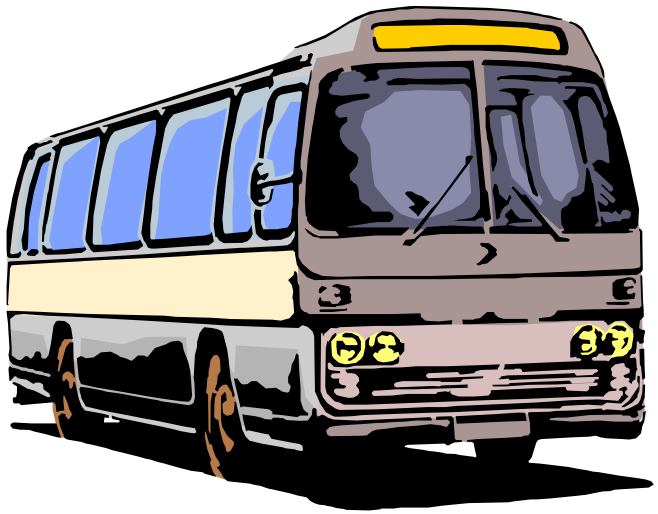
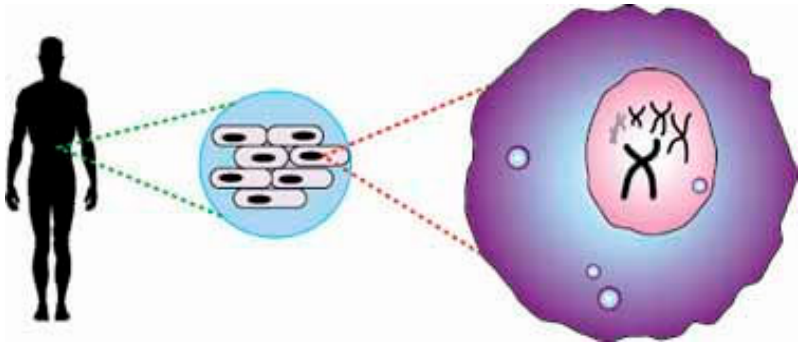
Introduction to I/O

- Where does the data for our CPU and memory come from or go to?
- Computers communicate with the outside world via I/O devices.
  - Input devices supply computers with data to operate on.
  - Results of computations can be sent to output devices.
- Today we'll talk a bit about I/O system issues.
  - I/O performance affects the overall system speed.
  - We'll look at some common devices and estimate their performance.
  - A bus connects different components of a computer system.



I/O is important!



- Many tasks involve reading and processing enormous quantities of data.
  - CCSO has two machines and 144GB of storage for a local web cache.
  - Institutions like banks and airlines have huge databases that must be constantly accessed and updated.
  - Celera Genomics is a company that sequences genomes, with the help of computers and 100 trillion bytes of storage!
- I/O is important for us small people too!
  - People use home computers to edit movies and music.
  - Large software packages may come on multiple compact discs.
  - Everybody and their grandparents surf the web!

I/O is slow!

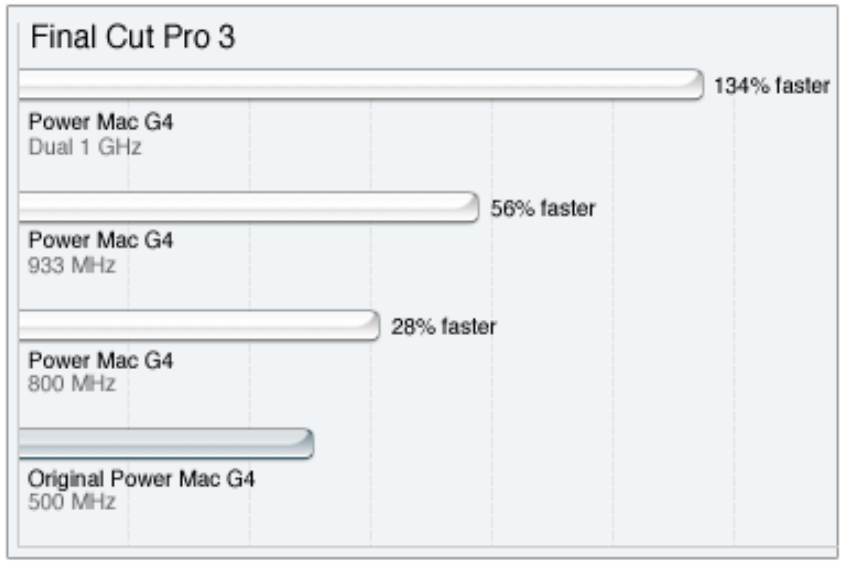
- How fast can a typical I/O device supply data to a computer?
  - A fast typist can enter 9-10 characters a second on a keyboard.
  - Common local-area network (LAN) speeds go up to 100 Mbit/s, which is about 12.5MB/s.
  - Today's hard disks provide a lot of storage and transfer speeds around 40MB per second.
- Unfortunately, this is excruciatingly slow compared to modern processors that can execute a billion instructions per second!
  - I/O performance has not increased as quickly as CPU performance, partially due to neglect and partially to physical limitations.
  - This is slowly changing, with faster networks, better I/O buses, RAID drive arrays, and other new technologies.

