CS61C : Machine Structures

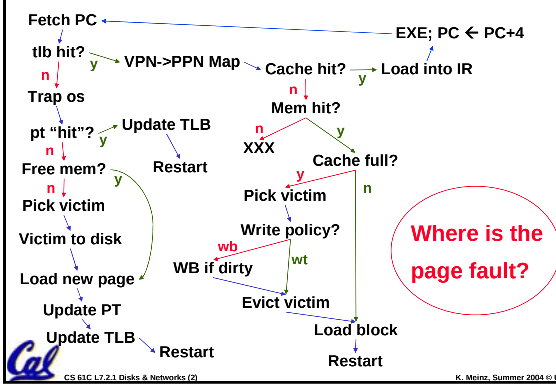
Lecture 7.2.1
Disks & Networks

2004-08-04

Kurt Meinz

inst.eecs.berkeley.edu/~cs61c

Cache, Proc and VM in IF



Administrative

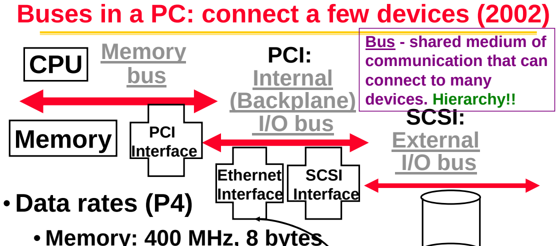
- Finish course material on Wed, Thurs.
- All next week will be review:
  - Review lectures (2 weeks/lecture)
  - No hw/labs*
    - Lab attendance still required. Checkoff points for showing up/finishing review material.*
- Schedule: P4 out tonight, MT3 on Friday, Final next Friday, P4 due next Sat*.

* Subject to change

Outline

- Buses
- Networks
- Disks
- RAID

Buses in a PC: connect a few devices (2002)



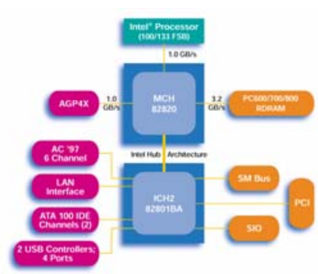
- Data rates (P4)
  - Memory: 400 MHz, 8 bytes ⇒ 3.2 GB/s (peak)
  - PCI: 100 MHz, 8 bytes wide ⇒ 0.8 GB/s (peak)
  - SCSI: "Ultra4" (160 MHz), "Wide" (2 bytes) ⇒ 0.3 GB/s (peak)

Ethernet Local Area Network

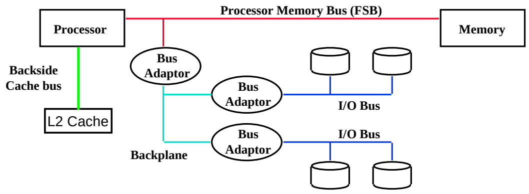
Gigabit Ethernet: ⇒ 0.125 GB/s (peak)

Main components of Intel Chipset: Pentium II/III

- Northbridge:
  - Handles memory
  - Graphics
- Southbridge: I/O
  - PCI bus
  - Disk controllers
  - USB controlers
  - Audio
  - Serial I/O
  - Interrupt controller
  - Timers



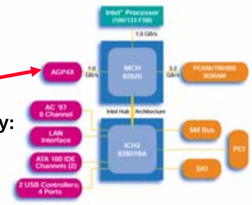
A Three-Bus System (+ backside cache)



- A small number of backplane buses tap into the processor-memory bus
  - FSB bus is only used for processor-memory traffic
  - I/O buses are connected to the backplane bus (PCI)
  - Advantage: load on the FSB is greatly reduced

What is DMA (Direct Memory Access)?

- Typical I/O devices must transfer large amounts of data to memory of processor:
  - Disk must transfer complete block
  - Large packets from network
  - Regions of frame buffer
- DMA gives external device ability to access memory directly:
  - much lower overhead than having processor request one word at a time.
- Issue: Cache coherence:
  - What if I/O devices write data that is currently in processor Cache?
    - The processor may never see new data!
  - Solutions:
    - Flush cache on every I/O operation (expensive)
    - Have hardware invalidate cache lines (remember "Coherence" cache misses?)



Outline

- Buses
- Networks
- Disks
- RAID

Why Networks?

