Microarchitecture Overview

Prof. Scott Rixner
Duncan Hall 3028
rixner@rice.edu

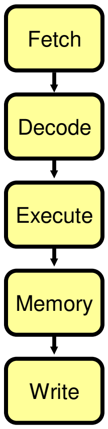
January 15, 2007

Performance

- **Make operations faster**
  - **Process improvements**
  - **Circuit improvements**
  - **Use more transistors to make a function faster**
- **Execute more operations in parallel**
  - **Pipelining**
  - **Superscalar execution**
  - **Multiple processors**

Pipelining

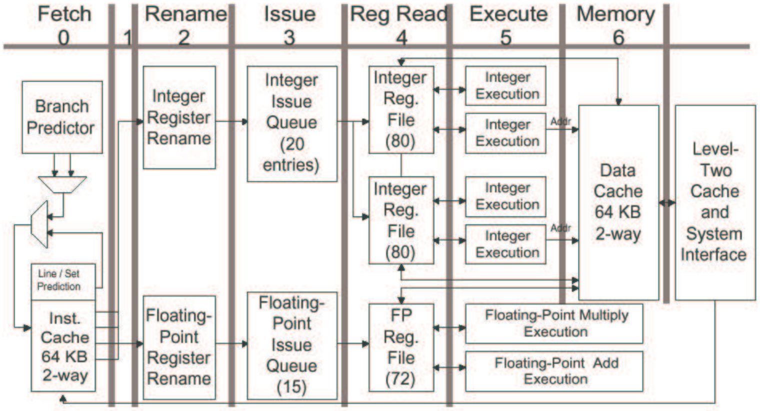
- Basic microprocessor pipeline
  - Instruction fetch (IF)
  - Instruction decode (ID)
  - Execute (EX)
  - Memory access (MEM)
  - Writeback (WB)



Superscalar Execution

- Branch prediction
- Instruction scheduling
  - Compiler reordering
  - In-order issue/completion
  - Out-of-order issue/completion
  - Register renaming

Alpha 21264 Pipeline



-IEEE Micro 3/99

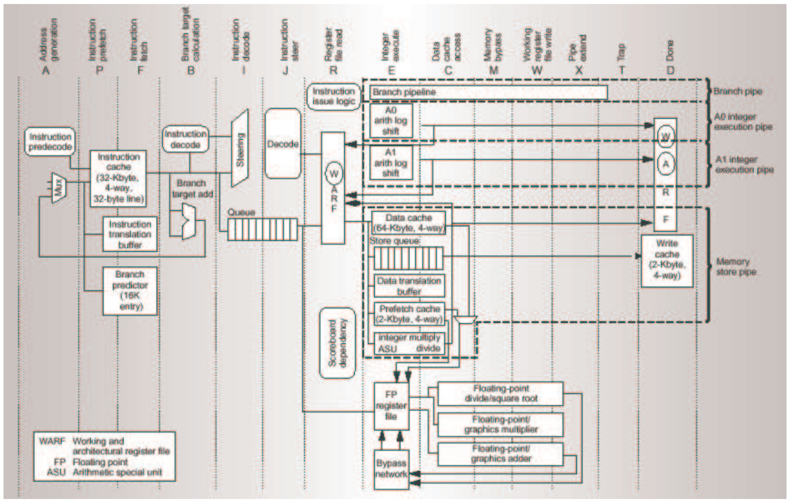
Pentium Pro Pipeline

- **IFU1**
  - **Fetch 32 bytes from L1 $**
- **IFU2**
  - **Find instructions (branches to BTB)**
- **IFU3**
  - **Align instructions**
- **DEC1**
  - **3 decoders generate μops**
- **DEC2**
  - **Move μops to dispatch queue**
- **RAT**
  - **Rename/allocate 40 registers**
- **ROB**
  - **Allocate reorder buffer (40)**
- **DIS**
  - **Dispatch μops to units**
- **EX**
  - **Execute**
- **RET1**
  - **Mark ROB for retirement**
- **RET2**
  - **Retire instructions**

