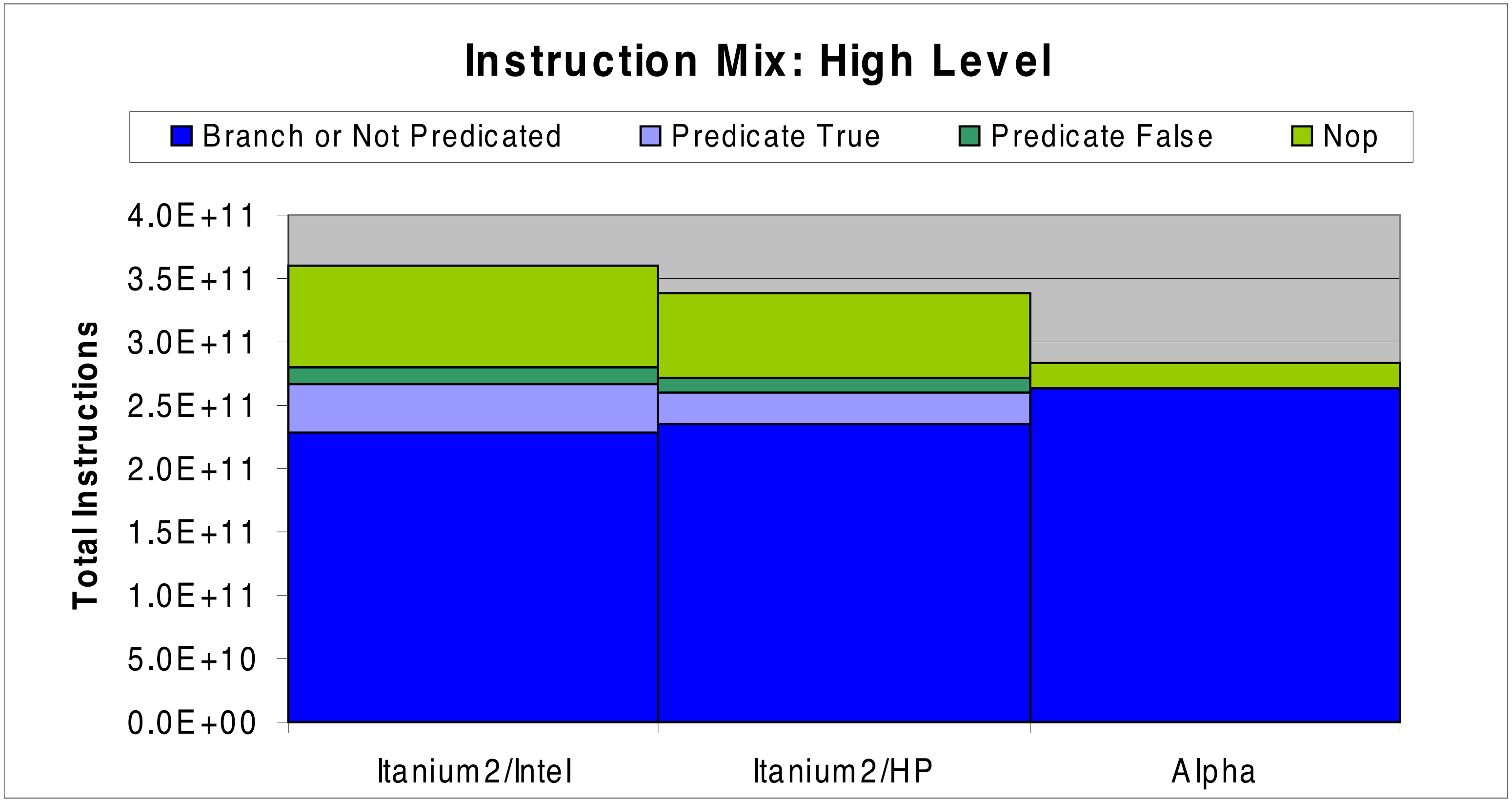
- **Itanium® 2 processor best of class on a wide range of applications**

Part I: ISA and Compiler Comparisons

Allan Knies

Itanium® Architecture and Performance Team

Intel Corporation
allan.knies@intel.com



- Number of ‘useful instructions’ (shown in blue) +/- 1% of Alpha
- Total instructions (blue+green) +20-30% of Alpha (due to NOPs)
- Bundling is main cause of extra NOPS

➢ **We’ll see that extra instr/nops are not substantially impacting perf**

Instruction Mix Details Introduction

Itanium/Alpha ISAs:

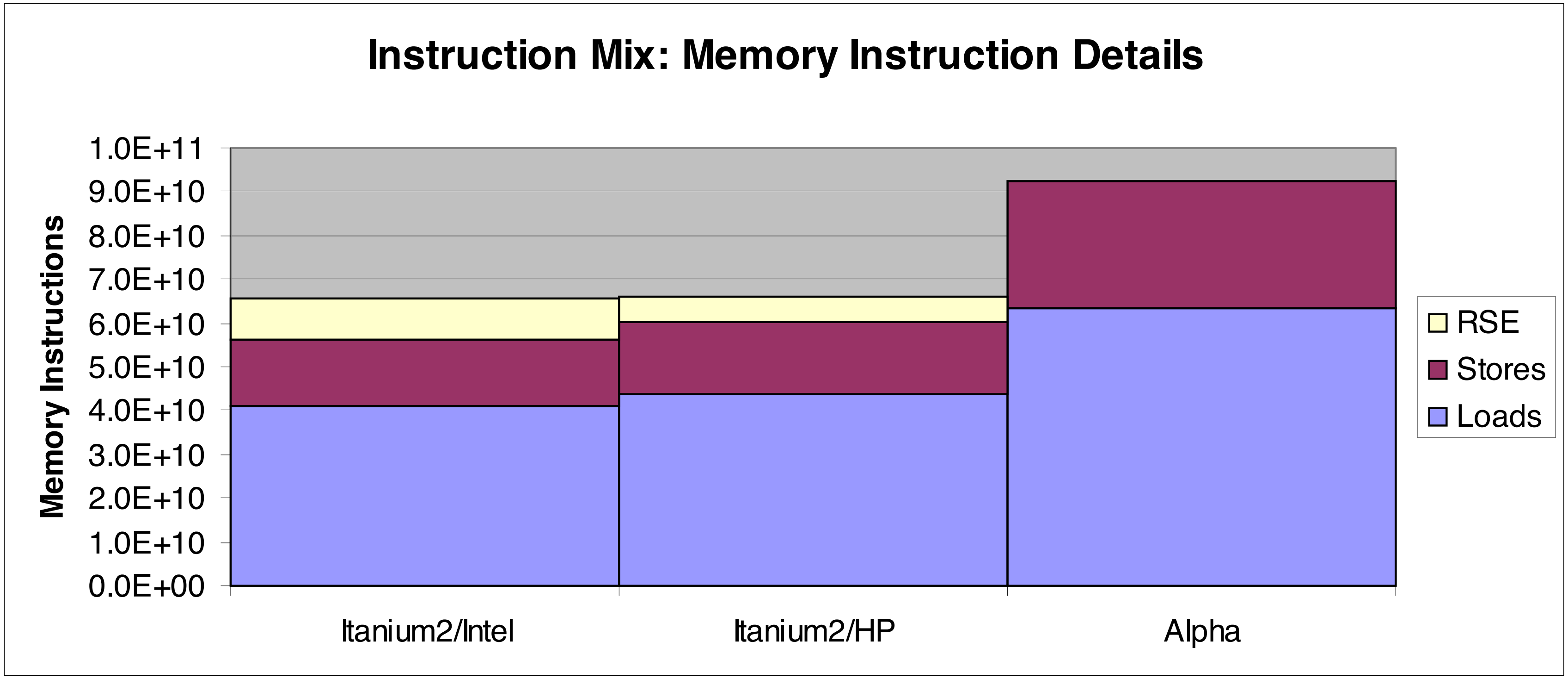
- Itanium® arch has 40% fewer memory operations and 30% fewer branches than Alpha (some impact from no-pgo on Alpha)
- Itanium arch has about 10% more ALU/compares/shifts than Alpha
- Itanium arch has about 20-30% NOPs – eventually expect this to be under 20%