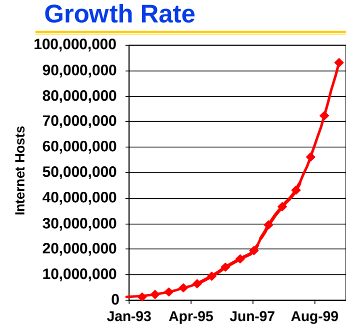
- Originally <u>sharing</u> I/O devices between computers
  (e.g., printers)
- Then Communicating <u>between</u> computers
  (e.g, file transfer protocol)
- Then Communicating <u>between</u> people
  (e.g., email)
- Then Communicating <u>between</u> networks of computers
  ⇒ File sharing, WWW, …

How Big is the Network (1999)?

~30	Computers in 273 Soda
~400	in inst.cs.berkeley.edu
~4,000	in eecs&cs .berkeley.edu
~50,000	in berkeley.edu
~5,000,000	in .edu
~46,000,000	in US (.com .net .edu .mil .us .org)
~56,000,000	in the world

Source: Internet Software Consortium

Growth Rate



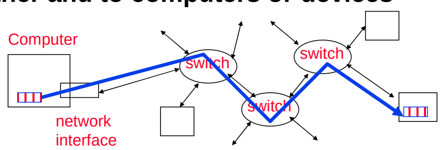
"Source: Internet Software Consortium (http://www.isc.org/)".

Ethernet Bandwidth

1983	3 mb/s
1990	10 mb/s
1997	100 mb/s
1999	1000 mb/s
2004	10 Gig E (to come!)

What makes networks work?

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



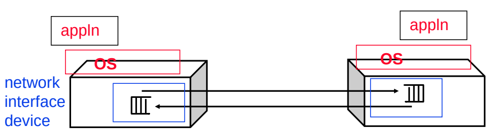
- **ability to name the components and to route packets of information - messages - from a source to a destination**
- **Layering, protocols, and encapsulation as means of abstraction (61C big idea)**

Typical Types of Networks

- **Local Area Network (Ethernet)**
  - **Inside a building: Up to 1 km**
  - **(peak) Data Rate: 10 Mbits/sec, 100 Mbits /sec,1000 Mbits/sec (1.25, 12.5, 125 MBytes/s)**
  - **Run, installed by network administrators**
- **Wide Area Network**
  - **Across a continent (10km to 10000 km)**
  - **(peak) Data Rate: 1.5 Mb/s to 10000 Mb/s**
  - **Run, installed by telecommunications companies (Sprint, UUNet[MCI], AT&T)**
- **Wireless Networks (LAN), ...**

ABCs of Networks: 2 Computers

- **Starting Point: Send bits between 2 computers**



- **Queue (First In First Out) on each end**
- **Can send both ways ("Full Duplex")**
- **Information sent called a "message"**
  - **Note: Messages also called packets**

A Simple Example: 2 Computers

- **What is Message Format?**
  - **Similar idea to Instruction Format**
  - **Fixed size? Number bits?**

Length	Data
8 bit	32 x Length bits

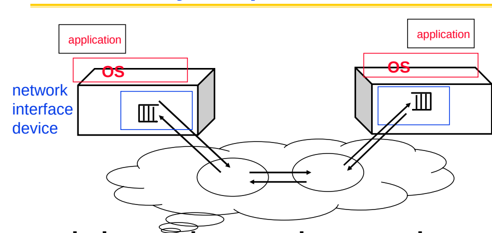
- **Header(Trailer): information to deliver message**
- **Payload: data in message**
- **What can be in the data?**
  - **anything that you can represent as bits**
  - **values, chars, commands, addresses...**

Questions About Simple Example

- **What if more than 2 computers want to communicate?**
  - **Need computer "address field" in packet to know which computer should receive it (destination), and to which computer it came from for reply (source) [just like envelopes!]**

Dest.	Source	Len	
Net ID	Net ID		CMD/ Address /Data
8 bits	8 bits	8 bits	32xn bits
Header			Payload

ABCs: many computers



- **switches and routers interpret the header in order to deliver the packet**
- **source encodes and destination decodes content of the payload**

Questions About Simple Example

- What if message is garbled in transit?
- Add redundant information that is checked when message arrives to be sure it is OK
- 8-bit sum of other bytes: called "Check sum"; upon arrival compare check sum to sum of rest of information in message

Math 55 talks about what a Check sum is…

Questions About Simple Example

- What if message never arrives?
- Receiver tells sender when it arrives (ack) [ala registered mail], sender retries if waits too long
- Don't discard message until get "ACK" (for ACKnowledgment); Also, if check sum fails, don't send ACK

Observations About Simple Example

- Simple questions such as those above lead to more complex procedures to send/receive message and more complex message formats
- Protocol: algorithm for properly sending and receiving messages (packets)