I/O speeds often limit system performance

- Many computing tasks are I/O-bound, and the speed of the input and output devices limits the overall system performance.
- This is another instance of Amdahl's Law. Improved CPU performance alone has a limited effect on overall system speed.

$$\text{Execution time after improvement} = \frac{\text{Time affected by improvement}}{\text{Amount of improvement}} + \text{Time unaffected by improvement}$$

- An old graph from Apple illustrates this pretty well; a system with *two* 1 GHz CPUs is just 50% faster than a machine with a single 933 MHz processor.

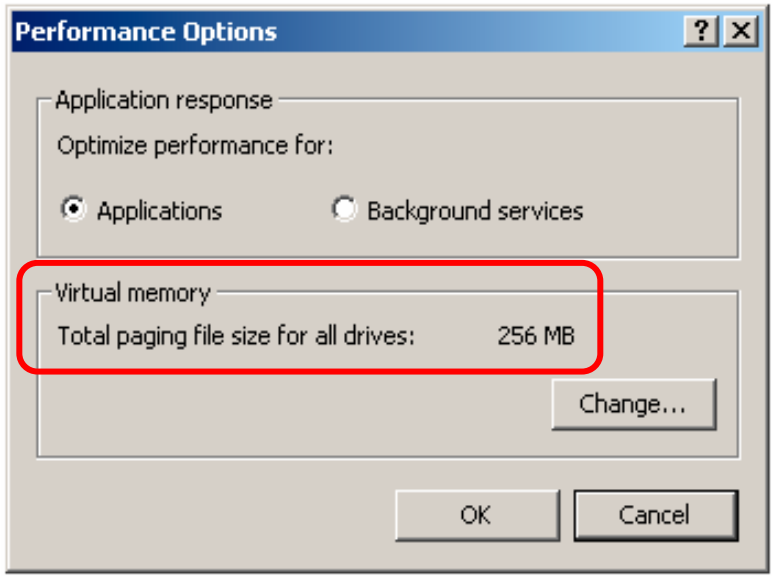


Measuring I/O performance

- There are three general performance measurements for I/O systems.
  - An application that accesses large quantities of data will demand high bandwidth or transfer speed.
  - Programs that access small amounts of data in frequent intervals may care more about the latency or delay.
  - Throughput, which is the number of transactions performed per unit time, accounts for latency, bandwidth and overhead times.
- Home network users can be affected by both bandwidth and latency.
  - If you download large files over Kazaa, bandwidth will be the limiting factor; a 4Mbit/s DSL line will outperform a 56Kbit/s modem.
  - If you send short instant messages, the latency becomes the limiting factor—it takes a long time for data to reach Swaziland.

