

inst.eecs.berkeley.edu/~cs61c
UCB CS61C : Machine Structures

Lecture 35 – Input / Output
2010-04-21



Lecturer SOE
Dan Garcia

We've merged 3 lectures into 1...
See 2008Sp for full slides (few Qs plz)

SKINPUT ... USE YOUR BODY AS INPUT!!

Chris Harrison has developed a system that allows you to touch your skin and control a computer. He shows examples, combined with a pico projector, of dialing a phone number on your hand, etc.



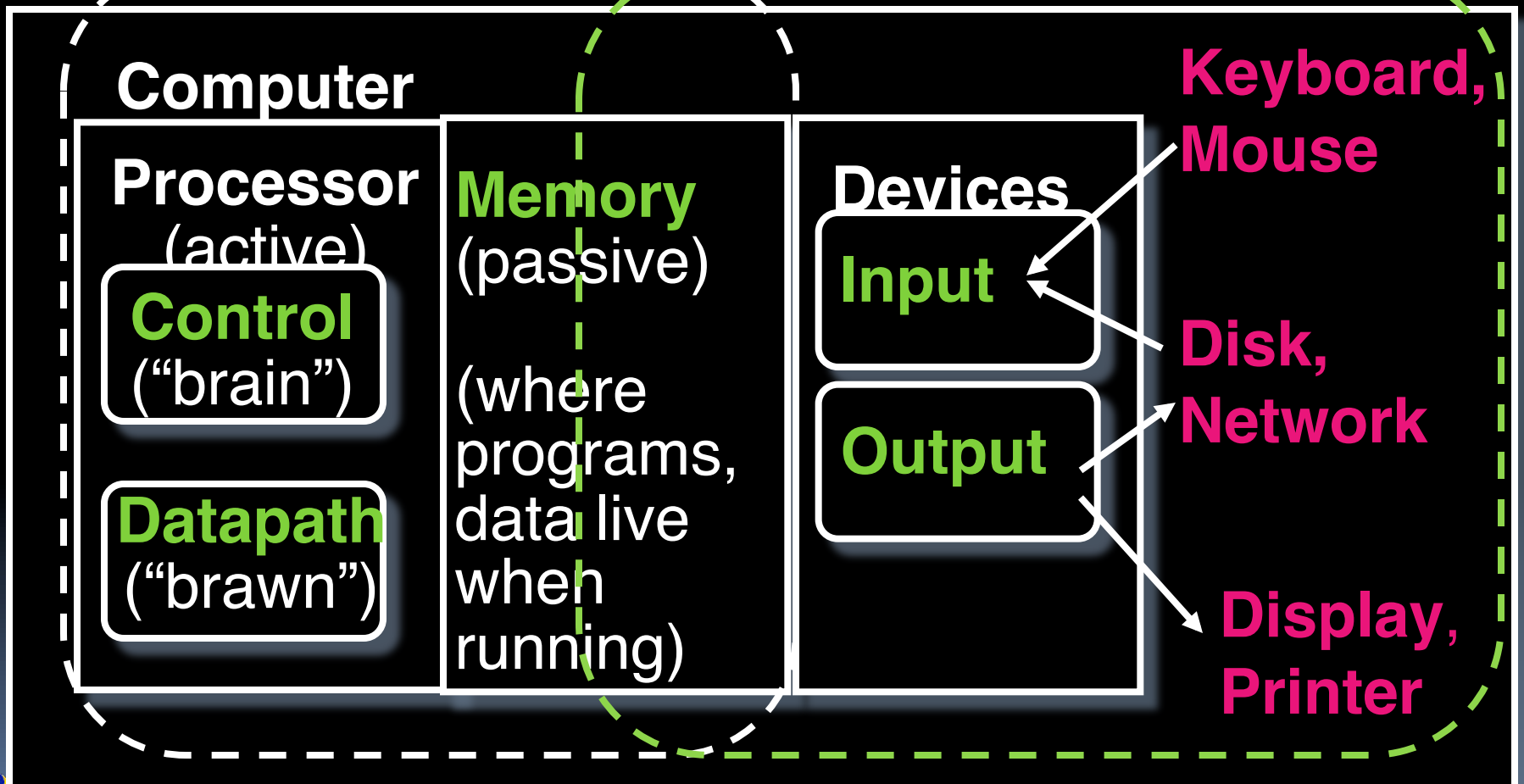
www.chrisharrison.net/projects/skinput/

I/O BASICS

Recall : 5 components of any Computer

Earlier Lectures

Current Lecture



Motivation for Input/Output

- I/O is how humans interact with computers
- I/O gives computers long-term memory.
- I/O lets computers do amazing things:



