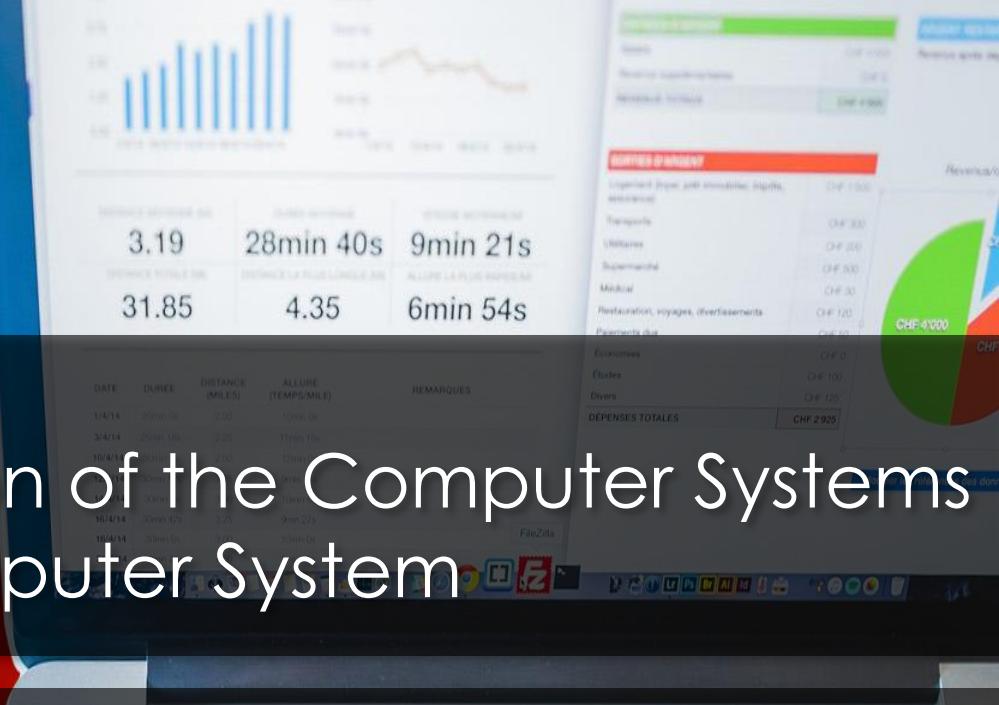


# Lecture 01

## History and the Evolution of the Computer Systems Components of a Computer System

**IT1020 Introduction to Computer Systems**  
**Part 01 – Computer Fundamentals**



# Pre-lecture activities

- History of computer systems
  - Major Milestones
- Diagram of Von Neumann
- Components of computer systems
- Storage devices

# Lecture Outline

1. Generations of Computers
  - Characteristics of each Generation
2. Components of Computers
  - Different types of components

# 1. Generations of the Computers

1<sup>st</sup> Generation computers(1944-1955)

2<sup>nd</sup> Generation computers (1955-1964)

3<sup>rd</sup> Generation computers (1964-1971)

4<sup>th</sup> Generation Computers (1971- Present )

5<sup>th</sup> Generation Computers (Present and Beyond)

# 1. Generations of the Computers

## 1<sup>st</sup> Generation Computers (1944-55)

### Main characteristics of this generation

- Used Thermion valves
- Large in size and very heavy in weight
- Power consumption was very high
- First Generation Computers relied on Machin Language
- Writing program on them was difficult or quite slow
- They were very expensive to operate, using a great deal of electricity, generated a lot of heat, which was often the cause of malfunctions

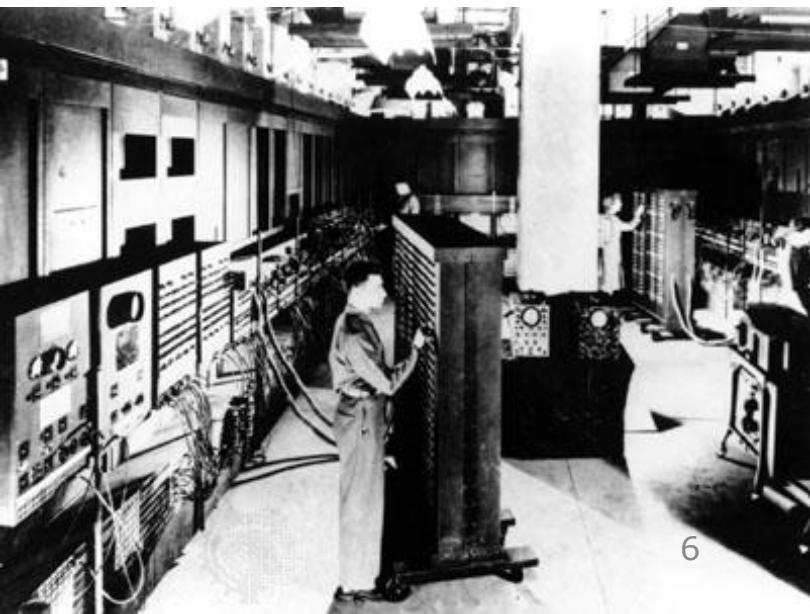


# 1. Generations of the Computers

## 1<sup>st</sup> Generation Computers

### Electronic Numerical Integrator and Calculator (ENIAC)

- 1946 : First electronic general purpose calculator, ENIAC was built in U.S, weighs 33 tons, consumes 150kw, and averages 5000 operations per second

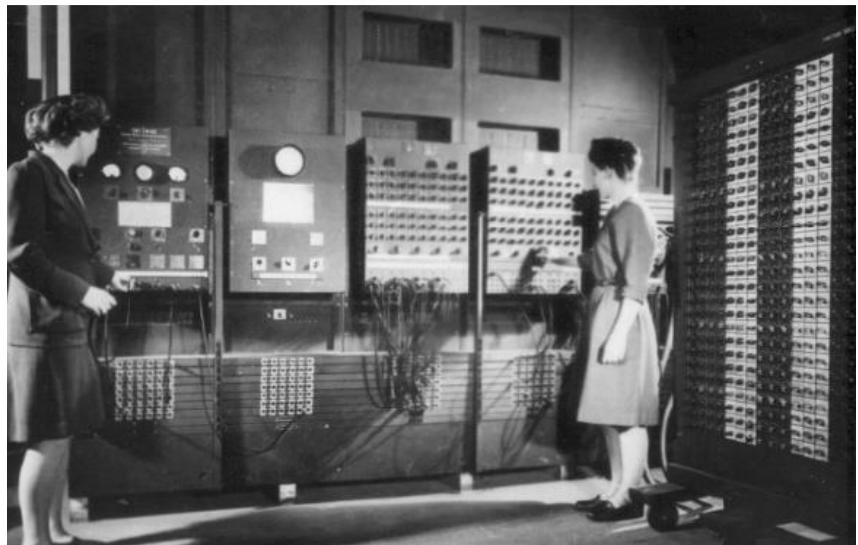


# 1. Generations of the Computers

## 1<sup>st</sup> Generation Computers

### Hard wired programming

- Early computers were programmed, using large number of switches in the console panel and plugging/unplugging cables
- It is called **hardwired programming**



Two women working with  
ENIAC computer  
United States Army Photo.

# 1. Generations of the Computers

## 1<sup>st</sup> to 2<sup>nd</sup> generation



## Von Neumann Architecture

- It was required re-wire and re-design the machine to run a different program. It was a manual and very tedious task
- Von Neumann proposed that programs and data can be stored in a **memory** device and instead of rewiring the machine we can change the program easily.



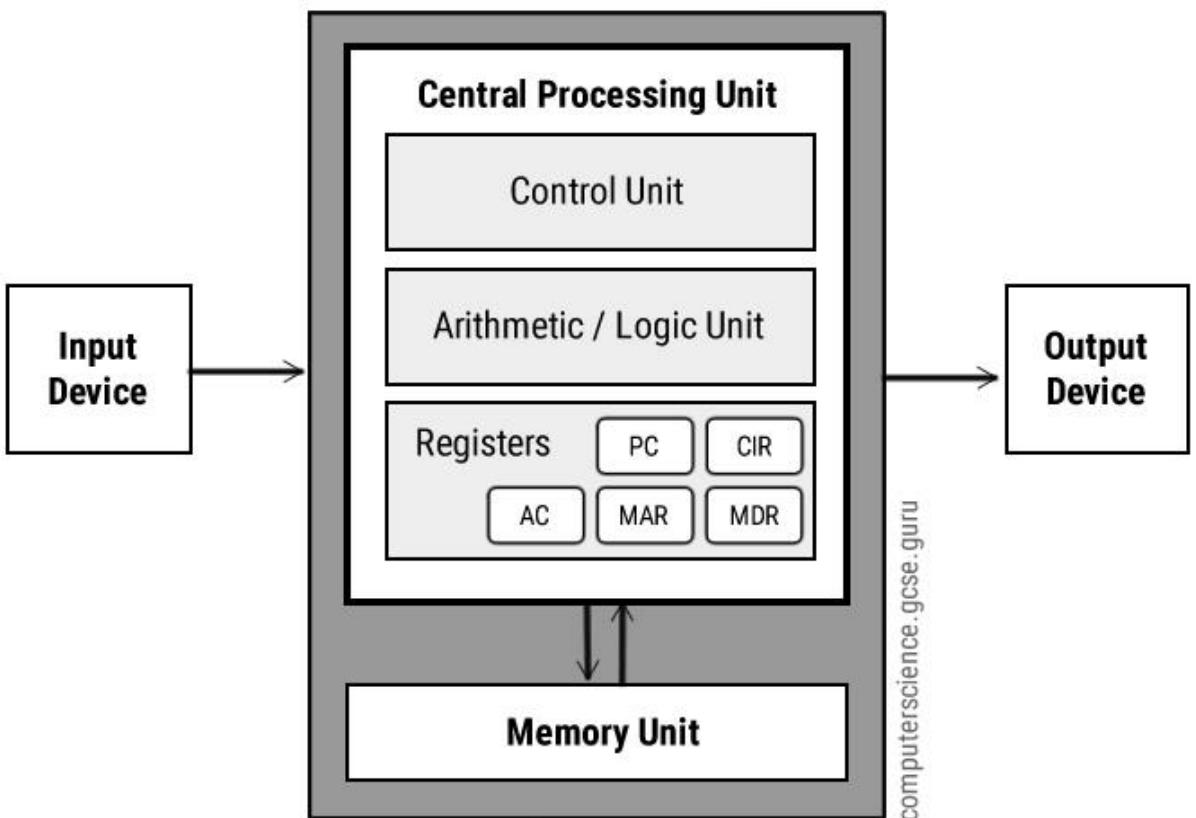
John Von Neumann

# 1. Generations of the Computers

## 1<sup>st</sup> to 2<sup>nd</sup> generation



## Von Neumann Architecture



computerscience.gcse.guru

- All computers share the same basic architecture, whether it be a multi-million dollar mainframe or a **Palm Pilot**.
- All have memory, an I/O system, and arithmetic/logic unit, and a control unit.

# 1. Generations of the Computers

## 1<sup>st</sup> to 2<sup>nd</sup> generation



## Von Neumann Architecture

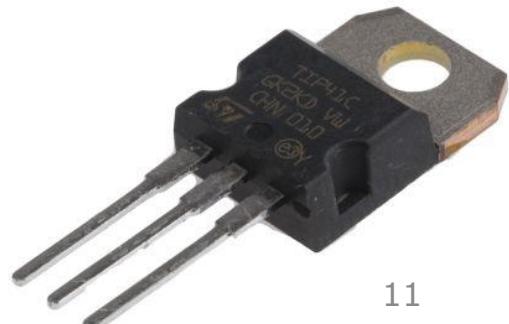
- The use of the binary number system
- A single sequentially addressed memory
- A separate arithmetic/logic unit for performing arithmetic and logical computations
- The stored program concept in which both the programs and its data are stored in memory.
- A controller that fetches instructions from memory and executes them.

# 1. Generations of the Computers

## 1<sup>st</sup> to 2<sup>nd</sup> generation

### Invention of Transistor

- 1947 : Transistor, essential storage device for computers invented at Bell Labs by American engineers William Shockley, John Bardeen and Walter Brattain .
- Transistors were much smaller, more rugged, cheaper to make and far more reliable than valves.



# 1. Generations of the Computers

## 2<sup>nd</sup> Generation Computers (1955-64)

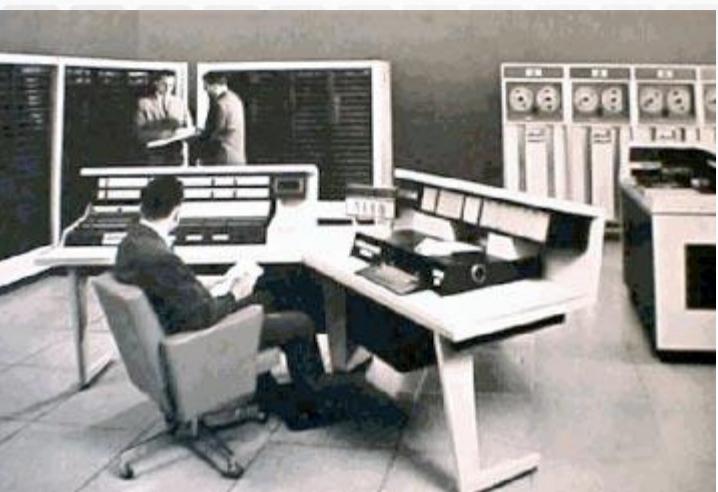
- Used transistors instead of Thermion valves.
- Comparatively higher operating speed.
- Size and weight of the computers decreased
- Manufacturing cost reduced
- The concepts of Central Processing Unit (CPU), memory, programming language and input and output units were developed.
- High-level programming languages introduced
- Development of software for computers
- Computer industry experienced rapid growth.

# 1. Generations of the Computers

## 2<sup>nd</sup> Generation Computers (1955-64)



**IBM 1620**

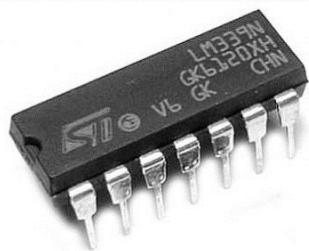


**IBM 1401**

# 1. Generations of the Computers

## 3<sup>rd</sup> Generation Computers(1964-71)

- Integrated Circuits (ICs) were used (A single IC has many transistors, resistors and capacitors built on a single thin slice of silicon.)
- The size of the computer got further reduced
- High Level Languages were developed in this generation
- Large IC companies were started. (INTEL started in 1968, AMD started in 1969)
- The computers were low cost, large memory and processing speed was very high.



# 1. Generations of the Computers

## 3<sup>rd</sup> Generation Computers(1964-71)

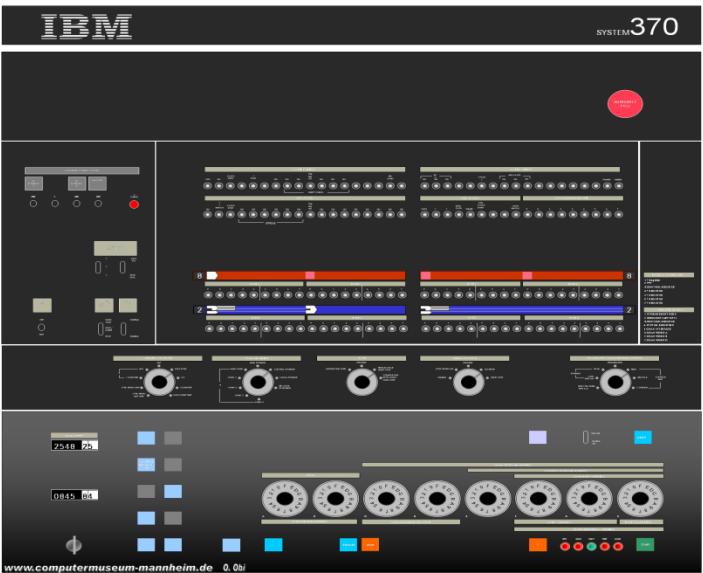
- Substantial **operating systems** were developed to manage and share the computing resources and time sharing operating systems were developed. These greatly improved the efficiency of computers.
- Computers had by now pervaded most areas of business and administration.
- Allowed the device to run many different applications at one time.

# 1. Generations of the Computers

## 3<sup>rd</sup> Generation Computers(1964-71)



IBM System/360



IBM System/370

# 1. Generations of the Computers

## 4<sup>th</sup> Generation Computers (1971-)

- Personal computers were developed and IBM launched
- the Power PC and Pentium introduced the 8088 and 8086 microprocessors. (Most of the computers at present are belong to this generation)
- It uses large scale Integrated Circuits (LSIC) built on a single silicon **chip** called **microprocessors**.
- Memory chips are in megabit range



# 1. Generations of the Computers

## 4<sup>th</sup> Generation Computers (1971-)



- On the software side, more powerful operating systems are available such as Unix.
- Fourth generation languages (4GLs) make the development process much easier and faster.
- Applications software has become cheaper and easier to use.
- Software development techniques have vastly improved.

# 1. Generations of the Computers

## 4<sup>th</sup> Generation Computers (1971-)



*Microprocessor*

# 1. Generations of the Computers

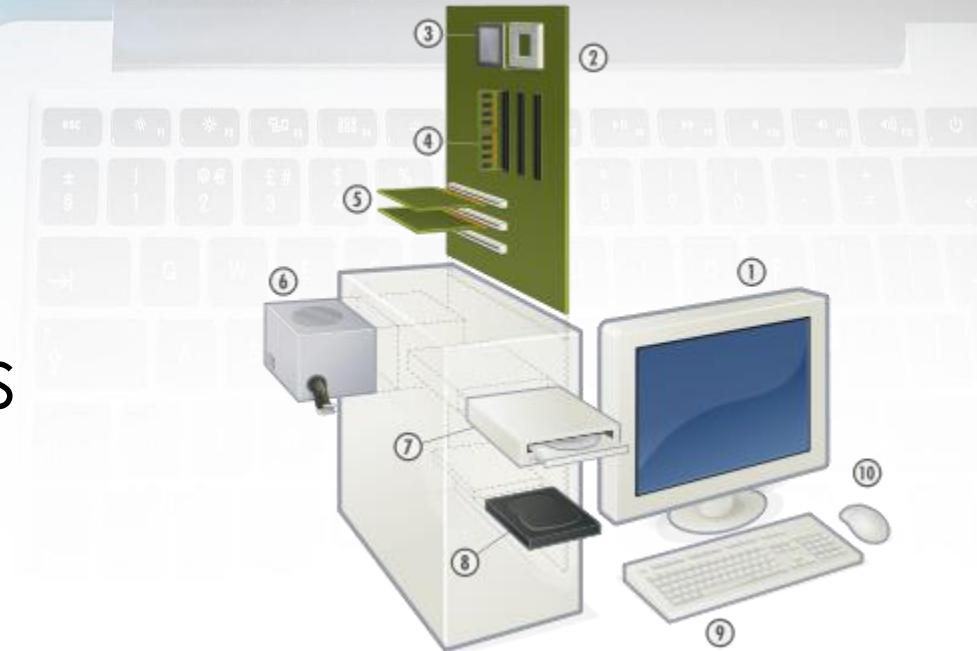
## 5<sup>th</sup> Generation Computers

### Present and beyond

- Fifth generation computing devices, based on Artificial Intelligence (AI).
- Are still in development, though there are some applications, such as voice recognition.
- The use of parallel processing and superconductors is helping to make artificial intelligence a reality.
- The goal of fifth-generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

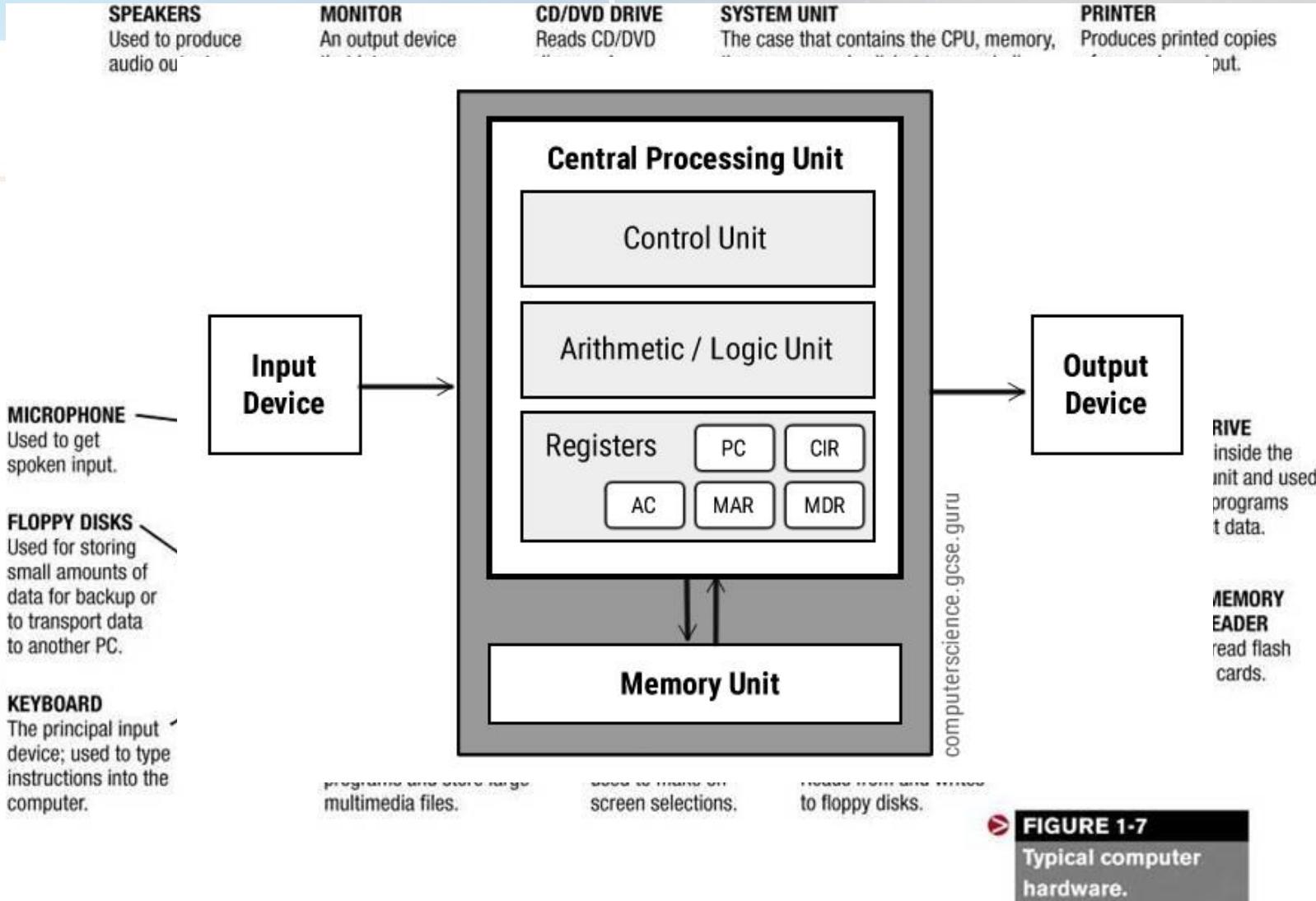
## 2. Components of the Computer

1. Input devices
2. Output devices
3. Processing devices
4. Storage devices
5. Other devices
  - Motherboard, Expansion cards, Power supply



