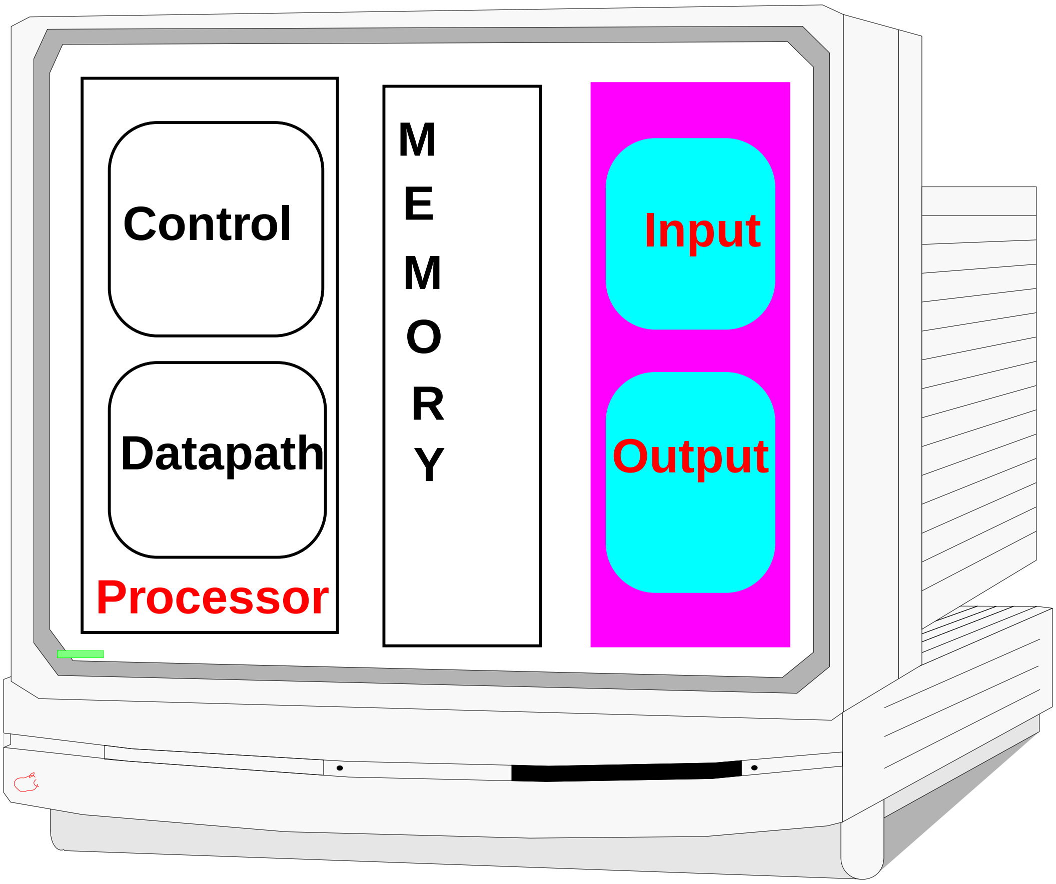
Outline of Lecture

- Various I/O Devices
- Importance of I/O
- User Response Time
- I/O Interfacing
- Type of Storage Devices

The Five Components of a Computer System



Why is I/O Important?

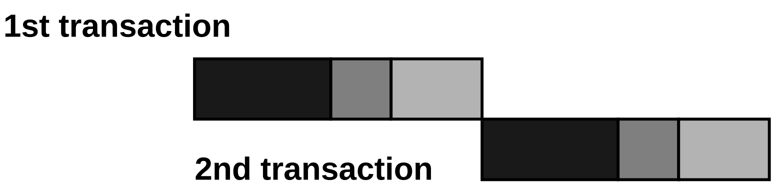
- Customers are more ***interested*** in response time than CPU time (CPU time does not include I/O performance).
- CPU performance: 50% to 100% improvement per year.
- I/O system performance is limited by mechanical delays (e.g., disk drivers) < 10% improvement per year (MB/sec).