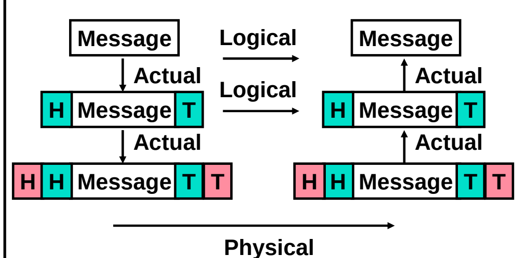
Software Protocol to Send and Receive

- SW Send steps
  1: Application copies data to OS buffer
  2: OS calculates checksum, starts timer
  3: OS sends data to network interface HW and says start
- SW Receive steps
  3: OS copies data from network interface HW to OS buffer
  2: OS calculates checksum, if OK, send ACK; if not, delete message (sender resends when timer expires)
  1: If OK, OS copies data to user address space, & signals application to continue

Protocol for Networks of Networks?

- Internetworking: allows computers on independent and incompatible networks to communicate reliably and efficiently;
  - Enabling technologies: SW standards that allow reliable communications without reliable networks
  - Hierarchy of SW layers, giving each layer responsibility for portion of overall communications task, called protocol families or protocol suites
- Abstraction to cope with complexity of communication vs. Abstraction for complexity of computation

Protocol Family Concept



Protocol Family Concept

- Key to **protocol families** is that communication occurs **logically** at the same level of the protocol, called **peer-to-peer**…

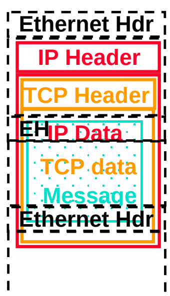
  …but is **implemented via services at the next lower level**
- **Encapsulation:** carry higher level information within lower level "envelope"
- **Fragmentation:** break packet into multiple smaller packets and reassemble

Protocol for Network of Networks

- **Transmission Control Protocol/Internet Protocol (TCP/IP)**
  - This protocol family is the basis of the Internet, a WAN protocol
  - IP makes best effort to deliver
  - TCP guarantees delivery
  - TCP/IP so popular it is used even when communicating locally: even across homogeneous LAN

TCP/IP packet, Ethernet packet, protocols

- Application sends message
- TCP breaks into 64KB segments, adds 20B header
- IP adds 20B header, sends to network
- If Ethernet, broken into 1500B packets with headers, trailers (24B)
- All Headers, trailers have length field, destination, ...

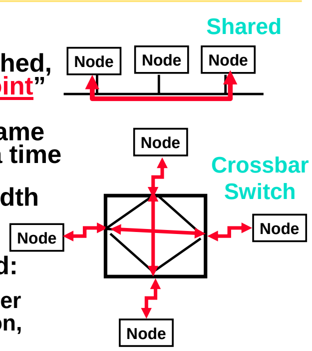


Overhead vs. Bandwidth

- Networks are typically advertised using peak bandwidth of network link: e.g., 100 Mbits/sec Ethernet ("100 base T")
- Software overhead to put message into network or get message out of network often limits useful bandwidth
- Assume overhead to send and receive = 320 microseconds (μs), want to send 1000 Bytes over "100 Mbit/s" Ethernet
  - Network transmission time: 1000Bx8b/B /100Mb/s = 8000b / (100b/μs) = 80 μs
  - Effective bandwidth: 8000b/(320+80)μs = 20 Mb/s

Shared vs. Switched Based Networks

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



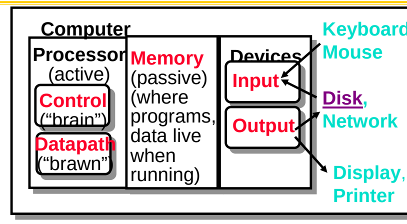
And in conclusion…

- Protocol suites allow heterogeneous networking
  - Another form of principle of abstraction
  - Protocols ⇒ operation in presence of failures
  - Standardization key for LAN, WAN
- Integrated circuit ("Moore's Law") revolutionizing network switches as well as processors
  - Switch just a specialized computer
- Trend from shared to switched networks to get faster links and scalable bandwidth

