Computer Science 146
Computer Architecture

Fall 2019

Harvard University

Instructor: Prof. David Brooks
dbrooks@eecs.harvard.edu

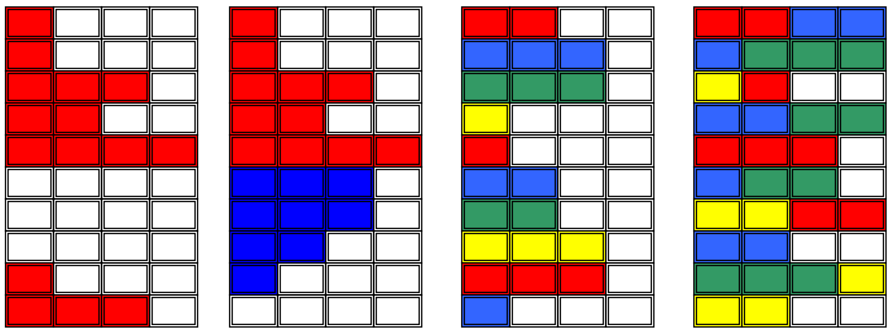
Lecture 21: Multithreading and I/O

Lecture Outline

- HW#5 on webpage
- Project Questions?
- Tale of two of multithreaded x86's
  - Intel Pentium 4 multithreading
  - MemoryLogix MLX1 multithreading
- Storage and I/O
  - Storage Technology (H&P 7.1-7.2)
  - I/O Busses (7.3)
  - RAID (H&P 7.4-7.5)

Multithreading Paradigms

SuperScalar | Coarse MT | Fine MT | Simultaneous MT

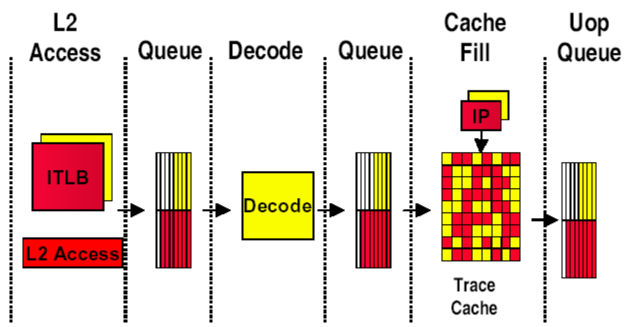


Pentium 3, Alpha EV6/7 | IBM Pulsar | | Intel P4-HT, EV8, Others?

Coarse vs. Fine Grained MT

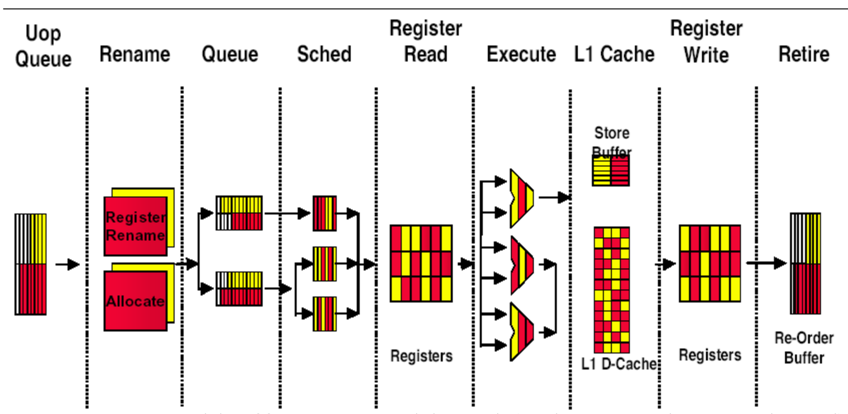
- Coarse-Grained
  - Makes sense for in-order/shorter pipelines
  - Switch threads on long stalls (L2 cache misses)
  - Threads don’t interfere with each other much
  - Can’t improve utilization on L1 misses/bpred mispredicts
- Fine-grained
  - Out-of-order, deep pipelines
  - Instructions from multiple threads in stage at a time, miss or not
  - Improves utilization in all scenarios
  - Individual thread performance suffers due to interference