The overall performance of a computer system will ***not improve*** greatly because of the I/O bottleneck.

Historically, I/O got little attention from computer architects because most computer systems are ***time-shared***. Thus, the CPU could work on a different process while waiting for I/O response. This is no longer true as most machines are used by single users.

Response Time Vs. Productivity

- **Interactive environments: each interaction or *transaction* has 3 parts.**
  - → ***Entry time:*** **time for the computer user to enter a command.**
  - → ***System response time:*** **Time between the user entry and the when the computer system replies.**
  - → ***Think time:*** **Time from computer response until user begins next command.**

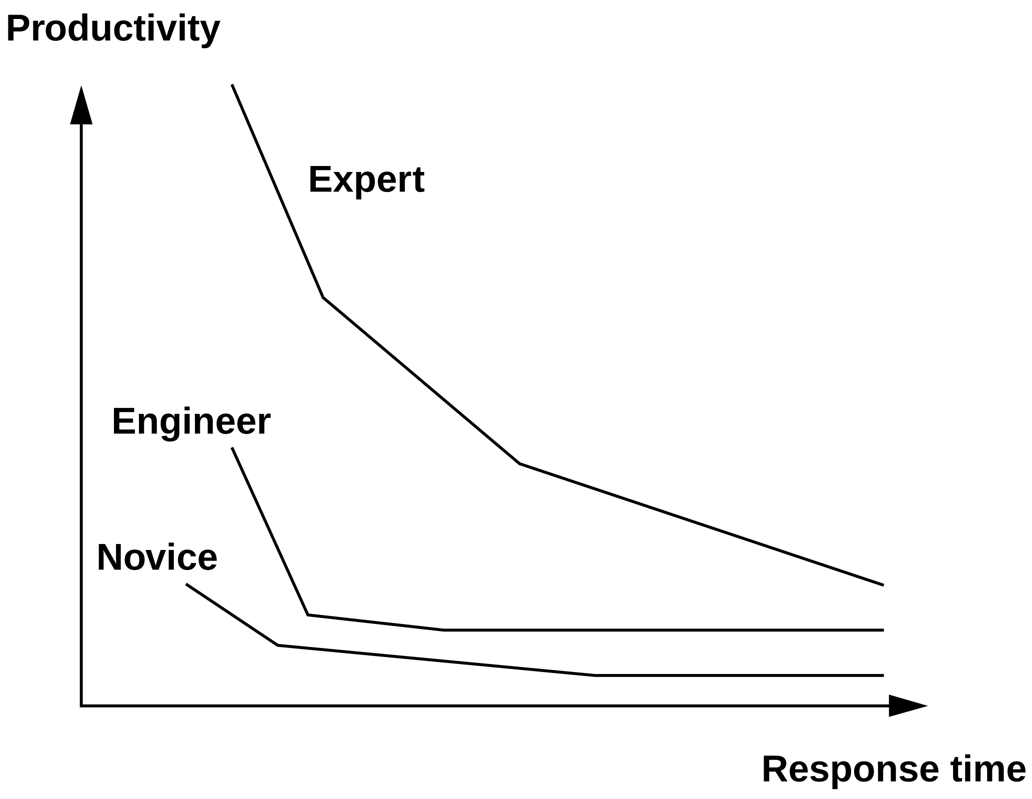


- **The user productivity is *inversely proportional* to transaction time.**

User Response Time

- A variety of different users (e.g., banks, supercomputers, engineers

☞ People use less time to think when given a faster response.



☞ It is extremely important to improve the response time of I/O devices.

Impact of I/O on System Performance

Suppose we have a benchmark that executes in 100 seconds of elapsed time, where 90 seconds is CPU time and the rest if I/O time. If CPU time improves by 50% per year forthe next five years but I/O time doesn't improve, how much faster will our program run at the end of five years?