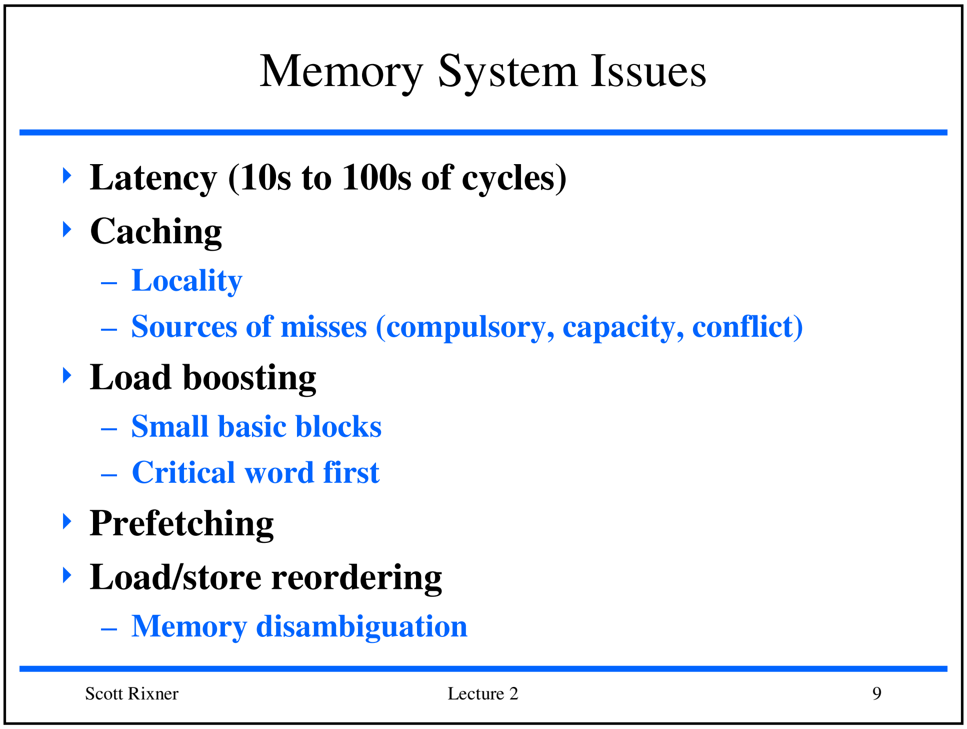
Pentium 4 Pipeline

- **Stages 1-2**
  - **Trace cache next instruction pointer**
- **Stages 3-4**
  - **Trace cache fetch**
- **Stage 5**
  - **Drive (wire delay!)**
- **Stages 6-8**
  - **Allocate and Rename**
- **Stages 10-12**
  - **Schedule instructions**
  - **Memory/fast ALU/slow ALU & general FP/simple FP**
- **Stages 13-14**
  - **Dispatch**
- **Stages 15-16**
  - **Register access**
- **Stage 17**
  - **Execute**
- **Stage 18**
  - **Set flags**
- **Stage 19**
  - **Check branches**
- **Stage 20**
  - **Drive (more wire delay!)**

UltraSPARC III Pipeline

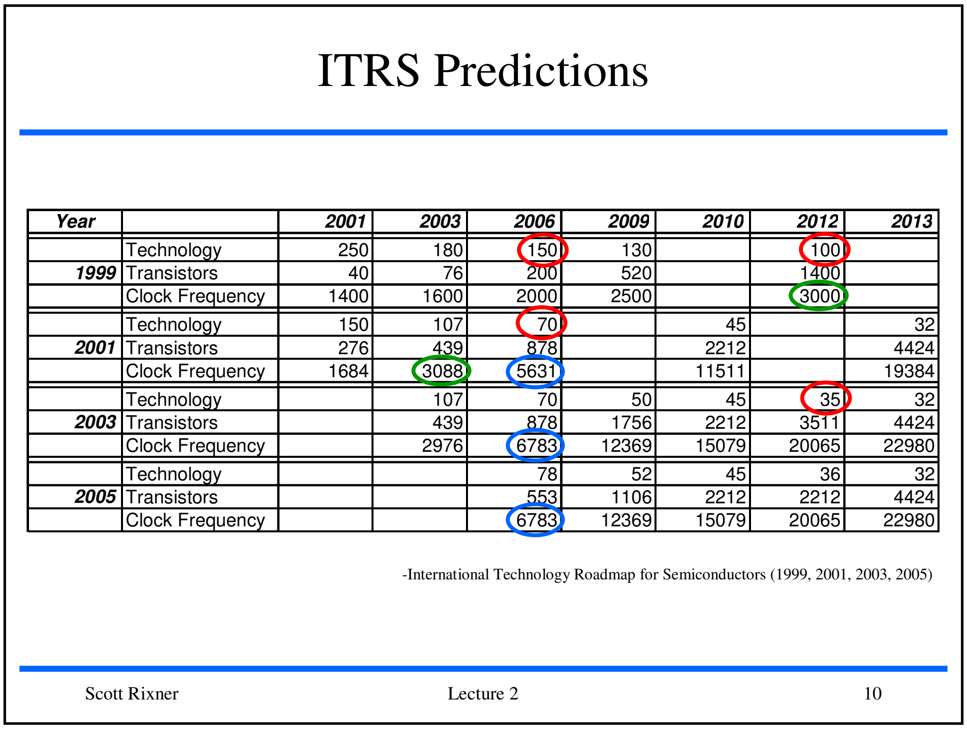

Address generation
A
Instruction prefetch
P
Instruction fetch
F
Branch target calculation
B
Instruction decode
I
Instruction steer
J
Register file read
R
Integer execute
E
Data cache access
C
Memory bypass
M
Working register file write
W
Pipe extend
X
Trap
T
Done
D
Instruction predecode
Mux
Instruction cache (32-Kbyte, 4-way, 32-byte line)
Instruction translation buffer
Branch predictor (16K entry)
Instruction decode
Branch target add
Steering
Queue
Decode
Instruction issue logic
W A R F
Scoreboard dependency
Branch pipeline
A0 arith log shift
A1 arith log shift
Data cache (64-Kbyte, 4-way)
Store queue
Data translation buffer
Prefetch cache (2-Kbyte, 4-way)
integer multiply ASU divide
FP register file
Bypass network
Floating-point divide/square root
Floating-point/ graphics multiplier
Floating-point/ graphics adder
W
A
R
F
Write cache (2-Kbyte, 4-way)
Branch pipe
A0 integer execution pipe
A1 integer execution pipe
Memory store pipe
WARF Working and architectural register file
FP Floating point
ASU Arithmetic special unit

