EECS 252 Graduate Computer Architecture

Lec 10 –Simultaneous Multithreading

David Patterson
Electrical Engineering and Computer Sciences
University of California, Berkeley

http://www.eecs.berkeley.edu/~pattrsn
http://vlsi.cs.berkeley.edu/cs252-s06

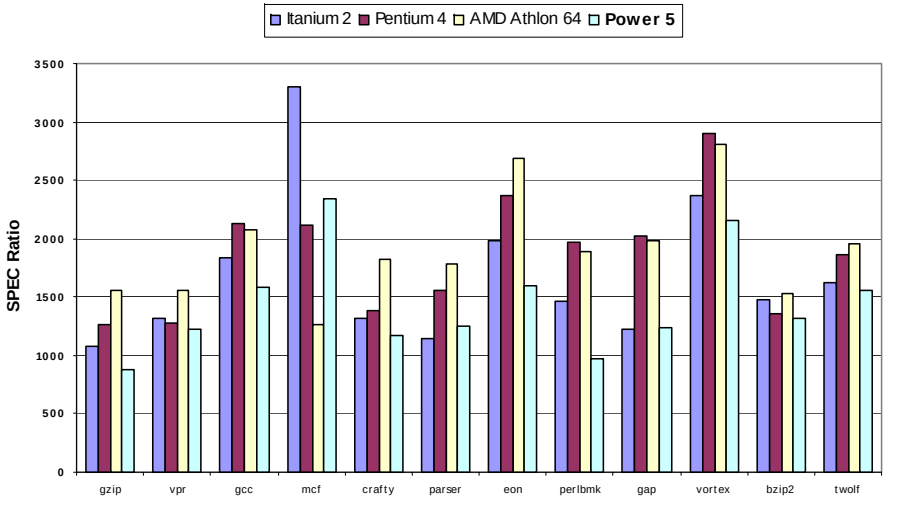
Review from Last Time

- Limits to ILP (power efficiency, compilers, dependencies …) seem to limit to 3 to 6 issue for practical options
- Explicitly parallel (Data level parallelism or Thread level parallelism) is next step to performance
- Coarse grain vs. Fine grained multihreading
  - Only on big stall vs. every clock cycle
- Simultaneous Multithreading if fine grained multithreading based on OOO superscalar microarchitecture
  - Instead of replicating registers, reuse rename registers
- Balance of ILP and TLP decided in marketplace

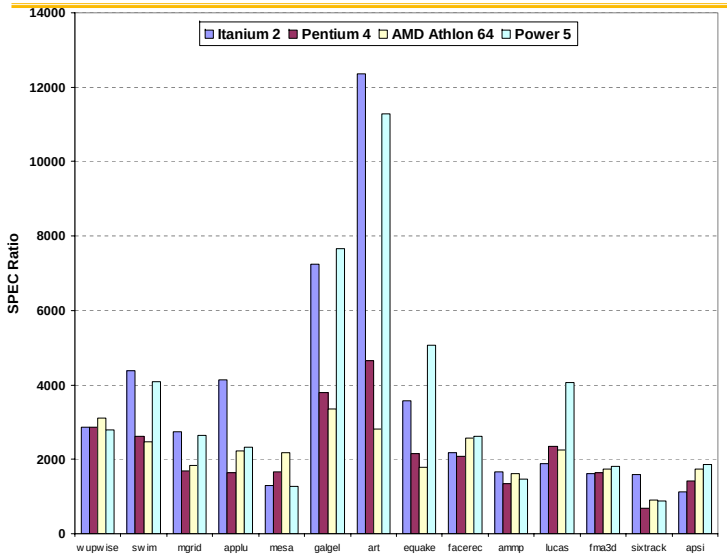
Head to Head ILP competition

Processor	Micro architecture	Fetch / Issue / Execute	Functional Units	Clock Rate (GHz)	Transistors, Die size	Power
Intel Pentium 4 Extreme	Speculative dynamically scheduled; deeply pipelined; SMT	3/3/4	7 int. 1 FP	3.8	125 M, 122 mm^2	115 W
AMD Athlon 64 FX-57	Speculative dynamically scheduled	3/3/4	6 int. 3 FP	2.8	114 M, 115 mm^2	104 W
IBM Power5 (1 CPU only)	Speculative dynamically scheduled; SMT; 2 CPU cores/chip	8/4/8	6 int. 2 FP	1.9	200 M, 300 mm^2 (est.)	80W (est.)
Intel Itanium 2	Statically scheduled VLIW-style	6/5/11	9 int. 2 FP	1.6	592 M, 423 mm^2	130 W