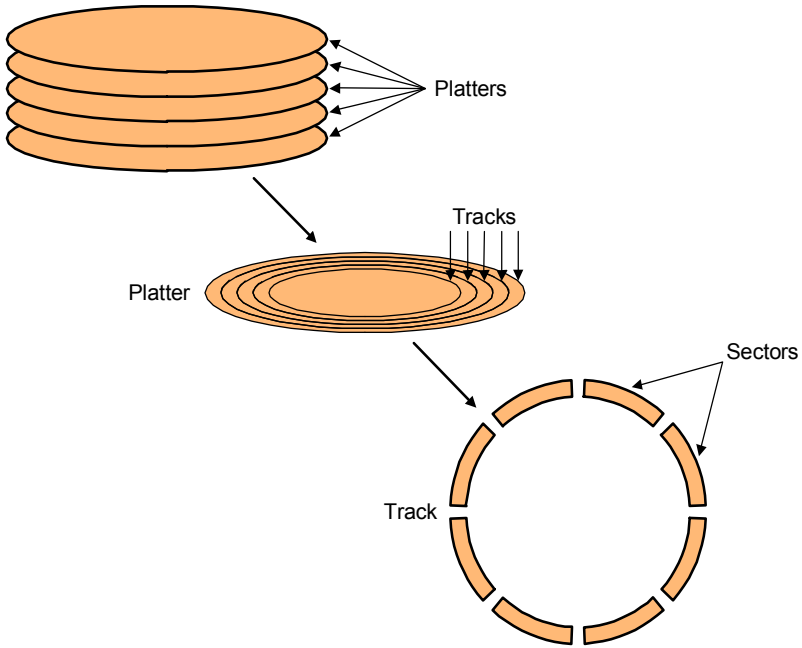
Common I/O devices

- Hard drives are almost a necessity these days, so their speed has a big impact on system performance.
  - They store all the programs, movies and assignments you crave.
  - Virtual memory systems let a hard disk act as a large (but slow) part of main memory.
- Networks are also ubiquitous nowadays.
  - They give you access to data from around the world.
  - Hard disks can act as a cache for network data. For example, web browsers often store local copies of recently viewed web pages.



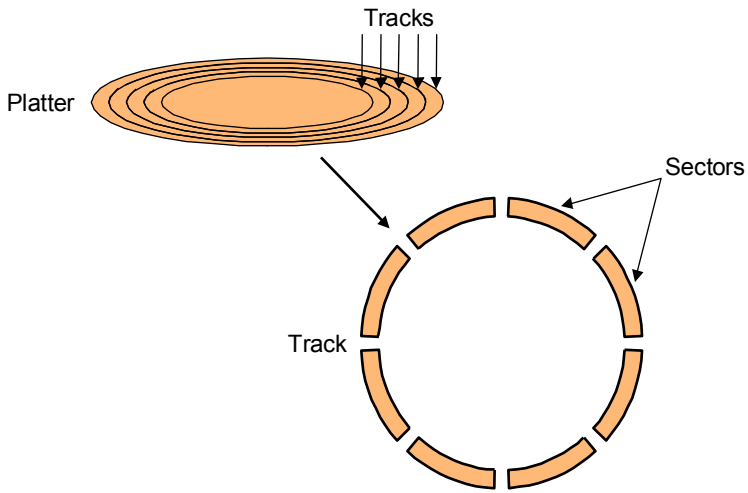
Hard drives

- Figure 8.4 in the textbook shows the ugly guts of a hard disk.
  - Data is stored on double-sided magnetic disks called platters.
  - Each platter is arranged like a record, with many concentric tracks.
  - Tracks are further divided into individual sectors, which are the basic unit of data transfer.
  - Each surface has a read/write head like the arm on a record player, but all the heads are connected and move together.
- A 75GB IBM Deskstar has roughly:
  - 5 platters (10 surfaces),
  - 27,000 tracks per surface,
  - 512 sectors per track, and
  - 512 bytes per sector.



Accessing data on a hard disk

- Accessing a sector on a track on a hard disk takes a lot of time!
  - Seek time measures the delay for the disk head to reach the track.
  - A rotational delay accounts for the time to get to the right sector.
  - The transfer time is how long the actual data read or write takes.
  - There may be additional overhead for the operating system or the controller hardware on the hard disk drive.
- Rotational speed, measured in revolutions per minute or RPM, partially determines the rotational delay and transfer time.



Estimating disk latencies

- Manufacturers often report *average* seek times of 8-10ms, but those do not account for common disk access patterns.
  - For example, if the head is already on or near the desired track, then seek time is negligible. In other words, locality is important!
  - Actual average seek times are often just 2-3ms.
- Rotational delay depends partly on how fast the disk platters spin.
  - We can assume the disk spins halfway on average.