-IEEE Micro 5/99



Memory System Issues

- Latency (10s to 100s of cycles)
- Caching
  - Locality
  - Sources of misses (compulsory, capacity, conflict)
- Load boosting
  - Small basic blocks
  - Critical word first
- Prefetching
- Load/store reordering
  - Memory disambiguation

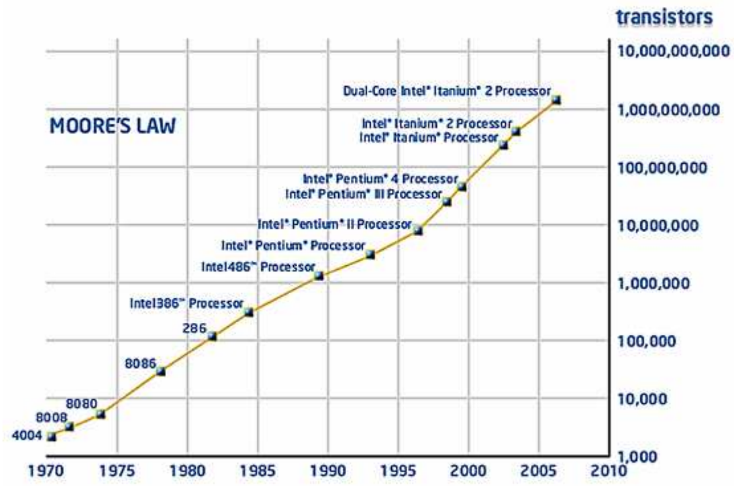


ITRS Predictions

Year		*2001*	*2003*	*2006*	*2009*	*2010*	*2012*	*2013*
	Technology	250	180	150	130		100	
1999	Transistors	40	76	200	520		1400	
	Clock Frequency	1400	1600	2000	2500		3000	
	Technology	150	107	70		45		32
2001	Transistors	276	439	878		2212		4424
	Clock Frequency	1684	3088	5631		11511		19384
	Technology		107	70	50	45	35	32
2003	Transistors		439	878	1756	2212	3511	4424
	Clock Frequency		2976	6783	12369	15079	20065	22980
	Technology			78	52	45	36	32
2005	Transistors			553	1106	2212	2212	4424
	Clock Frequency			6783	12369	15079	20065	22980

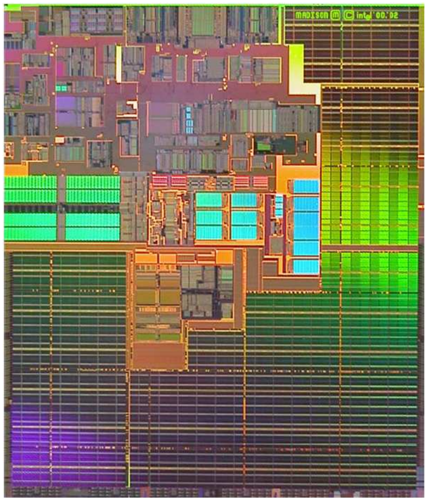
-International Technology Roadmap for Semiconductors (1999, 2001, 2003, 2005)

Transistor Counts



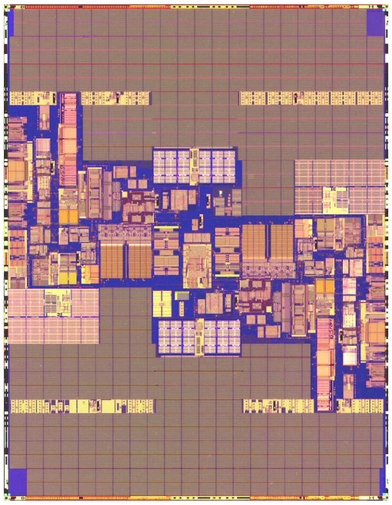
- Intel

2004: Intel Itanium 2



- Issues up to 8 ops per cycle
- 0.13 micron process
- 592 million transistors
- 432 mm^2 die
- 128-bit bus
- 16KB data cache
- 16KB instruction cache
- 9MB L3 cache (256KB L2)
- 1.6 GHz

2006: Intel Dual Core Itanium 2



- 2 Itanium 2 processors
- Each core
  - 2-way multithreading
  - Issues up to 8 ops per cycle
  - 16KB inst. & data L1 caches
  - 1MB inst. & 256KB data L2 caches
  - 12MB L3 cache
- Virtualization technology
- 0.09 micron process
- 1.72 billion transistors
  - Cores: 57M
  - L1/L2 caches: 106.5M
  - L3 caches: 1550M
  - Bus logic and I/O: 6.7M
- 596 mm^2 die
- 128-bit bus
- 1.6 GHz