Elapsed time = cpu time + I/O time

100 = 90 + I/O time

I/O time = 10 seconds

After n years	CPU time	I/O time	Elapsed time	% I/O time
0	90 seconds	10 seconds	100 seconds	10%
1	90/1.5 = 60 seconds	10 seconds	70 seconds	14%
2	60/1.5 = 40 seconds	10 seconds	50 seconds	20%
3	40/1.5 = 27 seconds	10 seconds	37 seconds	27%
4	27/1.5 = 18 seconds	10 seconds	28 seconds	36%
5	18/1.5 = 12 seconds	10 seconds	22 seconds	45%

The improvement in CPU performance over five years is 90/12 = 7.5

However, the improvement in elapsed time is only 100/22 = 4.5

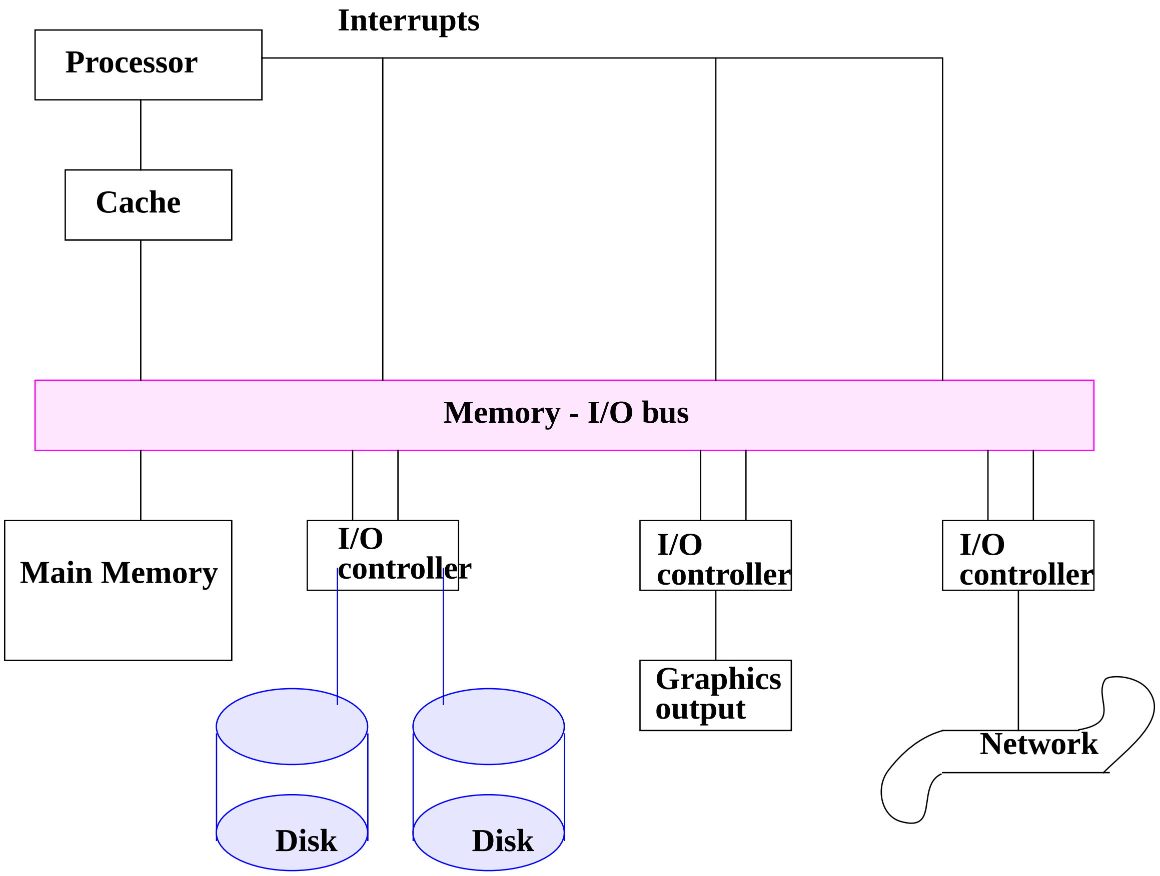
and the I/O time has increased from 10% to 45% of the elapsed time.

