Lecture 23

- Finish-up buses
- Storage

Example Bus Problems, cont.

2) Assume the following 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.

For this system, a single read can be performed in eight cycles or 80ns for an effective bandwidth of (12.5 x 10^6 reads/second) x (8 bytes/read) = 100MB/s.

A) If the memory was widened, such that 128-bit values could be read in 50ns, what is the new effective bandwidth?

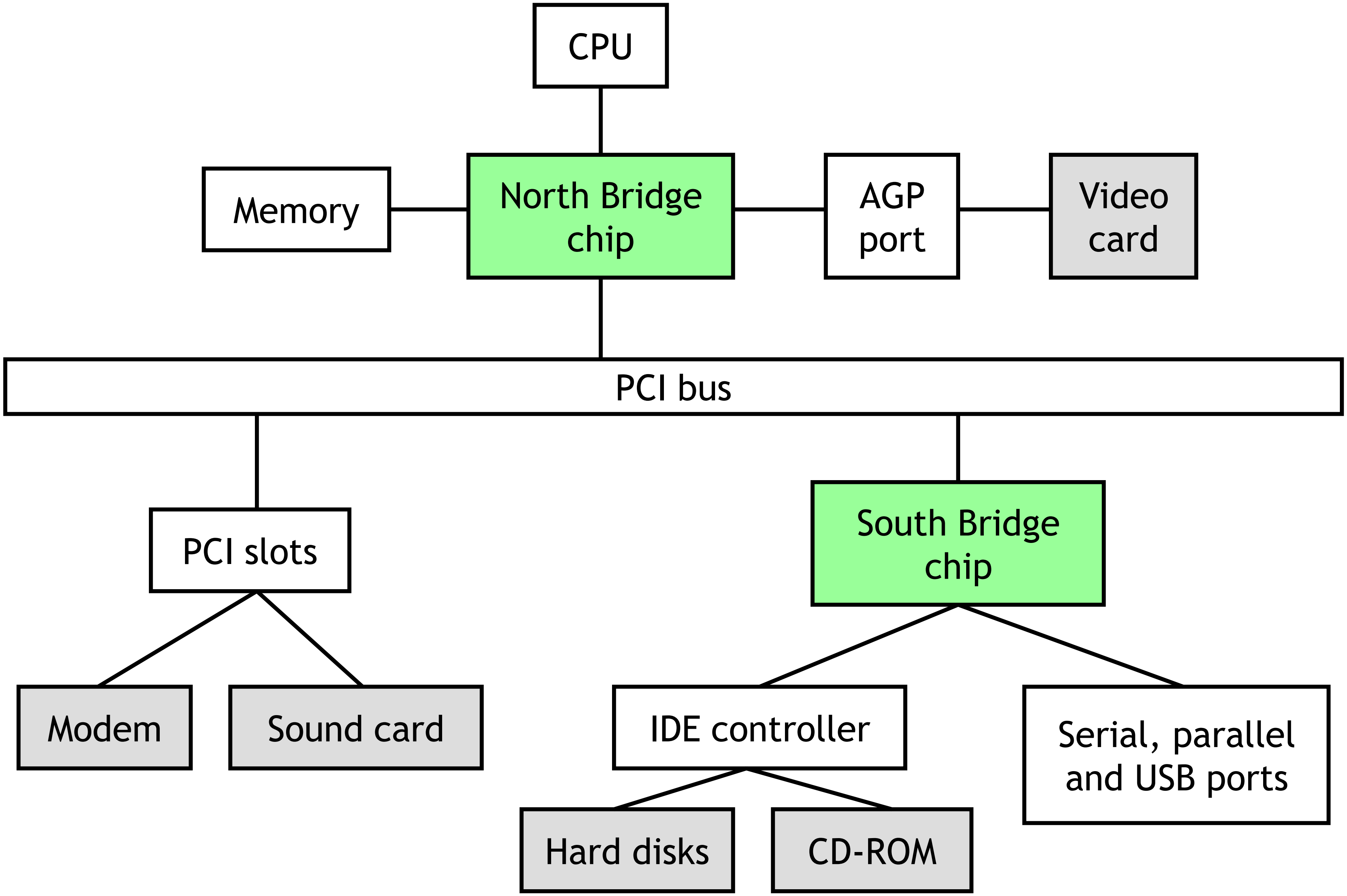
B) What is the bus utilization (fraction of cycles the bus is used) to achieve the above bandwidth?