Outline

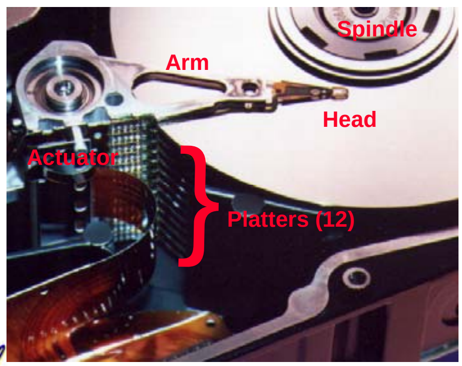
- Buses
- Networks
- Disks
- RAID

Magnetic Disks

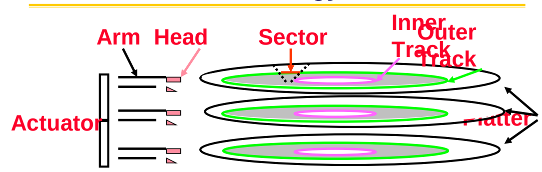


- Purpose:
  - Long-term, nonvolatile, inexpensive storage for files
  - Large, inexpensive, slow level in the memory hierarchy (discuss later)

Photo of Disk Head, Arm, Actuator

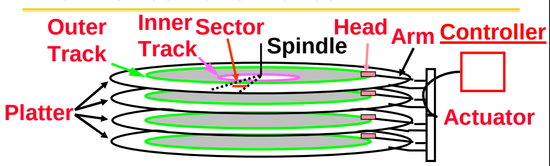


Disk Device Terminology



- 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



- 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

Data Rate: Inner vs. Outer Tracks

- To keep things simple, originally same # of sectors/track
  - Since outer track longer, lower bits per inch
- Competition decided to keep bits/inch (BPI) high for all tracks ("constant bit density")
  - More capacity per disk
  - More sectors per track towards edge
  - Since disk spins at constant speed, outer tracks have faster data rate
- Bandwidth outer track 1.7X inner track!

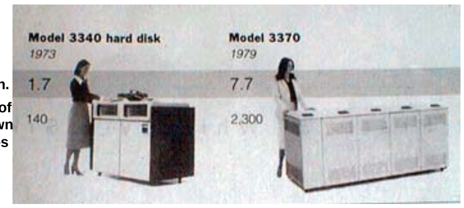
Disk Performance Model /Trends

- Capacity : + 100% / year (2X / 1.0 yrs)
  Over time, grown so fast that # of platters has reduced (some even use only 1 now!)
- Transfer rate (BW) : + 40%/yr (2X / 2 yrs)
- Rotation+Seek time : – 8%/yr (1/2 in 10 yrs)
- Areal Density
  - Bits recorded along a track: Bits/Inch (BPI)
  - # of tracks per surface: Tracks/Inch (TPI)
  - We care about bit density per unit area Bits/Inch^2
  - Called Areal Density = BPI x TPI
- MB/$: > 100%/year (2X / 1.0 yrs)
  - Fewer chips + areal density

Disk History (IBM)

Data density Mbit/sq. in.

Capacity of Unit Shown Megabytes



1973:
1. 7 Mbit/sq. in
0.14 GBytes

1979:
7. 7 Mbit/sq. in
2.3 GBytes

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 GBytes

1997:
1450 Mbit/sq. in
2.3 GBytes

1997:
3090 Mbit/sq. in
8.1 GBytes

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

Modern Disks: Barracuda 7200.7 (2004)

- 200 GB, 3.5-inch disk
- 7200 RPM; Serial ATA
- 2 platters, 4 surfaces
- 8 watts (idle)
- 8.5 ms avg. seek
- 32 to 58 MB/s Xfer rate
- $125 = $0.625 / GB

source: www.seagate.com;

Modern Disks: Mini Disks

- 2004 Toshiba Minidrive:
  - 2.1" x 3.1" x 0.3"
  - 40 GB, 4200 RPM, 31 MB/s, 12 ms seek
  - 20GB/inch^3 !!
  - Mp3 Players

Modern Disks: 1 inch disk drive!

- 2004 Hitachi Microdrive:
  - 1.7" x 1.4" x 0.2"
  - 4 GB, 3600 RPM, 4-7 MB/s, 12 ms seek
  - 8.4 GB/inch^3
  - Digital cameras, PalmPC
- 2006 MicroDrive?
  - 16 GB, 10 MB/s!
  - Assuming past trends continue

Modern Disks: 1 inch disk drive!

- Not magnetic but …

- 1gig Secure digital
  - Solid State NAND Flash
  - 1.2” x 0.9” x 0.08 (!!)
  - 11.6 GB/inch^3

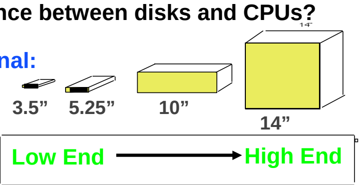
Outline

- Buses
- Networks
- Disks
- RAID

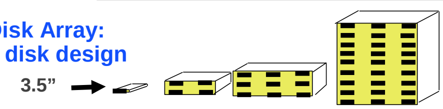
Use Arrays of Small Disks…

- Katz and Patterson asked in 1987:
  - Can smaller disks be used to close gap in performance between disks and CPUs?

