CMSC 411
Computer Systems Architecture
Lecture 3
Reliability and Performance

Alan Sussman
als@cs.umd.edu

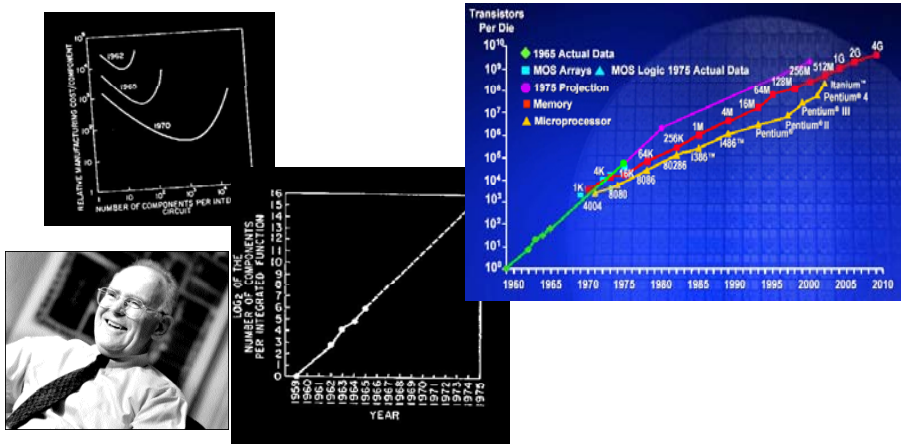
Administrivia

- Homework problems for Unit 1 posted
  - due next Thursday, 2/12
- Read Appendix A – Basic Pipelining
- Appendix B useful as a MIPS ISA reference
  - worth reading, but we'll only touch on parts of it in lecture as needed

Outline

- Moore's Law and Technology Trends
- Reliability and MTTF
- Performance
- MIPS – An ISA for Pipelining
  - 5 stage pipelining
  - Structural and Data Hazards
  - Forwarding
  - Branch Schemes
  - Exceptions and Interrupts
  - Conclusion

Moore's Law: 2X transistors / "year"



- "Cramming More Components onto Integrated Circuits"
  - Gordon Moore, Electronics, 1965
- # on transistors / cost-effective integrated circuit double every N months (12 ≤ N ≤ 24)

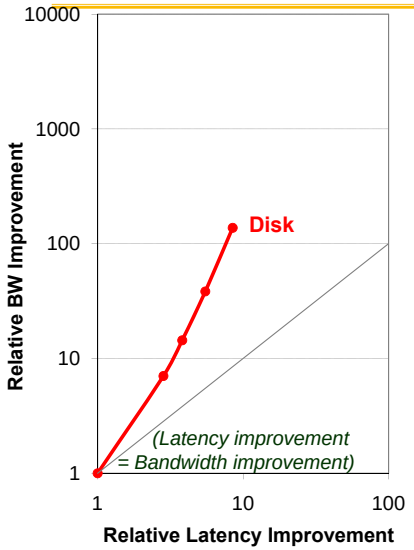
Tracking Technology Performance Trends

- Drill down into 4 technologies:
  - Disks,
  - Memory,
  - Network,
  - Processors
- Compare ~1980 Archaic (Nostalgic) vs. ~2000 Modern (Newfangled)
  - Performance Milestones in each technology
- Compare for Bandwidth vs. Latency improvements in performance over time
- Bandwidth: number of events per unit time
  - E.g., Mbits / second over network, Mbytes / second from disk
- Latency: elapsed time for a single event
  - E.g., one-way network delay in microseconds, average disk access time in milliseconds

Disks: Archaic(Nostalgic) v. Modern(Newfangled)

• CDC Wren I, 1983	• Seagate 373453, 2003
• 3600 RPM	• 15000 RPM (4X)
• 0.03 GBytes capacity	• 73.4 GBytes (2500X)
• Tracks/Inch: 800	• Tracks/Inch: 64000 (80X)
• Bits/Inch: 9550	• Bits/Inch: 533,000 (60X)
• Three 5.25" platters	• Four 2.5" platters (in 3.5" form factor)
• Bandwidth: 0.6 MBytes/sec	• Bandwidth: 86 MBytes/sec (140X)
• Latency: 48.3 ms	• Latency: 5.7 ms (8X)
• Cache: none	• Cache: 8 MBytes