C) If utilization were 100% (achievable by adding additional memories), what effective bandwidth would be achieved?

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.

Buses in modern PCs

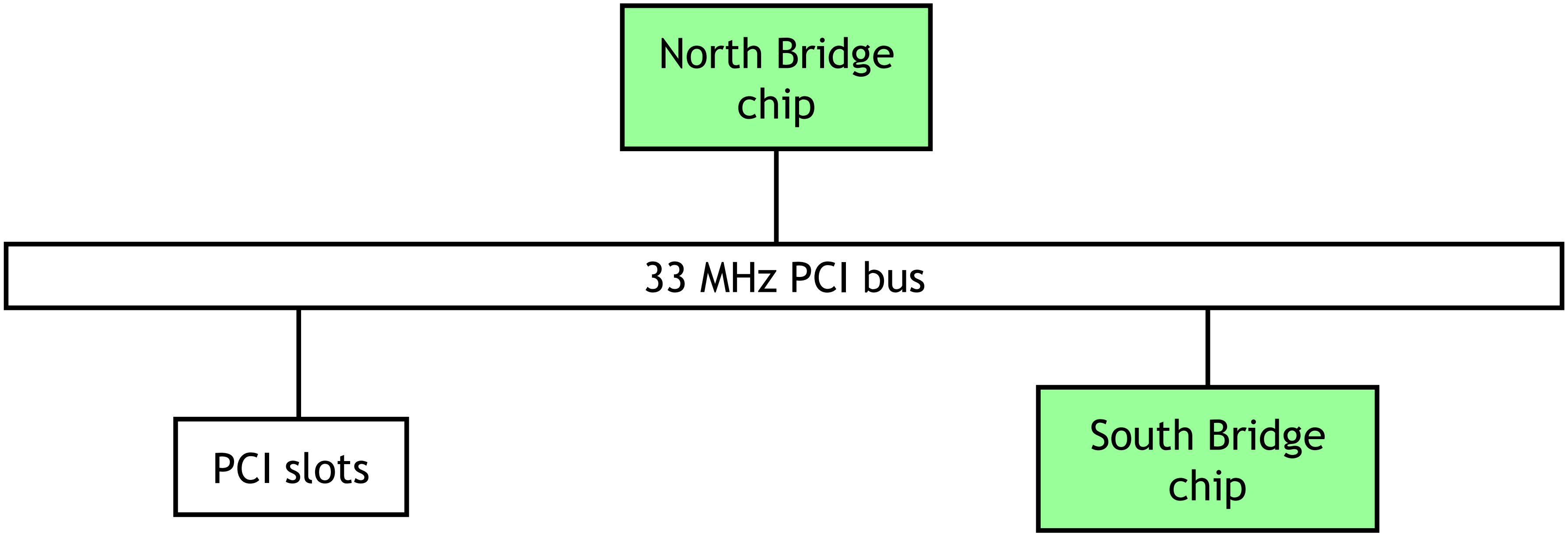


PCI

- Peripheral Component Interconnect is a synchronous 32-bit bus running at 33MHz, although it can be extended to 64 bits and 66MHz.
- The maximum bandwidth is about 132 MB/s.

33 million transfers/second x 4 bytes/transfer = 132MB/s

- Cards in the motherboard PCI slots plug directly into the PCI bus.
- Devices made for the older and slower ISA bus standard are connected via a “south bridge” controller chip, in a hierarchical manner.