Read pressure of synthetic hand and control synthetic arm and hand of fireman



Control propellers, fins, communicate in BOB (Breathable Observable Bubble)

- Computer without I/O like a car w/no wheels; great technology, but gets you nowhere



I/O Device Examples and Speeds

- I/O Speed: bytes transferred per second
(from mouse to Gigabit LAN: **7 orders of mag!**)

Device	Behavior	Partner	Data Rate (KB/s)
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Voice output	Output	Human	5.00
Floppy disk	Storage	Machine	50.00
Laser Printer	Output	Human	100.00
Magnetic Disk	Storage	Machine	10,000.00
Wireless Network	I or O	Machine	10,000.00
Graphics Display	Output	Human	30,000.00
Wired LAN Network	I or O	Machine	125,000.00



When discussing transfer rates, use 10^x

Instruction Set Architecture for I/O

- **What must the processor do for I/O?**
 - Input: reads a sequence of bytes
 - Output: writes a sequence of bytes
- **Some processors have special input and output instructions**
- **Alternative model (used by MIPS):**
 - Use loads for input, stores for output
 - Called "Memory Mapped Input/Output"
 - A portion of the address space dedicated to communication paths to I/O devices (no mem there)
 - Instead, they correspond to registers in I/O devices



Processor-I/O Speed Mismatch

- 1GHz microprocessor can execute 1 billion load or store instructions per second, or 4,000,000 KB/s data rate
 - I/O devices data rates range from 0.01 KB/s to 125,000 KB/s
- **Input: device may not be ready to send data as fast as the processor loads it**
 - Also, might be waiting for human to act
- **Output: device not be ready to accept data as fast as processor stores it**
- **What to do?**



Processor Checks Status before Acting

- Path to device generally has 2 registers:
 - Control Register, says it's OK to read/write (I/O ready) [think of a flagman on a road]
 - Data Register, contains data
- Processor reads from Control Register in loop, spins while waiting for device to set Ready bit in Control reg (0 \Rightarrow 1) to say its OK
- Processor then loads from (input) or writes to (output) data register
 - Load from or Store into Data Register resets Ready bit (1 \Rightarrow 0) of Control Register
- This is called "**Polling**"



What is the alternative to polling?

- Wasteful to have processor spend most of its time “spin-waiting” for I/O to be ready
- Would like an unplanned procedure call that would be invoked only when I/O device is ready
- Solution: use **exception mechanism** to help I/O. **Interrupt** program when I/O ready, return when done with data transfer