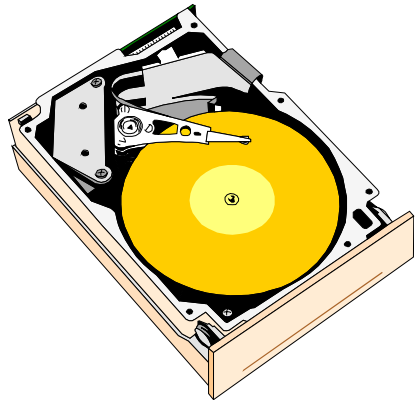
$$\text{average rotational delay} = 0.5 \times \text{rotations} / \text{rotational speed}$$

  - For example, a 5400 RPM disk has an average rotational delay of:

$$0.5 \text{ rotations} / (5400 \text{ rotations/minute}) = 5.55\text{ms}$$

Estimating disk times

- The overall response time is the sum of the seek time, rotational delay, transfer time and overhead.
- Assume a disk has the following specifications.
  - An average seek time of 9ms
  - A 5400 RPM rotational speed
  - A 10MB/s average transfer rate
  - 2ms of overheads

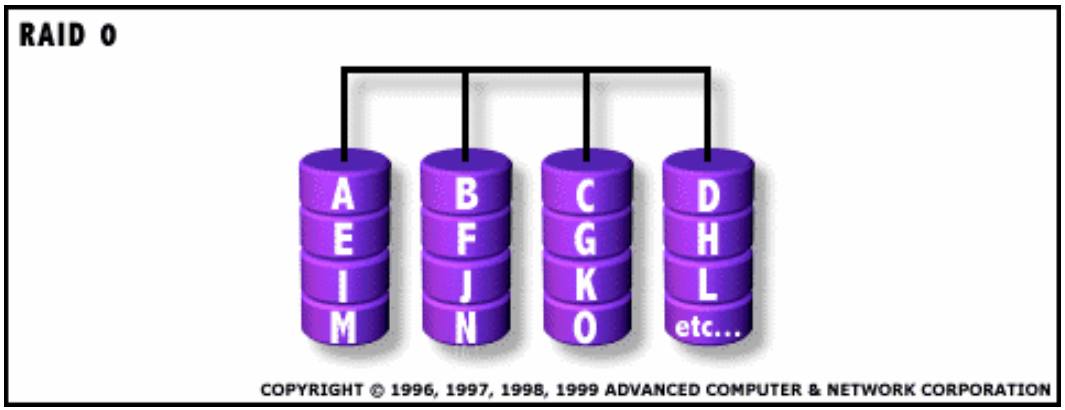


- How long does it take to read a 1,024 byte sector?
  - The average rotational delay is 5.55ms.
  - The transfer time will be about (1024 bytes / 10 MB/s) = 0.1ms.
  - The response time is then 9ms + 5.55ms + 0.1ms + 2ms = 16.7ms. That's 16,700,000 cycles for a 1GHz processor!
- One possible measure of throughput would be the number of random sectors that can be read in one second.

(1 sector / 16.7ms) × (1000ms / 1s) = 60 sectors/second.

Parallel I/O

- Many hardware systems use parallelism for increased speed.
  - Pipelined and superscalar processors include extra hardware so they can execute multiple instructions simultaneously.
  - Dividing memory into banks lets us access several words at once.
- A redundant array of inexpensive disks or RAID system allows access to several hard drives at once, for increased bandwidth.
  - The picture below shows a single data file with fifteen sectors denoted A-O, which are "striped" across four disks.
  - This is reminiscent of interleaved main memories from last week.



Networks

- Typical networks have even lower transfer speeds than hard disks.
  - Home PCs with cable modems get around 1-2MB/s, but hard disks can usually transfer at least 10MB/s.
  - People often save long downloads to their local hard disk, so the disk is serving as a cache for the network.
- Latency is a very important issue in networking. It takes time for bits to travel across states, countries and oceans!