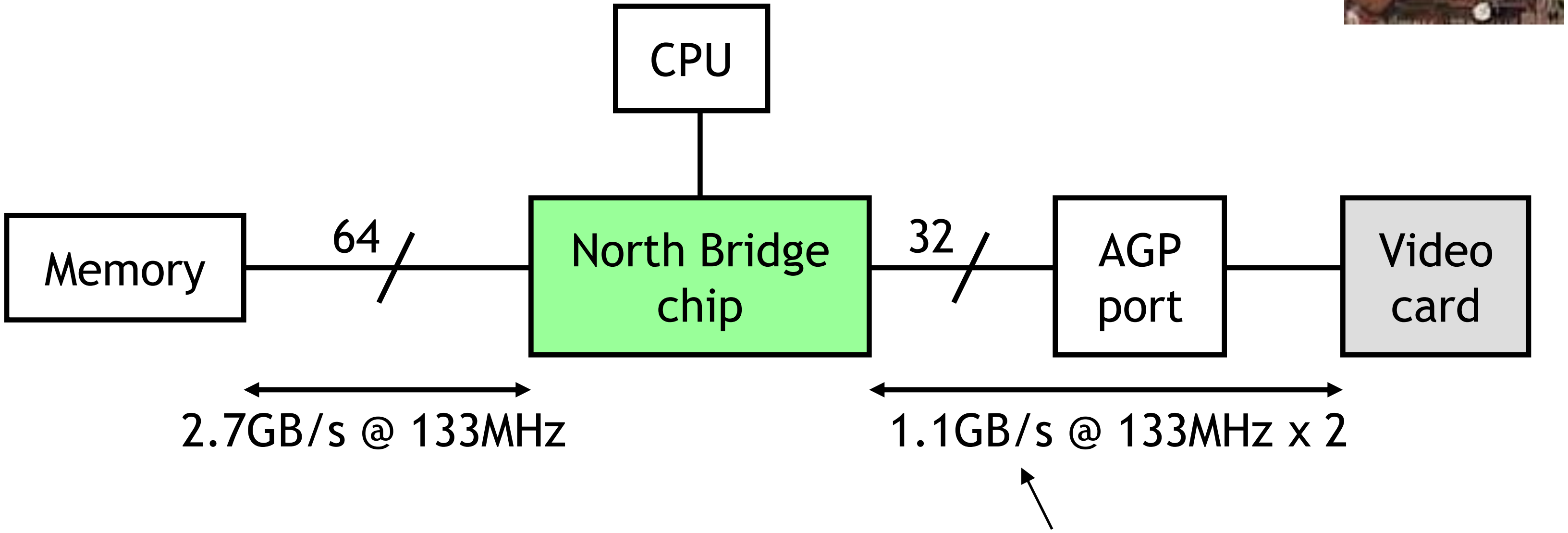


Frequencies

- CPUs actually operate at two frequencies.
  - The internal frequency is the clock rate inside the CPU, which is what we've been talking about so far.
  - The external frequency is the speed of the processor bus, which limits how fast the CPU can transfer data.
- The internal frequency is usually a multiple of the external bus speed.
  - A 2.167 GHz Athlon XP sits on a 166 MHz bus (166 x 13).
  - A 2.66 GHz Pentium 4 might use a 133 MHz bus (133 x 20).
    - *You may have seen the Pentium 4's bus speed quoted at 533MHz. This is because the Pentium 4's bus is "quad-pumped", so that it transfers 4 data items every clock cycle.*
- Processor and Memory data rates far exceed PCI's capabilities:
  - With an 8-byte wide "533 MHz" bus, the Pentium 4 achieves 4.3GB/s
  - A bank of 166MHz Double Data Rate (DDR-333) Memory achieves 2.7GB/s

The North Bridge

- To achieve the necessary bandwidths, a "frontside bus" is often dedicated to the CPU and main memory.
  - "bus" is actually a bit of a misnomer as, in most systems, the interconnect consists of point-to-point links.
  - The video card, which also need significant bandwidth, is also given a direct link to memory via the Accelerated Graphics Port (AGP).
- All this CPU-memory traffic goes through the "north bridge" controller, which can get very hot (hence the little green heatsink).



External buses

- External buses are provided to support the frequent plugging and un-plugging of devices
  - As a result their designs significantly differ from internal buses

- Two modern external buses, Universal Serial Bus (USB) and FireWire, have the following (desirable) characteristics:
  - Plug-and-play standards allow devices to be configured with software, instead of flipping switches or setting jumpers.
  - Hot plugging means that you don't have to turn off a machine to add or remove a peripheral.
  - The cable transmits power! No more power cables or extension cords.
  - Serial links are used, so the cable and connectors are small.