Latency Lags Bandwidth (for last ~20 years)



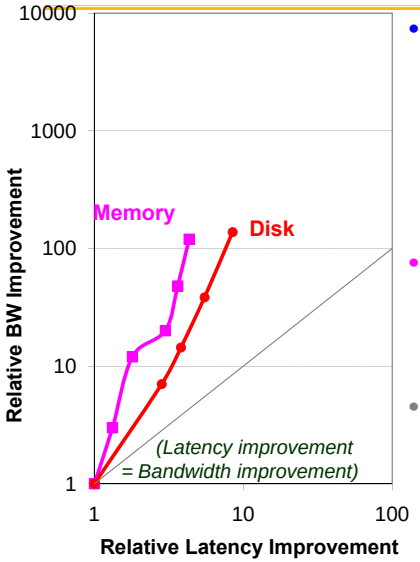
- Performance Milestones
- Disk: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)

(latency = simple operation w/o contention
BW = best-case)

Memory: Archaic (Nostalgic) v. Modern (Newfangled)

• 1980 DRAM (asynchronous)	• 2000 Double Data Rate Synchr. (clocked) DRAM
• 0.06 Mbits/chip	• 256.00 Mbits/chip (4000X)
• 64,000 xtors, 35 mm^2	• 256,000,000 xtors, 204 mm^2
• 16-bit data bus per module, 16 pins/chip	• 64-bit data bus per DIMM, 66 pins/chip (4X)
• 13 Mbytes/sec	• 1600 Mbytes/sec (120X)
• Latency: 225 ns	• Latency: 52 ns (4X)
• (no block transfer)	• Block transfers (page mode)

Latency Lags Bandwidth (last ~20 years)



- **Performance Milestones**
- **Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM** (4x,120x)
- **Disk: 3600, 5400, 7200, 10000, 15000 RPM** (8x, 143x)

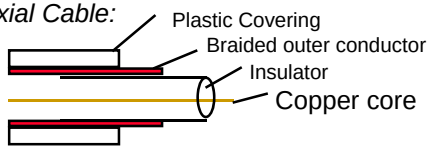
(latency = simple operation w/o contention
BW = best-case)

LANs: Archaic (Nostalgic)v. Modern (Newfangled)

- **Ethernet 802.3**
- **Year of Standard: 1978**
- **10 Mbits/s link speed**
- **Latency: 3000 µsec**
- **Shared media**
- **Coaxial cable**

- **Ethernet 802.3ae**
- **Year of Standard: 2003**
- **10,000 Mbits/s (1000X) link speed**
- **Latency: 190 µsec (15X)**
- **Switched media**
- **Category 5 copper wire**

Coaxial Cable:

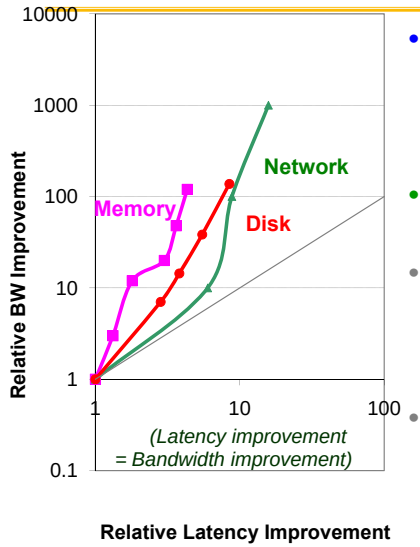


"Cat 5" is 4 twisted pairs in bundle

Twisted Pair:

Copper, 1mm thick,
twisted to avoid antenna effect

Latency Lags Bandwidth (last ~20 years)



- **Performance Milestones**
- **Ethernet: 10Mb, 100Mb, 1000Mb, 10000 Mb/s** (16x,1000x)
- **Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM** (4x,120x)
- **Disk: 3600, 5400, 7200, 10000, 15000 RPM** (8x, 143x)