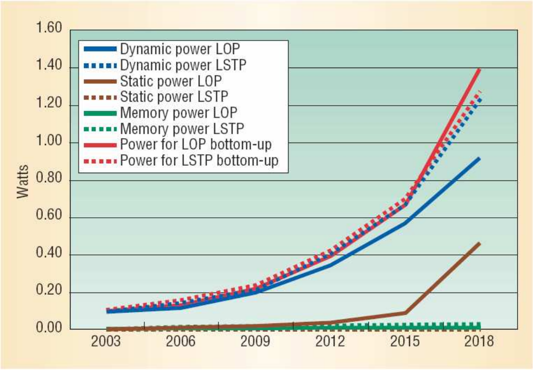
Future Challenges

- 2000
  - Interconnect
  - Design
- 2002
  - Design productivity
  - Power management
  - Multicore organization
  - I/O bandwidth
  - Circuit and process technology
- 2004
  - Drivers
    - Systems on chip
    - Mixed-signal chips
    - Embedded memory
  - Design
  - Test

Interconnect

- **Local**
  - **Within a module**
  - **Scales with technology**
- **Global**
  - **Connect major functional areas**
  - **Does not scale well with technology**
  - **On-chip interconnection networks**
- **Architectural impact?**

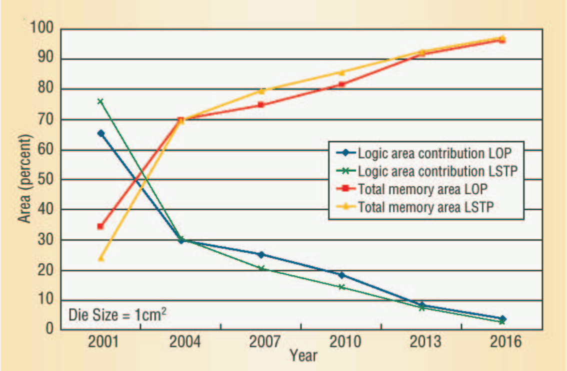
Power Trends



-Computer 1/04

- **“Low power” devices will consume too much power**

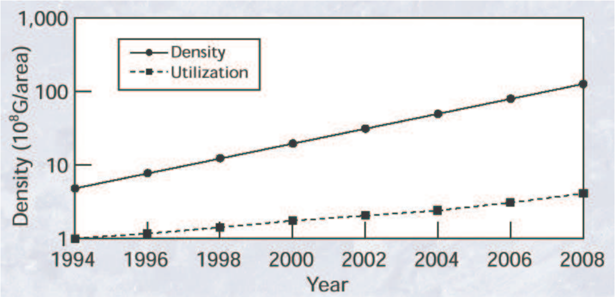
“Power Gap”



-Computer 1/02

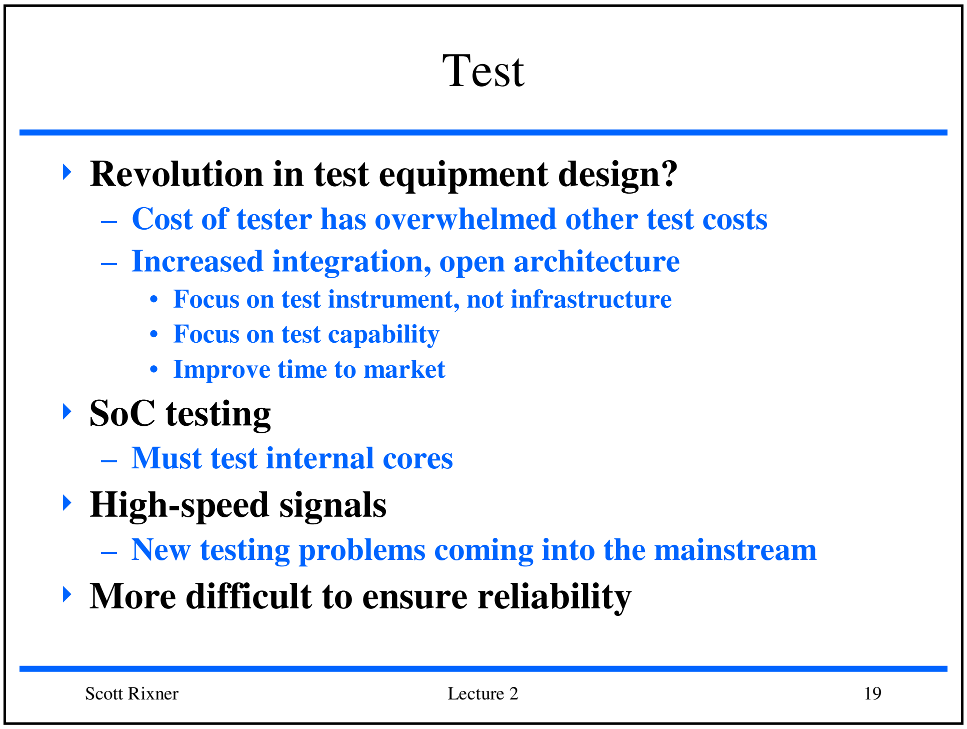
- Low-power devices will be dominated by memory
- Why?
- What is the architectural impact of this?