Pentium 4 Front-End



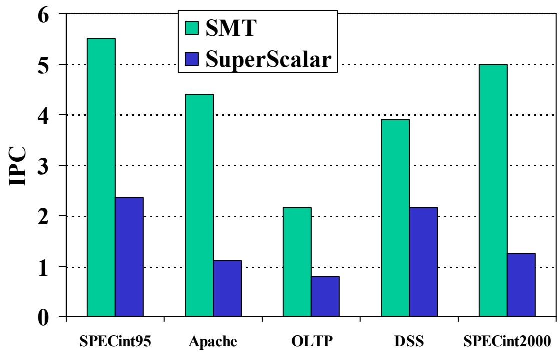
•Front end resources arbitrate between threads every cycle
•ITLB, RAS, BHB(Global History), Decode Queue are duplicated

Pentium 4 Backend



•Some queues and buffers are partitioned (only ½ entries per thread)
•Scheduler is oblivious to instruction thread Ids (limit on # per scheduler)

Multithreading Performance



9-stage pipeline
128KB I/D Cache
6 IntALU (4 Load/Store)
4 FPALUs
8 SMT threads
8-insn fetch (2 threads)

- Intel claiming ~20-30% increase on P4
- Lot of debate exists on performance benefits

Another look at multithreading…

- As an aside…
  - Embedded microprocessors (cell phones, PDAs)
    - Currently dominated by ARM ISA
    - Why not x86?
      + Huge software base, device drivers, programming skills..
      + Compatibility with PC/x86 platform
      – Die size, power dissipation of x86 vs. ARM

ARM vs. x86: X86 Core sizes are ~10x larger

Processors	Total Cache (KB)	Die Size (mm^2)	Est Core Size (mm^2)	Core Size Ratio	Typical Speed (MHz)
Intel ULV PIII-M	544	80	~34	~13	>1000
AMD Duron	192	55	~37	~14	>1000
Transmeta 5800	640	55	~25	~10	>800
VIA C3	192	52	~31	~12	>800
ARM 1026EJ-S	**32**	**4.6**	**2.6**	**1**	**>400**

Why are x86 cores so large?

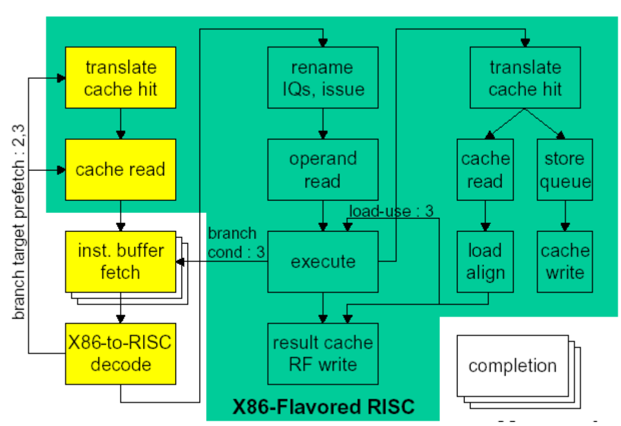
Why are x86 cores so large?

- X86 designed for peak frequency, performance
  - Large multi-level caches, TLBs (>30-50% of die)
  - Superscalar, speculative execution
  - Branch prediction tables, trace caches
  - Aggressive circuit designs
    - Large transistors (tune up transistor sizes for performance)
    - Replicated logic for speed
- Many x86 features not in ARM architecture (yet)
  - FPU, MMX, SSE (20-30% of core)
  - System features (I/O, tasks, MP)
  - Variable-length instruction decode (1-16 bytes)