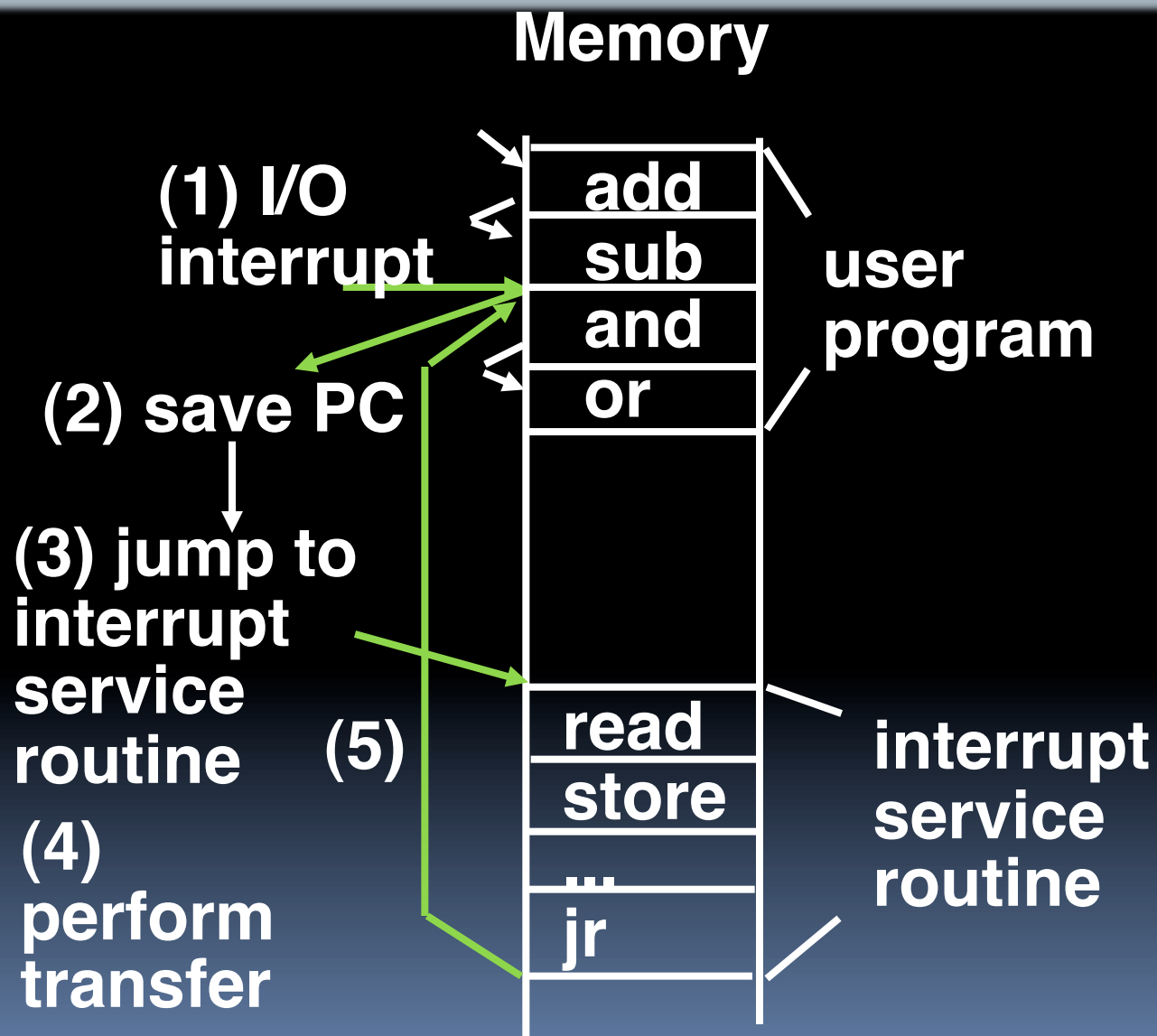


I/O Interrupt

- An I/O interrupt is like overflow exceptions except:
 - An I/O interrupt is “asynchronous”
 - More information needs to be conveyed
- An I/O interrupt is asynchronous with respect to instruction execution:
 - I/O interrupt is not associated with any instruction, but it can happen in the middle of any given instruction
 - I/O interrupt does not prevent any instruction from completion



Interrupt-Driven Data Transfer



Administrivia

- **Project 2 graded face-to-face, check web page for scheduling**
- **Project 3 (Cache simulator) out**
 - You may work in pairs for this project
- **Try the performance competition!**
 - You may work in pairs for this too
 - Do it for fun!
 - Do it to shine!
 - Do it to test your mettle!
 - Do it for EPA!



Upcoming Calendar

Week #	Mon	Wed	Thu Lab	Fri
#13 This week		I/O P3 out	VM	Performance
#14 Last week o' classes	Inter-machine Parallelism	Summary, Review, Evaluation	Parallel	Intra-machine Parallelism (Scott) P3 due
#15 RRR Week				Perf comp due 11:59pm
#16 Finals Week Review Sun May 9 3-6pm 10 Evans				Final Exam 8-11am in Hearst Gym



NETWORKS

The Internet (1962)

Founders

- JCR Licklider, as head of ARPA, writes on "intergalactic network"
- 1963 : ASCII becomes first universal computer standard
- 1969 : Defense Advanced Research Projects Agency (DARPA) deploys 4 "nodes" @ UCLA, SRI, Utah, & UCSB
- 1973 Robert Kahn & Vint Cerf invent TCP, now part of the Internet Protocol Suite

Internet growth rates

- Exponential since start!

The composite image includes a portrait of JCR Licklider, a diagram of the ARPANET network with nodes labeled SRI, UCSB, UCLA, and Utah, and an ASCII alphabet table. The diagram shows a central node 'SRI' connected to 'UCSB' and 'UCLA'. 'UCSB' is connected to '360' and '3'. 'UCLA' is connected to 'UCLA' and 'Sigma7'. 'Utah' is connected to 'PDP10'. The ASCII table is in the top right corner.

ASCII Alphabet			
A	1000001	N	1001110
B	1000010	O	1001111
C	1000011	P	1010000
D	1000100	Q	1010001
E	1000101	R	1010010
F	1000110	S	1010011
G	1000111	T	1010100
H	1001000	U	1010101
I	1001001	V	1010110
J	1001010	W	1010111
K	1001011	X	1011000
L	1001100	Y	1011001
M	1001101	Z	1011010

"Lick"

Vint Cerf

Revolutions like this don't come along very often

www.greatachievements.org/?id=3736

en.wikipedia.org/wiki/Internet_Protocol_Suite

Why Networks?

- Originally sharing I/O devices between computers
 - E.g., printers
- Then communicating between computers
 - E.g., file transfer protocol
- Then communicating between people
 - E.g., e-mail
- Then communicating between networks of computers
 - E.g., file sharing, www, ...