HP/Intel Compiler:

- HP compiler uses more memory and ALU ops than Intel
- Implies HP more conservative with registers – we'll see impact later

➢ **Itanium® architecture trades more 'easy' instructions (NOP, alu, cmp) for reducing the 'hard' instructions (branch, load)**

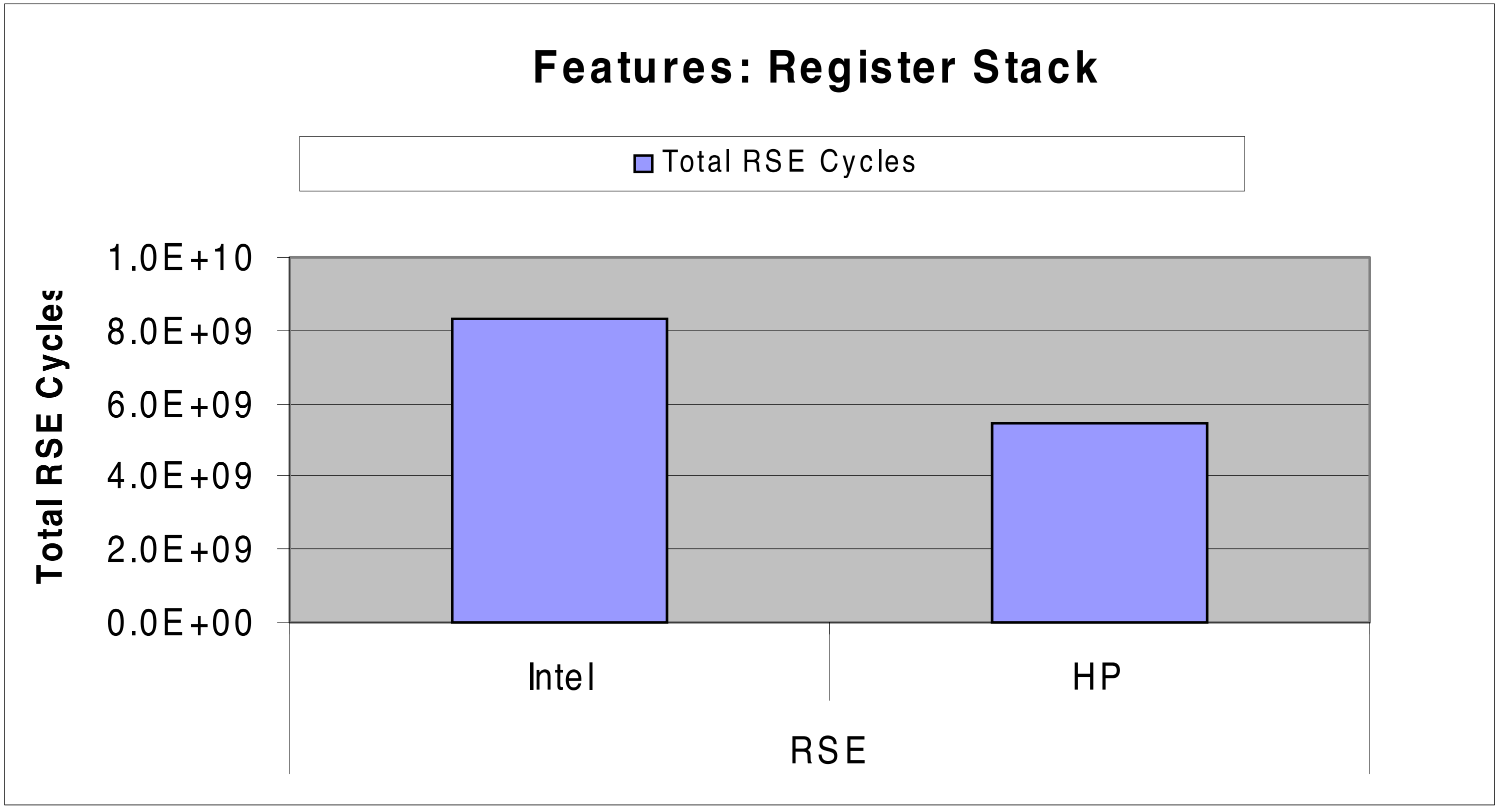
➢ **More than one way to get good performance from a compiler**



- Alpha has 1.4x mem refs of Itanium® architecture (incl/RSE ops)
- RSE ops are easy to optimize in the future, if needed
- HP compiler is more consv with regs, but has more ALU/reloads

➢ **Large register file, good compiler technology, and RSE pay off**

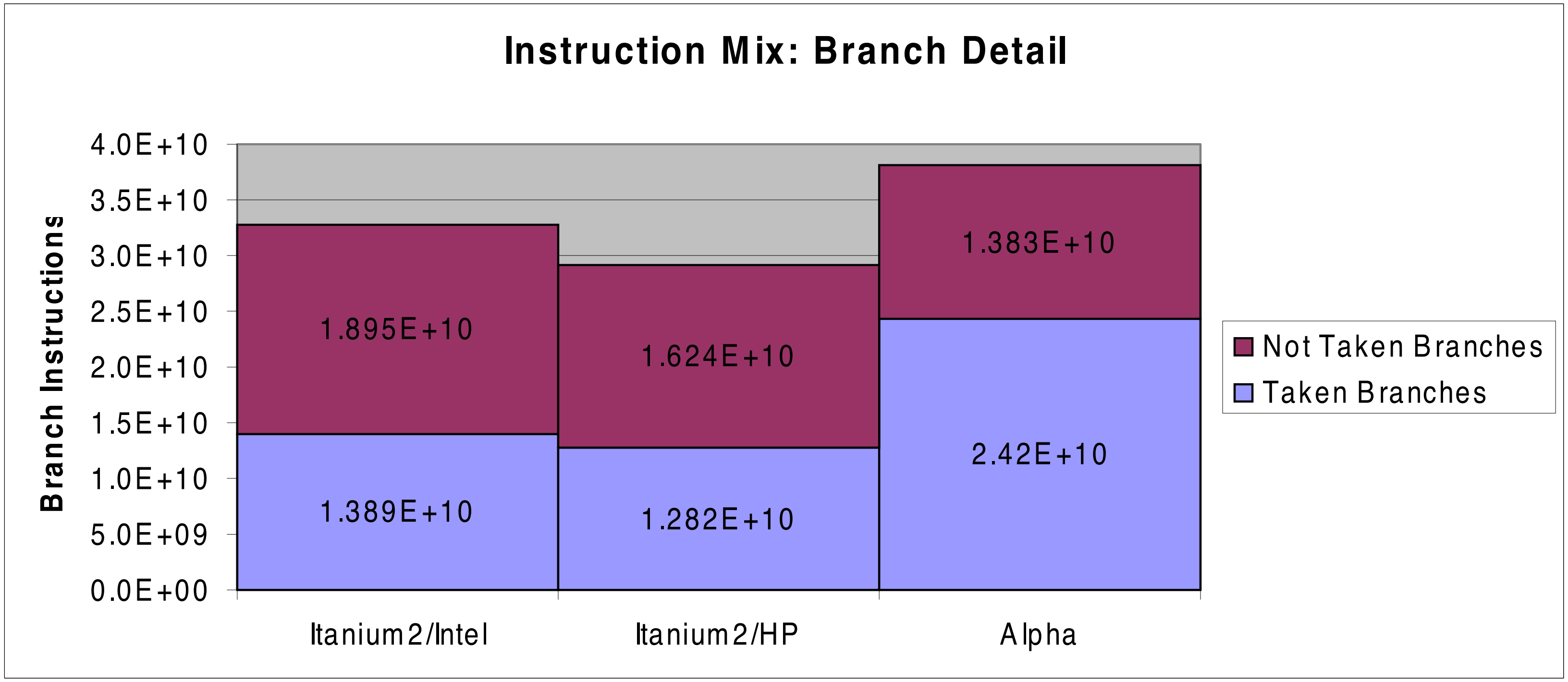
➢ **HP has lower RSE costs, but more reloads/alu ops**