“Design Productivity Gap”



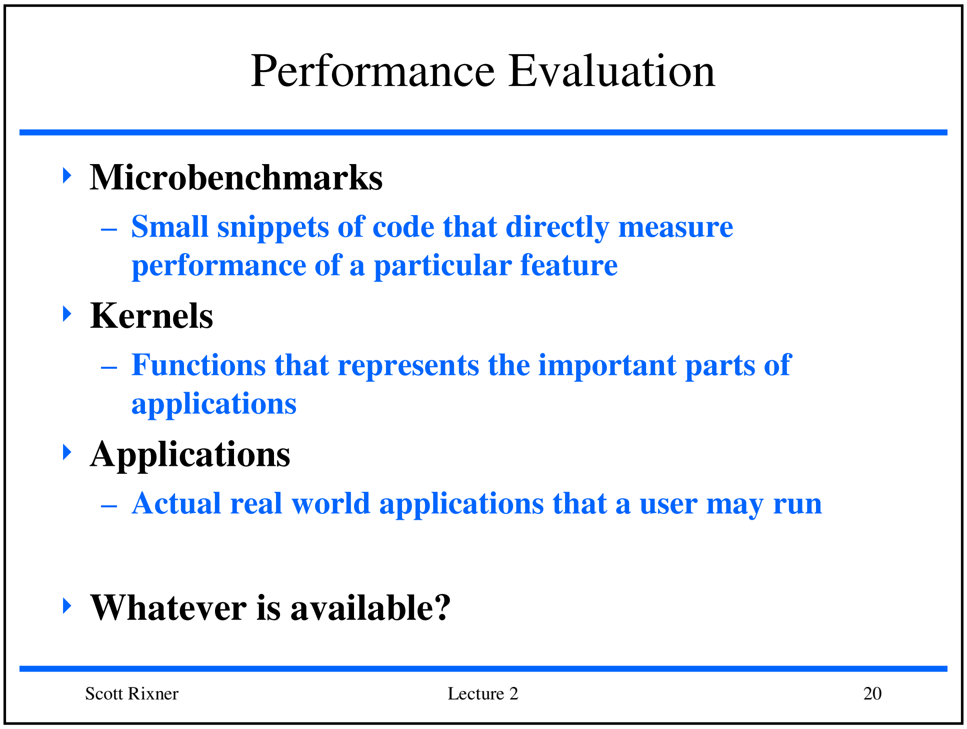
-Computer 1/99

- Communications-centric design
- Robustness and Scalability
- Cost optimization and cooptimization
- What about the architect?



Test

- Revolution in test equipment design?
  - Cost of tester has overwhelmed other test costs
  - Increased integration, open architecture
    - Focus on test instrument, not infrastructure
    - Focus on test capability
    - Improve time to market
- SoC testing
  - Must test internal cores
- High-speed signals
  - New testing problems coming into the mainstream
- More difficult to ensure reliability



Performance Evaluation

- Microbenchmarks
  - Small snippets of code that directly measure performance of a particular feature
- Kernels
  - Functions that represents the important parts of applications
- Applications
  - Actual real world applications that a user may run
- Whatever is available?

Performance Comparison

- **"X is *n* times faster than Y" means:**

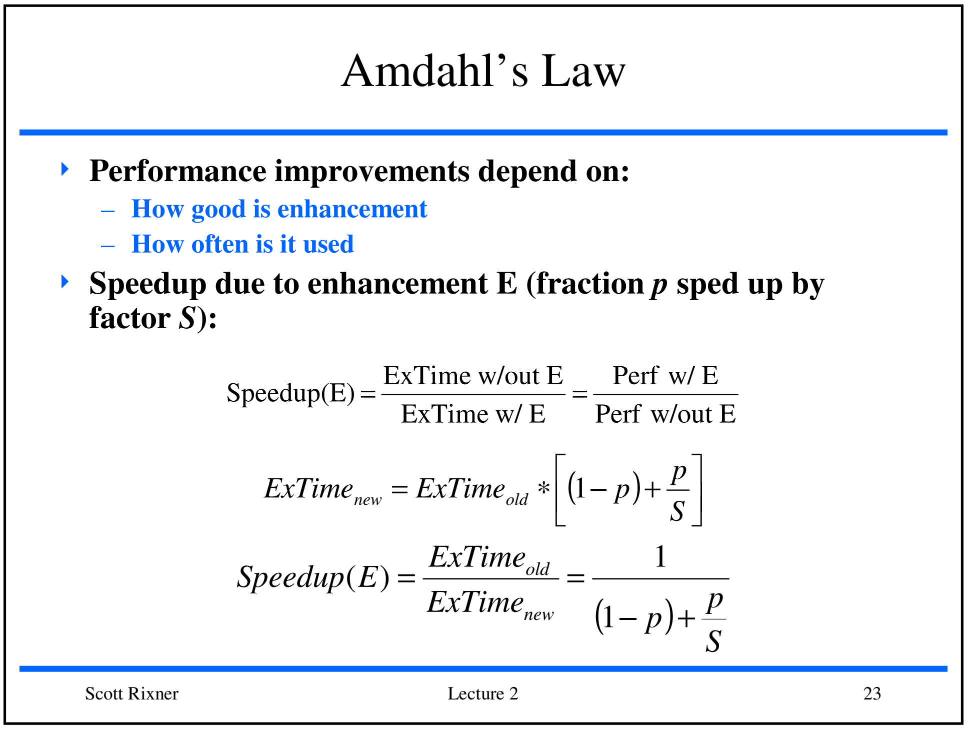
$$\frac{ExTime(Y)}{ExTime(X)} = \frac{Perf(X)}{Perf(Y)} = n$$