MemoryLogix MLX1: Tiny Multithreaded 586 core

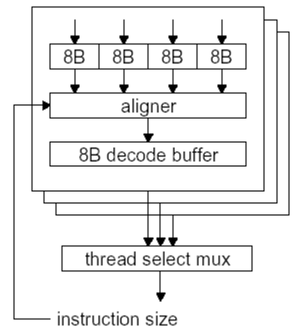
- Presented at Microprocessor Forum 2002
- MLX1 Design Goals
  - "Synthesizable" x86 core + Support FPU, MMX
  - Support variable size cache and TLBs using 1-port SRAMs
  - 2.5x the size of ARM10 core (<1/4 size of mobile x86)
  - 2.5x system performance of single-threaded core/MHz
- MLX1 Design Strategy
  - Simple and Small
    - Scalar RISC pipeline
    - Optimize for frequently used x86 instructions (92%)
    - Features to map x86 + Java instructions (microcode for complex insns)
  - High performance by SMT
    - Threads share register file, non-blocking 8-bank unified cache

MLX1: Multi-Fetch, Scalar Execute Pipeline



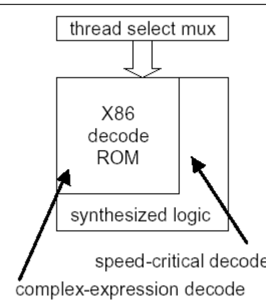
Multi-fetch Using 3 Identical Fetch Units

- Each fetch unit
  - Operates independently
  - Holds four 8-byte blocks
  - Prefetches up to 3 blocks from sequential path
  - Prefetches 2 blocks from target path as condition is evaluated
- Cache-location aware logic
  - Determines cache location of the next sequential 64B line
  - Remembers cache location of two previous 64B lines



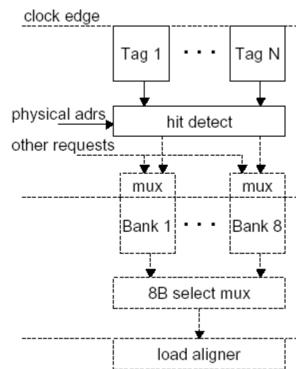
Threads Share a Single-Instruction Decoder

- ROM-based decode
  - 512 words x D wide
  - 430 MHz, 0.05mm^2 (0.13um) or
  - 700 MHz, 0.1mm^2 fast SRAM
  - 1-cycle size decode is speed critical
- Thread switch can occur
  - Thread's decode buffer not full
  - Thread's issue queue full
  - After a branch, 4-cycle load
  - After a serialization instruction
  - After 8th consecutive decode
- Threads Share 64 Rename Registers



Threads Share a Unified Cache

- 3-cycle access
  - 1: Tag access for hit check
  - 2: Data access when needed
  - 3: Data alignment & routing
  - Prefetches to reduce penalty
- Up to 8 accesses/cycle
  - 8 banks/line, 8 bytes/bank
  - Support multiple instruction fetch
- N-way set associative
  - N can be any integer
  - True LRU replacement
  - Set partitioning and locking (reduce multithread conflicts)



MLX1 Summary

- MLX1– Tiny x86 core
  - In 0.13um:
  - 3.5mm^2(core)+ 1.0mm^2 (MMX) + 1.5mm^2 (FPU) = 6.0mm^2
  - Compared to 146mm^2 for a Pentium 4
  - Can multithreading buy back the performance?
    - Sounds interesting
    - Depends on workloads
    - Are there enough embedded workloads that are throughput oriented?

Motivation: Who Cares About I/O?

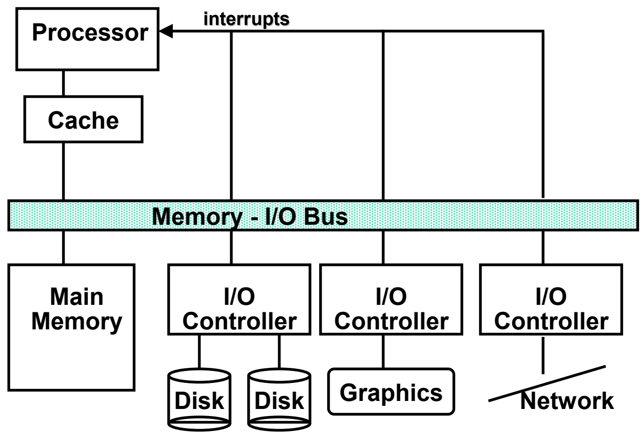
- CPU Performance: 57% per year
- I/O system performance limited by *mechanical* delays (disk I/O):
  < 10% increase per year (IO per sec)
- Amdahl's Law: system speed-up limited by the slowest part!
  - 10% IO & 10x CPU => 5x Performance (lose 50%)
  - 10% IO & 100x CPU => 10x Performance (lose 90%)
  - Need fast disk accesses (VM swaps, file reading, networks, etc)
- I/O bottleneck:
  - Increasing fraction of time in I/O (relative to CPU)
  - Similar to Memory Wall problem
- Why not context switch on I/O operation?
  - Must find threads to context switch to
  - Context-switching requires more memory