- Total time spent in RSE is only 3%-4% of overall execution
- 1½ -3 cycles per call/return for RSE spill/fill activity
- 1-3 instructions per subroutine setup for RSE

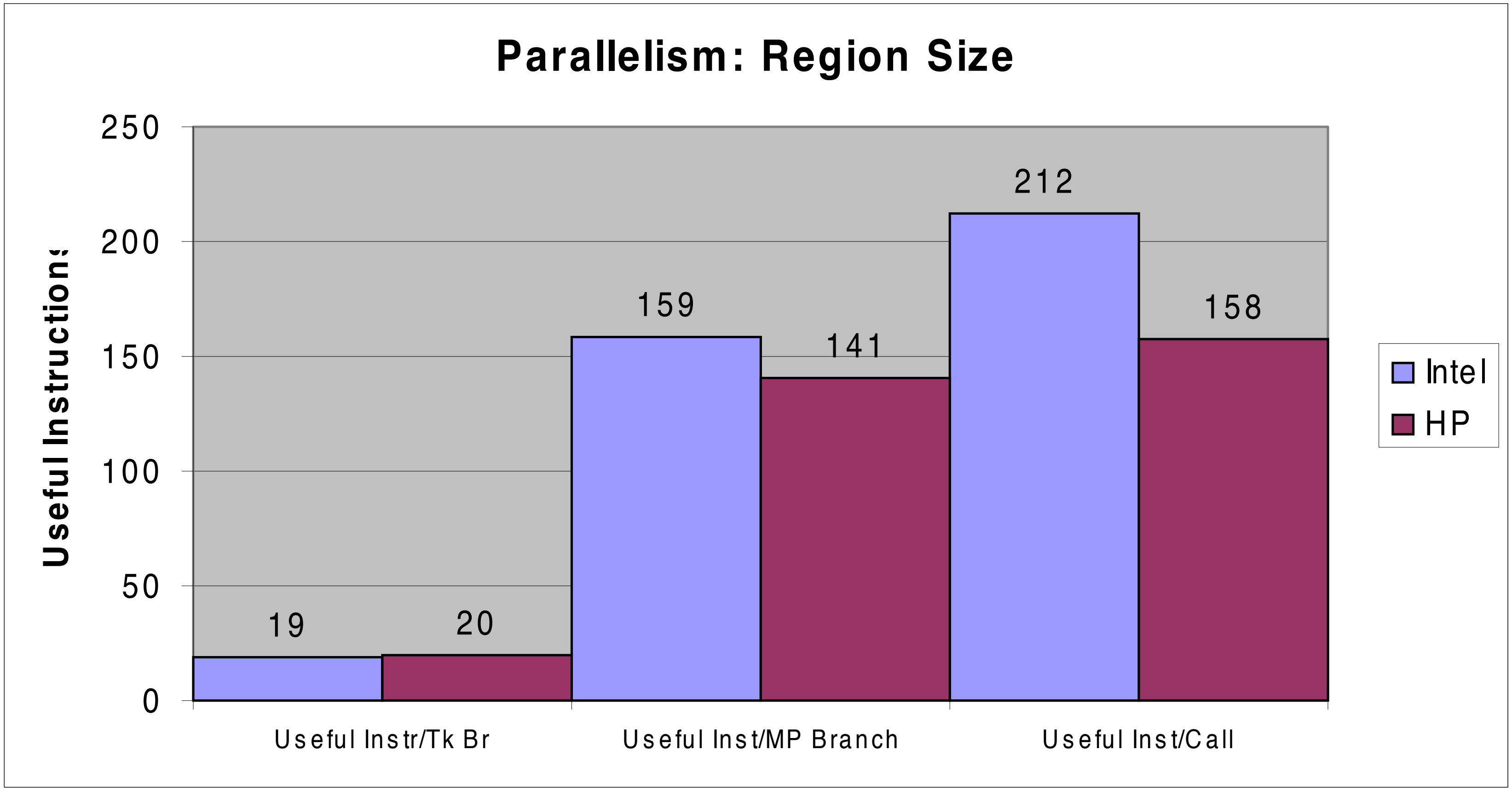
- Intel compiler has fewer calls, but more cycles/call