- **Same definition applies to other metrics, such as throughput**

Example

Plane	DC to Paris	Speed	Passengers	Throughput (PMPH)
Boeing 747	6.5 hours	610 mph	470	266,700
Concorde	3 hours	1350 mph	132	178,200

- **Performance metrics**
  - **Time to run the task (travel time for each passenger)**
  - **Throughput (person miles per hour = PMPH)**
- **Comparisons**
  - **Speed of Concorde > 747**
  - **Throughput of 747 > Concorde**



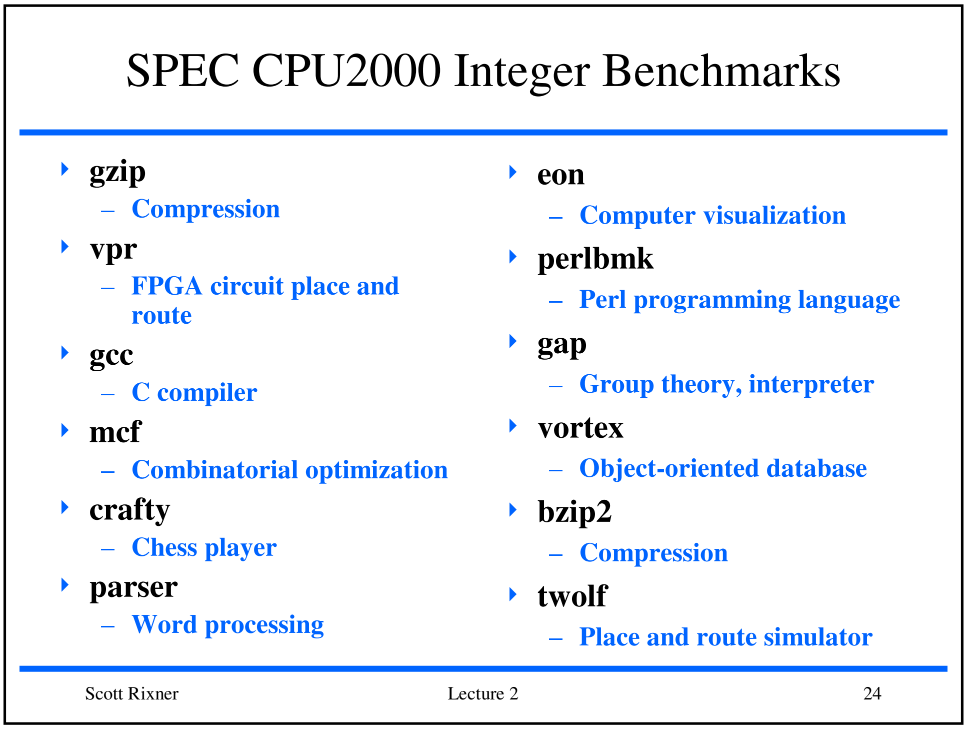
Amdahl’s Law

- **Performance improvements depend on:**
  - **How good is enhancement**
  - **How often is it used**
- **Speedup due to enhancement E (fraction *p* sped up by factor *S*):**

$$\text{Speedup(E)} = \frac{\text{ExTime w/out E}}{\text{ExTime w/ E}} = \frac{\text{Perf w/ E}}{\text{Perf w/out E}}$$