The World Wide Web (1989)

- “System of interlinked hypertext documents on the Internet”
- History
 - 1945: Vannevar Bush describes hypertext system called “memex” in article
 - 1989: Tim Berners-Lee proposes, gets system up ‘90
 - ~2000 Dot-com entrepreneurs rushed in, 2001 bubble burst
- **Wayback Machine**
 - Snapshots of web over time
- **Today : Access anywhere!**

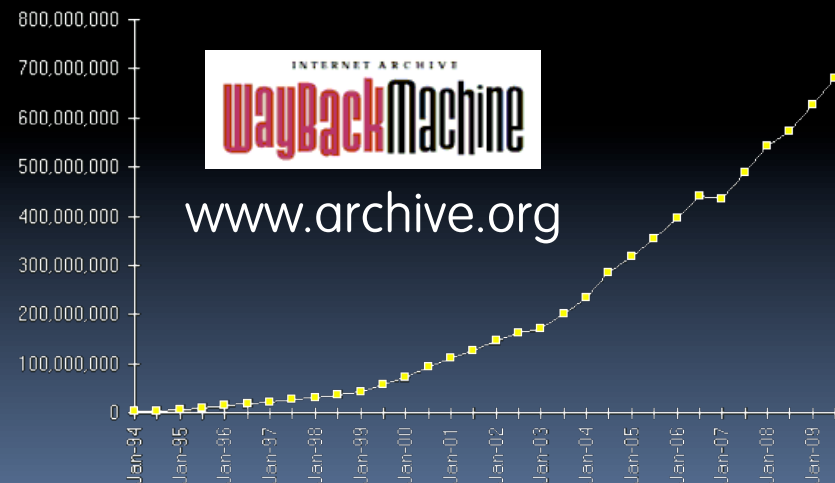


Tim Berners-Lee



World's First web server in 1990

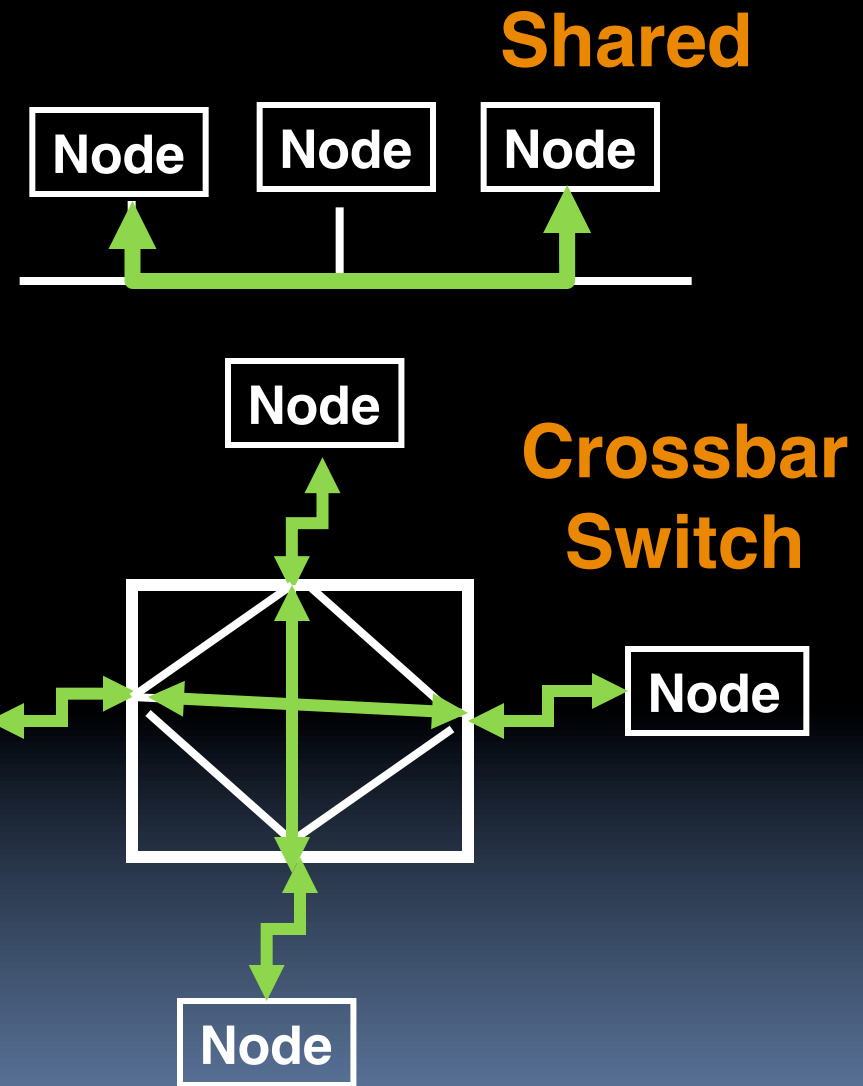
Internet Domain Survey Host Count



Source: Internet Systems Consortium (www.isc.org)