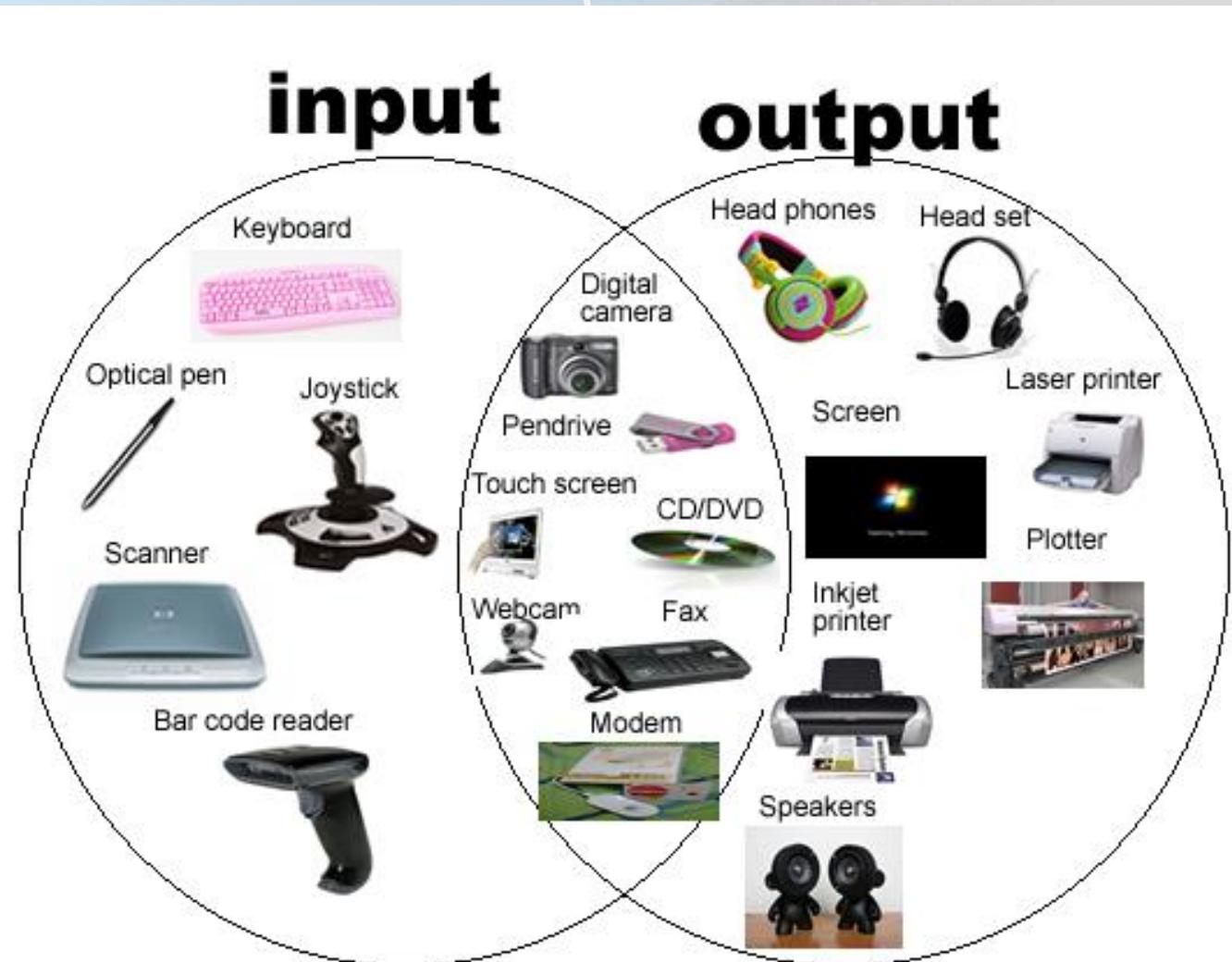
# 2. Components of the Computer

## Basic layout

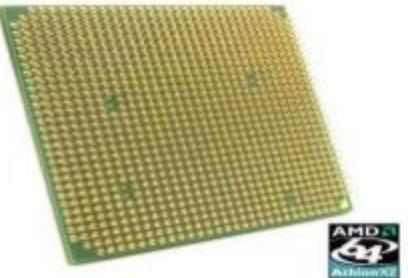


# 2. Components of the Computer

## Input and output devices



# 2. Components of the Computer Processing devices



# 2. Components of the Computer Storage Devices – Features

## 1. Volatility

- Volatile storage
- Non-Volatile storage

## 2. Accessibility

- Random access
- Sequential access

## 3. Mutability

- Read/write storage or mutable storage
- Read only storage
- Slow write, fast read storage

## 4. Addressability

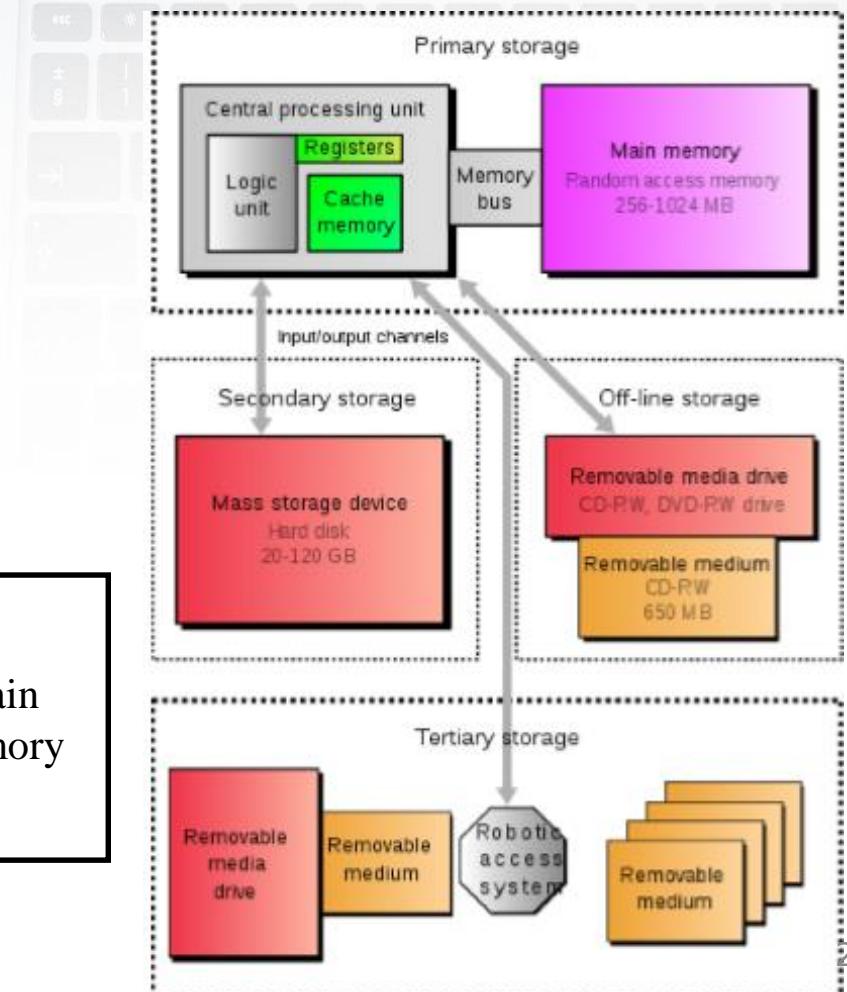
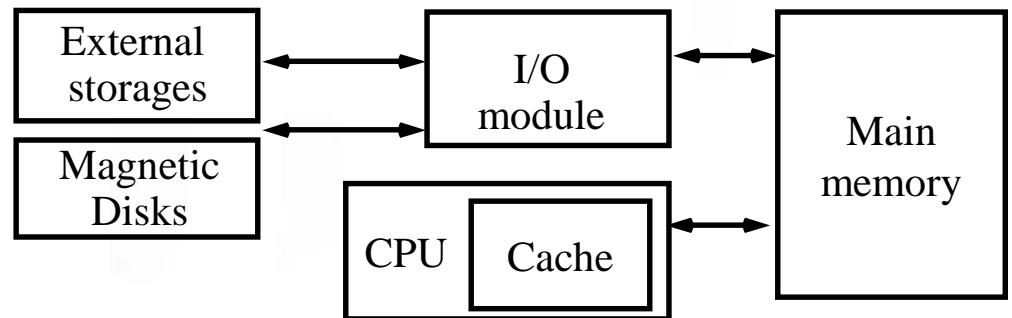
- Location addressable
- File addressable
- Content addressable



# 2. Components of the Computer Storage Devices – Types

There are four type of storage:

1. Primary Storage
2. Secondary Storage
3. Tertiary Storage
4. Off-line Storage



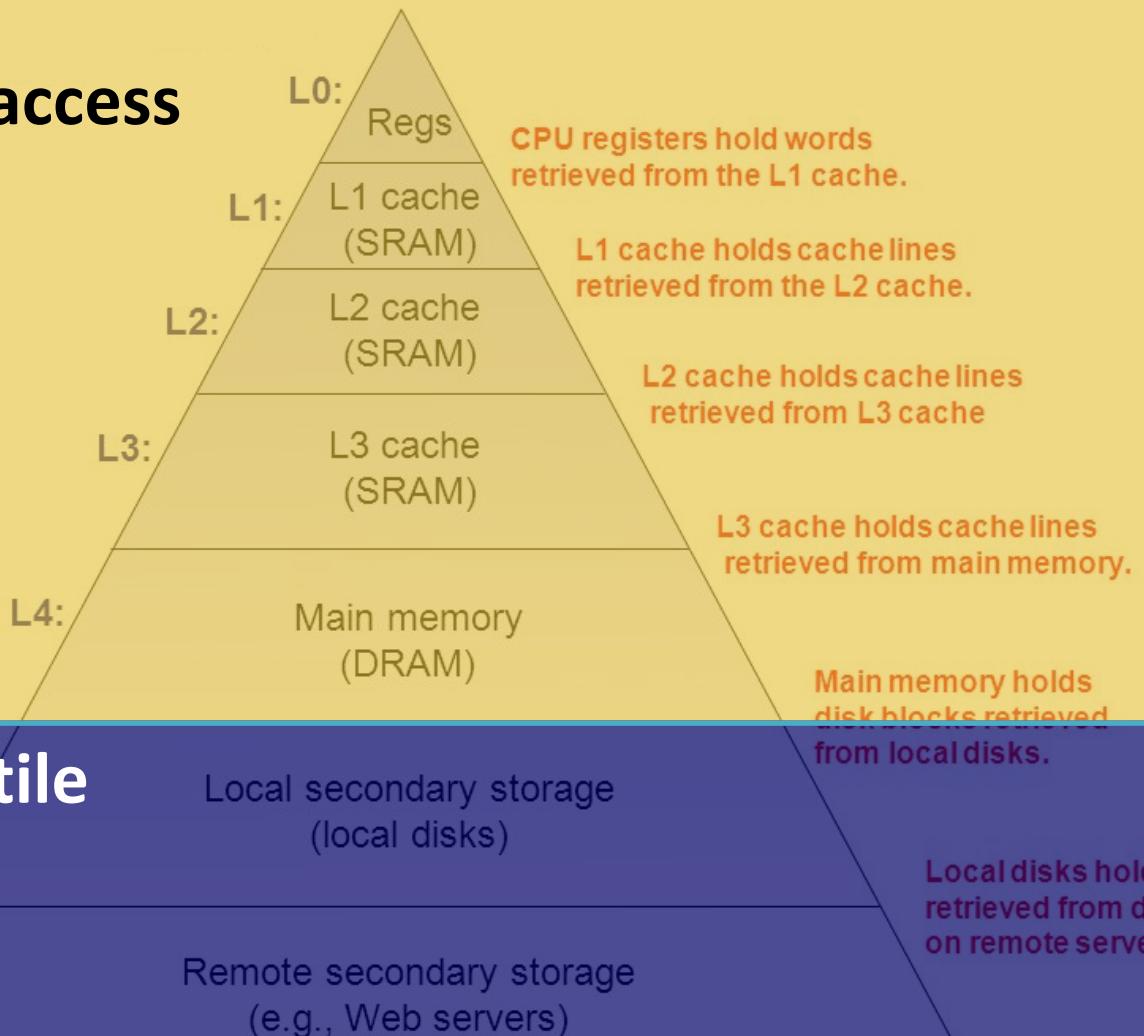
# 2. Components of the Computer Storage Devices – Hierarchy

## Volatile Random access

Smaller,  
faster,  
and  
costlier  
(per byte)  
storage  
devices

Larger,  
slower,  
and  
cheaper

Non-volatile  
(per byte)  
Storage devices



# 2. Components of the Computer Storage Devices – Primary storage

## Primary storage types

1. Registers
2. Main Memory
3. Cache

# 2. Components of the Computer

## Storage Devices – Primary storage

### Registers

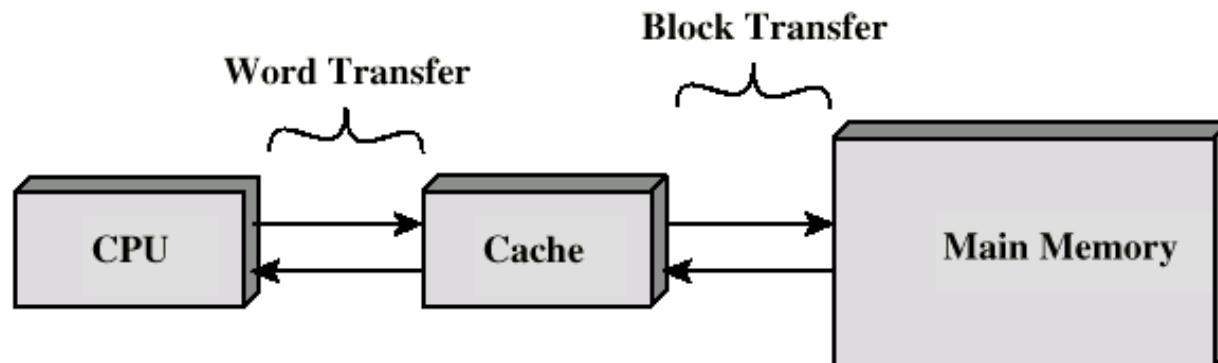
- very small amount of very fast memory that is built into the CPU
- This is to speed up its operations by providing quick access to commonly used values.
- Fastest memory in computer.
- Registers are normally measured by the number of bits they can hold, for example, an 8-bit register or a 32-bit register.
- Registers can also be classified into
  1. general purpose
  2. special purpose.

# 2. Components of the Computer

## Storage Devices – Primary storage

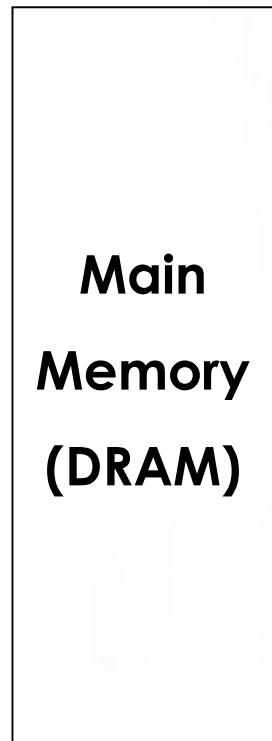
### Cache

- Small amount of fast memory (Faster than RAM, static memory)
- Sits between normal main memory and CPU
- May be located on CPU chip or module.
- Cache works on the principle of locality of reference.

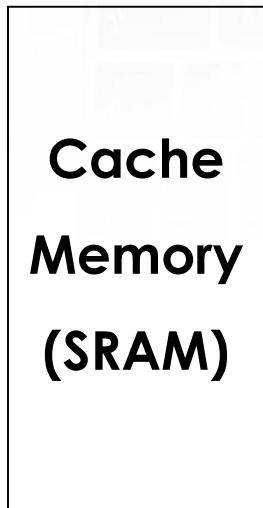


## 2. Components of the Computer Storage Devices – Primary storage

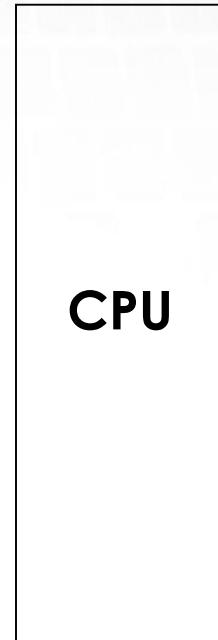
### Cache – a The operation of cache memory



1. Cache fetches data from next to current addresses in main memory
4. If not, the CPU has to fetch next instruction from main memory - a much slower process



- 2.CPU checks to see whether the next instruction it requires is in cache
3. If it is, then the instruction is fetched from the cache – a very fast position



# 2. Components of the Computer

## Storage Devices – Primary storage

### Memory – technologies

#### Static memory

- No charges to leak
- No refreshing needed when powered
- More complex construction
- Larger implementation per bit
- More expensive
- Faster than dynamic memory

#### Dynamic memory

- Bits stored as charge in capacitors
- Level of charge determines value - charges leak
- Need refreshing even when powered (need refresh circuits)
- Simpler construction, smaller per bit
- Less expensive
- Slower than Static memory

# 2. Components of the Computer

## Storage Devices – Primary storage

### Memory – types

<u>Read Only Memory (ROM)</u>	<u>Random Access Memory (RAM)</u> Also called read/write memory.
<ul style="list-style-type: none"><li>▪ Non-volatile in nature</li><li>▪ These cannot be accidentally changed</li><li>▪ Use static memory</li><li>▪ Faster than dynamic memory</li><li>▪ E.g. <b>BIOS chip</b><ul style="list-style-type: none"><li>▪ Masked ROM</li><li>▪ Programmable ROM (PROM)</li><li>▪ Erasable PROM (EPROM)</li><li>▪ EEPROM</li></ul></li></ul>	<ul style="list-style-type: none"><li>▪ Volatile</li><li>▪ This is a semi conductor memory (dynamic memory)</li><li>▪ E.g. <b>Main memory</b><ul style="list-style-type: none"><li>▪ Main memory is usually called RAM. (misnamed because all semiconductor memory is random access)</li><li>▪ Main Memory can be made faster by using static memory. Then why don't we do that?</li><li>▪ Main memory is directly or indirectly connected to the central processing unit via a bus.</li><li>▪ The CPU continuously reads instructions stored in the main memory and executes them as required.</li></ul></li></ul>

## 2. Components of the Computer Storage Devices – Primary storage

### Memory – Main memory

- Main memory consists of a number of storage locations, each of which is identified by a unique address
- The ability of the CPU to identify each location is known as its addressability
- Each location stores a word i.e. the number of bits that can be processed by the CPU in a single operation. Word length may be typically 16, 24, 32 or as many as 64 bits.