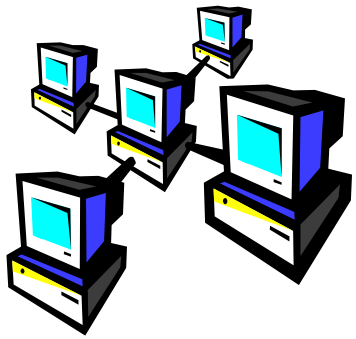
```
>ping www.uiuc.edu
Approximate round trip times in milli-seconds:
    Minimum = 104ms, Maximum =  115ms, Average =  112ms

>ping www.stanford.edu
Approximate round trip times in milli-seconds:
    Minimum = 160ms, Maximum =  170ms, Average =  164ms

>ping nus.edu.sg
Approximate round trip times in milli-seconds:
    Minimum = 410ms, Maximum =  437ms, Average =  420ms
```

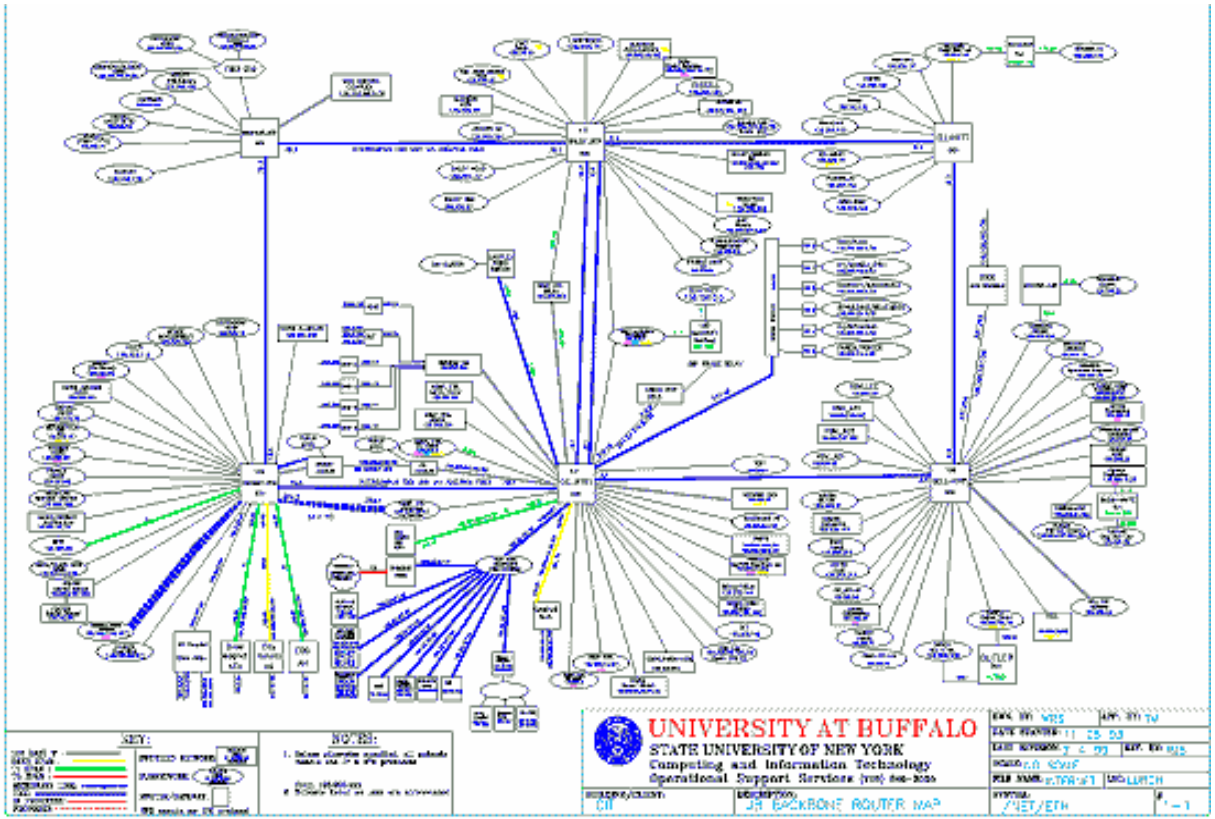
Network topologies

- The arrangement of a network affects performance.
- A linear topology connects machines in sequence.
  - Only one machine at a time may send data.
  - It takes a long time to get to the other end.
  - If one computer or link fails, the entire network goes down.
- In a star topology, all machines are connected to a central hub.
  - This can reduce the distance between any two computers on the network.
  - It also minimizes the effect of a failed node or link on other machines.