Linking Processor and Peripherals

- **I/O Design affected by many factors (expandability, resilience):**
  - access latency
  - throughput
  - connection between devices and the system
  - the memory hierarchy
  - the operating system

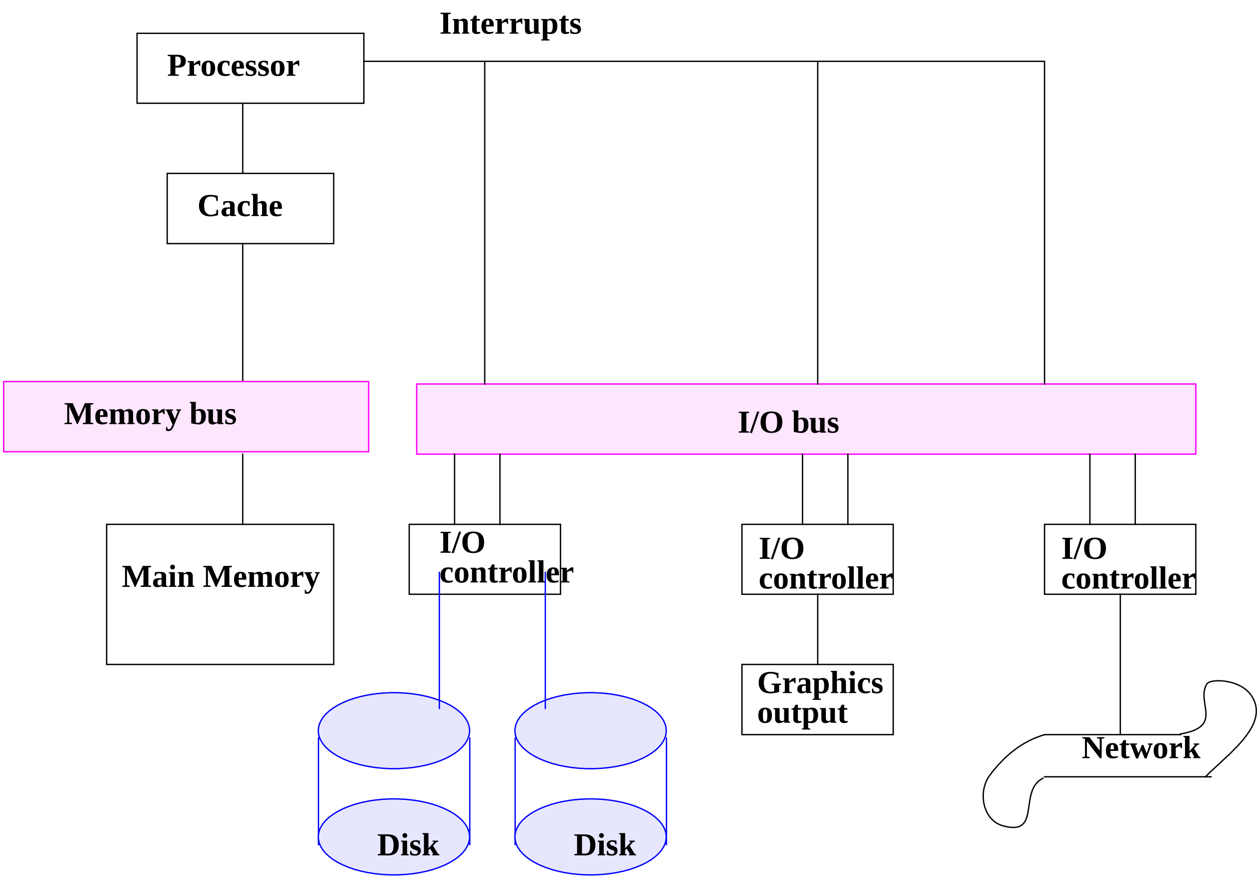
I/O Interface

- **Single bus:** This is the typical bus configuration in low-cost computer systems (supercomputers use optical buses to increase performance).
- I/O commands and CPU instruction and data fetches are ***multiplexed*** together on the same bus (bus could be ***saturated***).