# 2. Components of the Computer

## Storage Devices – Primary storage

### Memory – Main memory

- Program and data are stored in memory prior to execution.(This is called Stored Program Concept proposed by Von Neumann).
- Memory is a semiconductor device in modern computers (Magnetic core memories were used earlier)
- Main memory, primary storage are synonyms to memory. (RAM also denotes the same)

# 2. Components of the Computer

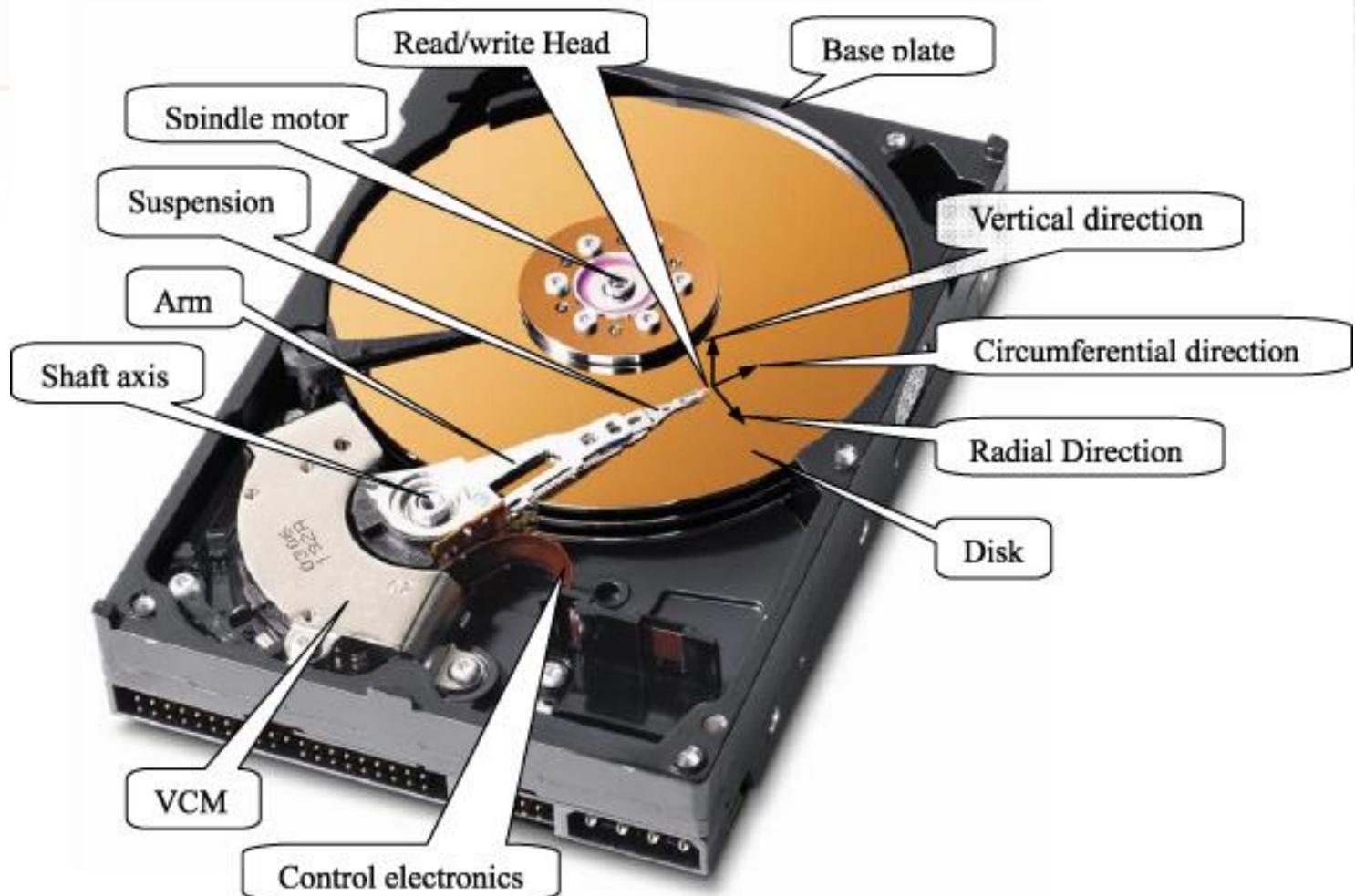
## Storage Devices – Primary storage

### Memory – Main memory

- Memory is byte addressable
- Each byte has a unique address
- Addresses start from zero and increment sequentially.
- Memory Refresh – Memory refresh is the process of periodically read data from an area of computer memory and immediately writing the read information to the same area with no modifications.

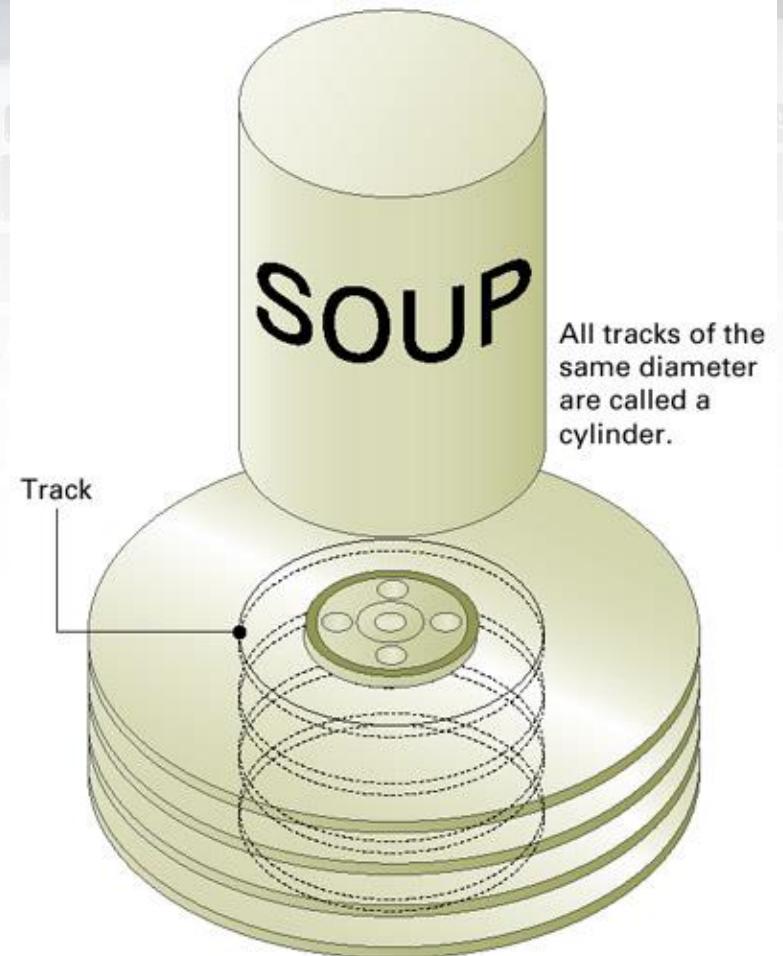
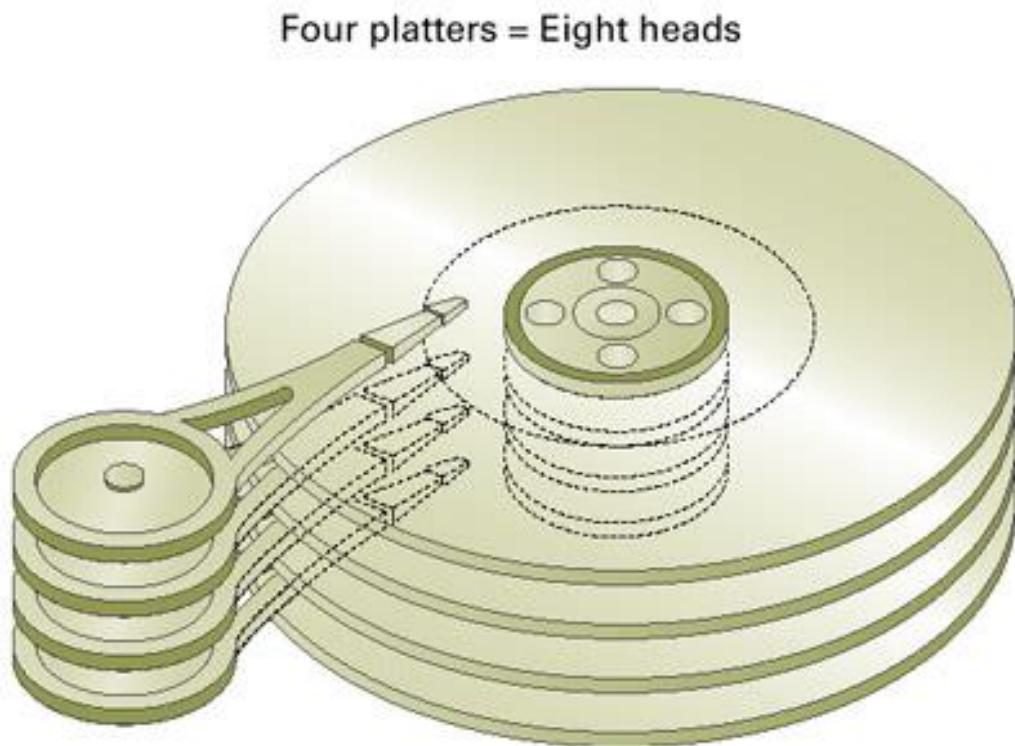
# 2. Components of the Computer Storage Devices – Secondary storage

## Hard disk



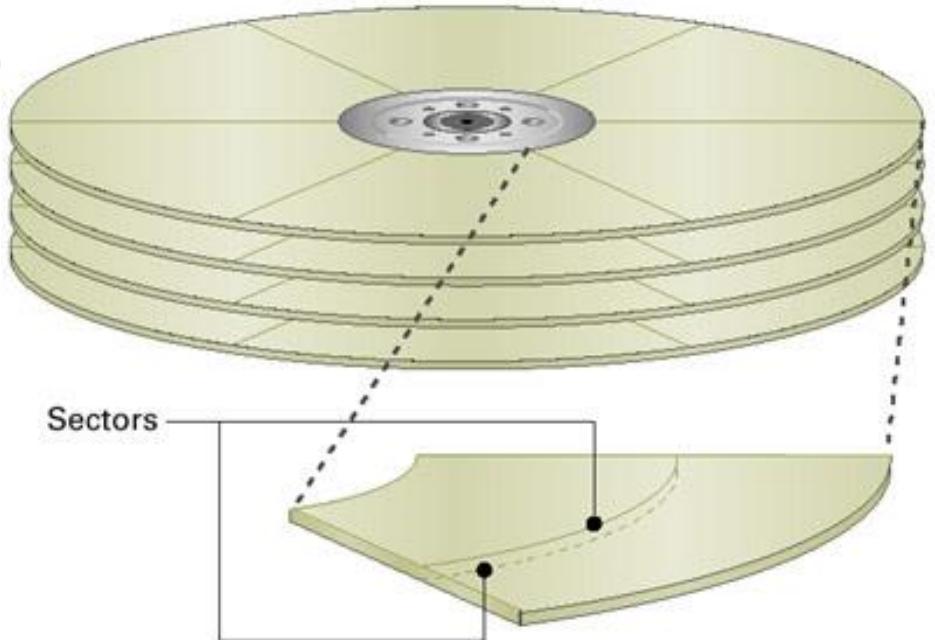
# 2. Components of the Computer Storage Devices – Secondary storage

## Hard disk

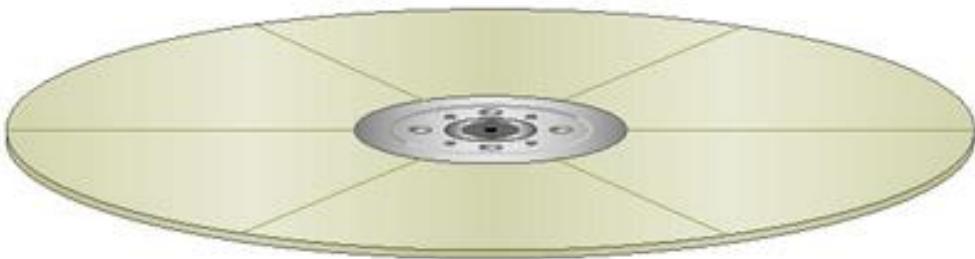


# 2. Components of the Computer Storage Devices – Secondary storage

## Hard disk



Six sectors per track  
(sectors/track)



## 2. Components of the Computer Storage Devices – Secondary storage

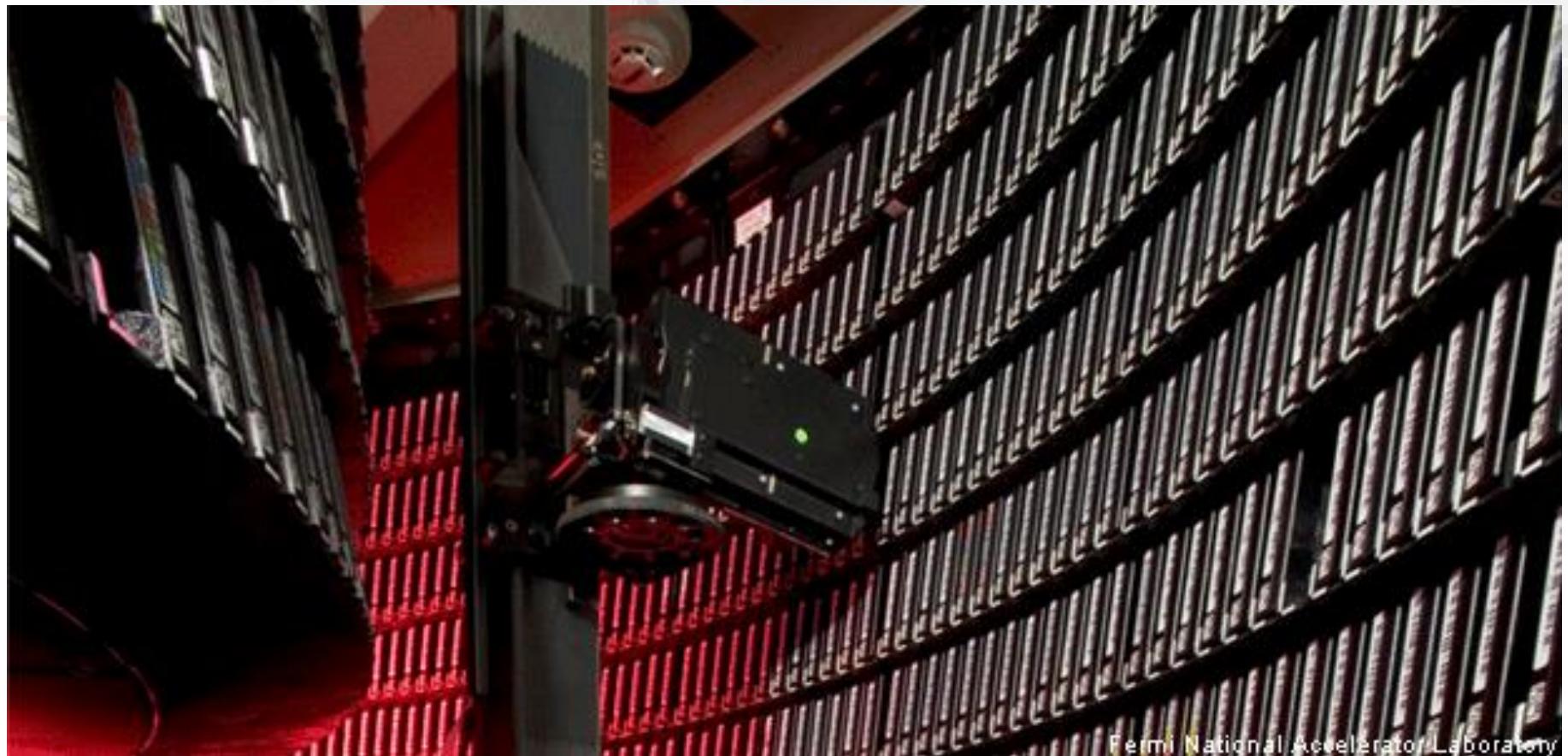
### Hard disk – performance parameters

- **Access time** – seek time + rotational delay+ transfer time
- **Seek time** – track selection time (moving the head on the desired sector on the track)
- **Rotational delay** – the time it takes for the head to reach the beginning of the sector
- **Transfer time** – the time required to transfer data

## 2. Components of the Computer Storage Devices – Tertiary Storage

- Typically it involves a robotic mechanism which will mount (insert) and dismount removable mass storage media into a storage device.
- It is a comprehensive computer storage system that is usually very slow, so it is usually used to archive data that is not accessed frequently.
- This is primarily useful for extraordinarily large data stores, accessed without human operators.

# 2. Components of the Computer Storage Devices – Tertiary Storage



Fermi National Accelerator Laboratory

## 2. Components of the Computer Storage Devices – Offline Storage

- Also known as Disconnected storage
- Is a computer data storage on a medium or a device that is not under the control of a processing unit
- It must be inserted or connected by a human operator before a computer can access it again
- Examples
  - Floppy Disk
  - CD/DVD/Blue-ray
  - USB Flash Drive
  - Memory Cards

# 2. Components of the Computer Storage Devices – Offline Storage

- A soft magnetic disk.
- Floppy disks are portable.
- Floppy disks are slower to access than hard disks and have less storage capacity, but they are much less expensive.
- Can store data up to 1.44MB.
- Two common sizes: 5 ¼" and 3 ½".



## Memory Card

- An electronic flash memory storage disk commonly used in consumer electronic devices such as digital cameras, MP3 players, mobile phones, and other small portable devices.
- Memory cards are usually read by connecting the device containing the card to your computer, or by using a USB card reader.



Secure Digital card (SD)

MINI SD

Compact Flash

Memory Stick

- A small, portable flash memory card that plugs into a computer's USB port and functions as a portable hard drive.
- Flash drives are available in sizes such as 256MB, 512MB, 1GB, 5GB, and 16GB and are an easy way to transfer and store information.



# 2. Components of the Computer Storage Devices – Offline Storage

	CD	DVD
Stands for	Compact Disc ✓	Digital Versatile Disc ✓
Purpose	CDs are made with the purpose of holding audio files as well as program files.	DVDs are made with the purpose of holding video files, movies, substantial amount of programs, etc. ✓
Media type	Optical disc ✓	Optical disc ✓
Capacity	Typically up to 700 MiB (up to 80 minutes audio)	DVD can range from 4.7 GB to 17.08 GB. ✓
Types	CD-R, CD-RW, CD-Text ETC. ✓	DVD-RW, DVD+RW, DVD-RAM and Blu-Ray. ✓

# 2. Components of the Computer Storage Devices

## Other Storage techniques

### Cloud Storage

- Cloud storage means "the storage of data online in the cloud," wherein a data is stored in and accessible from multiple distributed and connected resources that comprise a cloud.
- Cloud storage can provide the benefits of greater accessibility and reliability; rapid deployment; strong protection for data backup, archival and disaster recovery purposes.

## 2. Components of the Computer Storage Devices

### Other Storage techniques

#### Cloud Storage

- Examples:
  - ✓ Google Drive
  - ✓ Flickr
  - ✓ Microsoft Sky Drive



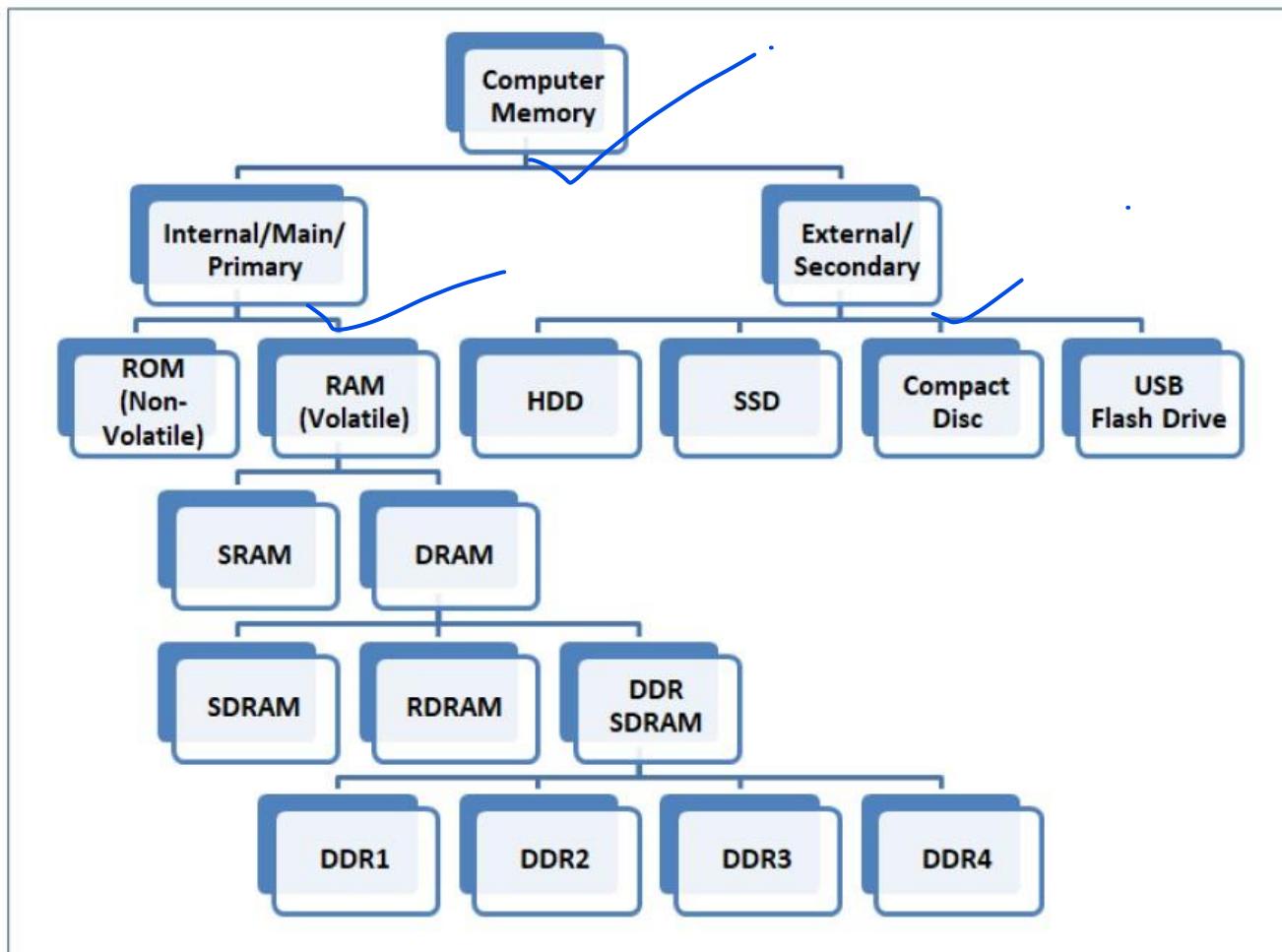
Google Drive



SkyDrive®

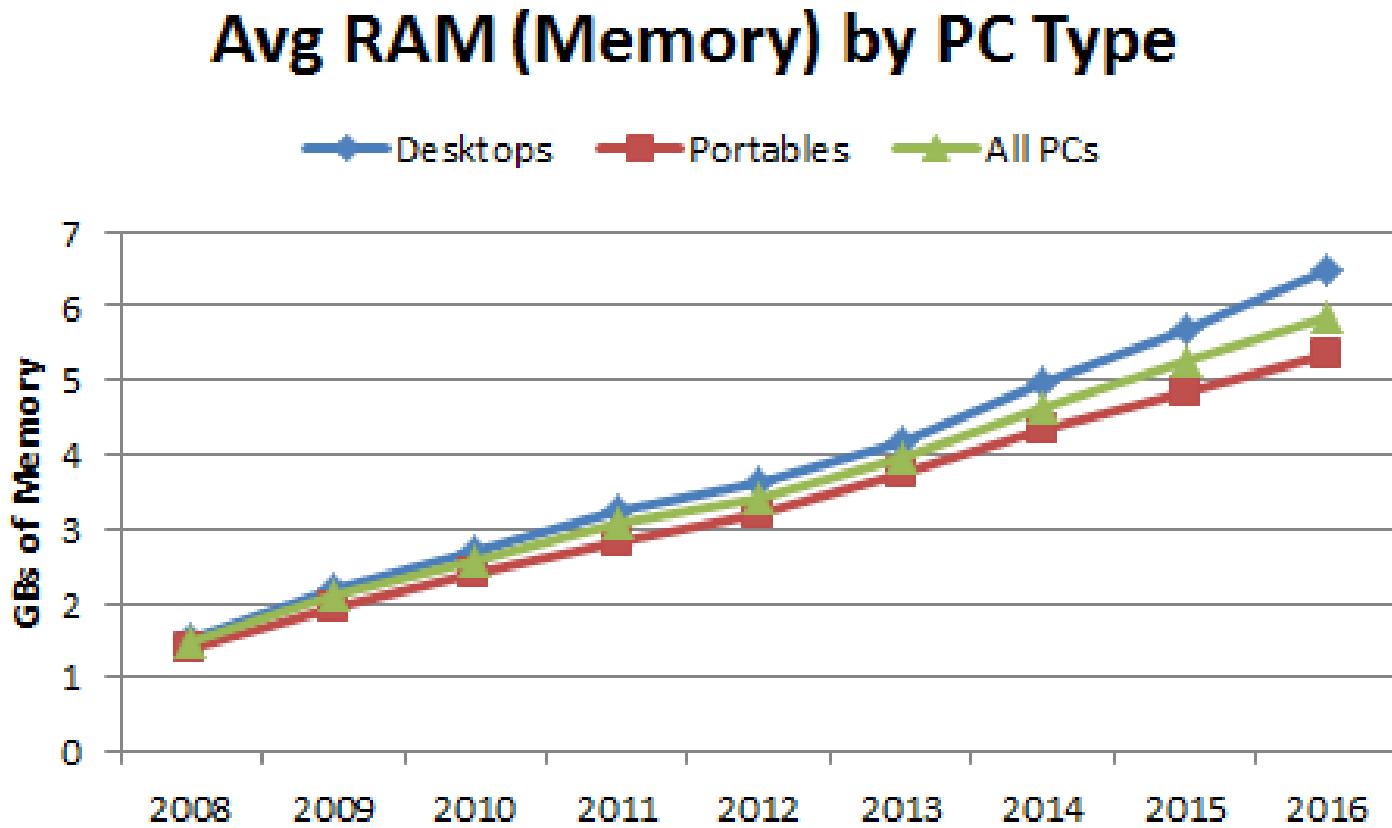
# 2. Components of the Computer Storage Devices