Tree topology

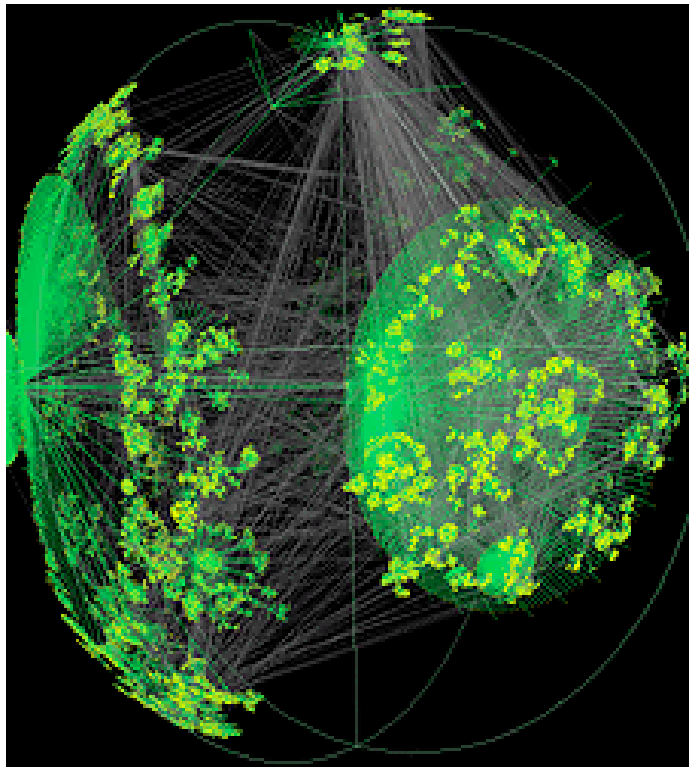
- A tree topology is also popular.
  - Individual units represent one leaf or one small subtree.
  - Larger subtrees may be complete departments or divisions.
  - The entire organization connects all of these departments.



(www.cybergeography.org)

The Internet

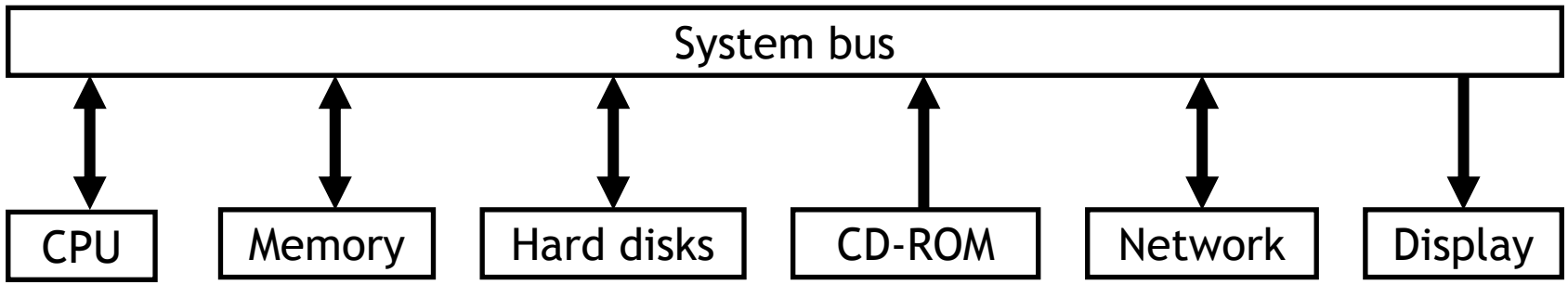
- The Internet is just one big graph.
  - There are many possible paths between any two machines.
  - This redundancy increases reliability, and data can also be sent along multiple paths for higher speed.
- UIUC accesses the Internet through several vendors, for both increased bandwidth and reliability.