I/O Interface

- **2 separate buses:** Most high performance computers use separate buses for I/O devices and memory access (I/O transactions does not slow down the memory access).



I/O Address

How Does the CPU address an I/O device that needs to send or receive data?

Memory Mapped I/O (Motorola)

- **Portions of the address space are assigned to I/O devices. Reads (`Load`) and writes (`Store`) to those addresses are interpreted as commands to the I/O device and cause data to be transferred to/from the I/O device. Thus, commands to the I/O devices are just accesses to those memory locations.**

ROM
RAM
I/O

Isolated I/O (Intel)

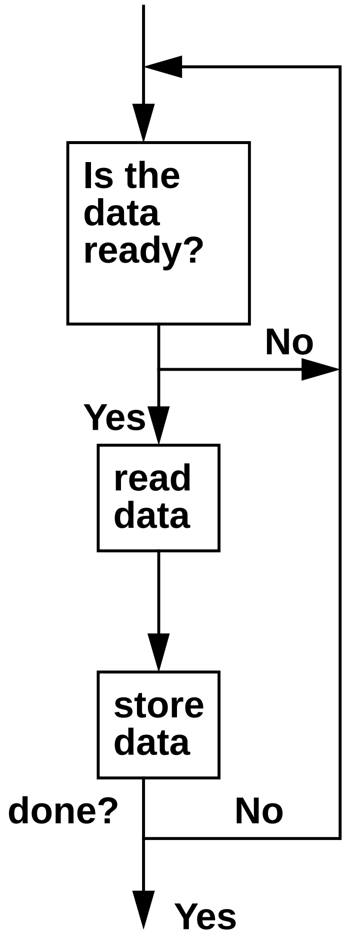
- **The CPU has distinct I/O opcodes (`IN & OUT`) (instructions) - freeing the memory for use by the program. Each of these opcodes is associated with the address of an interface register.**

Communication with I/O

How Does the CPU Sends and Receives Data from the I/O device?

(1) Programmed I/O (Polling)

- **The CPU periodically checks status bits (e.g., whether a character has been printed or not) to see if it is time for the next I/O operation.**



Busy wait loop - not an efficient way to use the CPU unless the I/O device is very fast.

Communication with I/O

(2) Interrupt Driven I/O

- **It is used by many computer systems. It allows the CPU to work on some other process while waiting for the I/O device. The CPU is only halted - *interrupted by the I/O device* - during the actual transfer.**

- **1000 transfers cause 1000 interrupts and 1000 interrupt services - The interrupt overhead typically takes as much time as the actual transfer itself.**

- **It can cause a large overhead on the operating system because of large number of context-switching - (1) I/O interrupt, (2) Save PC and registers, (3) Jump to interrupt service routine.**

Communication with I/O

(3) Direct Memory Access

- Interrupt-driven I/O relieves the CPU from waiting for every I/O event, but there are still many CPU cycles spent in transferring data (for a disk block of 2048 words, we require at least 2048 `Loads` or 2048 `Stores`) and 2048 interrupts.

- ***Direct memory access*** (DMA) is a hardware that is added to most computer systems to allow the transfer of a large number of words without the intervention of the CPU.

- DMA is a specialized processor that transfers data between memory and an I/O device while the CPU goes on with other tasks.

How DMA Works

- The CPU gives the starting address, direction, and length count to the DMA. Then issues "start". The DMA does the rest without any further CPU intervention. Thus, the CPU is only interrupted ounce during any data transfer.