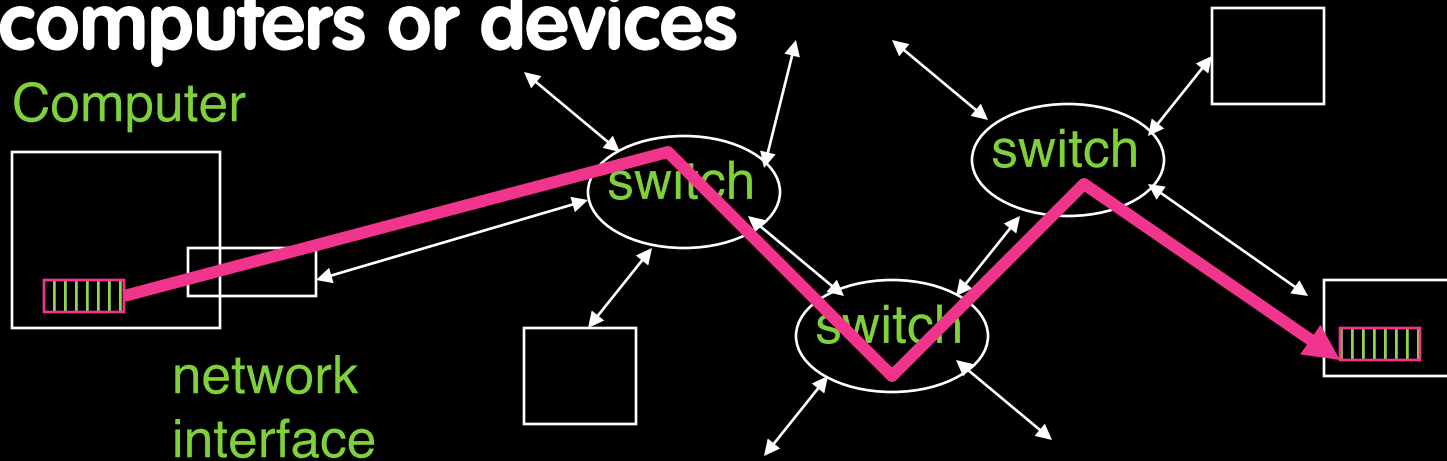
Shared vs. Switched Based Networks

- Shared vs. Switched:
 - Switched: pairs ("point-to-point" connections) communicate at same time
 - Shared: 1 at a time (CSMA/CD)
- Aggregate bandwidth (BW) in switched network is many times shared:
 - point-to-point faster since no arbitration, simpler interface



What makes networks work?

- **links** connecting **switches** to each other and to computers or devices



- ability to **name** the components and to **route** packets of information - messages - from a source to a destination

- Layering, redundancy, protocols, and encapsulation as means of **abstraction (61C big idea)**