## Evolution of storages

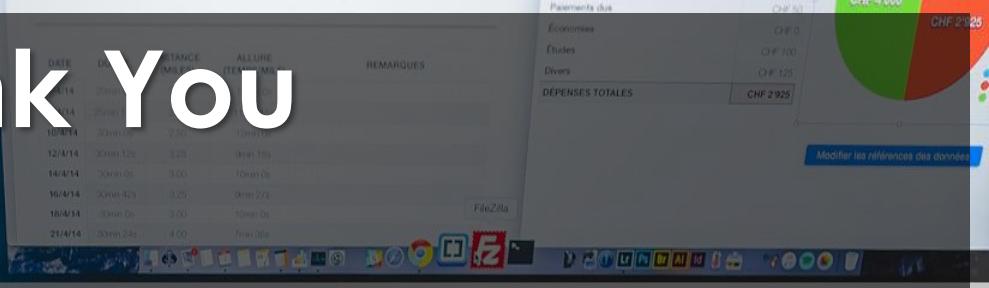


# 2. Components of the Computer Storage Devices

## Evolution of storages



# Thank You



## Lecture01 - Post-lecture activities

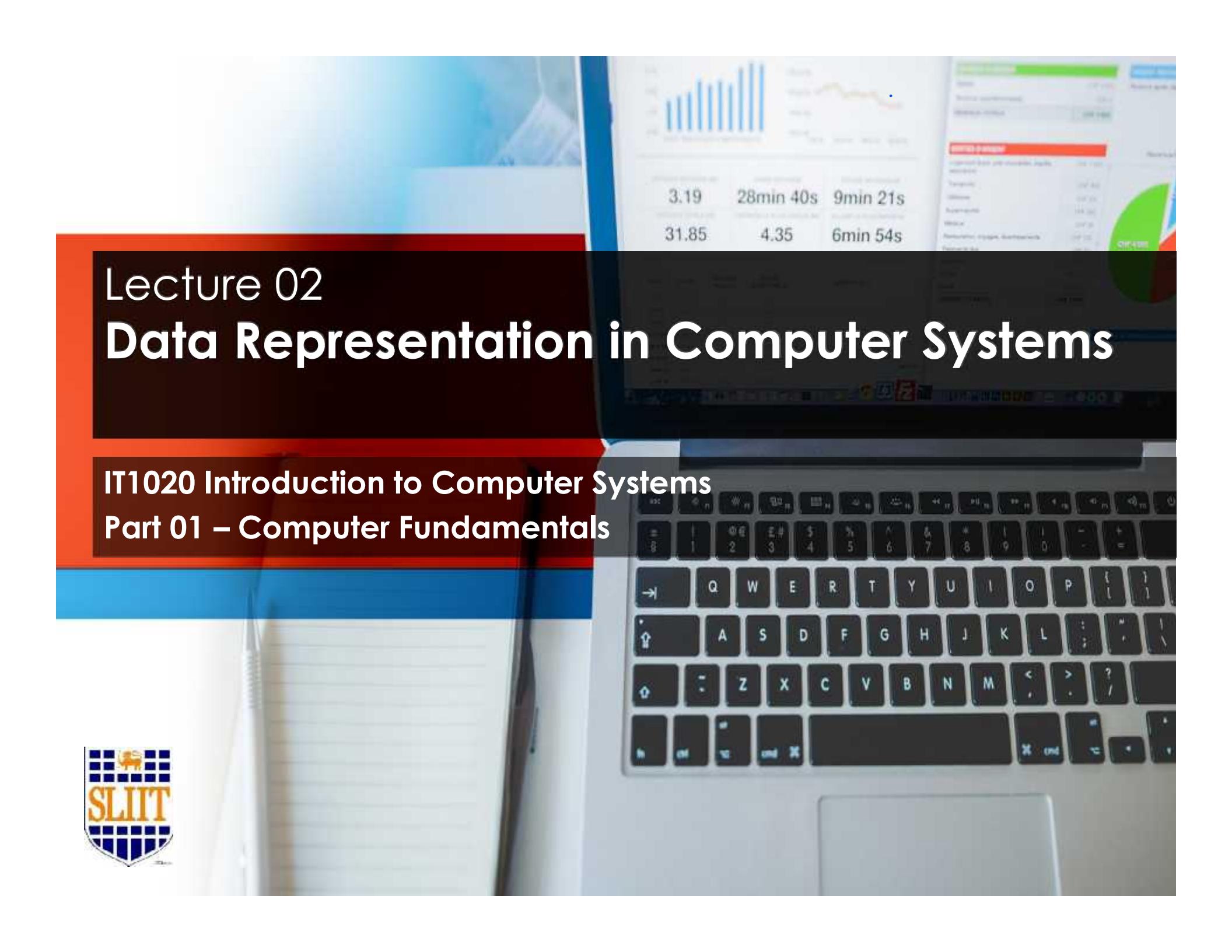
Find facts and create graph of  
CPU Ops, Speed, Cores, Disk space, RAM speed

## Lecture 02

Data representations

## Lecture02 - Pre-lecture activities

Data & Information, Different Types of Data,  
ASCII/Unicode



Lecture 02

# Data Representation in Computer Systems

IT1020 Introduction to Computer Systems

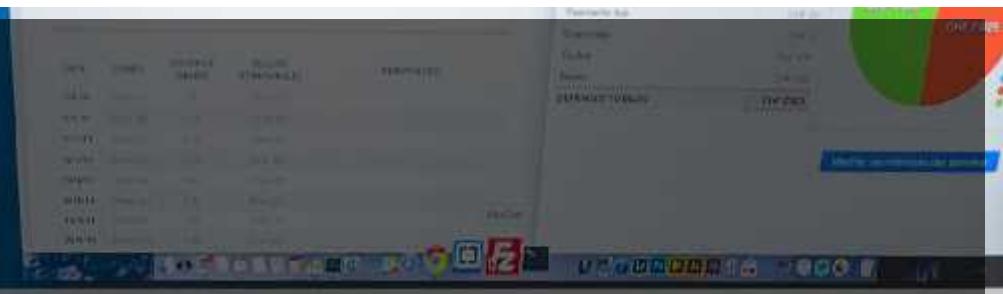
Part 01 – Computer Fundamentals



# Pre-lecture activities

- Data & Information
- Different Types of Data
- ASCII/UNICODE

# Content



- 1. Data and Information
- 2. Data types
- 3. Data Representation in Computers
- 4. Error checking
- 5. Data Compression

# 1. Data and Information

## Related terms

### What is a System?

- A system is a group of interrelated parts working together to produce output.
- A system takes input, processes it, and produces output.



### Who uses IS?

- IS users
- Input
- Output

**Information System**

# 1. Data and Information

## Related terms

### Software Engineering

- The engineering discipline of constructing Information Systems

### Information and Communication Technology (ICT)

- The technology used to engineer ISs

# 1. Data and Information

- Data is a collection of raw facts
  - Numbers
  - Words
  - Measurements
  - Observations
  - Description of things
- Information is the processed outcome of data.
  - It is derived from data

Saman	Kamal
Nuwan	
Maths	English
60	50
90	70
40	100

Saman Maths	60
Kamal Maths	100
Nuwan Maths	40
Saman English	50
Kamal English	70
Nuwan English	80

# 1. Data and Information

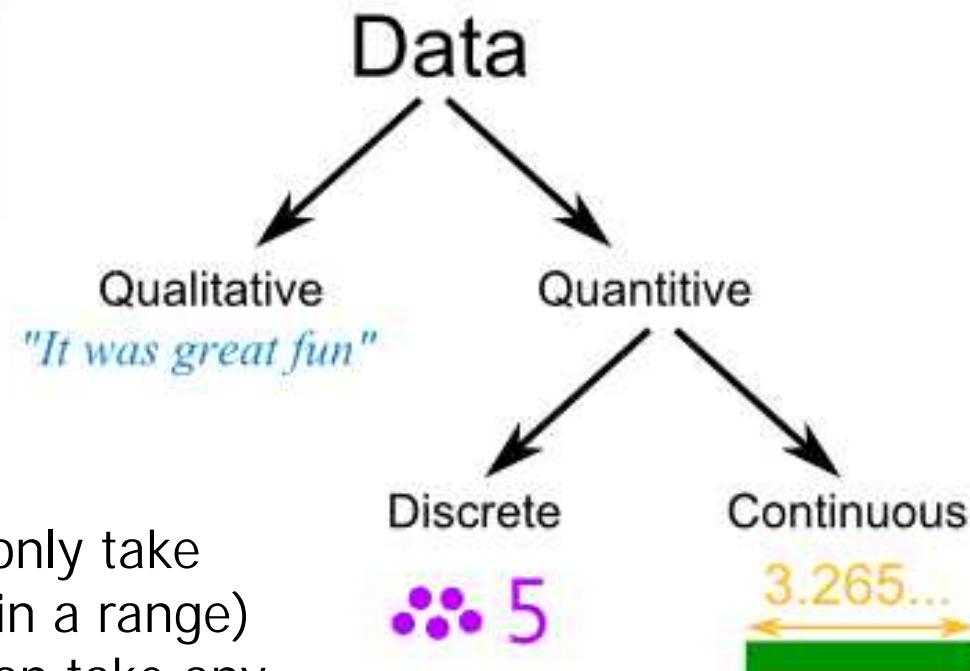
## What is Information?

- Information provide meaningful values to the receiver
  - ✓ **Timely** - Information should be available when required
  - ✓ **Accuracy** - Information should be correct
  - ✓ **Completeness** - Information should be complete

## 2. Data Types

### Qualitative vs. Quantitative Data

- Quantitative Data : Numerical Information
- Qualitative Data : Descriptive data



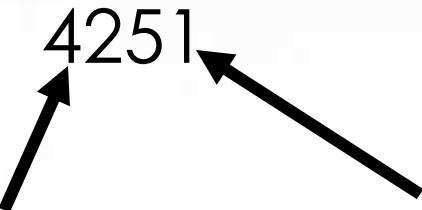
Discrete data: Can only take certain values (within a range)  
Continuous Data: Can take any value (within a range)

## 2. Data Types

### Numeric data

- **Integer numbers**

- Whole numbers, + or -

- 

4251

Most significant digit      Least significant digit

-582

- **Real (Decimal) numbers**

- All numbers including everything between integers
  - 0.23, 0,  $5\frac{1}{2}$ , -2.3,

## 2. Data Types

### Character Data

**Single character**

**Numeric**

0, 1, 2, .....

**Alphabetic**

A, B, C, a, b, c, .....

**Special**

#, @, %, (, \$, &, .....

**Multiple characters  
(String)**

**Numeric**

349, 53.781, .....

**Alphabetic**

Cat, Software, .....

**Special**

(#%\$), .....

**Alpha-numerical**

DIT451789#, .....

## 2. Data Types Summary

What are the  
other data types?

## 2. Data Representation

- Data representation refers to the form in which data is **stored**, **processed**, and **transmitted**
- Devices such as smartphones, iPods, and computers store data in digital formats that can be handled by **electronic circuitry**.
- **Digitization** is the process of converting information, such as text, numbers, photo, or music, into digital data that can be manipulated by electronic devices.

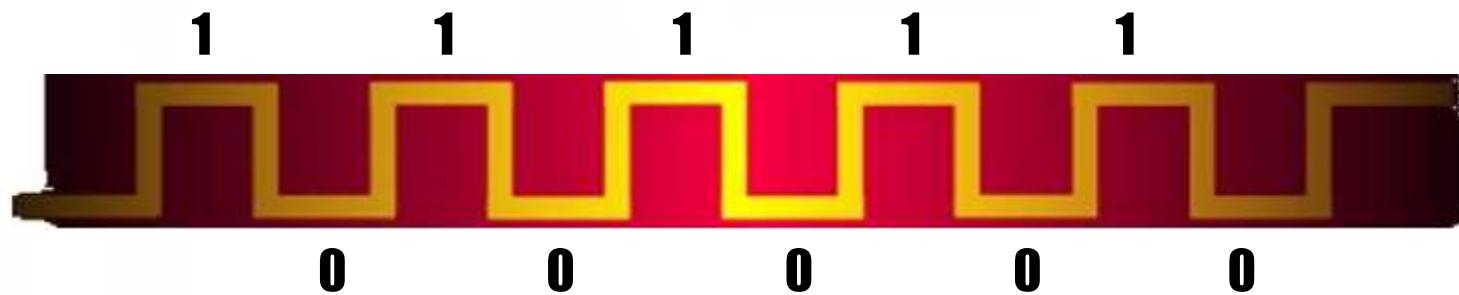
## 2. Data Representation

- The **Digital Revolution** has evolved through four phases, beginning with big, expensive, standalone computers, and progressing to today's digital world in which small, inexpensive digital devices are everywhere.

# 3. Data Representation in Computers

## How do computers represent data?

- Computers are digital
  - Recognize only **two discrete states**: **on** or **off**
  - Computers are electronic devices powered by electricity, which has only two states, on or off
  - Binary number system (0,1) is used for processing



What is the meaning of “Digital”?

What are the advantages of using binary number system? <sub>14</sub>

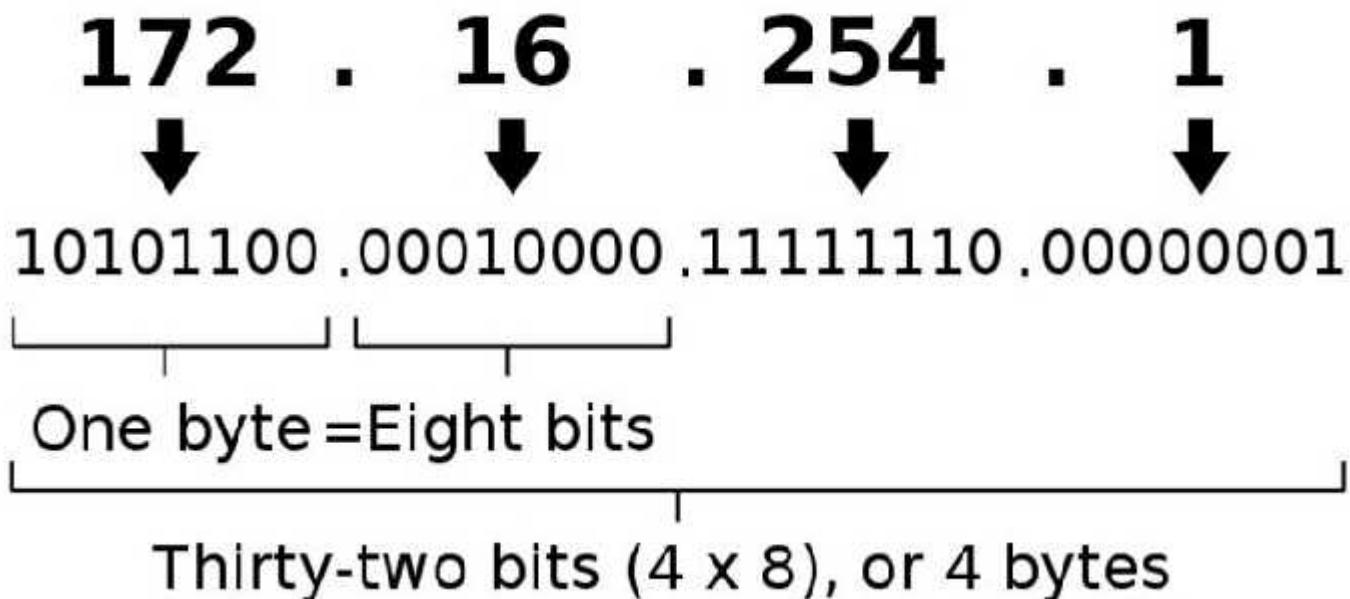
# 3. Data Representation in Computers

## How do computers represent data?

Example of the use of binary numbers in data communication

- IP addressing

An IPv4 address (dotted-decimal notation)



# 3. Data Representation in Computers

## How do we store data in a computer?

honda	789	234	baldwin	112	1545
toyota	1470	366	oscar	189	2141
dallas	2932	990	love	415	2989
miami	211	230	American	111	1096
coupe	599	334	history	272	2716
sedan	1693	524	white	284	3058
2000	896	641	black	221	2327
2004	3711	1124	time	145	907
black	2391	972	john 2007	391	4545
gold	709	508	action 2007	272	2601



Binary code representation of the data:

```
1100110011001000000  
00110010101110010001  
10000110000101110000  
00100110000101110000  
1101101110011001110011  
0111001001011101111111  
001000000111000001100  
1000001101001011101111  
001100001011100000110  
1110011001110010000000  
0011001010111001000101
```



### 3. Data Representation in Computers

## Binary number system

- The memory is made up of BITS and BYTES
- Single bit can hold a binary digit (0 or 1)
  - 8 Bits = 1 Byte
  - 1024 Bytes = 1KB
  - 1024 KB = 1 MB      104 KB: Kilobyte (KB or Kbyte)
  - 1024 MB = 1 GB      50 Mbps: Megabits/sec
  - 1024 GB = 1 TB

$$2^{10} = 1024$$

Terminology related to bits and bytes is extensively used to describe **storage capacity** and **network access speed**.