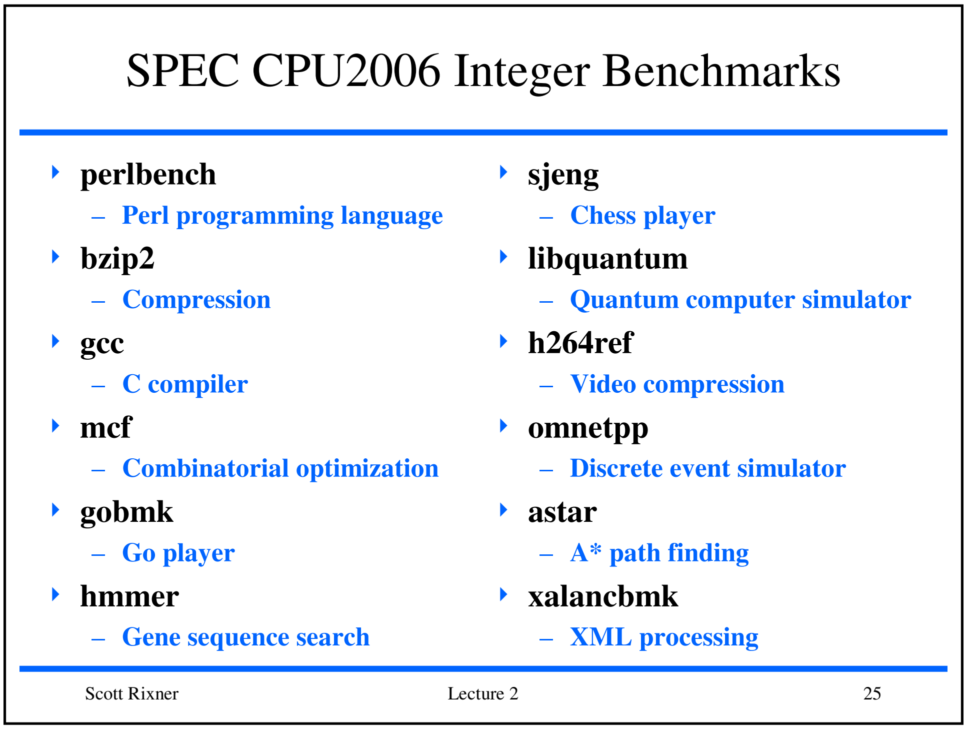
$$ExTime_{new} = ExTime_{old} * \left[(1-p)+\frac{p}{S}\right]$$

$$Speedup(E) = \frac{ExTime_{old}}{ExTime_{new}} = \frac{1}{(1-p)+\frac{p}{S}}$$



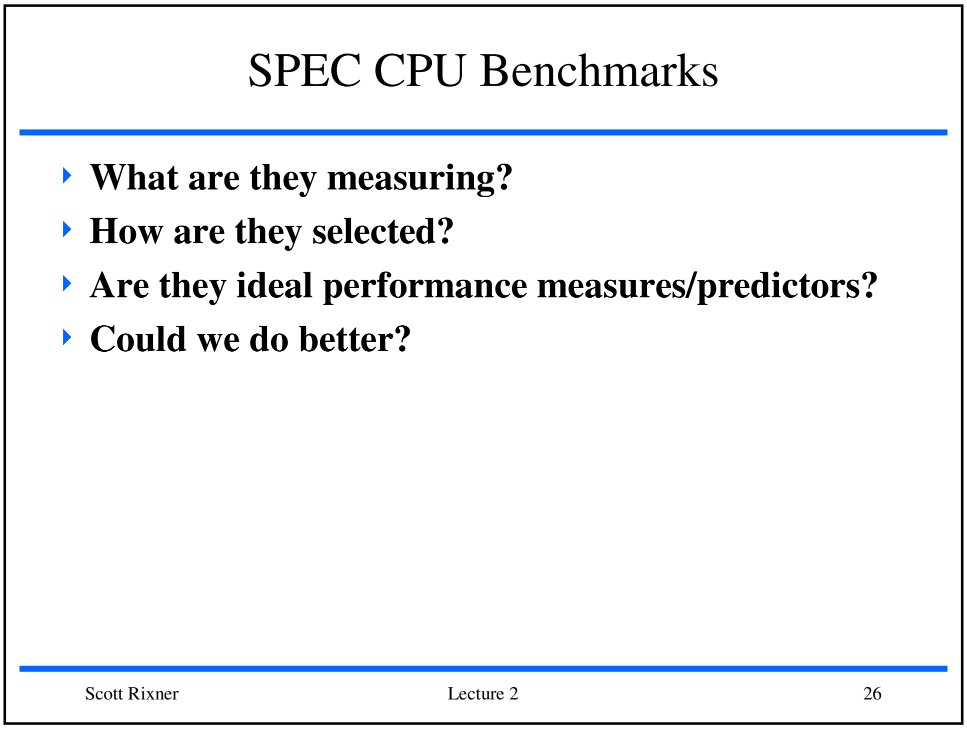
SPEC CPU2000 Integer Benchmarks

- **gzip**
  - **Compression**
- **vpr**
  - **FPGA circuit place and route**
- **gcc**
  - **C compiler**
- **mcf**
  - **Combinatorial optimization**
- **crafty**
  - **Chess player**
- **parser**
  - **Word processing**
- **eon**
  - **Computer visualization**
- **perlbmk**
  - **Perl programming language**
- **gap**
  - **Group theory, interpreter**
- **vortex**
  - **Object-oriented database**
- **bzip2**
  - **Compression**
- **twolf**
  - **Place and route simulator**