(www.caida.org)

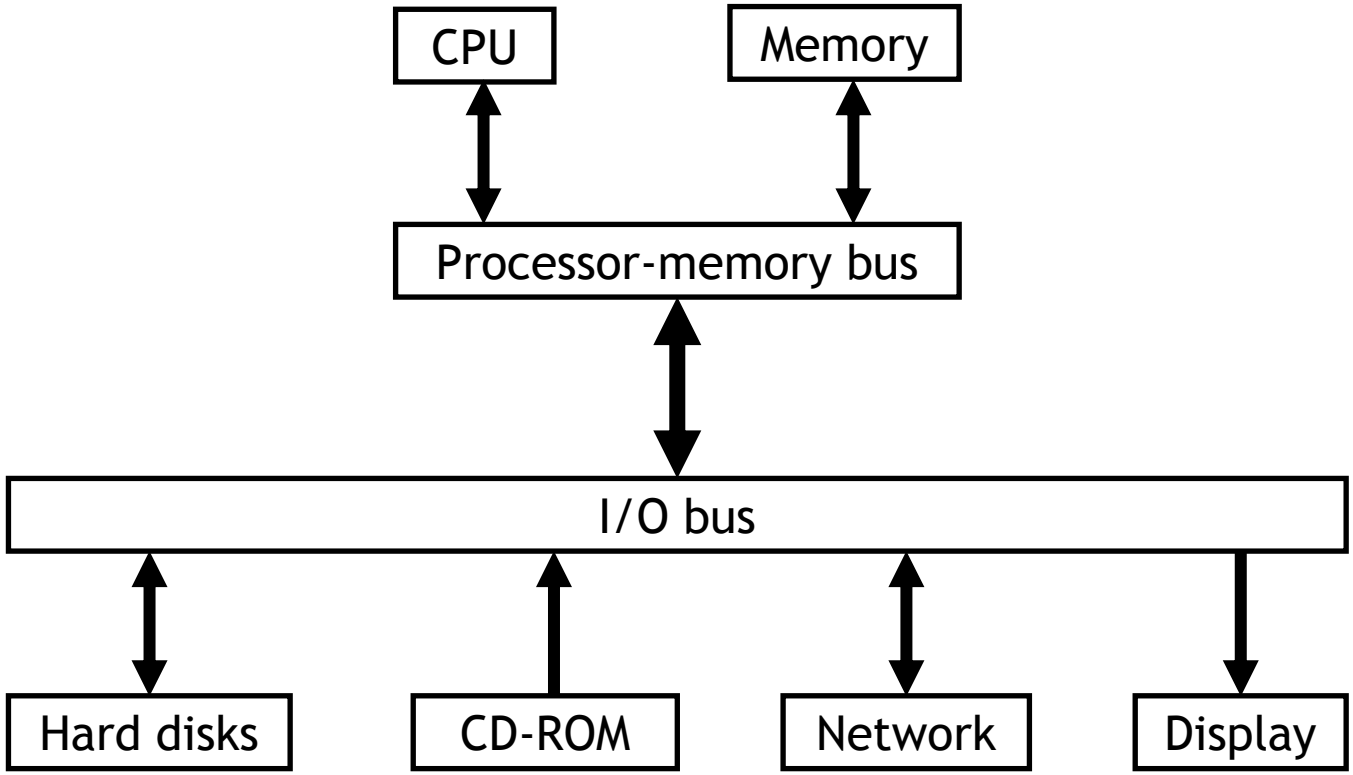
Computer buses

- Coming back to our world, each computer has several small "networks" inside, called buses, to connect processors, memory, and I/O devices.
- The simplest kind of bus is linear, as shown below.
  - All devices share the same bus.
  - Only one device at a time may transfer data on the bus.
- Simple is not always good!
  - With many devices, there might be a lot of contention.
  - The distance from one end of the bus to the other may also be relatively long, increasing latencies.



Hierarchical buses

- We could split the bus into different segments.
  - Since the CPU and memory need to communicate so often, a shorter and faster processor-memory bus can be dedicated to them.
  - A separate I/O bus would connect the slower devices to each other, and eventually to the processor.