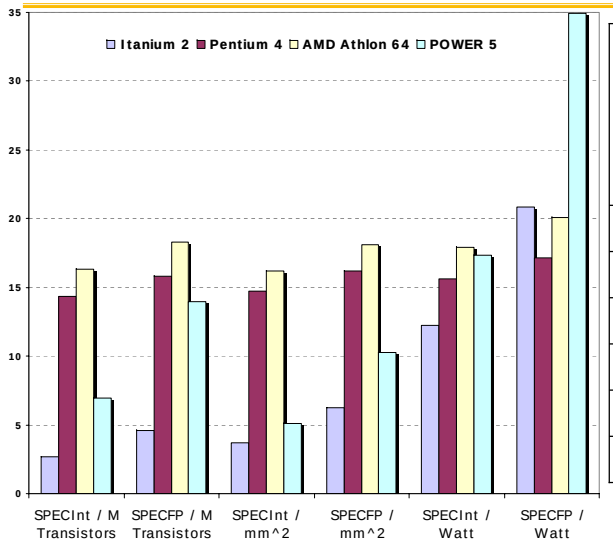
Performance on SPECint2000



Performance on SPECfp2000



Normalized Performance: Efficiency



Rank	Itanium 2	Pentium 4	Athlon	Power 5
Int/Trans	4	2	1	3
FP/Trans	4	2	1	3
Int/area	4	2	1	3
FP/area	4	2	1	3
Int/Watt	4	3	1	2
FP/Watt	2	4	3	1

No Silver Bullet for ILP

- No obvious over all leader in performance
- The AMD Athlon leads on SPECInt performance followed by the Pentium 4, Itanium 2, and Power5
- Itanium 2 and Power5, which perform similarly on SPECFP, clearly dominate the Athlon and Pentium 4 on SPECFP
- Itanium 2 is the most inefficient processor both for Fl. Pt. and integer code for all but one efficiency measure (SPECFP/Watt)
- Athlon and Pentium 4 both make good use of transistors and area in terms of efficiency,
- IBM Power5 is the most effective user of energy on SPECFP and essentially tied on SPECINT

Limits to ILP

- Doubling issue rates above today's 3-6 instructions per clock, say to 6 to 12 instructions, probably requires a processor to
  - Issue 3 or 4 data memory accesses per cycle,
  - Resolve 2 or 3 branches per cycle,
  - Rename and access more than 20 registers per cycle, and
  - Fetch 12 to 24 instructions per cycle.
- Complexities of implementing these capabilities likely means sacrifices in maximum clock rate
  - E.g, widest issue processor is the Itanium 2, but it also has the slowest clock rate, despite the fact that it consumes the most power!

Limits to ILP

- Most techniques for increasing performance increase power consumption
- The key question is whether a technique is *energy efficient*: does it increase power consumption faster than it increases performance?
- Multiple issue processors techniques all are energy inefficient:
  1. Issuing multiple instructions incurs some overhead in logic that grows faster than the issue rate grows
  2. Growing gap between peak issue rates and sustained performance
- Number of transistors switching = f(peak issue rate), and performance = f(sustained rate), growing gap between peak and sustained performance ⇒ increasing energy per unit of performance

Commentary

- Itanium architecture does not represent a significant breakthrough in scaling ILP or in avoiding the problems of complexity and power consumption
- Instead of pursuing more ILP, architects are increasingly focusing on TLP implemented with single-chip multiprocessors
- In 2000, IBM announced the 1st commercial single-chip, general-purpose multiprocessor, the Power4, which contains 2 Power3 processors and an integrated L2 cache
  - Since then, Sun Microsystems, AMD, and Intel have switch to a focus on single-chip multiprocessors rather than more aggressive uniprocessors.
- Right balance of ILP and TLP is unclear today
  - Perhaps right choice for server market, which can exploit more TLP, may differ from desktop, where single-thread performance may continue to be a primary requirement

And in conclusion …

- Limits to ILP (power efficiency, compilers, dependencies …) seem to limit to 3 to 6 issue for practical options
- Explicitly parallel (Data level parallelism or Thread level parallelism) is next step to performance
- Coarse grain vs. Fine grained multihreading
  - Only on big stall vs. every clock cycle
- Simultaneous Multithreading if fine grained multithreading based on OOO superscalar microarchitecture
  - Instead of replicating registers, reuse rename registers
- Itanium/EPIC/VLIW is not a breakthrough in ILP
- Balance of ILP and TLP unclear in marketplace