I/O Device Characteristics

- Types:
  - Input: Read Only
  - Output: write only
  - Storage: Both

Device	**Type**	**Partner**	**Peak Data Rate KB/S**
Mouse	I	Human	0.01
CRT	O	Human	60,000
Modem	I/O	Machine	2-8
LAN	I/O	Machine	500-600
Tape	Storage	Machine	2000
Disk	Storage	Machine	2000-10,000

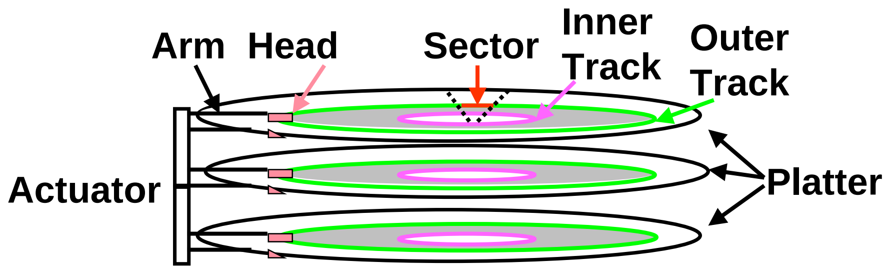
I/O Systems



Storage Technology Drivers

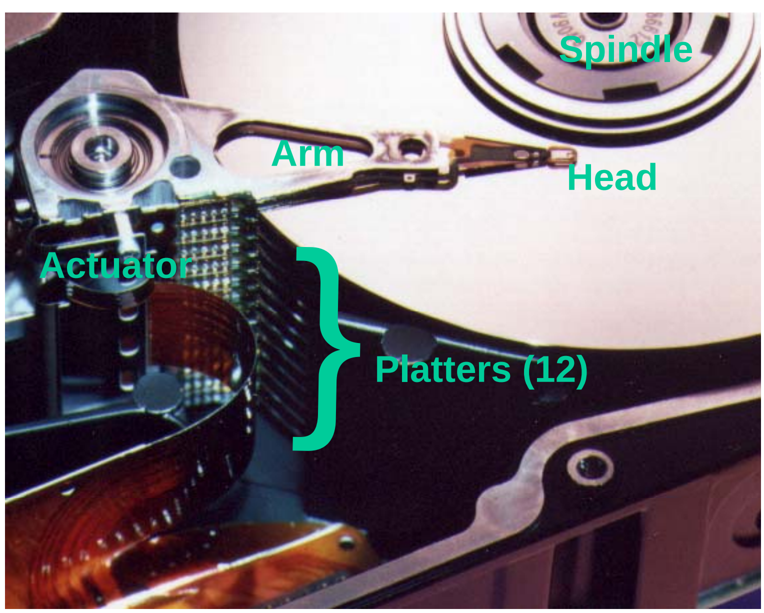
- Driven by the prevailing computing paradigm
  - 1950s: migration from batch to on-line processing
  - 1990s: migration to ubiquitous computing
    - computers in phones, books, cars, video cameras, …
    - nationwide fiber optical network with wireless tails
- Effects on storage industry:
  - Embedded storage
    - smaller, cheaper, more reliable, lower power
  - Data utilities
    - high capacity, hierarchically managed storage