➢ **Register stack provides very low overhead call/return support**



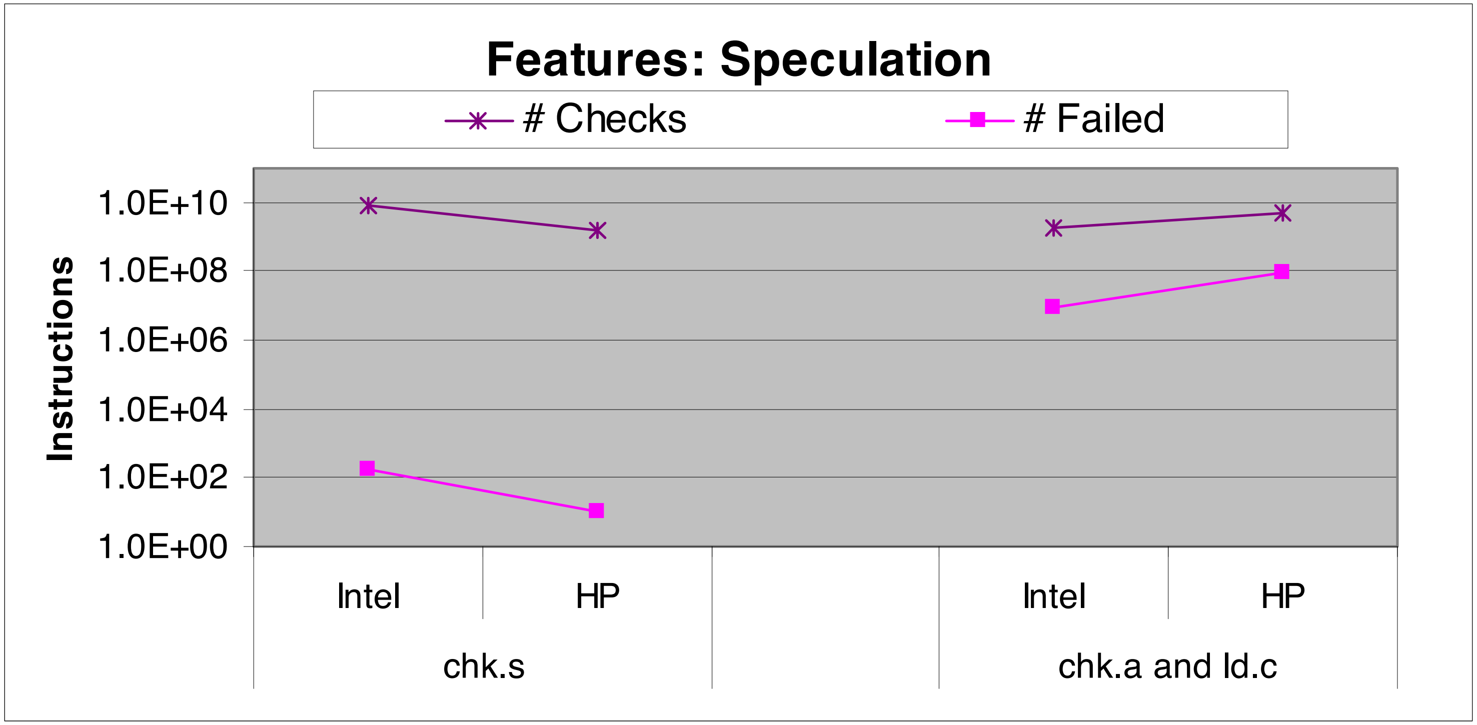
- HP compiler generates 13% fewer br’s than Intel, 9% more mispredictions
- HP compiler generates 31% fewer branches than Alpha
- Itanium-based binaries fewer tk br’s than Alpha, data skewed by lack of PGO

➢ **Itanium® architecture reduces the # branches and branch mispredicts**
➢ **HP/Intel compilers both reduce # branches – but with different focus**



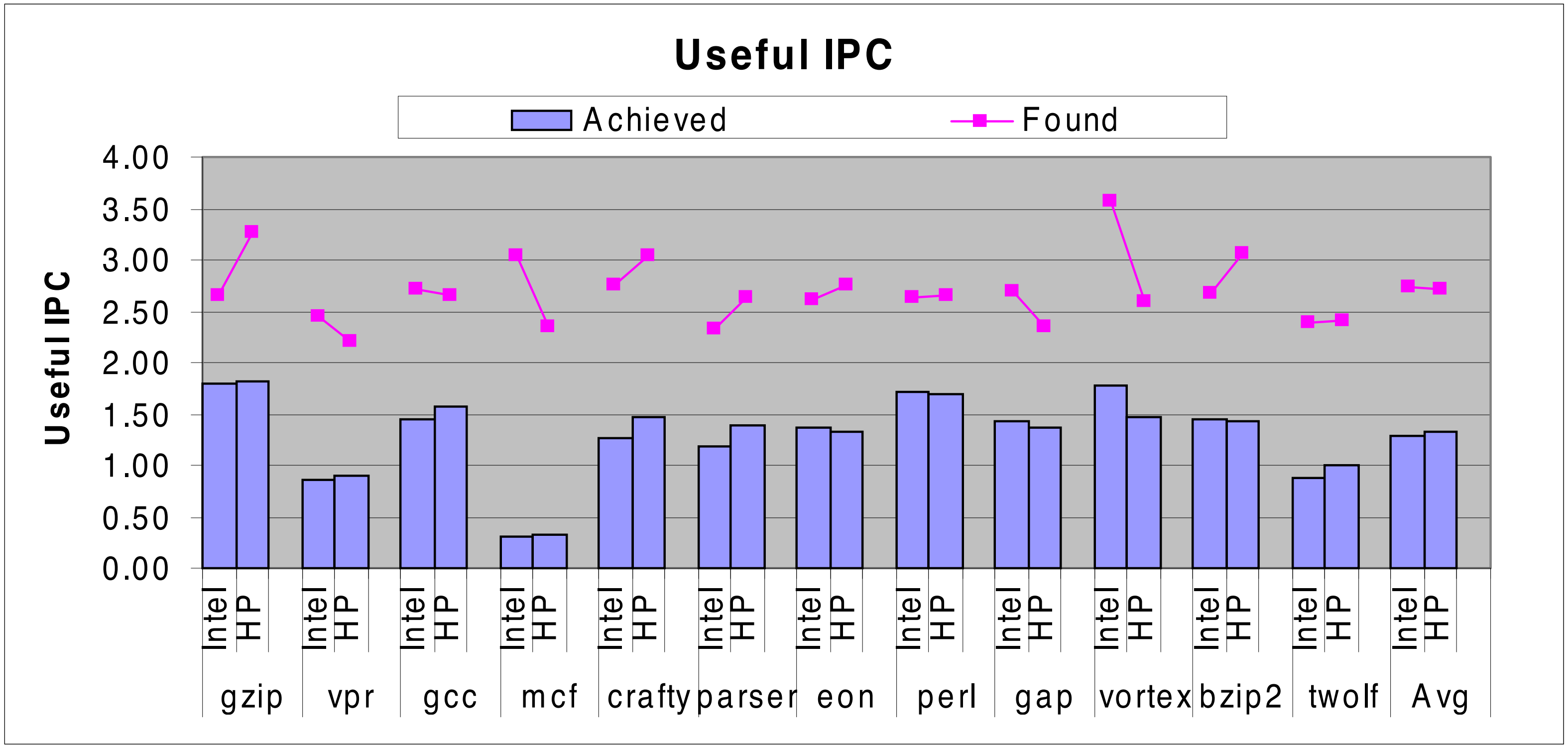
- Useful instructions per taken branch is very high
- Useful instructions per mispredicted branch slightly better for Intel
- Useful instructions/call very high – Intel/HP compilers very aggr inlining

➢ **Advanced compilers reduce stress on br prediction/Icache HW**
➢ **Trading Istream size for regularity improves HW efficiency**



- About 20-30% of loads are speculative in Intel binaries
- Data shows tiny penalty for chk.s usage despite high usage rate
- Intel has 10x more chk.s than HP, HP uses ‘no recovery model’ selectively (per benchmark decision)
- HP has 10x more chk.a/ld.c then Intel, recovery less than 1% time

➢ **Speculation heavily used, but causes little overhead**



- Useful IPC computed using ‘unstalled IPC’
- Compilers find 2.5-3.0 IPC in integer apps (even beyond SPEC)
- Dynamic delays reduce this to 1.3 achieved for CPU2000 integer