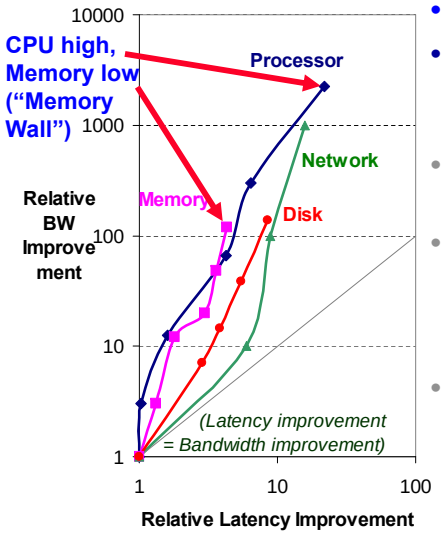
(latency = simple operation w/o contention
BW = best-case)

CPUs: Archaic (Nostalgic) v. Modern (Newfangled)

- **1982 Intel 80286**
- **12.5 MHz**
- **2 MIPS (peak)**
- **Latency 320 ns**
- **134,000 xtors, 47 mm²**
- **16-bit data bus, 68 pins**
- **Microcode interpreter, separate FPU chip**
- **(no caches)**

- **2001 Intel Pentium 4**
- **1500 MHz(120X)**
- **4500 MIPS (peak) (2250X)**
- **Latency 15 ns (20X)**
- **42,000,000 xtors, 217 mm²**
- **64-bit data bus, 423 pins**
- **3-way superscalar, Dynamic translate to RISC, Superpipelined (22 stage), Out-of-Order execution**
- **On-chip 8KB Data caches, 96KB Instr. Trace cache, 256KB L2 cache**

Latency Lags Bandwidth (last ~20 years)



- **Performance Milestones**
- **Processor: '286, '386, '486, Pentium, Pentium Pro, Pentium 4** (21x,2250x)
- **Ethernet: 10Mb, 100Mb, 1000Mb, 10000 Mb/s** (16x,1000x)
- **Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM** (4x,120x)
- **Disk : 3600, 5400, 7200, 10000, 15000 RPM** (8x, 143x)

Rule of Thumb for Latency Lagging BW

- **In the time that bandwidth doubles, latency improves by no more than a factor of 1.2 to 1.4**
  (and capacity improves faster than bandwidth)
- **Stated alternatively:**
  Bandwidth improves by more than the square of the improvement in Latency

6 Reasons Latency Lags Bandwidth

1. Moore's Law helps BW more than latency

- **Faster transistors, more transistors, more pins help Bandwidth**
  - » **MPU Transistors: 0.130 vs. 42 M xtors (300X)**
  - » **DRAM Transistors: 0.064 vs. 256 M xtors (4000X)**
  - » **MPU Pins: 68 vs. 423 pins (6X)**
  - » **DRAM Pins: 16 vs. 66 pins (4X)**
- **Smaller, faster transistors but communicate over (relatively) longer wires: limits latency**
  - » **Feature size: 1.5 to 3 vs. 0.18 micron (8X,17X)**
  - » **MPU Die Size: 35 vs. 204 mm^2 (ratio sqrt ⇒ 2X)**
  - » **DRAM Die Size: 47 vs. 217 mm^2 (ratio sqrt ⇒ 2X)**

6 Reasons Latency Lags Bandwidth (cont'd)

2. Distance limits latency

- **Size of DRAM block ⇒ long bit and word lines ⇒ most of DRAM access time**
- **Speed of light and computers on network**

3. Bandwidth easier to sell ("bigger=better")

- **E.g., 10 Gbits/s Ethernet ("10 Gig") vs. 10 µsec latency Ethernet**
- **4400 MB/s DIMM ("PC4400") vs. 50 ns latency**
- **Even if just marketing, customers now trained**
- **Since bandwidth sells, more resources thrown at bandwidth, which further tips the balance**