SPEC CPU2006 Integer Benchmarks

- **perlbench**
  - **Perl programming language**
- **bzip2**
  - **Compression**
- **gcc**
  - **C compiler**
- **mcf**
  - **Combinatorial optimization**
- **gobmk**
  - **Go player**
- **hmmer**
  - **Gene sequence search**
- **sjeng**
  - **Chess player**
- **libquantum**
  - **Quantum computer simulator**
- **h264ref**
  - **Video compression**
- **omnetpp**
  - **Discrete event simulator**
- **astar**
  - **A* path finding**
- **xalancbmk**
  - **XML processing**



SPEC CPU Benchmarks

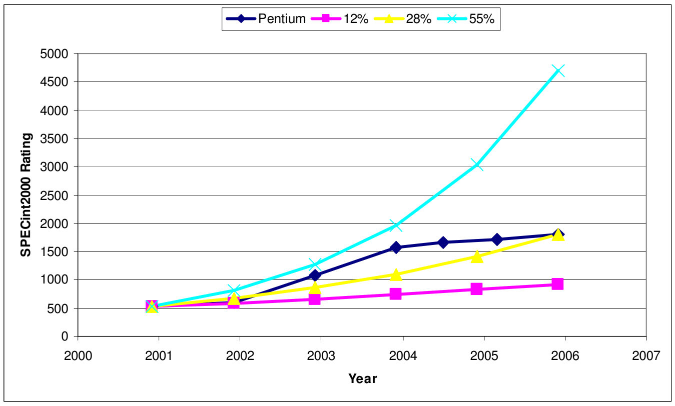
- **What are they measuring?**
- **How are they selected?**
- **Are they ideal performance measures/predictors?**
- **Could we do better?**

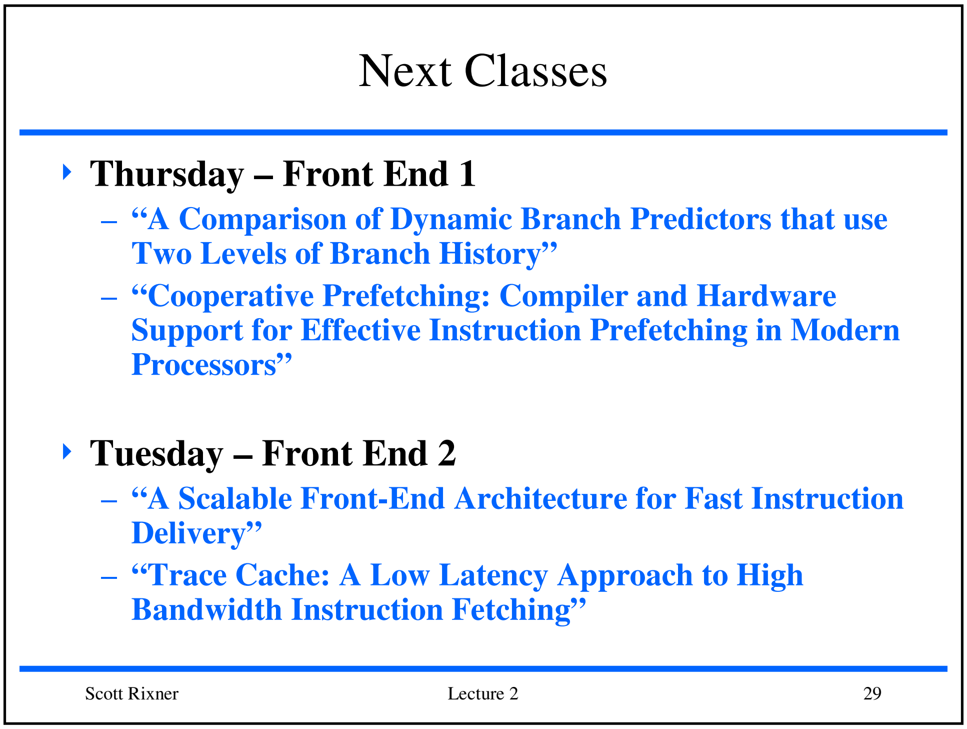
SPEC2000 Results

- **SPEC2000 ratios are speedups over a 300MHz Sun Ultra 5 times 100**

Processor	SPEC2000 Int	SPEC2000 FP	Power (W)
Alpha 21364	904	1,279	155
AMD Opteron 254	1,789	2,132	92
AMD Opteron 280	1,499	1,752	95
IBM Power4+	1,077	1,598	100
IBM Power5	1,470	2,839	120
Intel Itanium 2	1,490	2,801	130
Intel XeonMP	1,388	1,314	140
Intel Xeon	1,810	1,909	130

Pentium Performance Scaling





Next Classes

- **Thursday – Front End 1**
  - **“A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History”**
  - **“Cooperative Prefetching: Compiler and Hardware Support for Effective Instruction Prefetching in Modern Processors”**
- **Tuesday – Front End 2**
  - **“A Scalable Front-End Architecture for Fast Instruction Delivery”**
  - **“Trace Cache: A Low Latency Approach to High Bandwidth Instruction Fetching”**