I/O Devices

- Very diverse devices
  - behavior (i.e., input vs. output)
  - partner (who is at the other end?)
  - data rate

Device	Behavior	Partner	Data rate (KB/sec)
Keyboard	input	human	0.01
Mouse	input	human	0.02
Voice input	input	human	0.02
Scanner	input	human	400.000
Voice output	output	human	0.60
Line Printer	output	human	1.00
Laser Printer	output	human	200.000
Graphics display	output	human	60,000.00
Modem	input or output	machine	2.00 - 8.00
Network/LAN	input or output	machine	500.00 - 6000.00
Floppy disk	storage	machine	100.00
Optical disk	storage	machine	1000.00
Magnetic tape	storage	machine	2000.00
Magnetic disk	storage	machine	2000.00 - 10,000.00

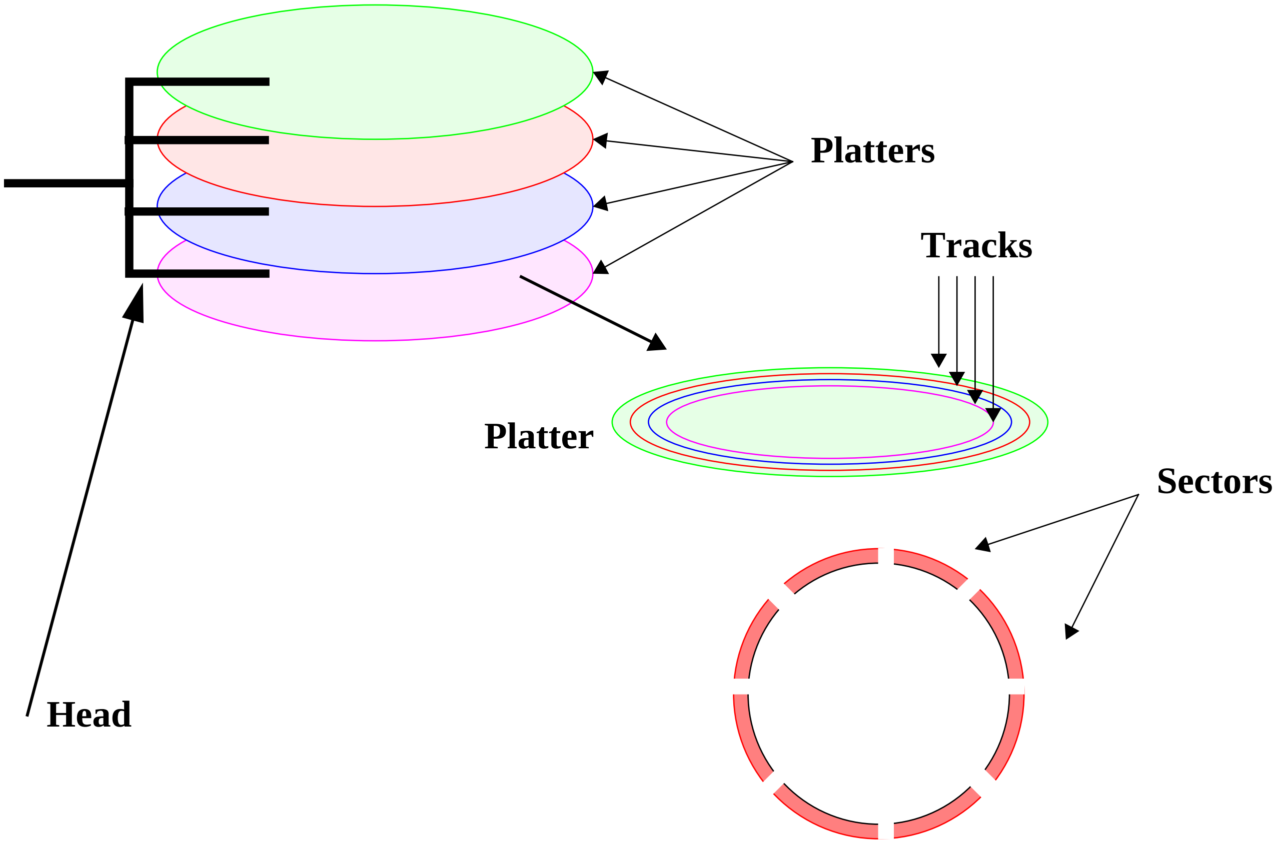
Types of Storage Devices

Magnetic Disks

They have dominated the secondary storage devices since their introduction to computer systems in 1965.

- ***Purpose:***
  - ➔ **Long-term, nonvolatile storage for files.**
  - ➔ **Large, cheap, slowest level in the memory hierarchy.**
- ***Characteristics:***
  - ➔ ***Seek time*** **(related to mechanical components) is the amount of time it takes to move to the appropriate sector. Typically it is 12 msec on average (positional latency {search track} and rotational latency {search sector}).**
  - ➔ ***Transfer rate*** **is the rate at which a block of bits, typically a sector, is transferred under the read/write head. Typical rates for disks are 1 - 10 MB/sec.**
  - ➔ ***Capacity*** **is in the Gigabytes (e.g., 2-4 Gbytes) (4x every 4 years) .**

I/O Example: Disk Drivers



- **To access data:**
  - Queueing Time: if other users are waiting
  - seek: position head over the proper track (8 to 20 ms. avg.)
  - rotational latency: wait for desired sector (.5/ RPM)
  - transfer: grab the data (one or more sectors) 2 to 15 MB/sec

☞ ***Disk Access Time= Queuing Time+ Average Seek Time + Average rotational Delay + Transfer Time + Control Overhead***

Rotational Delay