➢ **Differences in perf/heuristics shows headroom for both compilers**

➢ **Good IPC found by both compilers, room for uArch improvements**

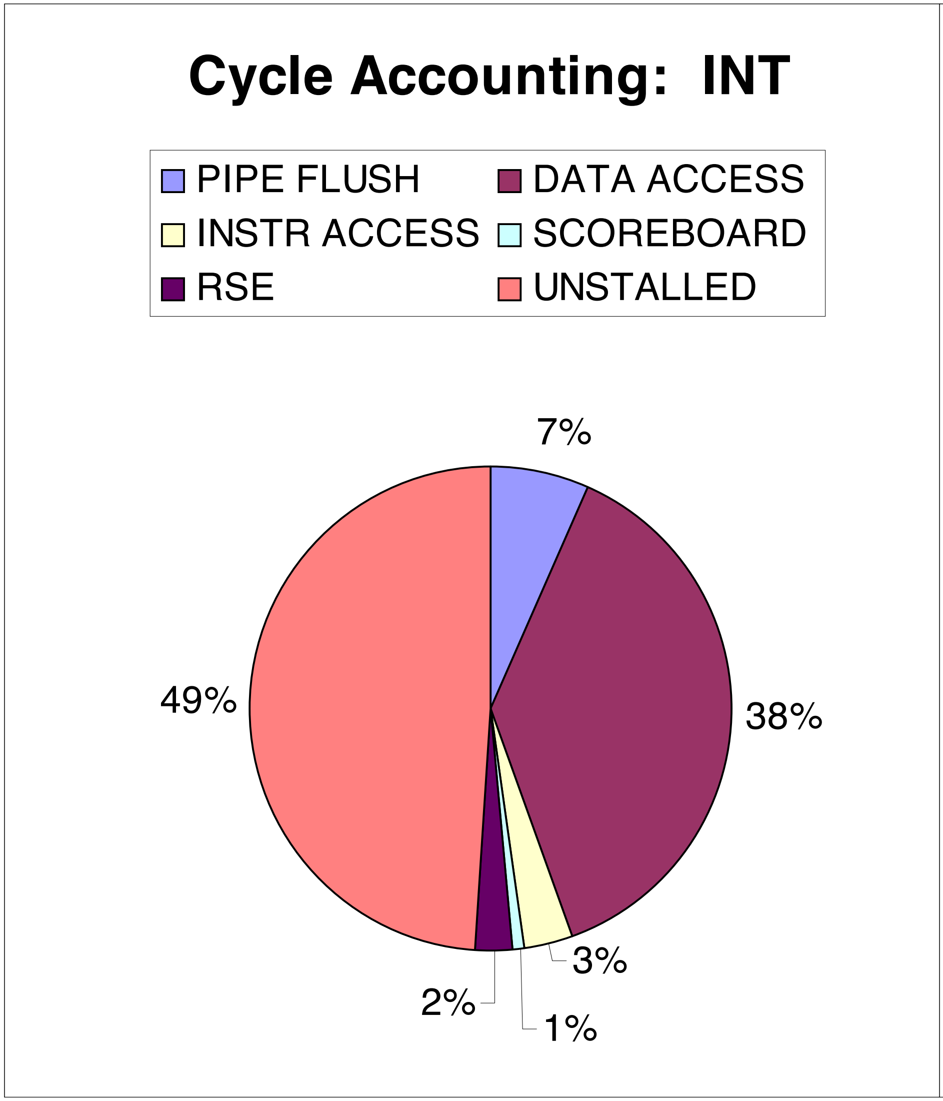
Notes

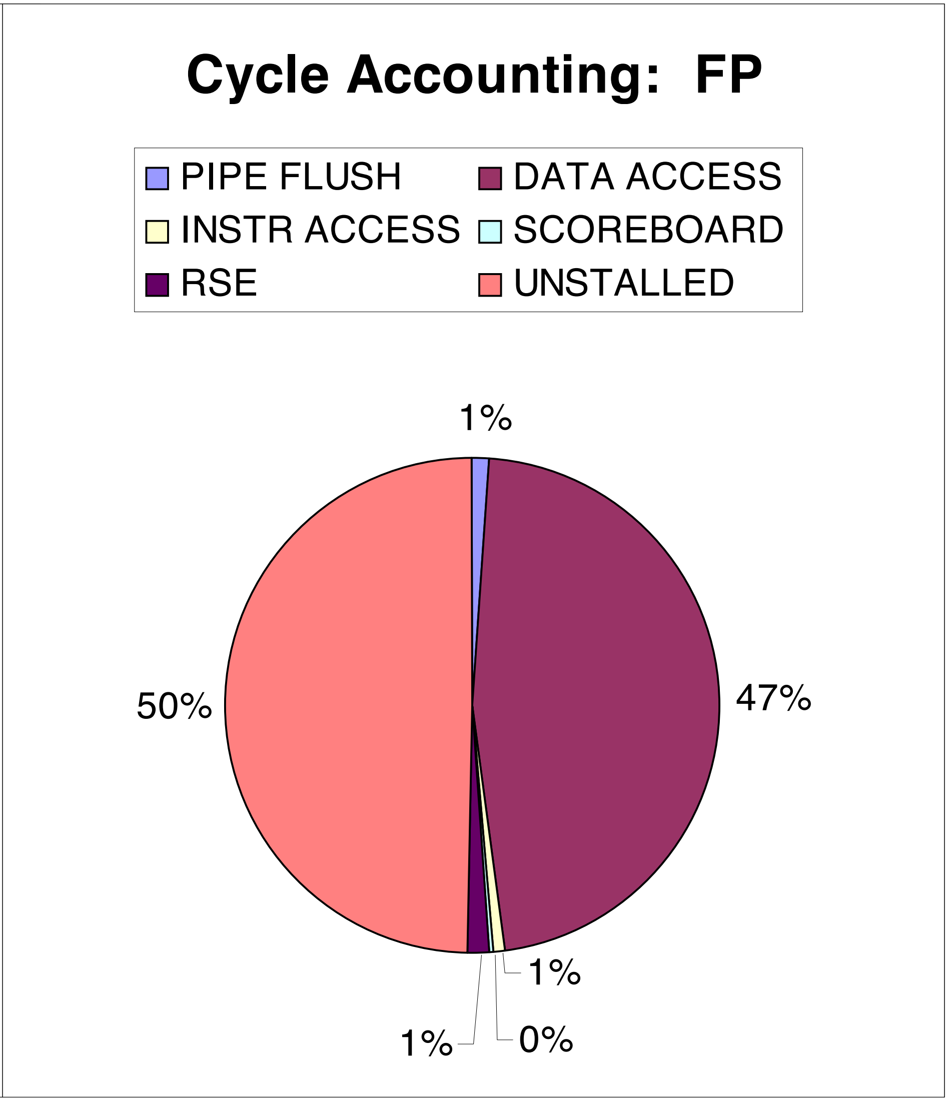
- Itanium-based binaries used for these stats are older than those used for the official SPEC submission (less than 10% difference)
- The results for Intel® Itanium® 2 processor in Part I are: one with the Intel compiler running on 64-bit MS OS and another with the HP compiler running under HP-UX
- Alpha ISA numbers via simulation, binaries used near peak (no profile guided optimization), tuned for 21264. Alpha data missing VPR and PERL – thus, left out of all averages in Part I.
- Results computed with arithmetic averages – data thus skewed towards long-running benchmarks