Disk Device Terminology

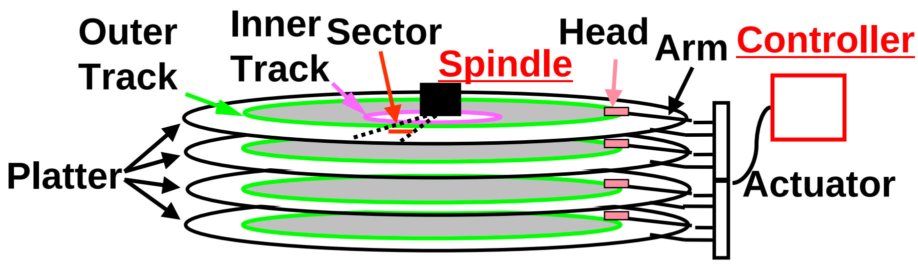


- Several platters, with information recorded magnetically on both surfaces (usually)
- Bits recorded in tracks, which in turn divided into sectors (e.g., 512 Bytes)
- Actuator moves head (end of arm,1/surface) over track ("seek"), select surface, wait for sector rotate under head, then read or write
  - "Cylinder": all tracks under heads

Photo of Disk Head, Arm, Actuator



Disk Device Performance



- Disk Latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead
- Seek Time? depends no. tracks move arm, seek speed of disk
- Rotation Time? depends on speed disk rotates, how far sector is from head
- Transfer Time? depends on data rate (bandwidth) of disk (bit density), size of request

Disk Device Performance

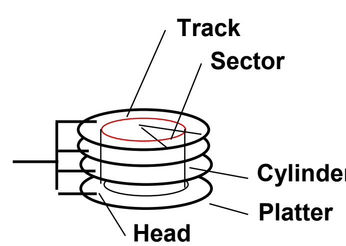
- Average distance sector from head?
- 1/2 time of a rotation
  - 10000 Revolutions Per Minute ⇒ 166.67 Rev/sec
  - 1 revolution = 1/ 166.67 sec ⇒ 6.00 milliseconds
  - 1/2 rotation (revolution) ⇒ 3.00 ms
- Average no. tracks move arm?
  - Sum all possible seek distances from all possible tracks / # possible
    - Assumes average seek distance is random
  - Disk industry standard benchmark

Data Rate: Inner vs. Outer Tracks

- To keep things simple, orginally kept same number of sectors per track
  - Since outer track longer, lower bits per inch
- Competition ⇒ decided to keep BPI the same for all tracks ("constant bit density")
  - ⇒ More capacity per disk
  - ⇒ More of sectors per track towards edge
  - ⇒ Since disk spins at constant speed, outer tracks have faster data rate
- Bandwidth outer track 1.7X inner track!
  - Inner track highest density, outer track lowest, so not really constant
  - 2.1X length of track outer / inner, 1.7X bits outer / inner

Devices: Magnetic Disks

- Purpose:
  - Long-term, nonvolatile storage
  - Large, inexpensive, slow level in the storage hierarchy
- Characteristics:
  - Seek Time (~8 ms avg)
    - positional latency
    - rotational latency
- Transfer rate
  - 10-40 MByte/sec
  - Block transfers
- Capacity
  - 100s of Gigabytes in 2002
  - Quadruples every 2 years



7200 RPM = 120 RPS => 8 ms per rev
ave rot. latency = 4 ms
128 sectors per track => 0.25 ms per sector
1 KB per sector => 16 MB / s

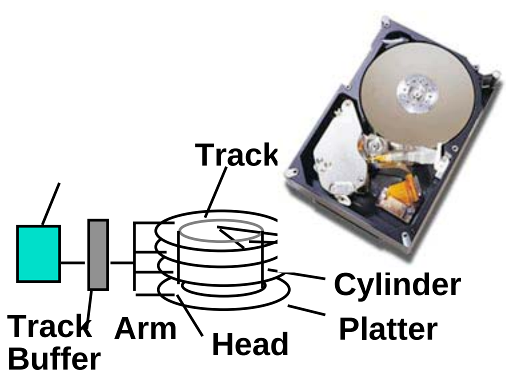
Response time
= Queue + Controller + Seek + Rot + Xfer