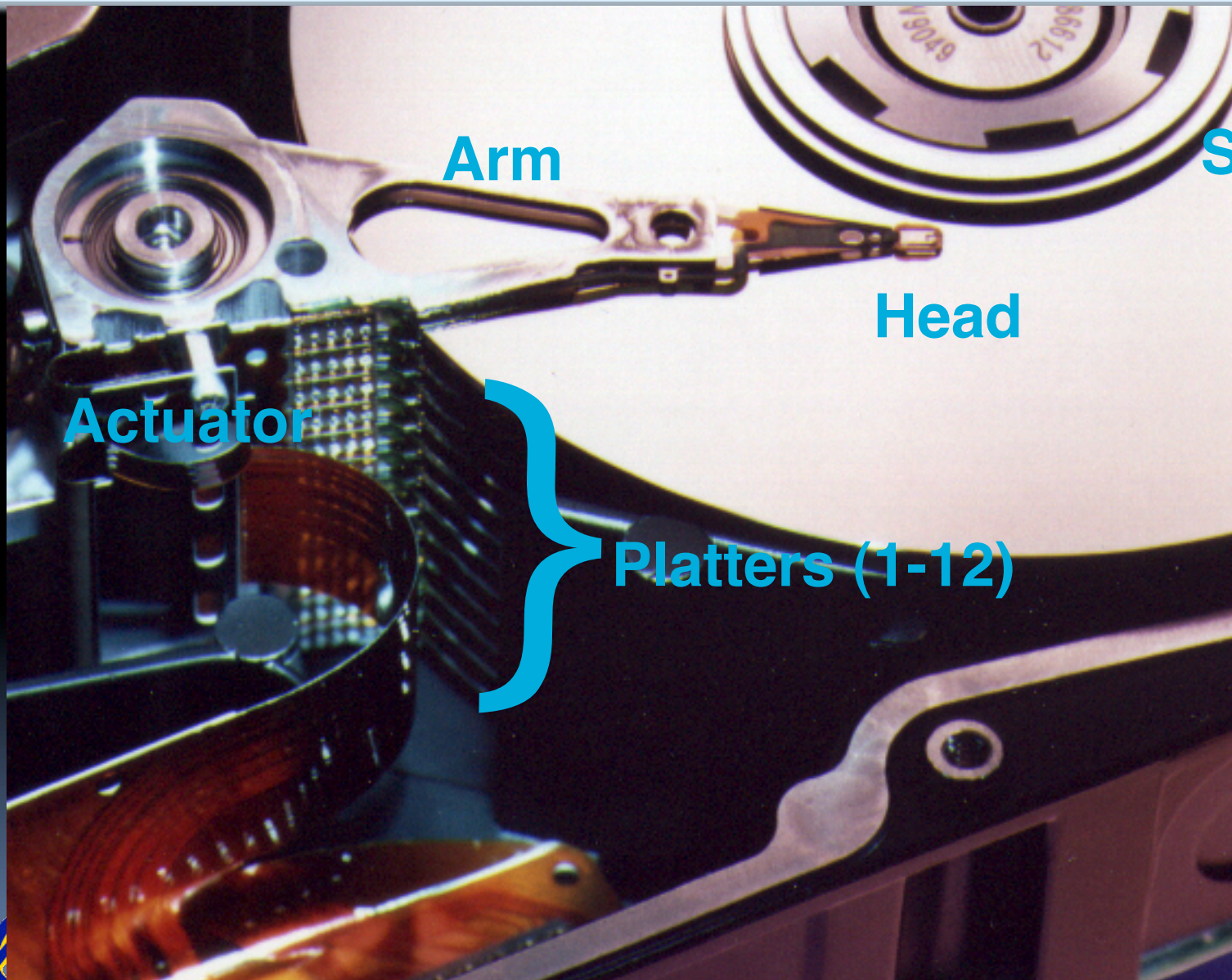
DISKS

Magnetic Disk – common I/O device

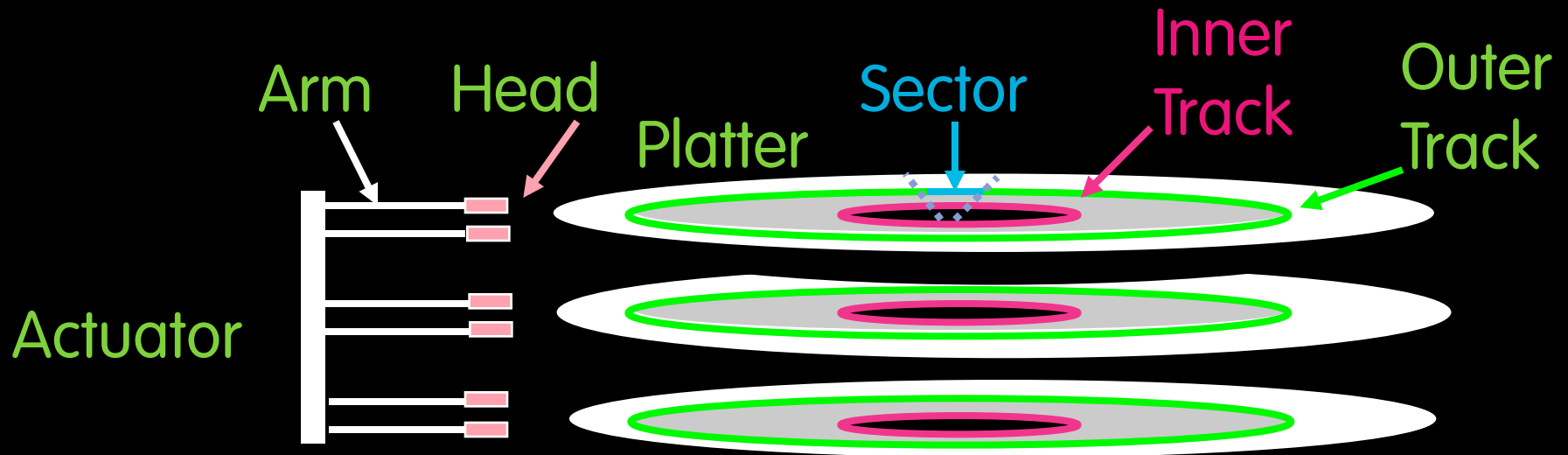
- **A kind of computer memory**
 - Information stored by magnetizing ferrite material on surface of rotating disk
 - similar to tape recorder except digital rather than analog data
- **Nonvolatile storage**
 - retains its value without applying power to disk.
- **Two Types**
 - Floppy disks – slower, less dense, removable.
 - Hard Disk Drives (HDD) – faster, more dense, non-removable.
- **Purpose in computer systems (Hard Drive):**
 - Long-term, inexpensive storage for files
 - “Backup” for main-memory. Large, inexpensive, slow level in the memory hierarchy (virtual memory)