Part II: Microarchitecture

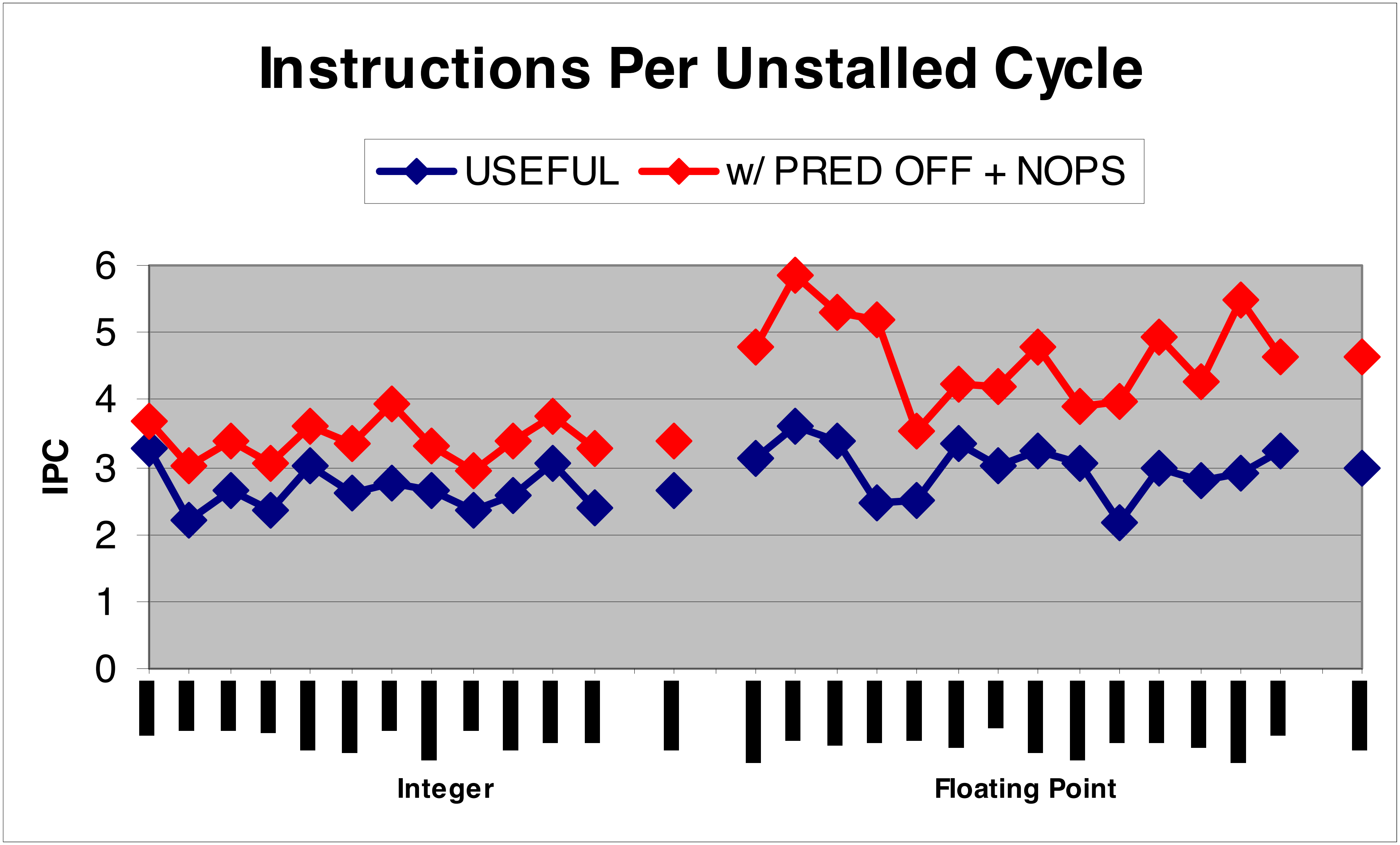
James McCormick

HP Colorado VLSI Laboratory

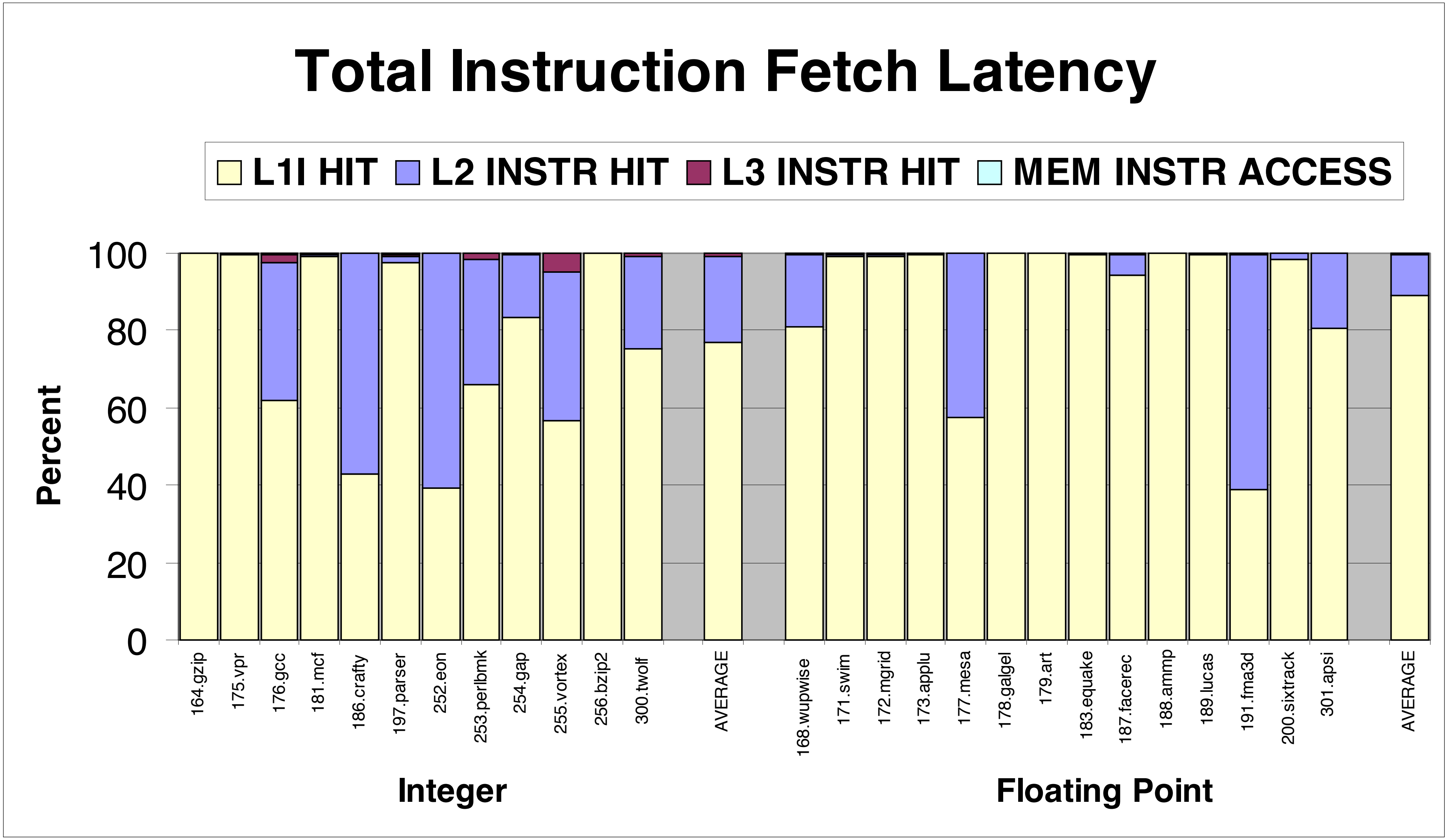