- **Once the head reaches the correct track, we must wait for the desired sector to come under the head. This time is called the *rotational delay or average rotational latency*.**
- **The average latency to the desired sector is half way around the track.**
- **The average disk rotation rate is 3600 RPM to 7200 RPM.**
- **Thus the average rotation delay is**

Disk rotational delay = $\frac{0.5 \text{ rotation}}{3600/60}$ **= 0.0083 sec. = 8.3ms**

and

Disk rotational delay = $\frac{0.5 \text{ rotation}}{7200/60}$ **= 0.0042 sec. = 4.2ms**

Example

What is the average time to read or write a 512-byte sector for a typical disk?

Advertised seek time = 12 ms

The disk rotates at 5400 RPM

Transfer rate is 5MB/sec

Controller overhead = 2ms

No Queuing delay

Answer

Average read/write time **= 12 msec + 0.5 / 5400 /60**

+ 0.5 KB / 5MB + 2 ms

= 12 + 5.6 + 0.1 + 2 = 19.7 ms

Disks are 100 times cheaper than DRAMs but are 100,000 times slower - architects tried to invent something in between with *NO* success (e.g., Solid State Disks {SSD}).

Magnetic Tapes Vs. Disks

- Magnetic tapes use the same technology as hard disks; thus, they track their density improvements.

- Inherent cost-performance difference between tapes and disks are based on their geometries:

  ➔ Fixed rotating platters with gaps (random access, limited area, 1 media/reader).

Versus

  ➔ Long strips wound on removable spools (Sequential access - slow, “unlimited” length, multiple tapes per reader).

Optical Disks Vs. Magnetic Tapes

- **Optical compact disks (CDs) are removable and inexpensive to manufacture - but it is a *read only* medium. It is a perfect medium for distributing information (e.g., software).**
- **When CDs become writable, they will have a greater impact.**

Further Reading

Chapter 8. **David A. Patterson and John L. Hennessy. *Computer Organization & Design: The Hardware / Software Interface*. Morgan Kaufman Publishers (page 638-656).**