Serial/Parallel

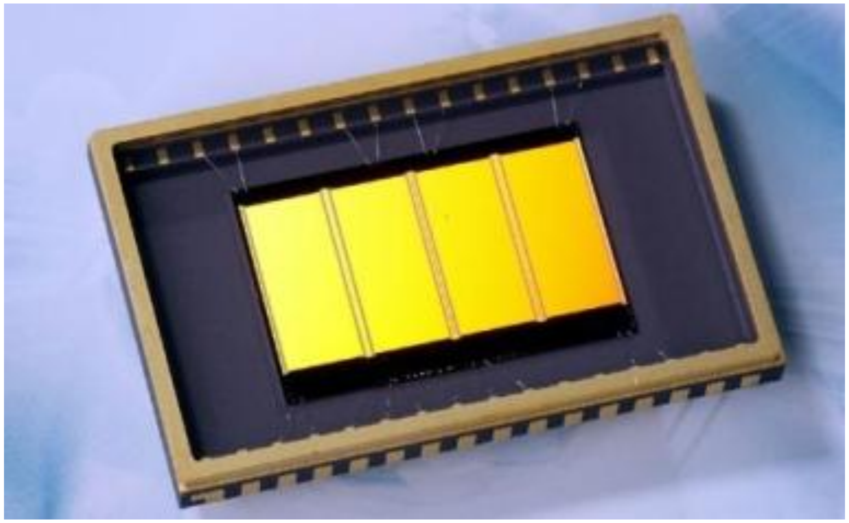
- Why are modern external buses serial rather than parallel?

- Generally, one would think that having more wires would increase bandwidth and reduce latency, right?
  - Yes, but only if they can be clocked at comparable frequencies.

- Two physical issues allow serial links to be clocked significantly faster:
  - On parallel interconnects, interference between the signal wires becomes a serious issue.
  - Skew is also a problem; all of the bits in a parallel transfer could arrive at slightly different times.

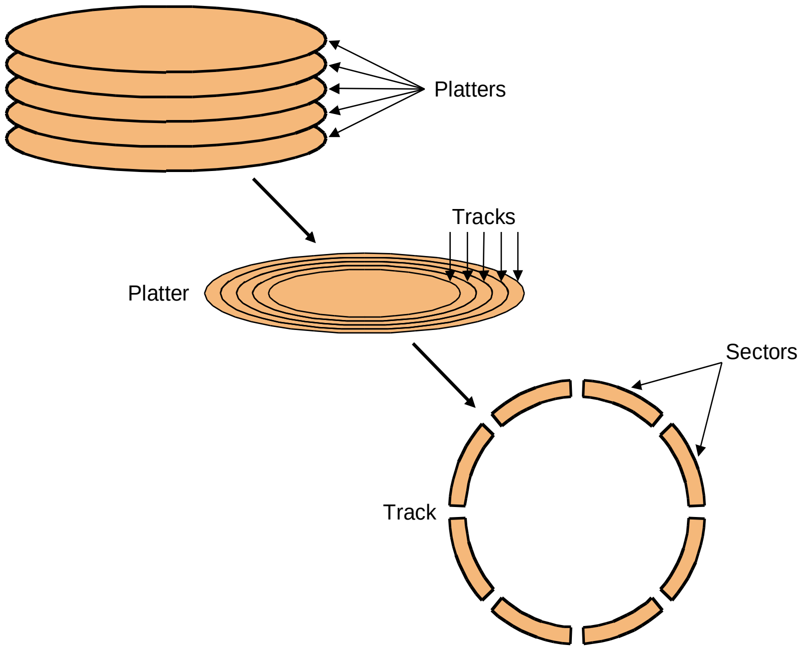
- Serial links are being increasingly considered for internal buses:
  - Serial ATA is a new standard for hard drive interconnects
  - PCI-Express (aka 3GI/O) is a PCI bus replacement that uses serial links