6 Reasons Latency Lags Bandwidth (cont'd)

4. Latency helps BW, but not vice versa

- Spinning disk faster improves both bandwidth and rotational latency
  - » 3600 RPM ⇒ 15000 RPM = 4.2X
  - » Average rotational latency: 8.3 ms ⇒ 2.0 ms
  - » Things being equal, also helps BW by 4.2X
- Lower DRAM latency ⇒ More access/second (higher bandwidth)
- Higher linear density helps disk BW (and capacity), but not disk Latency
  - » 9,550 BPI ⇒ 533,000 BPI ⇒ 60X in BW

6 Reasons Latency Lags Bandwidth (cont'd)

5. Bandwidth hurts latency

- Queues help Bandwidth, hurt Latency (Queuing Theory)
- Adding chips to widen a memory module increases Bandwidth but higher fan-out on address lines may increase Latency

6. Operating System overhead hurts Latency more than Bandwidth

- Long messages amortize overhead; overhead bigger part of short messages

Summary of Technology Trends

- For disk, LAN, memory, and microprocessor, bandwidth improves by square of latency improvement
  - In the time that bandwidth doubles, latency improves by no more than 1.2X to 1.4X
- Lag probably even larger in real systems, as bandwidth gains multiplied by replicated components
  - Multiple processors in a cluster or even in a chip
  - Multiple disks in a disk array
  - Multiple memory modules in a large memory
  - Simultaneous communication in switched LAN
- HW and SW developers should innovate assuming Latency Lags Bandwidth
  - If everything improves at the same rate, then nothing really changes
  - When rates vary, require real innovation

Outline

- Moore's Law and Technology Trends
- Reliability and MTTF
- Performance
- MIPS – An ISA for Pipelining
  - 5 stage pipelining
  - Structural and Data Hazards
  - Forwarding
  - Branch Schemes
  - Exceptions and Interrupts
  - Conclusion

Define and quantity dependability (1/3)

- How decide when a system is operating properly?
- Infrastructure providers now offer Service Level Agreements (SLA) to guarantee that their networking or power service would be dependable
- Systems alternate between 2 states of service with respect to an SLA:

1. Service accomplishment, where the service is delivered as specified in SLA
2. Service interruption, where the delivered service is different from the SLA

- Failure = transition from state 1 to state 2
- Restoration = transition from state 2 to state 1

Define and quantity dependability (2/3)

- *Module reliability* = measure of continuous service accomplishment (or time to failure). 2 metrics

1. *Mean Time To Failure* (*MTTF*) measures Reliability
2. *Failures In Time* (*FIT*) = 1/MTTF, the rate of failures
   - Traditionally reported as failures per billion hours of operation

- *Mean Time To Repair* (*MTTR*) measures Service Interruption
  - *Mean Time Between Failures* (*MTBF*) = MTTF+MTTR
- *Module availability* measures service as alternate between the 2 states of accomplishment and interruption (number between 0 and 1, e.g. 0.9)
- *Module availability = MTTF / (MTTF + MTTR)*

Example calculating reliability

- If modules have *exponentially distributed lifetimes* (age of module does not affect probability of failure), overall failure rate is the sum of failure rates of the modules
- Calculate FIT and MTTF for 10 disks (1M hour MTTF per disk), 1 disk controller (0.5M hour MTTF), and 1 power supply (0.2M hour MTTF):

$$FailureRate =$$

$$MTTF =$$

Example calculating reliability

- If modules have *exponentially distributed lifetimes* (age of module does not affect probability of failure), overall failure rate is the sum of failure rates of the modules
- Calculate FIT and MTTF for 10 disks (1M hour MTTF per disk), 1 disk controller (0.5M hour MTTF), and 1 power supply (0.2M hour MTTF):

$$\begin{aligned} FailureRate &= 10 \times (1/1{,}000{,}000) + 1/500{,}000 + 1/200{,}000 \\ &= (10 + 2 + 5)/1{,}000{,}000 \\ &= 17/1{,}000{,}000 \\ &= 17{,}000\,FIT \\ MTTF &= 1{,}000{,}000{,}000/17{,}000 \\ &\approx 59{,}000\,hours \end{aligned}$$