Conventional: 4 disk designs



Disk Array: 1 disk design



Replace Small Number of Large Disks with Large Number of Small Disks! (1988 Disks)

	IBM 3390K	IBM 3.5" 0061	x70	
Capacity	20 GBytes	320 MBytes	23 GBytes	
Volume	97 cu. ft.	0.1 cu. ft.	11 cu. ft.	9X
Power	3 KW	11 W	1 KW	3X
Data Rate	15 MB/s	1.5 MB/s	120 MB/s	8X
I/O Rate	600 I/Os/s	55 I/Os/s	3900 IOs/s	6X
MTTF	250 KHrs	50 KHrs	??? Hrs	
Cost	$250K	$2K	$150K	

Disk Arrays potentially high performance, high MB per cu. ft., high MB per KW, <u>but what about reliability?</u>

Array Reliability

- <u>Reliability</u> - whether or not a component has failed
  - measured as Mean Time To Failure (MTTF)
- Reliability of N disks = Reliability of 1 Disk ÷ N (assuming failures independent)
  - 50,000 Hours ÷ 70 disks = 700 hour
- Disk system MTTF: Drops from 6 years to 1 month!
- Disk arrays too unreliable to be useful!

<u>Redundant</u> Arrays of (Inexpensive) Disks

- Files are "striped" across multiple disks
- Redundancy yields high data availability
  - <u>Availability</u>: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array

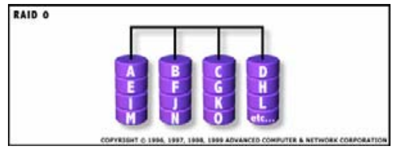
  ⇒ Capacity penalty to store redundant info

  ⇒ Bandwidth penalty to update redundant info

Berkeley History, RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software
- Today RAID is $27 billion dollar industry, 80% nonPC disks sold in RAIDs

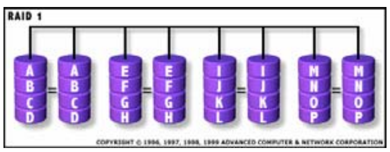
"RAID 0": Striping



- Assume have 4 disks of data for this example, organized in blocks
- Large accesses faster since transfer from several disks at once

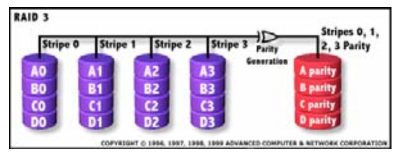
This and next 5 slides from RAID.edu, http://www.acnc.com/04_01_00.html

RAID 1: Mirror



- Each disk is fully duplicated onto its "mirror"
  - Very high availability can be achieved
- Bandwidth reduced on write:
  - 1 Logical write = 2 physical writes
- Most expensive solution: 100% capacity overhead

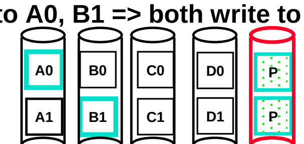
RAID 3: Parity



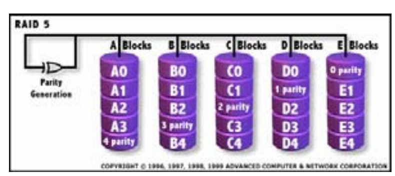
- Parity computed across group to protect against hard disk failures, stored in P disk
- Logically, a single high capacity, high transfer rate disk
- 25% capacity cost for parity in this example vs. 100% for RAID 1 (5 disks vs. 8 disks)

Inspiration for RAID 5

- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk (access all disks)
  - Option 2: since P has old sum, compare old data to new data, add the difference to P: 1 logical write = 2 physical reads + 2 physical writes to 2 disks
- Parity Disk is bottleneck for Small writes: Write to A0, B1 => both write to P disk



RAID 5: Rotated Parity, faster small writes



- Independent writes possible because of interleaved parity
  - Example: write to A0, B1 uses disks 0, 1, 4, 5, so can proceed in parallel
  - Still 1 small write = 4 physical disk accesses

Magnetic Disk Summary

- Magnetic Disks continue rapid advance: 60%/yr capacity, 40%/yr bandwidth, slow on seek, rotation improvements, MB/$ improving 100%/yr?
  - Designs to fit high volume form factor
- RAID
  - Higher performance with more disk arms per $
  - Adds option for small # of extra disks
  - Today RAID is > $27 billion dollar industry, 80% nonPC disks sold in RAIDs; started at Cal