### 3. Data Representation in Computers

#### Binary number system

- Each BYTE can be addressed uniquely
- When the address is expressed in Binary, the number of maximum BITS used to write the address specifies the total number of locations available
- If **n** number of BITS are available then the total number of locations available is  $2^n$
- If we have 32 BITS then we can have 4GB of Memory ( $2^{32} = 4 \text{ GB}$ )

**What is the maximum amount of memory for 64bit computer?**

128 GB

# 3. Data Representation in Computers

## BCD (Binary Coded Decimal)

- 4 bit code (for numeric values only)

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000
11	0001 0001
12	0001 0010

# 3. Data Representation in Computers

## ASCII (American Standard Code for Information Interchange)

- ASCII
  - 7 bit code for all 128 characters
  - A=1000001
  - Fundamentally, computers just deal with numbers. The letters and other characters are stored by assigning a number for each one
- Extended ASCII
  - This system is an 8-bit system and allows the system to store up to 256 different characters

Symbol	Decimal	Binary
A	65	01000001
B	66	01000010
C	67	01000011

Symbol	Decimal	Binary
a	97	01100001
b	98	01100010
c	99	01100011

# 3. Data Representation in Computers

## ASCII

Symbol	Decimal	Binary
A	65	01000001
B	66	01000010
C	67	01000011
D	68	01000100
E	69	01000101
F	70	01000110
G	71	01000111
H	72	01001000
I	73	01001001
J	74	01001010
K	75	01001011
L	76	01001100
M	77	01001101
N	78	01001110
O	79	01001111
P	80	01010000
Q	81	01010001
R	82	01010010
S	83	01010011
T	84	01010100
U	85	01010101
V	86	01010110
W	87	01010111
X	88	01011000
Y	89	01011001
Z	90	01011010

Symbol	Decimal	Binary
a	97	01100001
b	98	01100010
c	99	01100011
d	100	01100100
e	101	01100101
f	102	01100110
g	103	01100111
h	104	01101000
i	105	01101001
j	106	01101010
k	107	01101011
l	108	01101100
m	109	01101101
n	110	01101110
o	111	01101111
p	112	01110000
q	113	01110001
r	114	01110010
s	115	01110011
t	116	01110100
u	117	01110101
v	118	01110110
w	119	01110111
x	120	01111000
y	121	01111001
z	122	01111010

# 3. Data Representation in Computers

## Unicode

- **Unicode** provides a unique number for every character
  - no matter what the platform
  - no matter what the program
  - no matter what the language

0D85	අ	SINHALA LETTER AYANNA = sinhala letter a
0D86	අ	SINHALA LETTER AAYANNA = sinhala letter aa
0D87	ඇ	SINHALA LETTER AEYANNA = sinhala letter ae
0D88	ඈ	SINHALA LETTER AEEYANNA = sinhala letter aae
0D89	ඉ	SINHALA LETTER IYANNA = sinhala letter i
0D8A	ඊ	SINHALA LETTER IIYANNA

0B85	அ	TAMIL LETTER A
0B86	ஆ	TAMIL LETTER AA
0B87	இ	TAMIL LETTER I
0B88	ஈ	TAMIL LETTER II
0B89	ு	TAMIL LETTER U
0B8A	ா	TAMIL LETTER UU
0B8B	ஃ	<reserved>
0B8C	ஃ	<reserved>
0B8D	ஃ	<reserved>
0B8E	எ	TAMIL LETTER E
0B8F	ே	TAMIL LETTER EE

How many bits are used to represent a character?  
How many characters can be represented?

22

255

# 3. Data Representation in Computers

## Unicode

### Independent vowels

0D85	අ	SINHALA LETTER AYANNA = sinhala letter a
0D86	අ	SINHALA LETTER AAYANNA = sinhala letter aa
0D87	ඇ	SINHALA LETTER AEYANNA = sinhala letter ae
0D88	ඈ	SINHALA LETTER AEEYANNA = sinhala letter aac
0D89	ඉ	SINHALA LETTER IYANNA = sinhala letter i
0D8A	ඊ	SINHALA LETTER IIYANNA

	0D8	0D9	0DA
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
A			

	0B8	0B9	0BA	0BB
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
A				

### Independent vowels

0B85	அ	TAMIL LETTER A
0B86	ஆ	TAMIL LETTER AA
0B87	இ	TAMIL LETTER I
0B88	ஈ	TAMIL LETTER II
0B89	உ	TAMIL LETTER U
0B8A	ஊ	TAMIL LETTER UU
0B8B	<reserved>	
0B8C	<reserved>	
0B8D	<reserved>	
0B8E	எ	TAMIL LETTER E
0B8F	ஏ	TAMIL LETTER EE

## 4. Error checking Parity

- The parity is used in error checking, to find errors that may occur during data storing/transmission
- A parity bit is a single bit added to a binary data transmission used to indicate if whether the 0's and 1's within that data transmission is an even or odd number

# 4. Error checking Parity

- There are two modes of parity
  - Odd parity:- The number of 1-bits (including the parity bit) must be an odd number
  - Even parity:- The number of 1-bits (including the parity bit) must be an even number
- A single bit is appended to each data chunk (either as the least or most significant bit)
  - Makes the number of 1 bits even/odd

## Example: even parity

1000000(1)  
1111101(0)  
1001001(1)

## Example: odd parity

1000000(0)  
1111101(1)  
1001001(0)

## 4. Error checking Parity – Example

- You receive a binary word “11000101”
  - **odd parity** is used
  - Most significant bit is used for parity
- Is the binary word correct?
- The answer is no:
  - There are 4 1-bits, which is an even number
  - We are using an odd parity
  - So there must be an error.

## 4. Error checking Parity – Example2

- Assume we are using **even parity** with 7-bit ASCII.
- The letter V in 7-bit ASCII is encoded as 0110101.
- How will the letter V be transmitted with parity?
  - Because there are four 1s (an even number), parity is set to zero.
  - This would be transmitted as: 0110101(0).

## 4. Error checking Parity – Exercises

- Add the odd parity bit as the least significant bit for the following data

– 0100101 (0)

– 1101101 (0)

– 1011001 (1)

– 0011010 (0)

– 1111011 (1)



## 4. Error checking Parity – Exercises

- Verify the correctness of the following data with the even parity bit on the most significant bit
  - 01010001 Wrong
  - 10100101 Correct
  - 01101101 Wrong
  - 01111011 Correct
  - 10011010 Correct

# 5. Data Compression

- To reduce file size and transmission times, digital data can be **compressed**.
- Data compression refers to any technique that recodes the data in a file so that it contains **fewer bits**.
- Compression is commonly referred to as **“zipping”**
- The process of reconstituting zipped files is called **extracting** or **unzipping**.
- Compressed files may end with a **.zip**, **.gz**, **.pkg**, or **.tar.gz**

# Summary

- 
1. Data and Information
  2. Data types
  3. Data Representation in Computers
  4. Error checking
  5. Data Compression



# Thank You

## Lecture 02 – Post-lecture activities

### Parity & Binary Exercises

## Lecture 03

### Computer Architecture

## Lecture 03 – Pre-lecture activities

### Diagram of 8086, Bus Architecture with 3 bus types

# **COMPUTER ARCHITECTURE**

**IT1020 Introduction to Computer Systems  
Part 01 – Computer Fundamentals**



# CONTENT

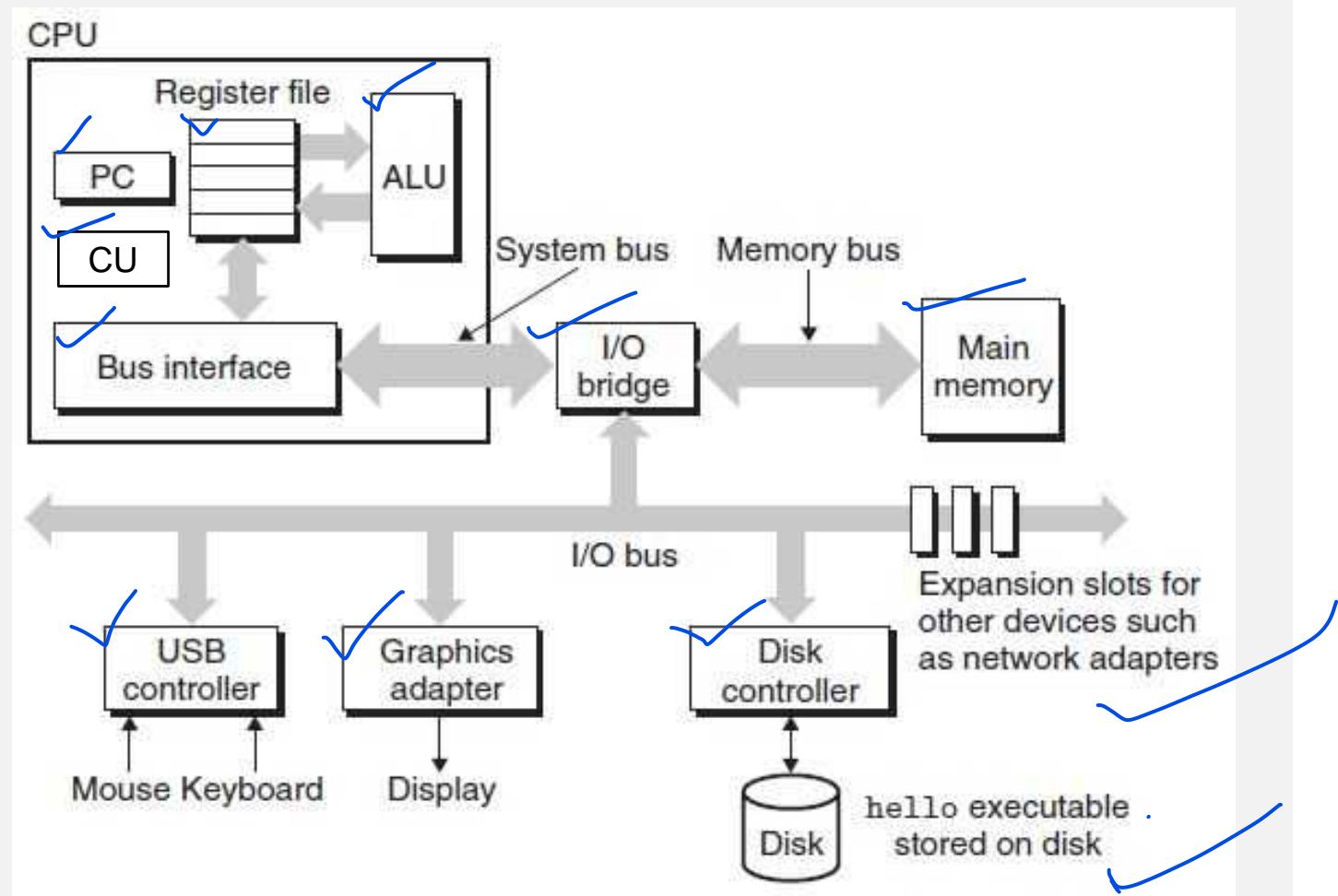
- This lecture covers:
  - fundamentals of computer organization,
  - the heart of a computer system, CPU, its structure and functionality,
  - how CPU is seen by programmers and CPU design engineers
  - instruction set architecture of a processor.

# COMPUTER ORGANIZATION

- Different groups of engineers see computer different manner
- **Hardware Engineers** see how different components work, their configuration, how they are organized – **Computer Organization**
- **Software Engineers** see how computer can be used for different applications, how their high level language codes can be executed in machine, How easily it can be used for programs – **Instruction Set Architecture**

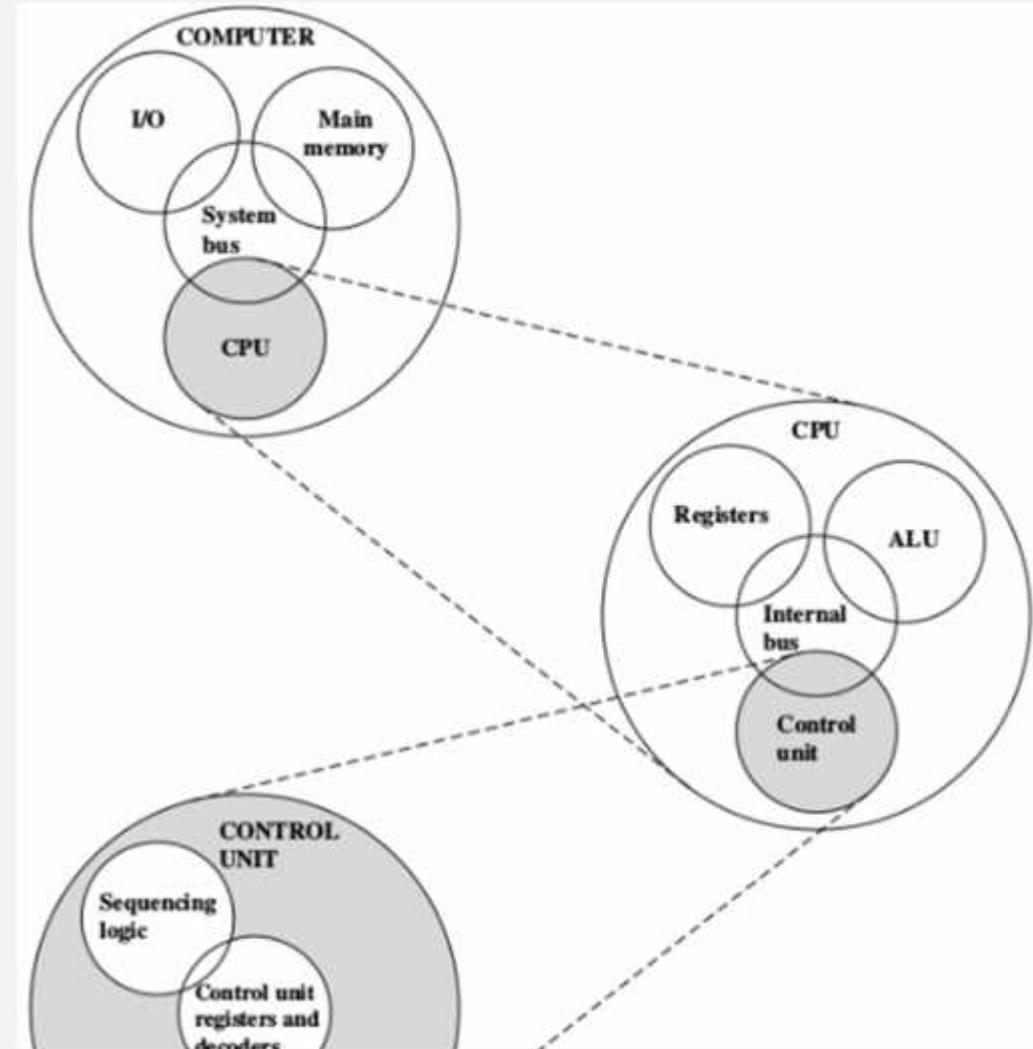
# COMPUTER ORGANIZATION

Computer organization is concerned with the way the hardware components operate and the way they are connected together to form the computer system.



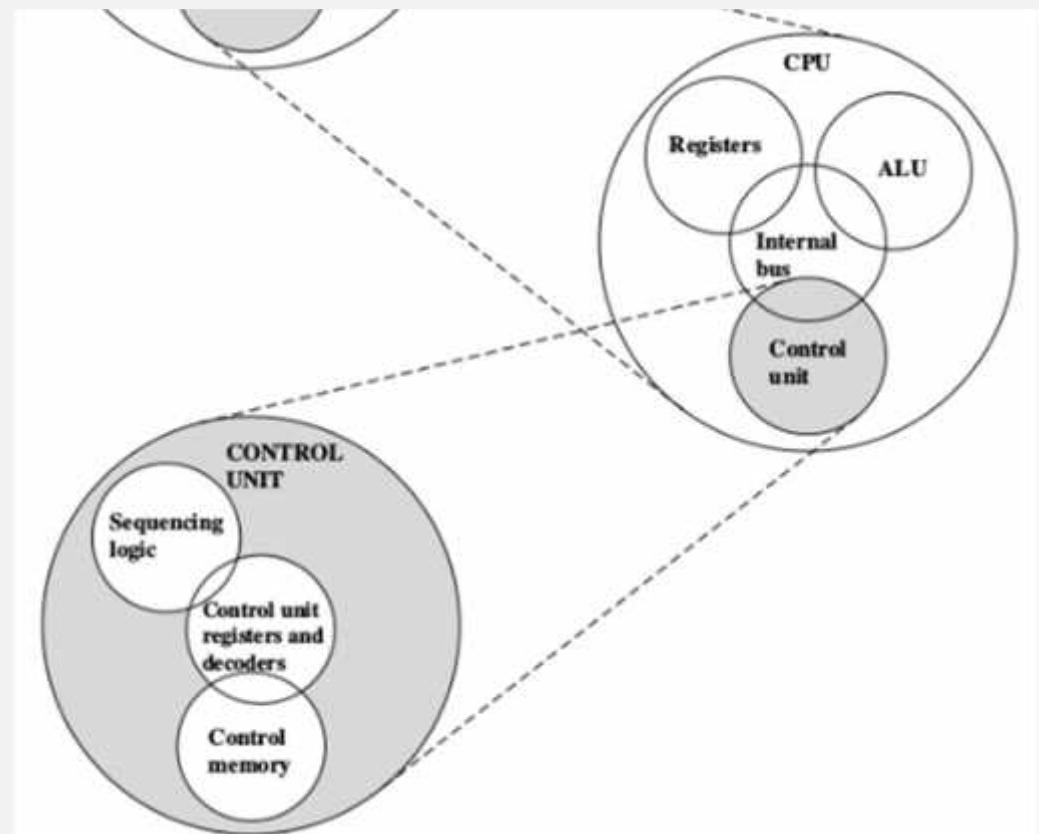
# COMPUTER ORGANIZATION

- **CPU:** Controls the operation of the computer and performs its data processing functions.
- **Main Memory:** Stores data and Instructions (Programs)
- **I/O:** Moves the data between the computer and its external environment.
- **System Interconnections (System Bus):** Mechanism that provides for communication among CPU, MM, and I/O.

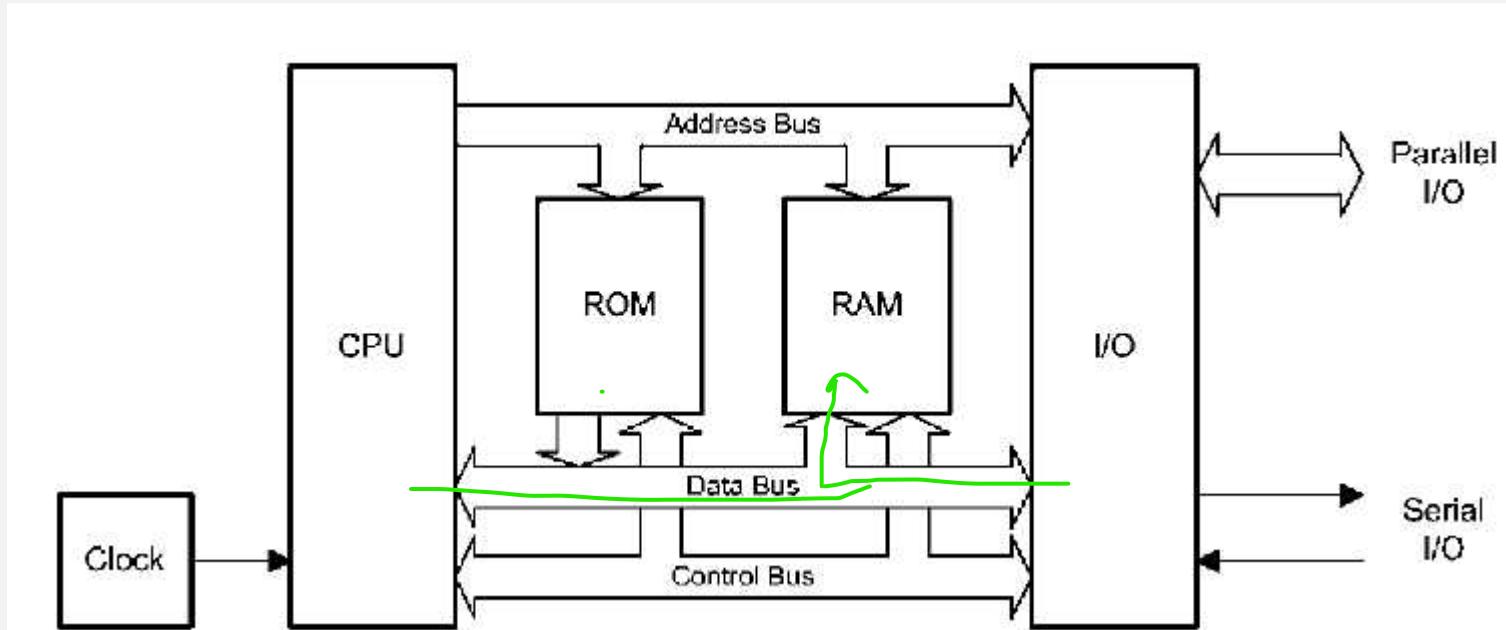


# COMPUTER ORGANIZATION

- **ALU:** Performs the computer's data processing functions
- **Registers:** Provides storage internal to the CPU
- **Control Unit:** A digital circuitry within the processor that coordinates the sequence of data movements into, out of, and between a processor's sub-units

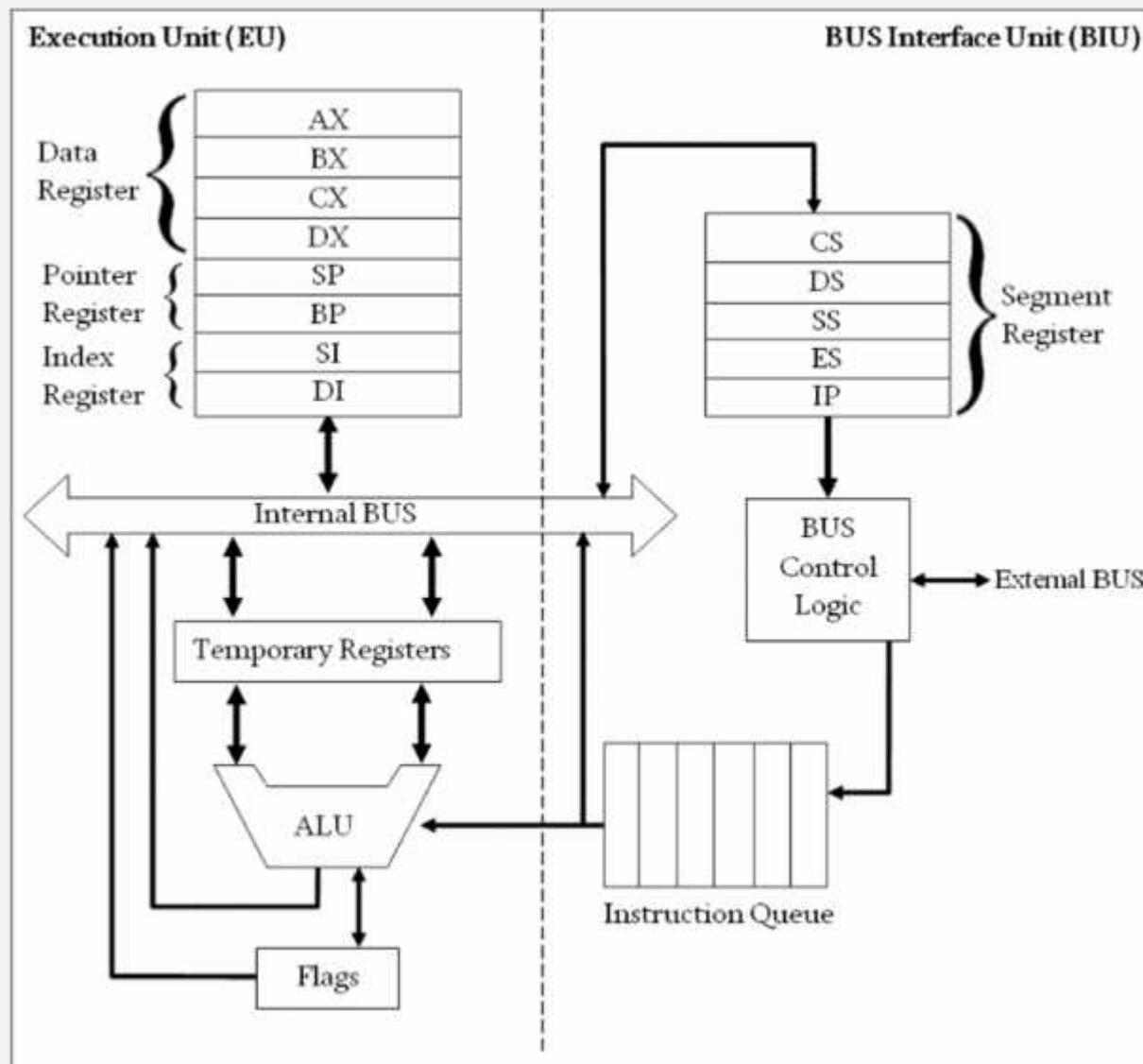


# MODERN COMPUTER



- **Address Bus:** CPU specify the address of MM or I/O, Uni-directional
- **Data Bus:** Transfer data/instruction to and from CPU
- **Control Bus:** Carries commands from CPU and status signals to CPU