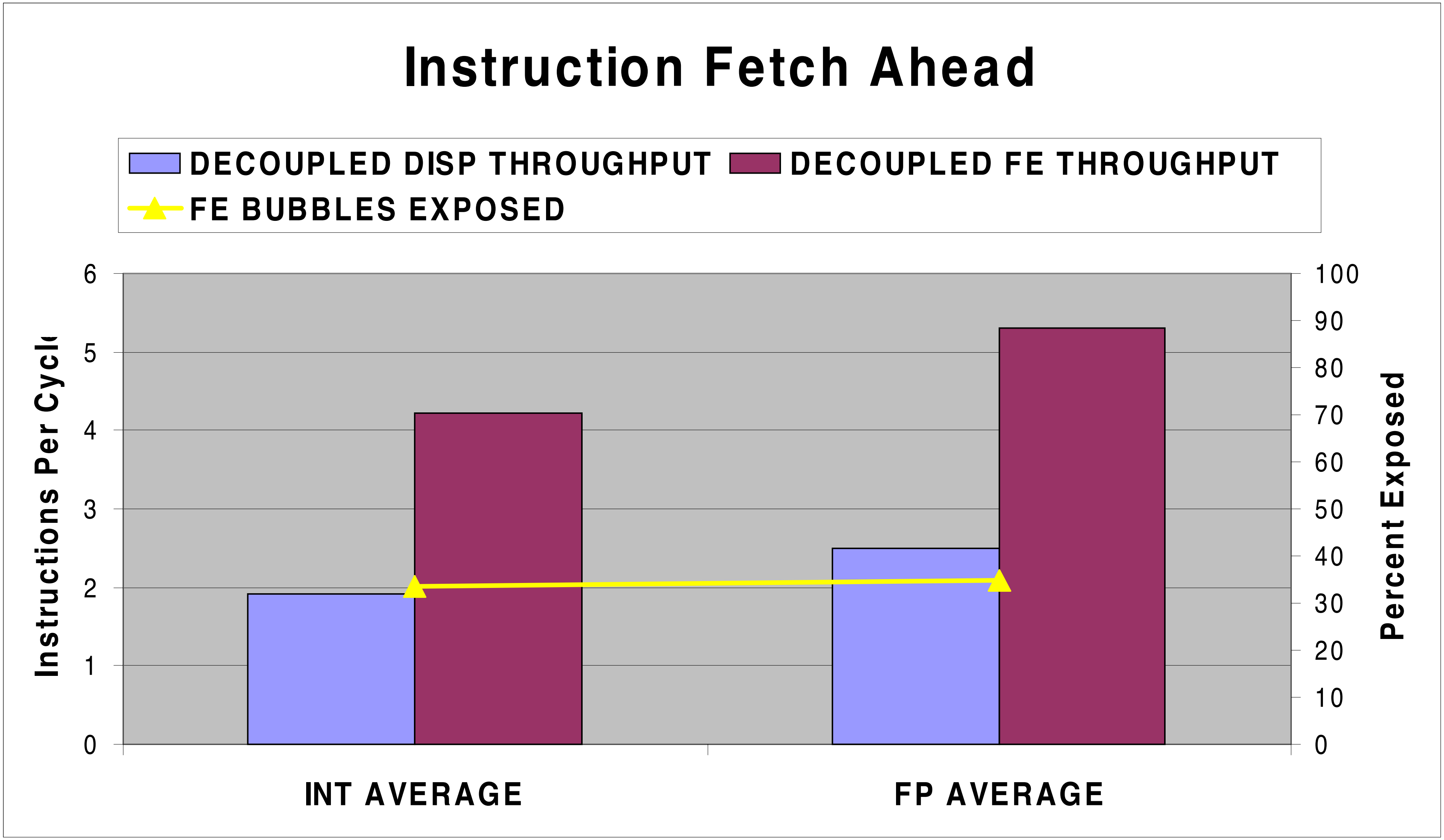
- Remaining: *unstalled execution* and *data access*
- Great performance