Photo of Disk Head, Arm, Actuator



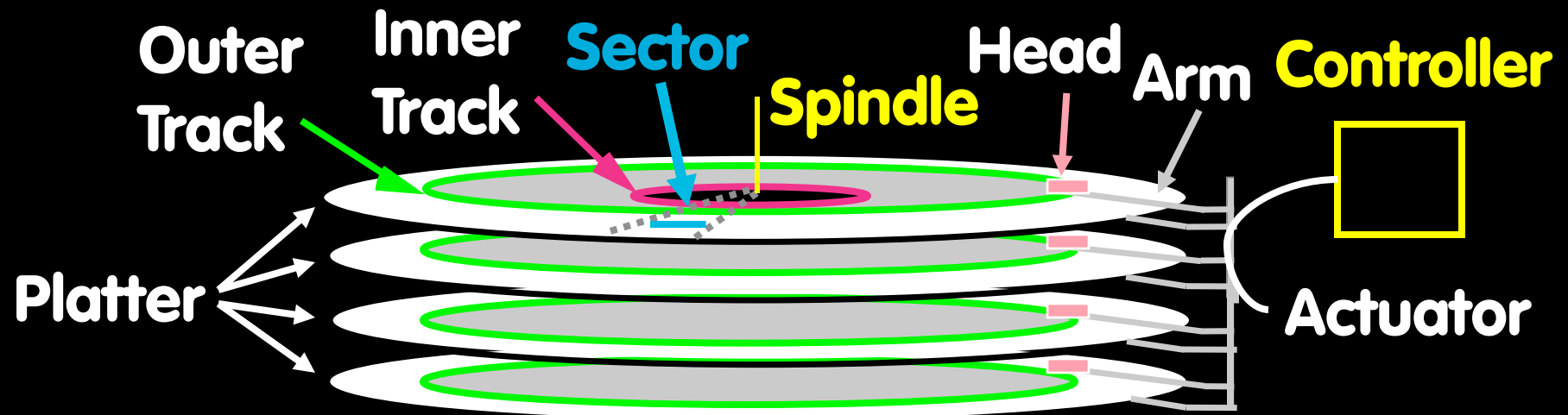
Disk Device Terminology



- 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) over track ("seek"), wait for sector rotate under head, then read or write



Disk Device Performance (1/2)



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

Disk Device Performance (2/2)

- **Average distance of sector from head?**
- **1/2 time of a rotation**
 - 7200 Revolutions Per Minute \Rightarrow 120 Rev/sec
 - 1 revolution = $1/120$ sec \Rightarrow 8.33 milliseconds
 - 1/2 rotation (revolution) \Rightarrow 4.17 ms
- **Average no. tracks to move arm?**
 - **Disk industry standard benchmark:**
 - Sum all time for all possible seek distances from all possible tracks / # possible
 - Assumes average seek distance is random
- **Size of Disk cache can strongly affect perf!**
 - Cache built into disk system, OS knows nothing



Where does Flash memory come in?

- **Microdrives and Flash memory (e.g., CompactFlash) are going head-to-head**
 - Both non-volatile (no power, data ok)
 - Flash benefits: durable & lower power (no moving parts, need to spin μ drives up/down)
 - Flash limitations: finite number of write cycles (wear on the insulating oxide layer around the charge storage mechanism). **Most $\geq 100K$, some $\geq 1M$ W/erase cycles.**
- **How does Flash memory work?**
 - NMOS transistor with an additional conductor between gate and source/drain which “traps” electrons. The presence/absence is a 1 or 0.



en.wikipedia.org/wiki/Flash_memory

What does Apple put in its iPods?