# 8086 ARCHITECTURE



# **8086 REGISTERS**

**They are grouped into several categories as follows:**

- Four **general-purpose** registers, AX, BX, CX, and DX.
- Four **special-purpose** registers, SP, BP, SI, and DI.
- Four segment registers, CS, DS, ES, and SS.
- The Program Counter/Instruction Pointer
- The status flag register, FLAGS

# 8086 REGISTERS

General Purpose	Special Purpose (Addressing registers)
<ul style="list-style-type: none"><li>• AX is the <b>accumulator</b>. It is used for all input/output operations and some arithmetic operations.</li><li>• BX is the <b>base register</b>. It can be used as an address register.</li><li>• CX is the <b>count register</b>. It is used by instructions which require to count (Ex: Looping)</li><li>• DX is the <b>data register</b>. It is used for some input/output and also when multiplying and dividing.</li></ul>	<ul style="list-style-type: none"><li>• The <b>addressing registers</b> are used in memory addressing operations, such as holding the source address of the memory and the destination address.</li><li>• SI is the <b>source index</b> and is used with extended addressing commands.</li><li>• DI is the <b>destination index</b> and is used in some addressing modes.</li><li>• BP is the <b>base pointer</b>.</li><li>• SP is the <b>stack pointer</b>.</li></ul>

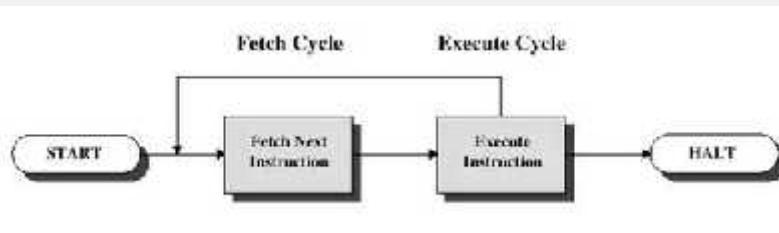
# 8086 REGISTERS

- The two status registers have 16 bits and are called the **Instruction Pointer (IP)** or **Program Counter (PC)** and the flag register (F)
- IP is the instruction pointer and contains the **address of the next instruction** of the program. This is called as Program Counter (PC)
- **Flag register** holds a collection of 16 different **conditions**. (Ex: result is zero or not, there is a “carry” etc)

# 8086 REGISTERS

- **Segments registers:** There are four areas of memory called segments, each of which are 16 bits and can thus address up to 64KB (from 0000h to FFFFh).
  - **Code segment** (cs register), where the program code is stored.
  - **Data segment** (ds register), where data from the program is stored
  - **Stack segment** (ss register), where the stack is stored.
  - **Extra segment** (es register), a spare segment
- All addresses are with reference to the segment registers.

# INSTRUCTION CYCLE



- The basic function performed by a computer is **execution of a program**
- Program is a well defined set of Instructions
- Processing a single instruction is called **instruction cycle**
- It has two steps referred as **fetch cycle** and **execute cycle**

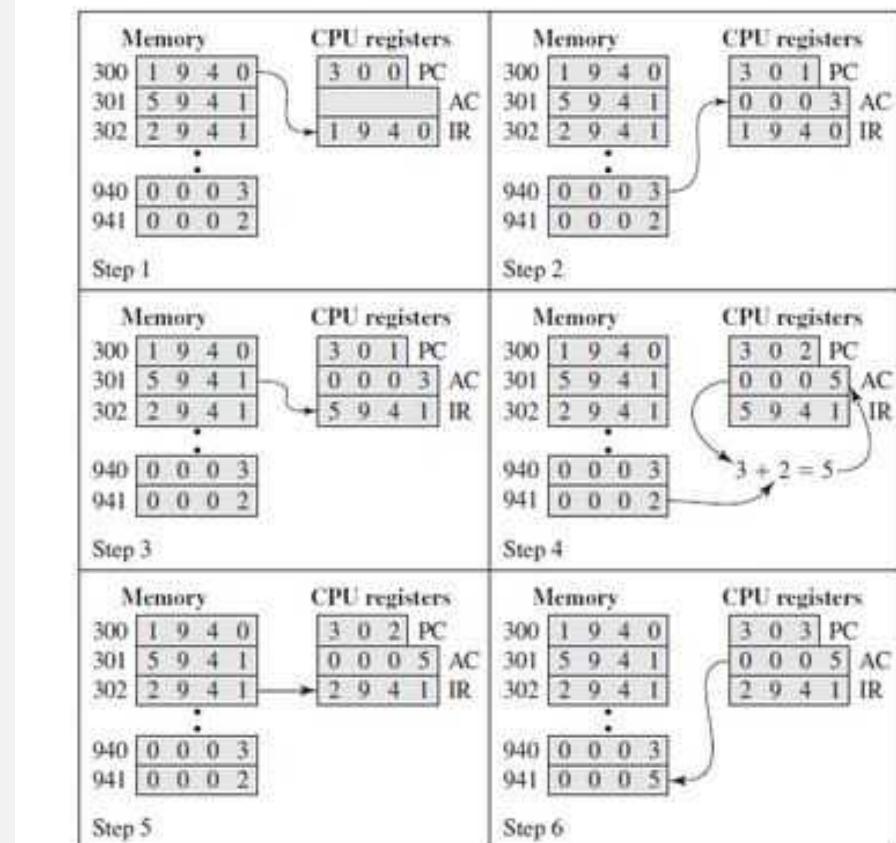
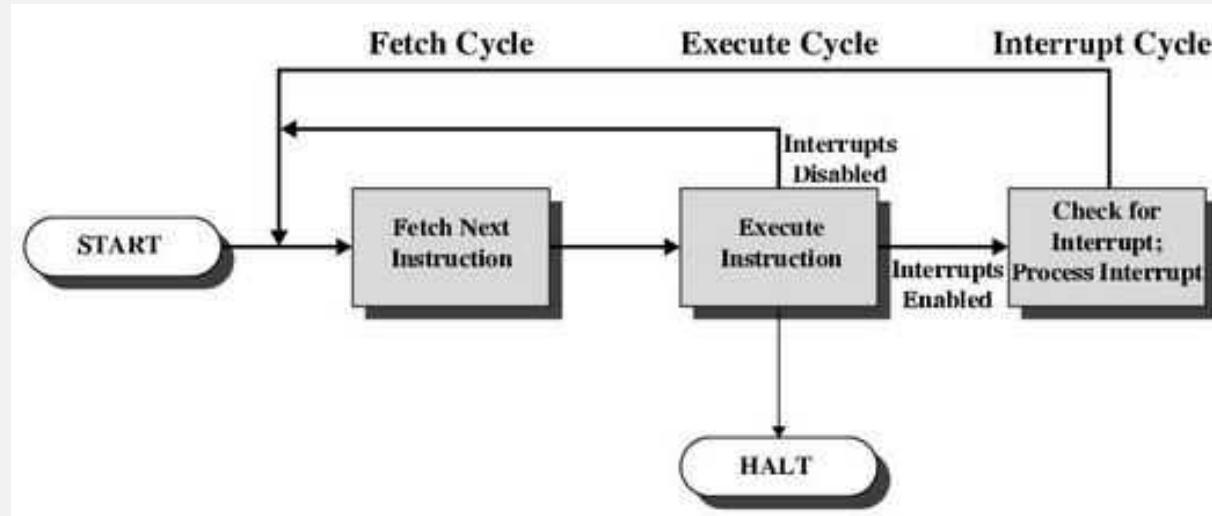


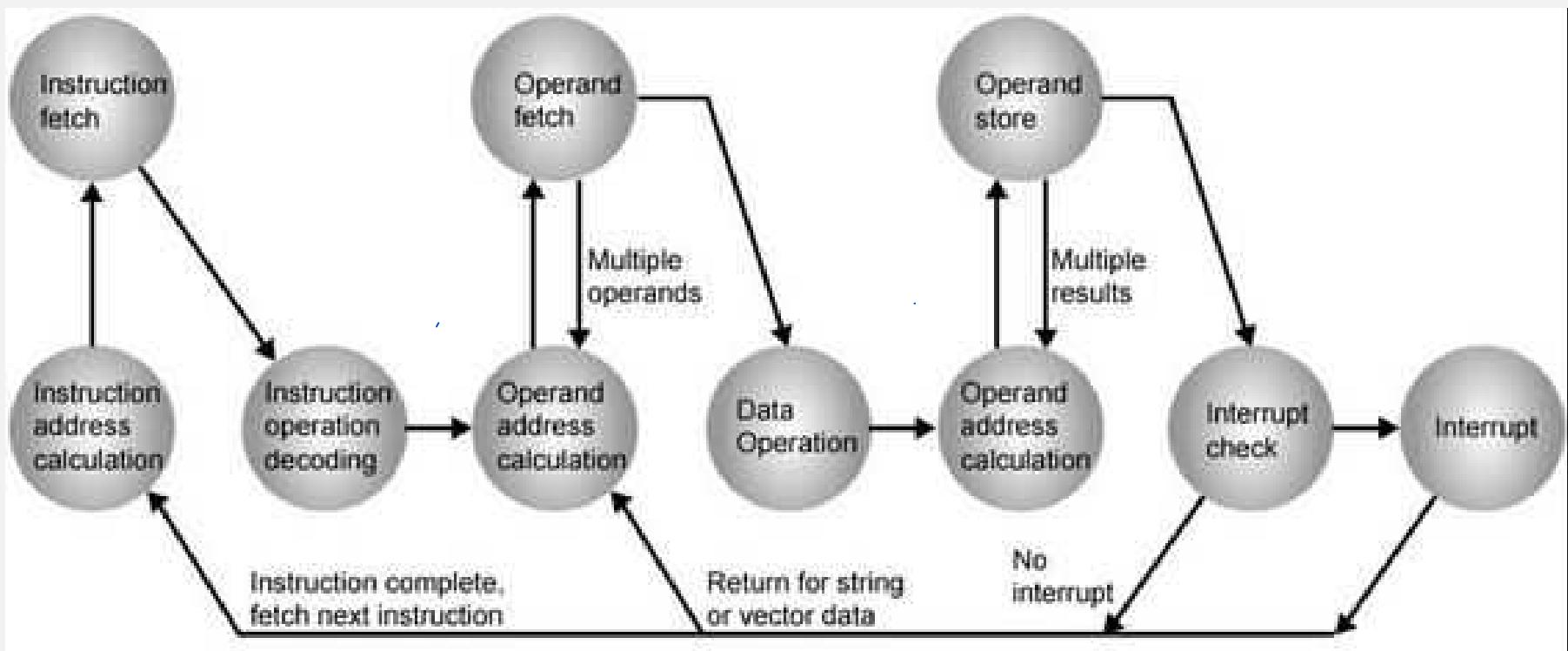
Figure 3.5 Example of Program Execution (contents of memory and registers in hexadecimal)

# INSTRUCTION CYCLE WITH INTERRUPT

- All computers provide a mechanism by which other modules (I/O, Memory) may **interrupt** the normal processing of the processor
- Processor engaged in executing other instructions while slow external devices (ex. I/O module) become ready and send **interrupt request** to processor
- Processor responds to such interrupt request by running **interrupt handler**



# STATE DIAGRAM OF INSTRUCTION CYCLE WITH INTERRUPT

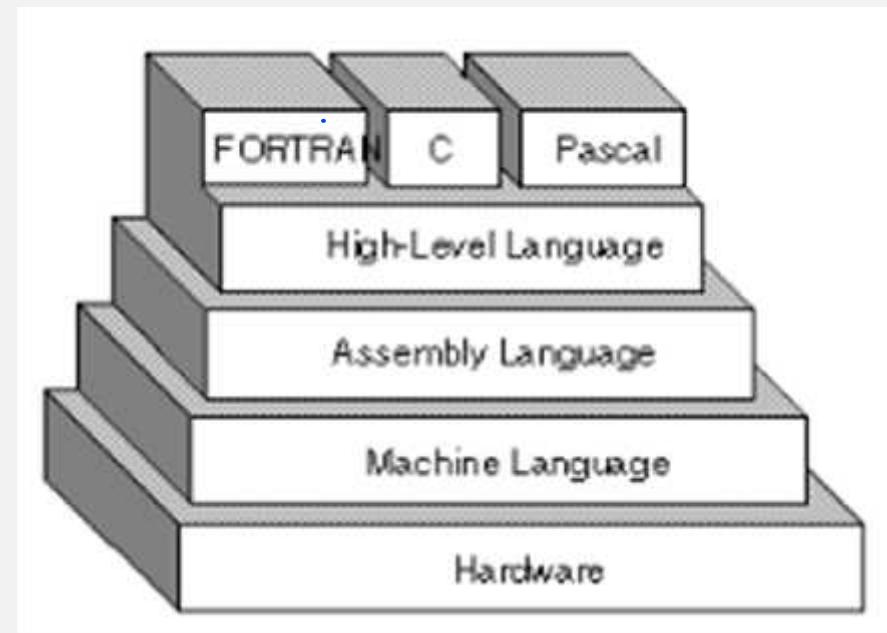


# INSTRUCTION SET ARCHITECTURE (ISA)

- Instruction Set is a part of the computer that pertains to internal (or low level) programming, which is basically machine language.
- The instruction set provides commands to the processor.
- Instructions instruct (give orders to) the processor what it needs to do. Consider as “words” in processor’s language.
- User codes, normally written in High Level Languages (closer to Natural Language) must convert to Machine Language to be run on the processor

# INSTRUCTION SET ARCHITECTURE (ISA)

- Assembly Language has rich set of mnemonics to represent Machine Language Instructions
- Assembly Language commands are in a human readable format.



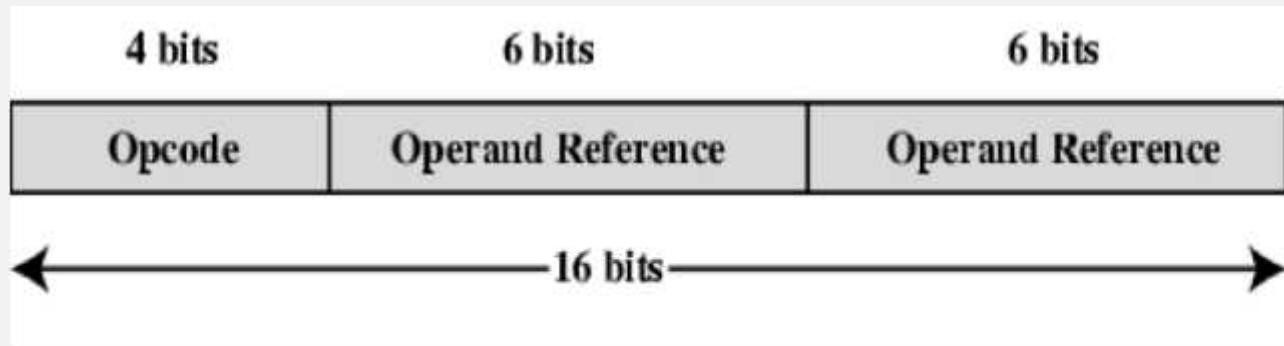
**MOU AX, DATA  
MOU DS, AX**

**MOU AL, NUM1  
ADD AL, NUM2**

# INSTRUCTION SET ARCHITECTURE (ISA)

- Typical ISA defines:
  - How to access data in Registers, Memory, and other I/O devices
  - Mechanisms to transfer data and instruction to and from processor
  - Operations such as additions, subtractions which processor can execute
  - Control mechanisms such as branch, jump
- To be effective as a programmer or processor designer, one should know how ISA works.

# INSTRUCTION FORMAT

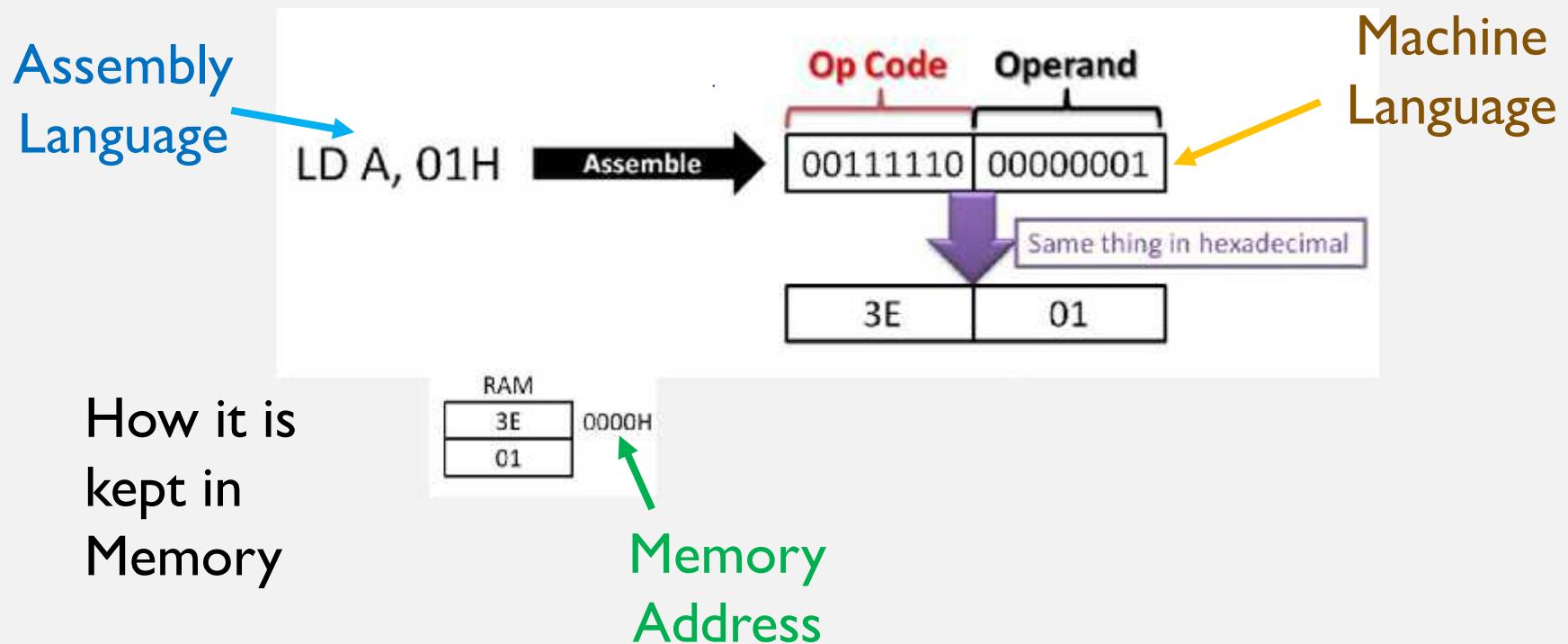


**Opcode:** Specifies the operation to be performed (Ex- ADD, SUB).

**Operand:** One or more inputs/outputs, or their source or destination

**Next instruction reference:** Tells the processor where to fetch the next instruction once the execution of this instruction is completed.