Outline

- Moore's Law and Technology Trends
- Reliability and MTTF
- Performance
- MIPS – An ISA for Pipelining
  - 5 stage pipelining
  - Structural and Data Hazards
  - Forwarding
  - Branch Schemes
  - Exceptions and Interrupts
  - Conclusion

Definition: Performance

- Performance is in units of things per second
  - bigger is better
- If we are primarily concerned with response time

$$\text{performance}(x) = \frac{1}{\text{execution\_time}(x)}$$

" X is n times faster than Y" means

$$n = \frac{\text{Performance}(X)}{\text{Performance}(Y)} = \frac{\text{Execution\_time}(Y)}{\text{Execution\_time}(X)}$$

Performance: What to measure

- Usually rely on benchmarks vs. real workloads
- To increase predictability, collections of benchmark applications, called *benchmark suites*, are popular
- SPECCPU: popular desktop benchmark suite
  - CPU only, split between integer and floating point programs
  - SPECint2000 has 12 integer, SPECfp2000 has 14 integer pgms
  - SPECCPU2006 announced Spring 2006 (after book written)
  - SPECSFS (NFS file server) and SPECWeb (WebServer) added as server benchmarks
- Transaction Processing Council measures server performance and cost-performance for databases
  - TPC-C Complex query for Online Transaction Processing
  - TPC-H models ad hoc decision support
  - TPC-W a transactional web benchmark
  - TPC-App application server and web services benchmark

How Summarize Suite Performance (1/3)

- Arithmetic average of execution time of all pgms?
  - But they vary by 4X in speed, so some would be more important than others in arithmetic average
- Could add a weights per program, but how pick weight?
  - Different companies want different weights for their products
- SPECRatio: Normalize execution times to reference computer, yielding a ratio proportional to performance =

$$\frac{\text{time on reference computer}}{\text{time on computer being rated}}$$

How Summarize Suite Performance (2/3)

- If program SPECRatio on Computer A is 1.25 times bigger than Computer B, then

$$1.25 = \frac{SPECRatio_A}{SPECRatio_B} = \frac{\frac{ExecutionTime_{reference}}{ExecutionTime_A}}{\frac{ExecutionTime_{reference}}{ExecutionTime_B}}$$

$$= \frac{ExecutionTime_B}{ExecutionTime_A} = \frac{Performance_A}{Performance_B}$$

  - Note that when comparing 2 computers as a ratio, execution times on the reference computer drop out, so choice of reference computer is irrelevant

How Summarize Suite Performance (3/3)

$$GeometricMean = \sqrt[n]{\prod_{i=1}^{n} SPECRatio_i}$$

Outline

- Moore's Law and Technology Trends
- Reliability and MTTF
- Performance
- MIPS – An ISA for Pipelining
  - 5 stage pipelining
  - Structural and Data Hazards
  - Forwarding
  - Branch Schemes
  - Exceptions and Interrupts
  - Conclusion

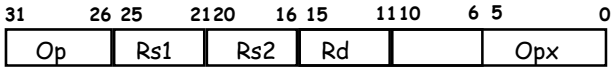
A "Typical" RISC ISA

- 32-bit fixed format instruction (4 formats)
- 32 32-bit GPR (R0 contains zero, DP take pair)
- 3-address, reg-reg arithmetic instruction
- Single address mode for load/store: base + displacement
  - no indirection
- Simple branch conditions
- Delayed branch

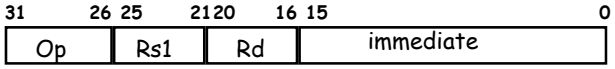
see: SPARC, MIPS, HP PA-Risc, DEC Alpha, IBM PowerPC, CDC 6600, CDC 7600, Cray-1, Cray-2, Cray-3

Example: MIPS

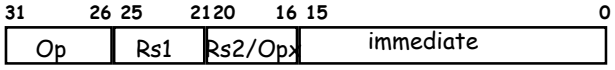
Register-Register



Register-Immediate



Branch



Jump / Call