Service time (Controller + Seek + Rot + Xfer)

Disk Performance Model /Trends

- Capacity
  + 100%/year (2X / 1.0 yrs)
- Transfer rate (BW)
  + 40%/year (2X / 2.0 yrs)
- Rotation + Seek time
  – 8%/ year (1/2 in 10 yrs)
- MB/$
  > 100%/year (2X / 1.0 yrs)
  Fewer support chips + increased areal density

State of the Art: Barracuda 180



Latency =

per access + *per byte* { **Queuing Time + Controller time + Seek Time + Rotation Time + Size / Bandwidth**

source: www.seagate.com

– 181.6 GB, 3.5 inch disk
– 12 platters, 24 surfaces
– 24,247 cylinders
– 7,200 RPM; (4.2 ms avg. latency)
– 7.4/8.2 ms avg. seek (r/w)
– 64 to 35 MB/s (internal)
– 0.1 ms controller time
– 10.3 watts (idle)

Disk Performance Example

- Calculate time to read 64 KB (128 sectors) for Barracuda 180 X using advertised performance
- Disk latency = average seek time + average rotational delay + transfer time + controller overhead

= 7.4 ms + 0.5 * 1/(7200 RPM)
+ 64 KB / (65 MB/s) + 0.1 ms

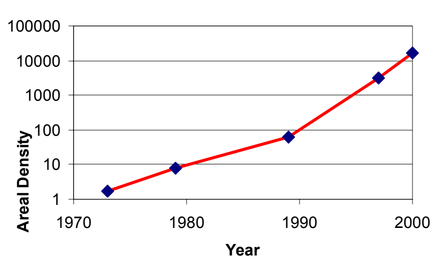
= 7.4 ms + 0.5 /(7200 RPM/(60000ms/M))
+ 64 KB / (65 KB/ms) + 0.1 ms

= 7.4 + 4.2 + 1.0 + 0.1 ms = 12.7 ms

- Transfer time (1.0/12.7=7.8%) is a small fraction of the total time

Disk Capacity Trends: Areal Density

- Bits recorded along a track
  - Metric is Bits Per Inch (BPI)
- Number of tracks per surface
  - Metric is Tracks Per Inch (TPI)
- Disk designs quote bit density per unit area
  - Metric is Bits Per Square Inch
  - Called Areal Density
  - Areal Density = BPI x TPI
  - Change slope 30%/yr to 60%/yr about 1991



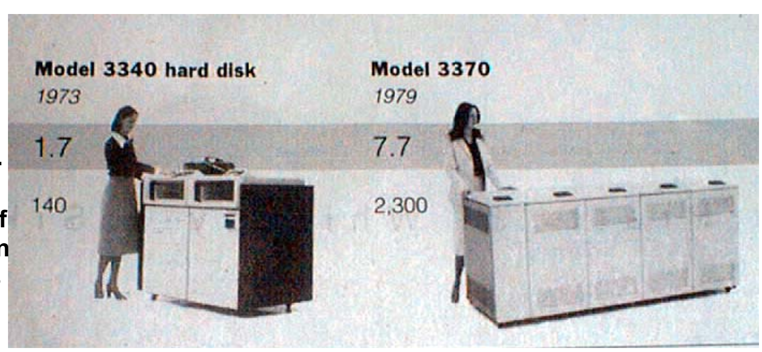
Historical Perspective

- 1956 IBM Ramac — early 1970s Winchester
  - Developed for mainframe computers, proprietary interfaces
  - Steady shrink in form factor: 27 in. to 14 in
- Form factor and capacity drives market, more than performance
- 1970s: Mainframes ⇒ 14 inch diameter disks
- 1980s: Minicomputers,Servers ⇒ 8",5 1/4" diameter
- PCs, workstations Late 1980s/Early 1990s:
  - Mass market disk drives become a reality
  - Pizzabox PCs ⇒ 3.5 inch diameter disks
  - Laptops, notebooks ⇒ 2.5 inch disks
- 2000s:
  - 1 inch for cameras, cell phones?