# INSTRUCTION FORMAT



# INSTRUCTION SET

## Sample Instructions

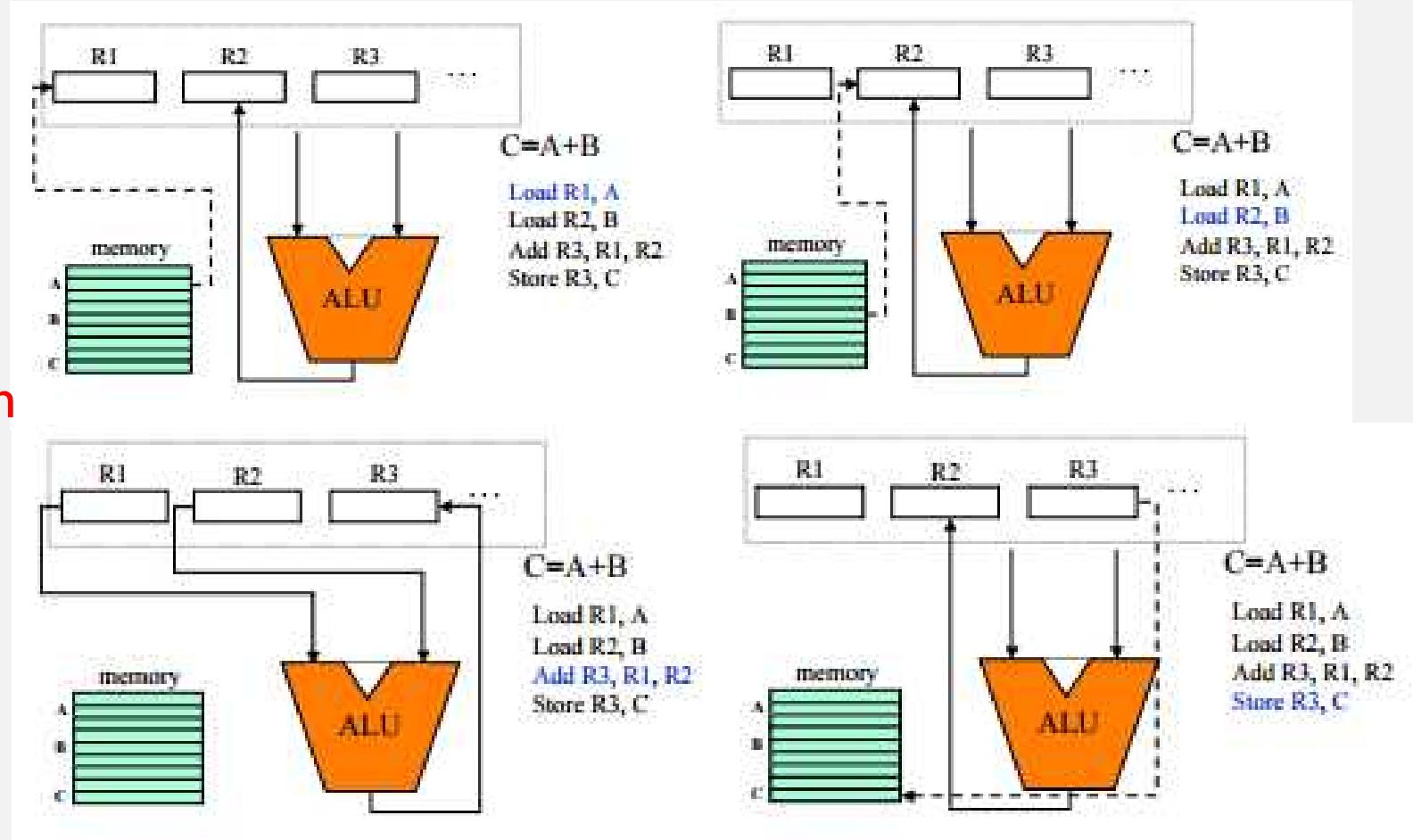
- **ADD** - Add two numbers.
- **CMP** - Compare numbers.
- **IN** - Input from port into AL or AX. Second operand is a port number.
- **JMP** - Unconditional Jump. Transfers control to another part of the program.
- **JNE** - Short Jump if first operand is Not Equal to second operand.
- **LOAD** - Load information from RAM to the CPU.
- **OUT** - Output information to device, e.g. monitor.
- **STORE** - Store information to RAM.

Do

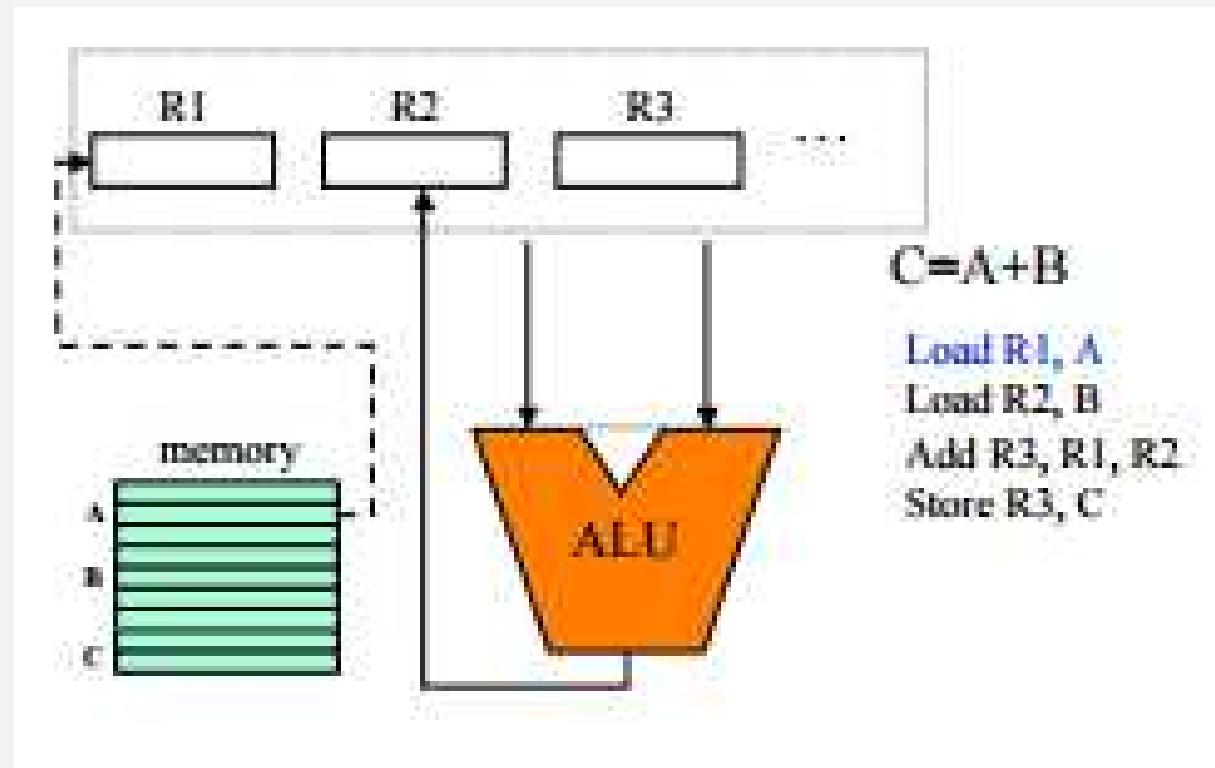
again

# EXECUTION OF INSTRUCTIONS

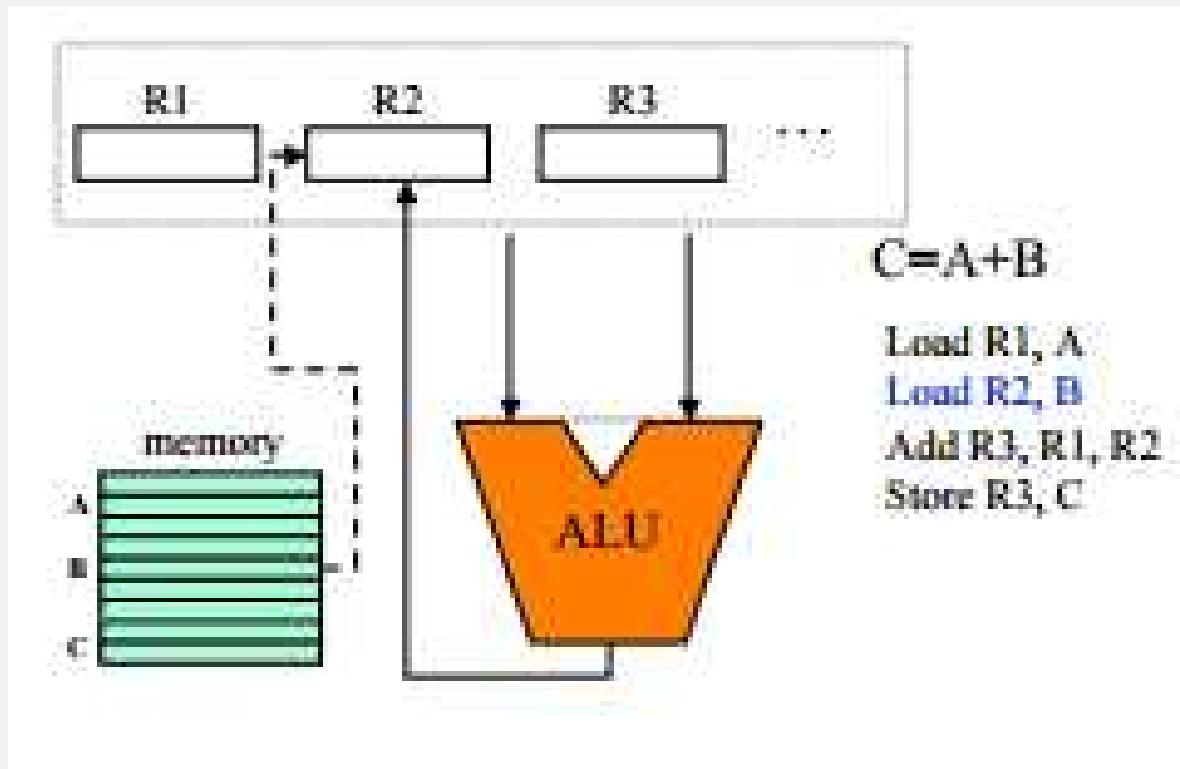
Processor  
Executes  
one  
instruction  
at a time.



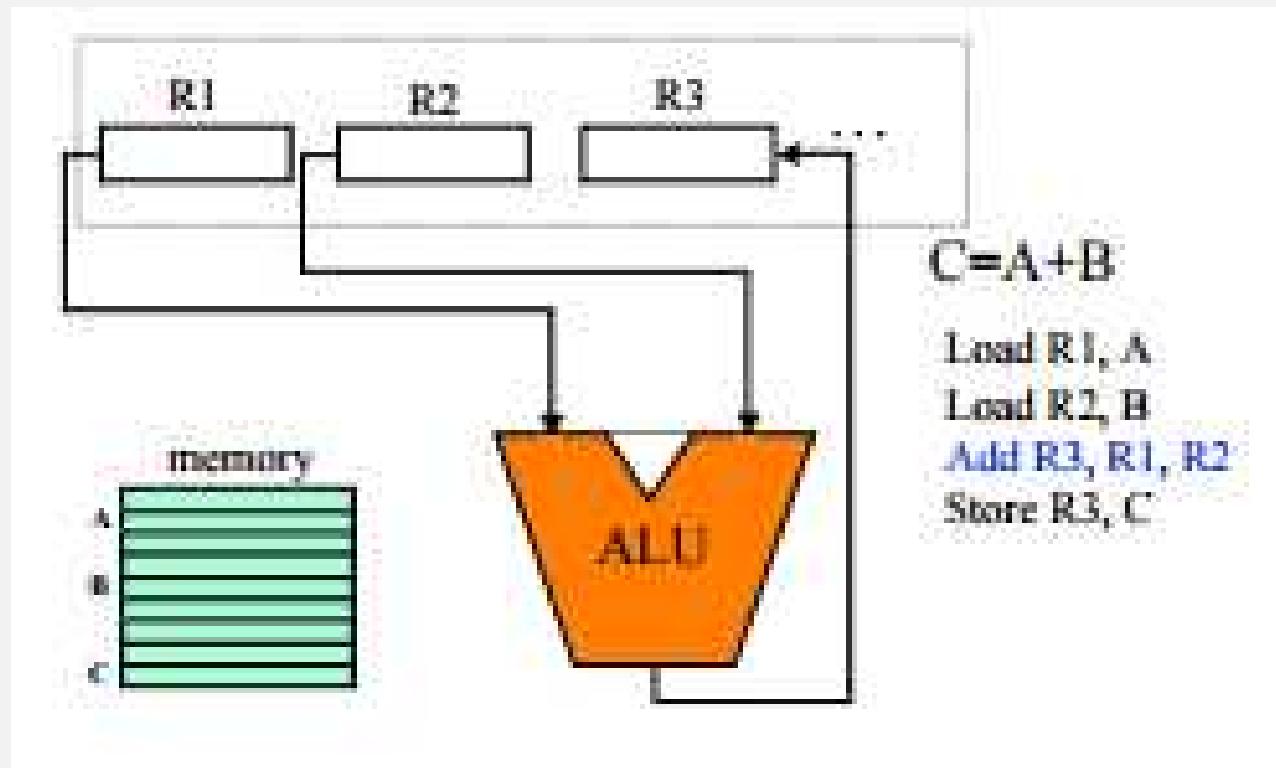
# EXECUTION OF INSTRUCTIONS



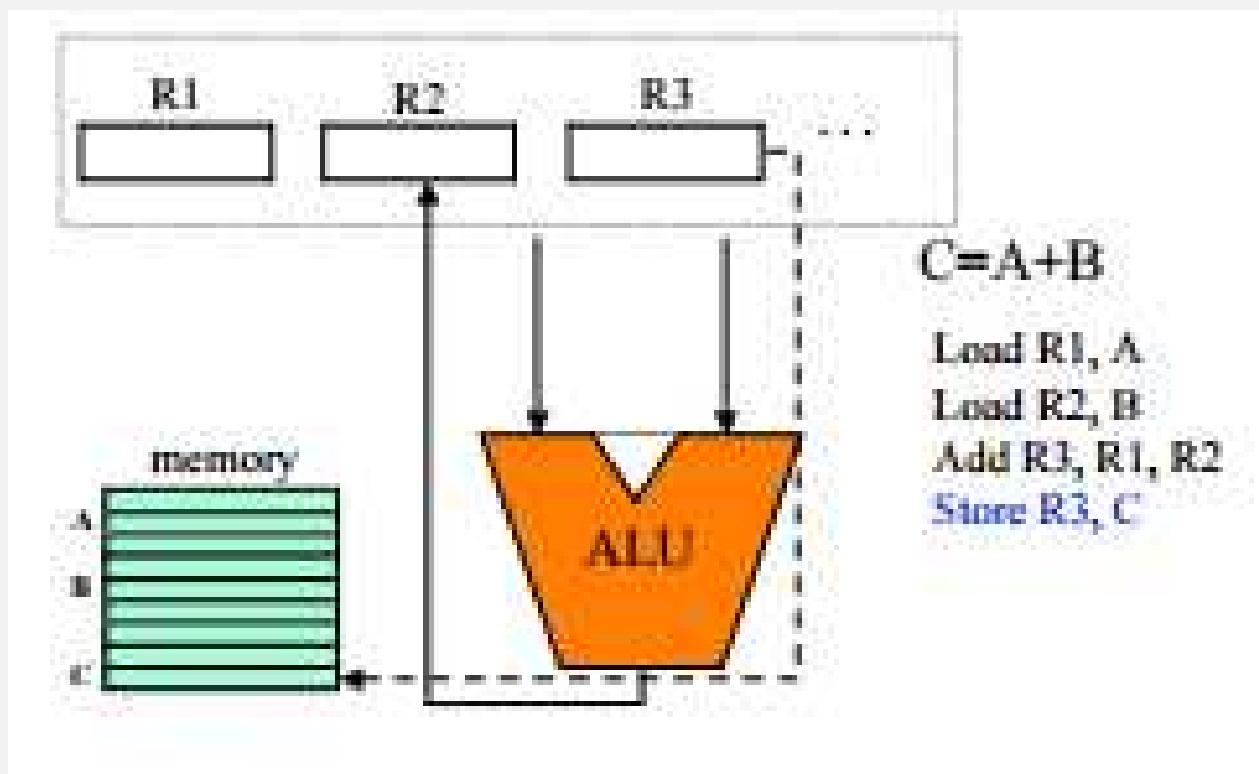
# EXECUTION OF INSTRUCTIONS



# EXECUTION OF INSTRUCTIONS



# EXECUTION OF INSTRUCTIONS



# CISC & RISC

- CISC:
  - Primary goal is to complete a task in as few lines of assembly as possible
  - Processor hardware complex; Needs less RAM to store the code; Instruction set is high-level, hence Compiler workload is low
  - Ex: MULT M1,M2

# CISC & RISC

- RISC:
  - Simple instructions that can be executed within one clock cycle
  - Processor hardware simple; Need more RAM; Instruction set Low-Level, hence Compiler workload high
  - Ex: LOAD A, M1
    - LOAD B, M2
    - PROD A, B
    - STORE M1,A

# CISC & RISC

CISC	RISC
<ul style="list-style-type: none"><li>Includes multi-clock complex instructions</li><li>Emphasis on hardware</li><li>Memory-to-memory: "LOAD" and "STORE" incorporated in instructions</li><li>Small code sizes, high cycles per second</li><li>Transistors used for storing complex instructions</li></ul>	<ul style="list-style-type: none"><li>Single-clock, reduced instruction only</li><li>Emphasis on hardware</li><li>Register to register: "LOAD" and "STORE" are independent instructions</li><li>Low cycles per second, large code sizes</li><li>Spends more transistors on memory registers</li></ul>

# **CISC & RISC**

<b>CISC</b>	<b>RISC</b>
<ul style="list-style-type: none"><li>• Intel X86 family,AMD processors are heavily used in desktop, laptop and server computers</li></ul>	<ul style="list-style-type: none"><li>• SPARC and Power PC are used in desktop computers and game consoles</li><li>• RISC processors are heavily used in real-time embedded systems such as mobile phones, washing machines, Routers</li><li>• Raspberry pi and Arduino</li><li>• IoT drives by RISC processors</li></ul>

# **THANK YOU**

## **Post-lecture activities**

Instruction Set of Core i5, ARM processor

## **Lecture 04**

Operating system

## **Lecture 04 – Pre-lecture activities**

Types of OSs, Windows, Linux, MacOS descriptions

---

IT1020 – Introduction to Computer Systems

Lecture 4

# Operating Systems





# Agenda

- What Operating Systems Do
- Computer-System Organization
- Operating-System Structure
- Operating-System Operations
- Process, Memory & Storage Management
- Protection and Security
- Computing Environments
- Open-Source Operating Systems



## Objectives

- To describe the basic organization of computer systems
- To provide a grand tour of the major components of operating systems
- To give an overview of the many types of computing environments



## What is an Operating System

- A program that acts as **an intermediary** between **a user** of a computer and the **computer hardware**
- Operating system **goals**:
  - **Execute user programs** and make solving user problems easier
  - Make the computer system **convenient to use**
  - **Manage** the computer hardware in an **efficient manner**

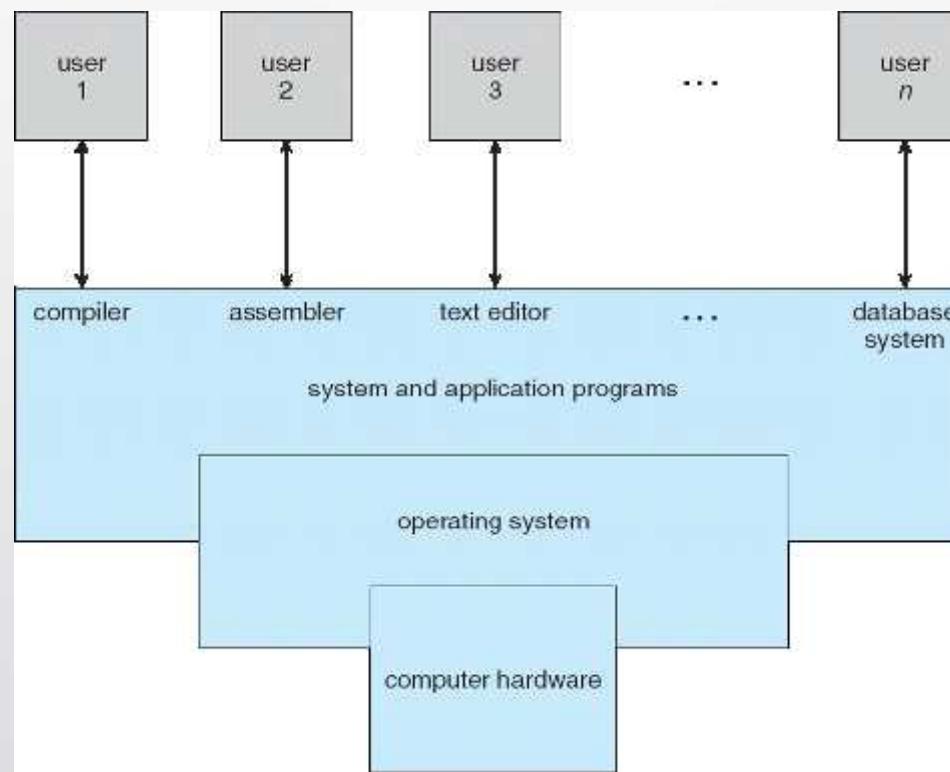


# Operating-System Structure

**Computer system can be divided into four components:**

- **Hardware** – provides basic computing resources
  - CPU, memory, I/O devices
- **Operating system**
  - Controls and coordinates use of hardware among various applications and users
- **Application programs** – define the ways in which the system resources are used to solve the computing problems of the users
  - Word processors, compilers, web browsers, database systems, video games
- **Users**
  - People, machines, other computers

# Operating-System Structure





## Operating System Definition

- OS as a **control program**
  - Controls **execution of programs** to prevent errors and improper use of the computer
- OS as a **resource allocator**
  - **Manages** all resources
  - Decides between conflicting requests for **efficient** and **fair** resource use



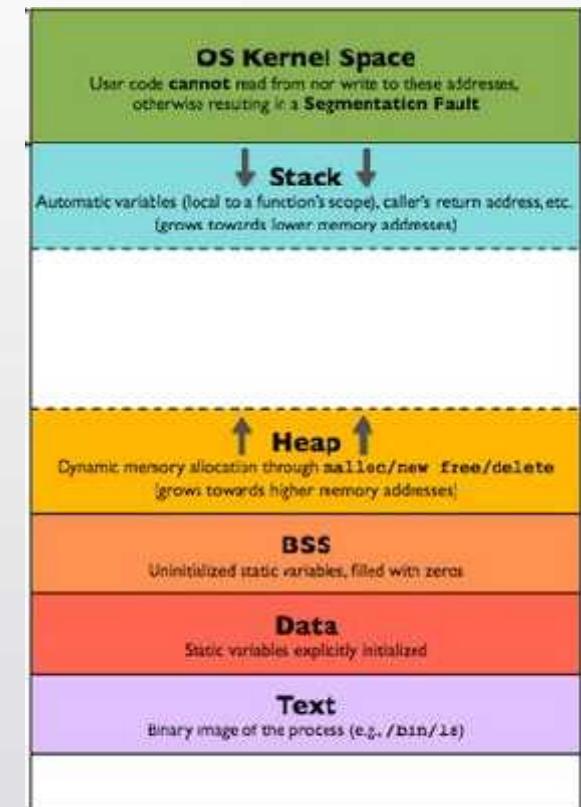
## Operating System Definition – Common Beliefs

- “Everything a vendor ships when you order an operating system”
- “The one program running at all times on the computer”
  - kernel.
- Everything else is either
  - a **system program** (ships with the operating system) , or
  - an **application program**.