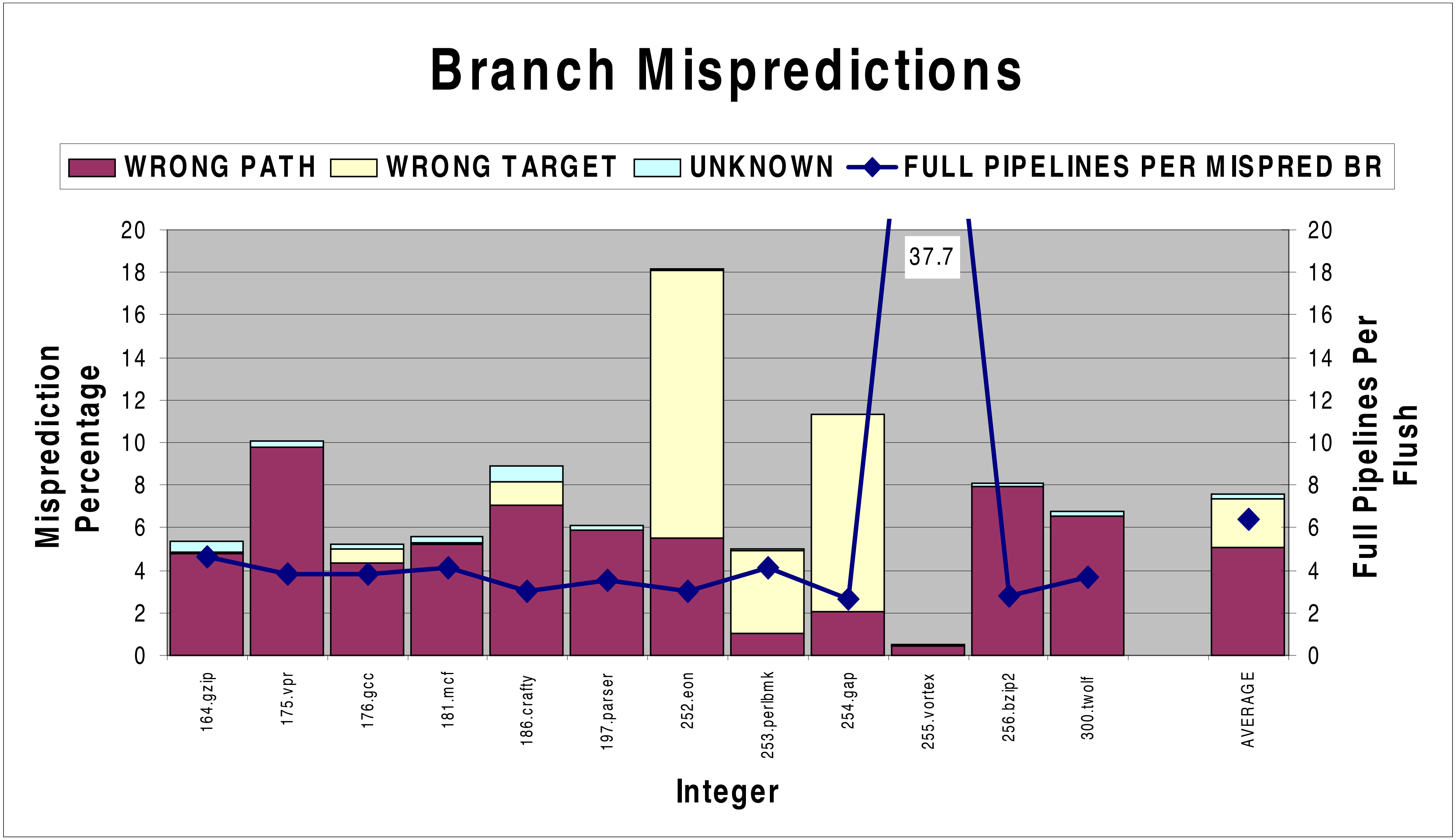
- High machine parallelism



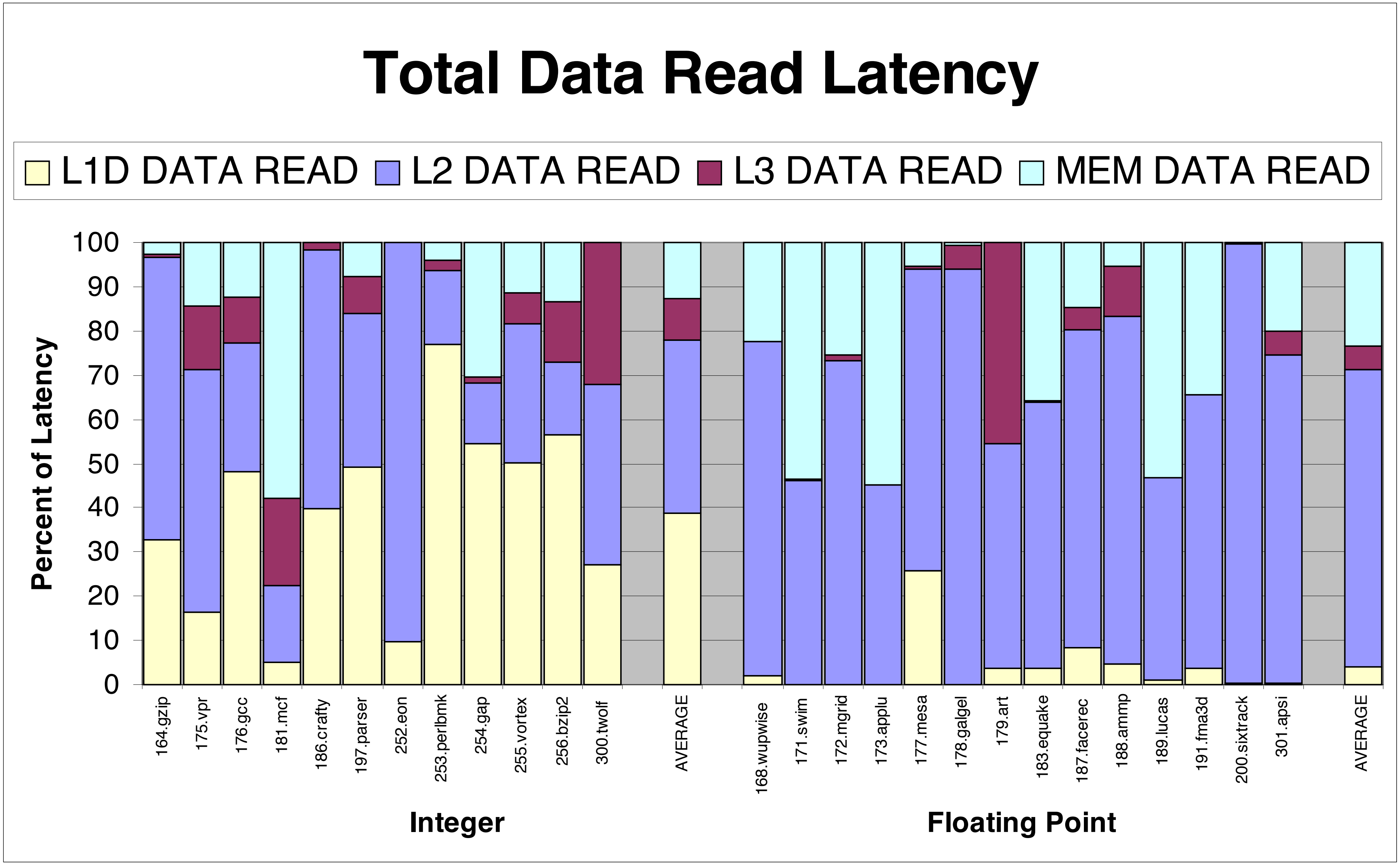
- Noticeable L1I misses
- Very small I-fetch component



- ➢ High FE throughput rate
- ➢ I-miss latency hidden



- ➢ High accuracy, low penalty
- ➢ Helps instruction fetch



- Large component, large opportunity

Summary

Itanium® 2 Processor Delivers Leadership Performance

- Architecture / Compilers
  - Expose lots of ILP to in-order pipeline
  - Replace difficult instructions with easy ones
  - RSE and large register file work well together
- CPU Design
  - Machine parallelism handles high ILP
  - Aggressive design reduces bottlenecks

Acknowledgements

Jason Cantin (U. Wisconsin):

Ran all the experiments to gather the Alpha ISA data with changes/alterations at our request. Without Jason, we would have no Alpha data of any kind. Thanks to his efforts to rerun all the data for us!

Hwansoo Han, Youngsoo Choi, Geetha Vederaman (Intel):

Intel data gathering, formatting, methodology, performance monitoring

Bryan Black, Ed Grochowski, Jim Callister, Carole Dulong (Intel):

Extensive draft review and comments

Caliper Team (HP):

Tool support

Backup/Reference Slides

Configuration Information

Intel® Itanium® 2 processor data for Intel systems/compilers:

- Binaries: Pre-production version of Intel C++ 6.0 Compiler, -O3 with interprocedural optimization and profile guided optimization
- Run on: Pre-production stepping of Itanium 2 processor 800Mhz/200Mhz core/bus, Intel 870 chipset, monitor kernel and user instructions and events

Intel® Itanium® 2 processor data for HP systems/compilers:

- Binaries: Pre-production version of HP compiler
- Run on: Pre-production stepping of Itanium 2 processor 1000Mhz/200Mhz core/bus, rx2600 prototype, monitor kernel and user instructions and events

Alpha ISA data:

- Run on: Functional simulator, system code not simulated
- Simulation details: http://www.cs.wisc.edu/multifacet/misc/spec2000cache-data/
- All data thanks to Jason Cantin at the University of Wisconsin.
- Binaries: **http://www.eecs.umich.edu/~chriswea/benchmarks/Com.cf**

  In general, these binaries are optimized for 21264, peak optimization. Usually, –g3 –fast –O4, but NO profile feedback. Compiler: DEC C V5.9-005 and DIGITAL C++ V6.1-027

Other remarks:

- All averages in slides left out PERL and VPR due to data not available for Alpha