Toshiba flash
1, 2GB



Samsung flash
4, 8GB



Toshiba 1.8-inch HDD
80, 160GB



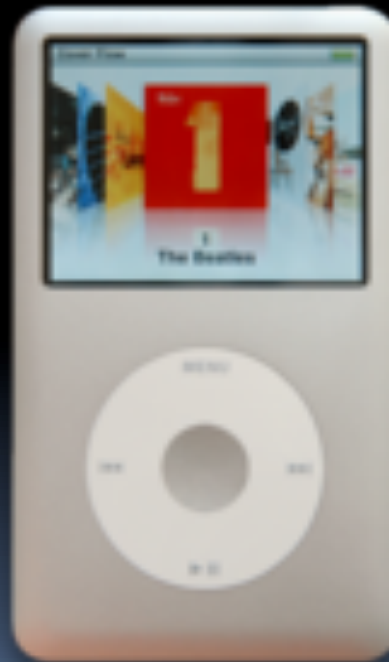
Toshiba flash
8, 16, 32GB



shuffle,



nano,



classic,



touch

RAID : Redundant Array of Inexpensive Disks

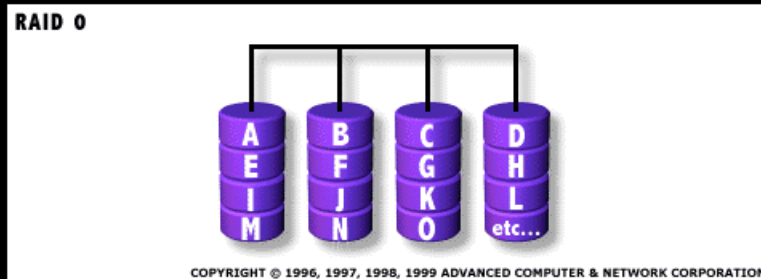
- Invented @ Berkeley (1989)
- A multi-billion industry
80% non-PC disks sold in RAIDs
- Idea:
 - Files are "striped" across multiple disks
 - Redundancy yields high data availability
 - Disks will still fail
 - Contents reconstructed from data redundantly stored in the array
 - ⇒ Capacity penalty to store redundant info
 - ⇒ Bandwidth penalty to update redundant info



Common RAID configurations

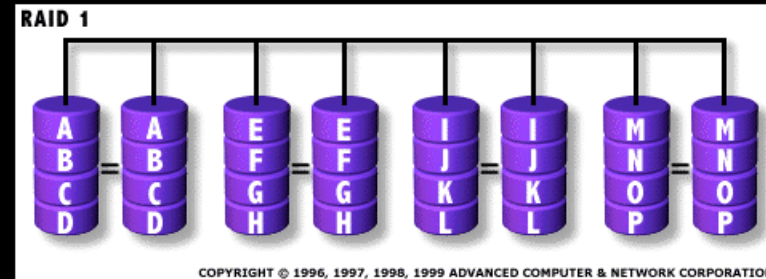
RAID 0

No redundancy, Fast access



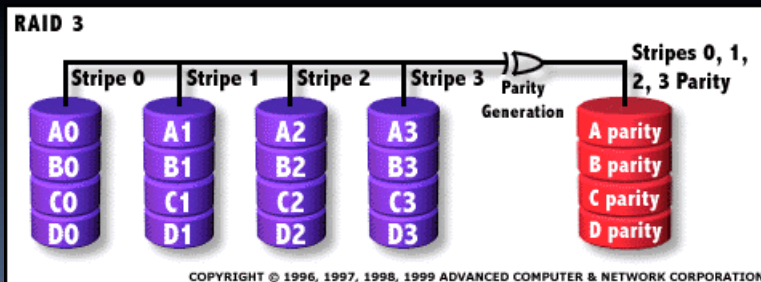
RAID 1

Mirror Data, most expensive sol'n



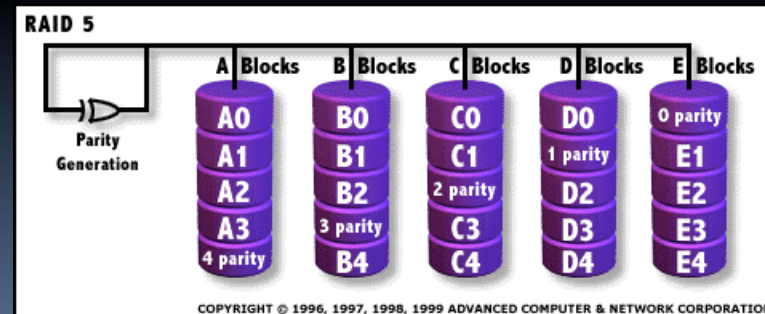
RAID 3

Parity drive protects against 1 failure



RAID 5

Rotated parity across all drives



“And in conclusion...”

- I/O gives computers their 5 senses
- I/O speed range is 100-million to one
- Processor speed means must synchronize with I/O devices before use
- Polling works, but expensive
 - processor repeatedly queries devices
- Interrupts works, more complex
 - devices causes an exception, causing OS to run and deal with the device
- I/O control leads to Operating Systems