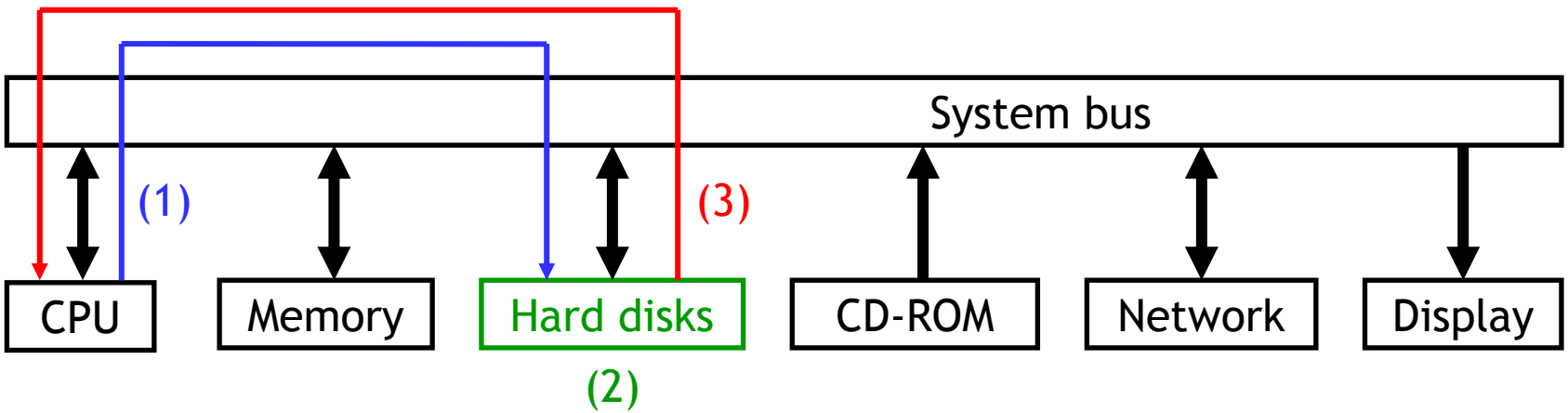


Synchronous and asynchronous buses

- A synchronous bus operates with a central clock signal.
  - Bus transactions can be handled easily with finite state machines.
  - However, the clock rate and bus length are inversely proportional; faster clocks mean less time for data to travel. This is one reason why PCs never have more than about five expansion slots.
  - All devices on the bus must run at the same speed, even if they are capable of going faster.
- An asynchronous bus does not rely on clock signals.
  - Bus transactions rely on complicated handshaking protocols so each device can determine when other ones are available or ready.
  - On the other hand, the bus can be longer and individual devices can operate at different speeds.
  - Many external buses like USB and Firewire are asynchronous.

Basic bus protocols

- It's convenient to think of I/O operations as memory operations.
  - The CPU reads from input devices like keyboards and CD-ROMs, and writes to output devices like monitors and printers.
  - An address specifies the exact sector, pixel, host, etc. to access.
- Using these terms, two devices can communicate over a bus as follows.
  1. An initiator sends an address and possibly data to a target.
  2. The target processes the request by "reading" or "writing" data.
  3. The target sends a reply over the bus back to the initiator.
- The bus width limits the number of bits transferred per cycle.



Example bus problems

- I/O problems always start with some assumptions about a system.
  - A CPU and memory share a 32-bit bus running at 100MHz.
  - The memory needs 50ns to access a 64-bit value from one address.
- Then, questions generally ask about the latency or throughput.
  - How long does it take to read one address of memory?
  - How many random addresses can be read per second?
- You need to find the total time for a single transaction.
  1. It takes one cycle to send a 32-bit address to the memory.
  2. The memory needs 50ns, or 5 cycles, to read a 64-bit value.
  3. It takes two cycles to send 64 bits over a 32-bit wide bus.
- Then you can calculate latencies and throughputs.
  - The time to read from one address is eight cycles or 80ns.
  - You can do 12.5 million reads per second, for an effective bandwidth of $(12.5 \times 10^6 \text{ reads/second}) \times (8 \text{ bytes/read}) = 100\text{MB/s}$.

Summary

- Peripheral devices are a crucial component of computer systems.
  - They deliver obscene amounts of data to and from a processor.
  - Buses connect devices, the processor and memory together.
- Unfortunately, I/O is often a bottleneck.
  - Physical limitations and poor designs can drag down a system.
  - Amdahl's law argues in favor of improving I/O performance.
- I/O performance metrics include bandwidth, latency and throughput.
- Next week we'll look at some more bus-related issues, drawing examples from current technologies like PCI, USB and Firewire.