Disk History

Data density Mbit/sq. in.

Capacity of Unit Shown Megabytes

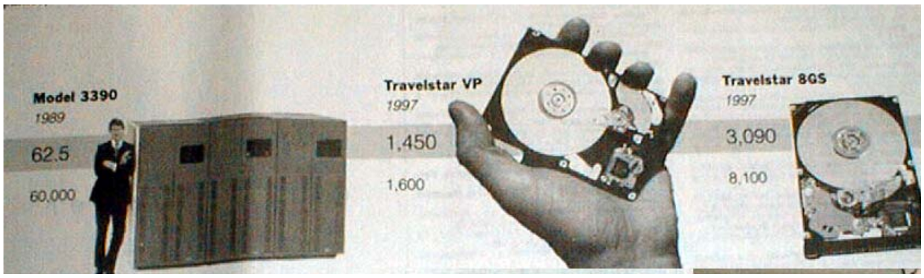


1973:
1. 7 Mbit/sq. in
140 MBytes

1979:
7. 7 Mbit/sq. in
2,300 MBytes

source: New York Times, 2/23/98, page C3,
"Makers of disk drives crowd even more data into even smaller spaces"

Disk History



1989:
63 Mbit/sq. in
60,000 MBytes

1997:
1450 Mbit/sq. in
2300 MBytes

1997:
3090 Mbit/sq. in
8100 MBytes

source: New York Times, 2/23/98, page C3,
"Makers of disk drives crowd even more data into even smaller spaces"

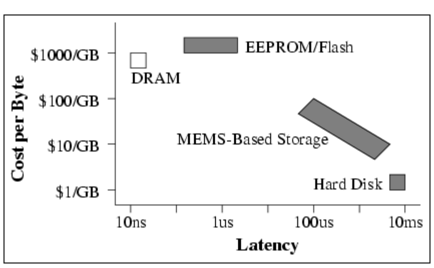
1 inch disk drive!

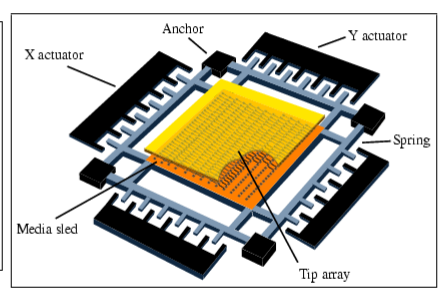
- 2000 IBM MicroDrive:
  - 1.7" x 1.4" x 0.2"
  - 1 GB, 3600 RPM,
    5 MB/s, 15 ms seek
  - Digital camera, PocketPCs
- 2003 MicroDrives, 4GB

What about FLASH

- Compact Flash Cards
  - Intel Strata Flash (16 Mb in 1 square cm.)
  - 100,000 write/erase cycles.
  - Standby current = 100uA, write = 45mA
  - Transfer @ 3.5MB/s, read access times in 65-150ns range
  - Compact Flash (2002) 256MB=$73 512MB=$170, 1GB=$560
  - Compact Flash (2004) 256MB=$39 512MB=$80 1GB=$146 2GB=$315 4GB=$800
- IBM/Hitachi Microdrive 4GB=$370
  - Standby current = 20mA, write = 250mA
  - Efficiency advertised in watts/MB
- Flash vs. Disks
  - Nearly instant standby wake-up time
  - Random access to data stored
  - Tolerant to shock and vibration (1000G of operating shock)

MEMS based storage?





Fixed Probe Tips, Moving Media
(4 – 11 GB in 8mm x 8mm Array)

From Schlosser et al, ASPLOS 2000

Next two lectures

- Monday:
  - Finish up with I/O Monday
    - I/O Buses
    - RAID Systems
  - Course Evaluations (need a volunteer to return them)
- Next Wednesday:
  - Google Cluster
  - Course Summary and Wrapup
  - Final Review (may schedule another review before final)
- Final Exam: Tue 05/25 (Boylston 105)