CS 252 Administrivia

- Next Reading Assignment: Vector Appendix
- Next Monday guest lecturer: Krste Asanovíc (MIT)
  - Designer of 1st vector microprocessor
  - Author of vector appendix for CA:AQA
  - Ph.D. from Berkeley in 1998, took CS 252 in 1991
  - Tenured Associate Professor at MIT
  - On sabbatical at UCB this academic year
- Next paper: “The CRAY-1 computer system”
  - by R.M. Russell, *Comm. of the ACM*, January 1978
  - Send comments on paper to TA by Monday 10PM
  - Post on wiki and read on Tuesday, 30 minutes on Wednesday
- Be sure to comment on vector vs. scalar speed, min. size vector faster than scalar loop, relative speed to other computers, clock rate, size of register state, memory size, no. functional units, and general impressions compared to today’s CPUs

Today's Discussion

- "Simultaneous Multithreading: A Platform for Next-generation Processors," Susan J. Eggers et al, *IEEE Micro*, 1997
- What were worse options than SMT for 1B transistors?
- What is the main extra hardware resource that SMT requires?
- What is "Vertical" and "Horizontal" waste?
- How does SMT differ from Multithreading?
- What unit is the bottleneck for SMT

Today's Discussion (con't)

- "Simultaneous Multithreading: A Platform for Next-generation Processors," Susan J. Eggers et al, *IEEE Micro*, 1997
- How many instructions fetched per clock cycle? From how many threads?
- How did it do priority?
- What assumption made about computer organization before add SMT?
  - When did they think it would ship?
  - How compare to slide 3?
  - What was memory hierarchy?

Today's Discussion (con't)

- "Simultaneous Multithreading: A Platform for Next-generation Processors," Susan J. Eggers et al, *IEEE Micro*, 1997
- What compare performance to?
- For what workloads?
- What performance advantages claimed?
  - What was performance metric?
- How compare to Wall's ILP limit claims?

Time travel …

- End of CS 252 in 2001 I told students to try to think about following architecture questions to think about in the future
- Which ones can we answer 5 years later?
- What do you think the answers are?

2001 252 Questions for Future [1/5]

- What did IA-64/EPIC do well besides floating point programs?
  - Was the only difference the 64-bit address v. 32-bit address?
  - What happened to the AMD 64-bit address 80x86 proposal?
- What happened on EPIC code size vs. x86?
- Did Intel Oregon increase x86 performance so as to make Intel Santa Clara EPIC performance similar?

2001 252 Questions for Future [2/5]

- Did Transmeta-like compiler-oriented translation survive vs. hardware translation into more efficient internal instruction set?
- Did ILP limits really restrict practical machines to 4-issue, 4-commit?
- Did we ever really get CPI below 1.0?
- Did value prediction become practical?
- Branch prediction: How accurate did it become?
  - For real programs, how much better than 2 bit table?
- Did Simultaneous Multithreading (SMT) exploit underutilized Dynamic Execution HW to get higher throughput at low extra cost?
  - For multiprogrammed workload (servers) or for parallelized single program?

2001 252 Questions for Future [3/5]

- Did VLIW become popular in embedded? What happened on code size?
- Did vector become popular for media applications, or simply evolve SIMD?
- Did DSP and general purpose microprocessors remain separate cultures, or did ISAs and cultures merge?
  - Compiler oriented?
  - Benchmark oriented?
  - Library oriented?
  - Saturation + 2's complement

2001 252 Questions for Future [4/5]

- Did emphasis switch from cost-performance to cost-performance-availability?
- What support for improving software reliability? Security?

2001 252 Questions for Future [5/5]

- 1985-2000: 1000X performance
  - Moore's Law transistors/chip => Moore's Law for Performance/MPU
- Hennessy: industry been following a roadmap of ideas known in 1985 to exploit Instruction Level Parallelism to get 1.55X/year
  - Caches, Pipelining, Superscalar, Branch Prediction, Out-of-order execution, …
- ILP limits: To make performance progress in future need to have explicit parallelism from programmer vs. implicit parallelism of ILP exploited by compiler, HW?
- Did Moore's Law in transistors stop predicting microprocessor performance? Did it drop to old rate of 1.3X per year?
  - Less because of processor-memory performance gap?