Disk Drives, Flash



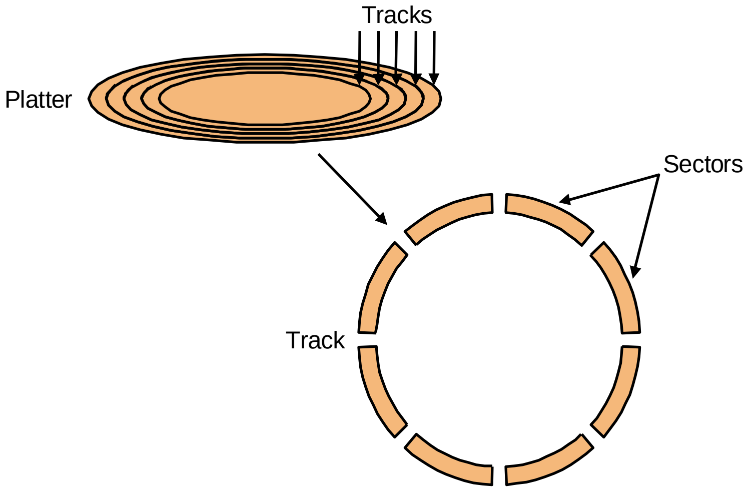
Hard drives

- 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 (seek time)

- Manufacturers often report *average* seek times of 8-10ms.
  - These times average the time to seek from any track to any other track.
- In practice, seek times are often much better.
  - For example, if the head is already on or near the desired track, then seek time is much smaller. In other words, locality is important!
  - Actual average seek times are often just 2-3ms.

Estimating Disk Latencies (rotational latency)

- Once the head is in place, we need to wait until the right sector is underneath the head.
  - This may require as little as no time (reading consecutive sectors) or as much as a full rotation (just missed it).
  - On average, for random reads/writes, we can assume that the disk spins halfway on average.

- Rotational delay depends partly on how fast the disk platters spin.

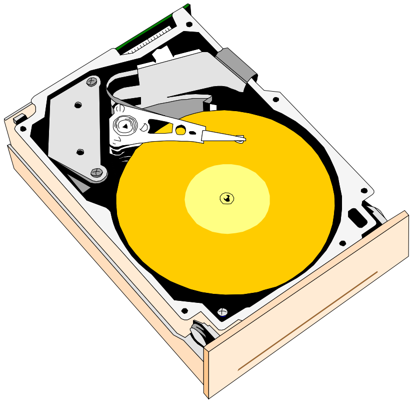
Average rotational delay = 0.5 x rotations x rotational speed

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

0.5 rotations / (5400 rotations/minute) = 5.55ms

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 random 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.

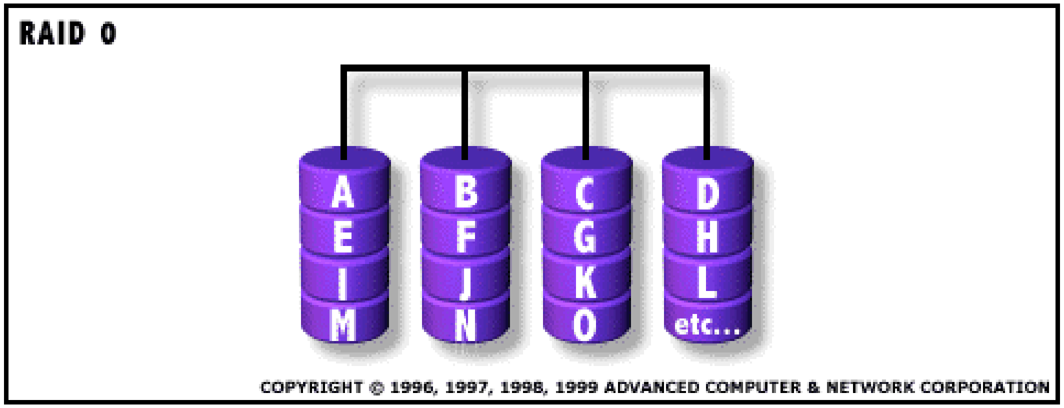
(1 sector / 16.7ms) x (1000ms / 1s) = 60 sectors/second.

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 3ms
  - A 6000 RPM rotational speed
  - A 10MB/s average transfer rate
  - 2ms of overheads
- How long does it take to read a random 1,024 byte sector?
  - The average rotational delay is:
  - The transfer time will be about:
  - The response time is then:
- How long would it take to read a whole track (512 sectors) selected at random, if the sectors could be read in any order?

Parallel I/O

- Many hardware systems use parallelism for increased speed.
  - Pipelined 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.


RAID 0
A
E
I
M
B
F
J
N
C
G
K
O
D
H
L
etc...
COPYRIGHT © 1996, 1997, 1998, 1999 ADVANCED COMPUTER & NETWORK CORPORATION