## Computer-Start up – How OS comes to work?

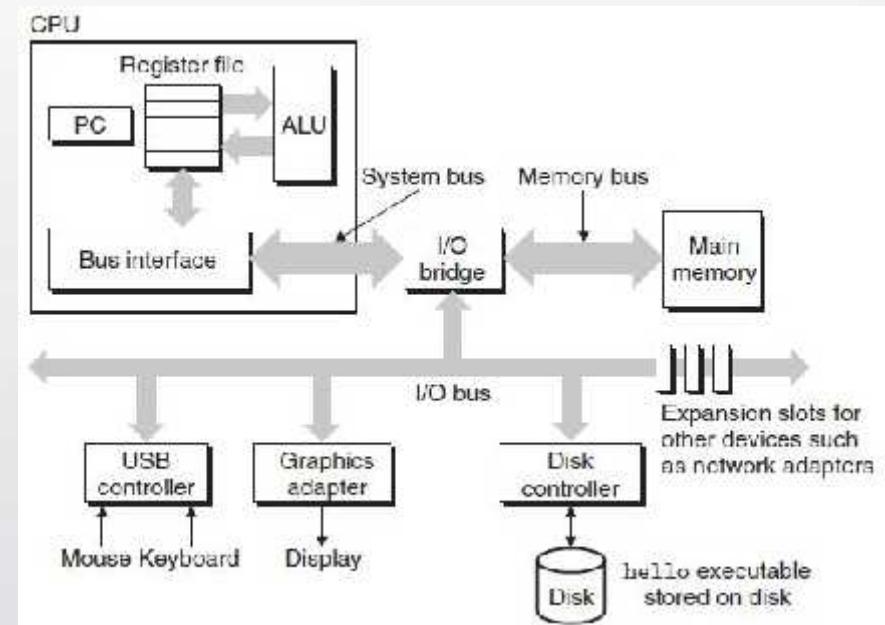
- **BIOS program** is loaded at power-up or reboot
  - Typically stored in EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

## System Memory Map



# Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a **local buffer**
- CPU moves data **from/to** main memory **to/from** local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an **interrupt**





## Computer System Operation: Interrupts

- **Interrupt** transfers control to the interrupt service routine
- Interrupt architecture must save the address of the interrupted instruction
- A **trap (or an exception)** is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**



## Operating-System Operations

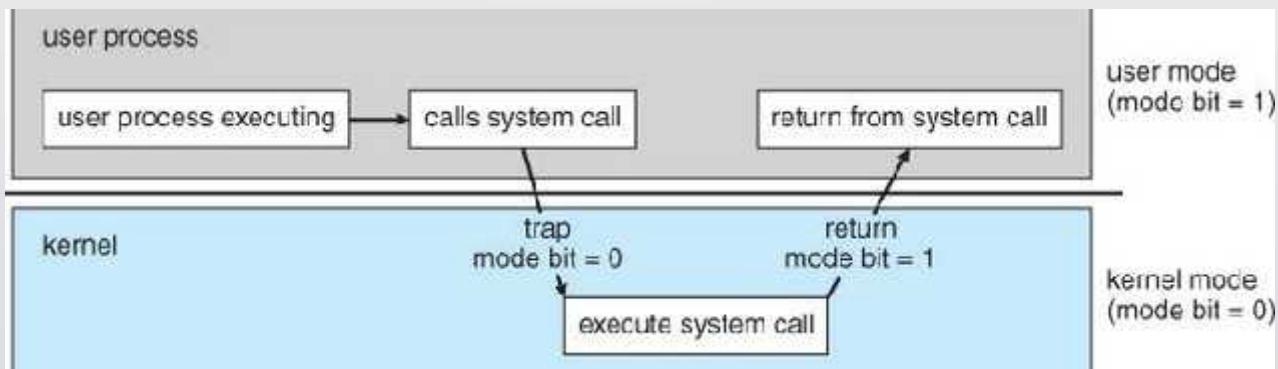
- **Multiprogramming (Batch system)** needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via **job scheduling**
  - When it has to wait (for I/O for example), OS switches to another job

# Operating-System Operations

- **Timesharing (multitasking)** is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory  
⇒ **process**
  - If several jobs ready to run at the same time ⇒ **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory

# Operating-System Operations

- Dual-mode operation allows OS to protect itself and other system components
  - User mode and kernel mode
  - Mode bit provided by hardware
  - Provides ability to distinguish when system is running user code or kernel code



Re do

## Operating-System Operations

- Some instructions designated as privileged, only executable in kernel mode
- **System call** changes mode to kernel, return from call resets it to user
- Increasingly CPUs support multi-mode operations
  - i.e. virtual machine manager (VMM) mode for guest VMs

A **system call** is the way that a computer program requests a service from the kernel. This may include hardware-related services (for example, accessing a hard disk drive), creation and execution of new processes, and communication with integral kernel services such as process scheduling.



## Operating System - Services

- Process Management
- Memory Management
- Storage Management
  - File system management
  - Storage space management
- Protection and Security

# Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a **passive entity**, process is an **active entity**.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion



## Process Management

- The operating system is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication
  - Providing mechanisms for deadlock handling



## Memory Management

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed



## Storage Management

- OS provides uniform, **logical view** of information storage
- Abstracts physical properties to logical storage unit - file
- Each medium is controlled by a device (i.e., disk drive, tape drive)
- Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)



## Storage Management

- File-System management
- Files usually organized into directories
- Access control on most systems to determine who can access what
- OS activities include
  - Creating and deleting files and directories
  - Primitives to manipulate files and directories
  - Mapping files onto secondary storage
  - Backup files onto stable (non-volatile) storage media



## Storage Management

- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling



## Protection and Security

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file



## Computing Environments – Traditional

- Stand-alone general purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers (thin clients)** are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks



## Computing Environments – Mobile

- Handheld smartphones, tablets, etc
- What is the functional difference between them and a “traditional” laptop?
- Extra feature – more OS features (GPS, gyroscope)
- Allows new types of apps like ***augmented reality***
- Use wireless, or cellular data networks for connectivity
- Leaders are **Apple iOS** and **Google Android**



# Computing Environments – Virtualization

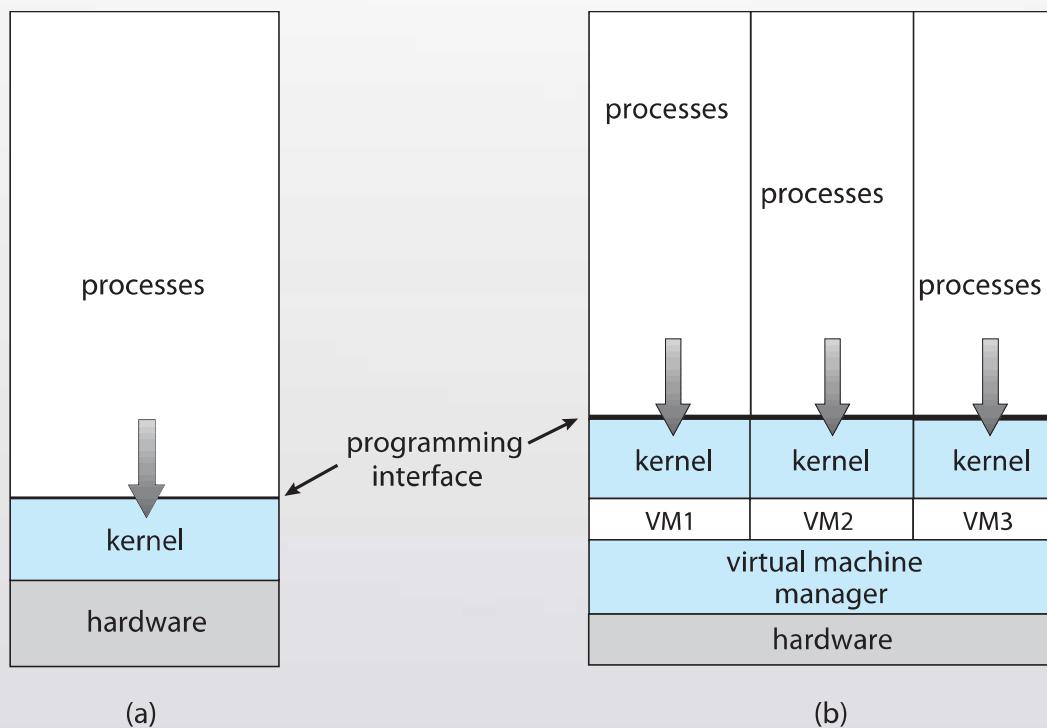
- Allows operating systems to run applications within other OSes
  - Vast and growing industry
- **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
  - Generally slowest method
  - When computer language not compiled to native code – **Interpretation**
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - **VMM** (virtual machine Manager) provides virtualization services



## Computing Environments – Virtualization

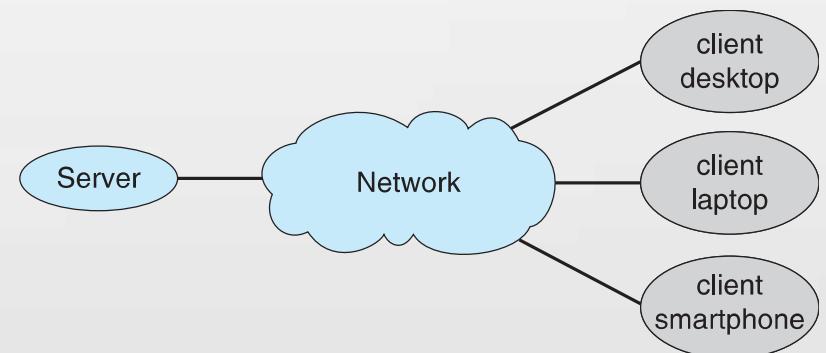
- Use cases involve laptops and desktops running multiple OSes for exploration or compatibility
  - Apple laptop running Mac OS X host, Windows as a guest
  - Developing apps for multiple OSes without having multiple systems
  - QA testing applications without having multiple systems
  - Executing and managing computer environments within data centers

# Computing Environments – Virtualization



# Computing Environments – Client-Server

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now servers, responding to requests generated by clients
- Compute-server system provides an interface to client to request services (i.e., database)
- File-server system provides interface for clients to store and retrieve files





## Computing Environments – Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for its functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components



## Computing Environments – Cloud Computing

- Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor)
- Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server)
- Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use)



## Computing Environments – Real-Time Embedded Systems

- Real-time embedded systems most prevalent form of computers
  - Vary considerable, special purpose, limited purpose OS, **Real-time OS**
  - Use expanding
- Many other special computing environments as well
  - Some have OSes, some perform tasks without an OS
- Real-time OS has well-defined fixed time constraints
  - Processing **must** be done within constraint
  - Correct operation only if constraints met



# Open-Source Operating Systems

- Operating systems made available in source-code format rather than just binary **closed-source**
- Counter to the **copy protection** and **Digital Rights Management (DRM)** movement
- Started by **Free Software Foundation (FSF)**, which has “copyleft” **GNU Public License (GPL)**
- Examples include **GNU/Linux** and **BSD UNIX** (including core of **Mac OS X**), and many more
- Can use VMM like VMware Player (Free on Windows), Virtualbox (open source and free on many platforms - <http://www.virtualbox.com>)
  - Use to run guest operating systems for exploration



Thank you





# Introduction to Computer Systems

---

LECTURE 5.1: LOGIC CIRCUITS & SIMPLIFICATION

# Objectives

---

After completing this module you will be able to:

- ❖ Understand what a logic circuit is.
- ❖ Understand how to simplify the digital logic circuits
- ❖ Understand how to simplify the circuits using Boolean Algebra
- ❖ Understand how to simplify the circuits using K-Map

# Semiconductors to Computers

---

Increasing level of complexity:

Transistors built from semiconductors

Logic gates built from transistors

Logic functions built from gates

Flip-flops built from logic gates

Counters and sequencers from flip-flops

Microprocessors from sequencers

Computers from microprocessors

# Logic Gates

---

Logic Gates are the building blocks of Digital Circuits

Logic Gate is an electronic circuit having one or more than one input and only one output

The relationship between the input and the output is based on a certain logic

Inputs and Outputs can be represented as binary variables (logic variables, Boolean variables)

# Logic Gates

---

Used algebraic or tabular forms to describe the manipulation and processing of binary information.

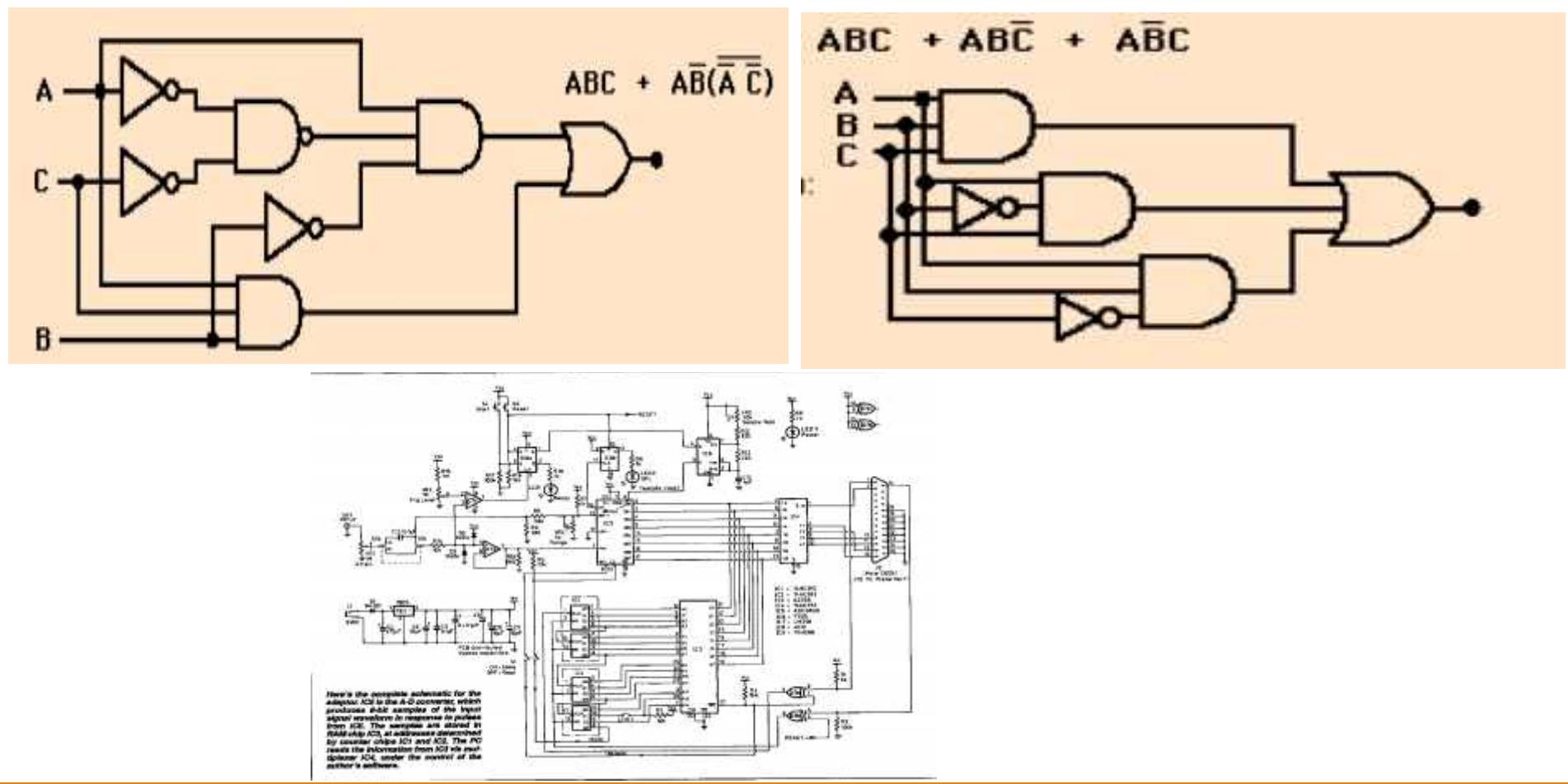
- Important advantages for two-valued Digital Circuit:
  - ❖ Mathematical Model – Boolean Algebra
  - ❖ Can help *design, analyse, simplify* Digital Circuits.



# Digital Logic Circuits

---

# Digital Logic Circuits





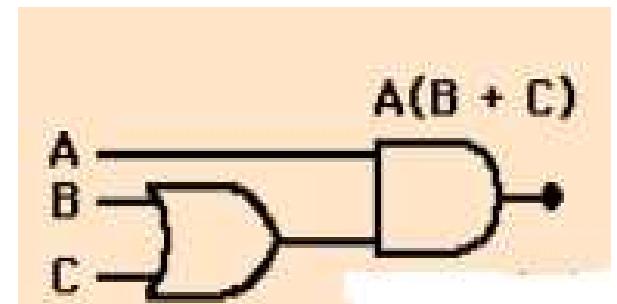
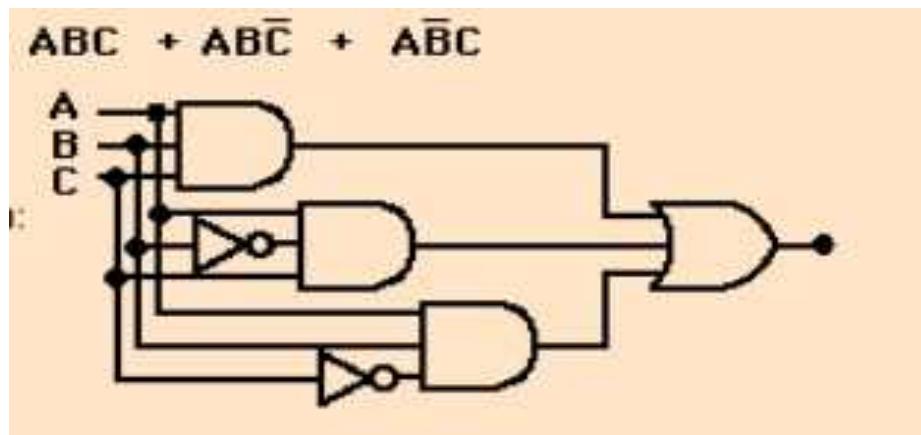
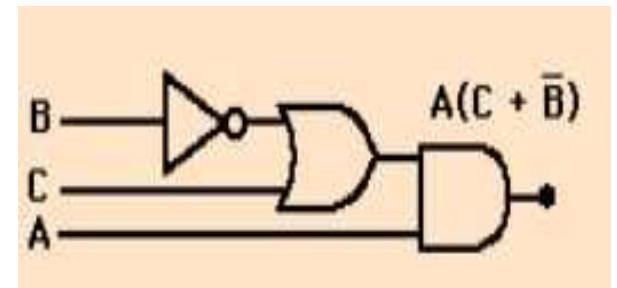
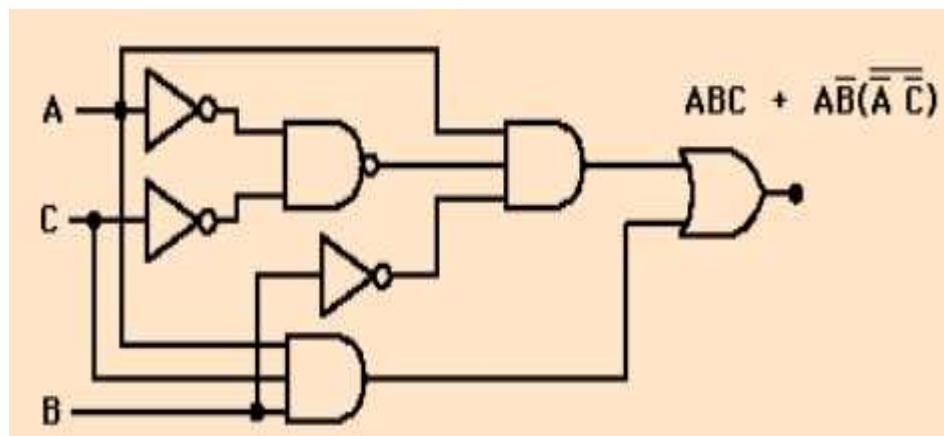
# Simplification of the Digital Logic Circuits

---



# Digital Logic Circuits

---



# Simplification of the Digital Logic Circuits

---

Why do we need simplification?

- ❖ To reduce the number of gates required to build the circuit and hence:

Reduce the cost

Reduce the power consumption

Reduce the space required

Reduce the propagation delay

# Simplification of the Digital Logic Circuits (Contd.)

---

## Methods to Simplify the Digital Logic Circuits

- ❖ Using Truth Table
- ❖ Using Mathematical Method

### Using Truth Table

- ❖ The problem is analyzed
- ❖ The truth table is developed
- ❖ Using the truth table the circuit is built.

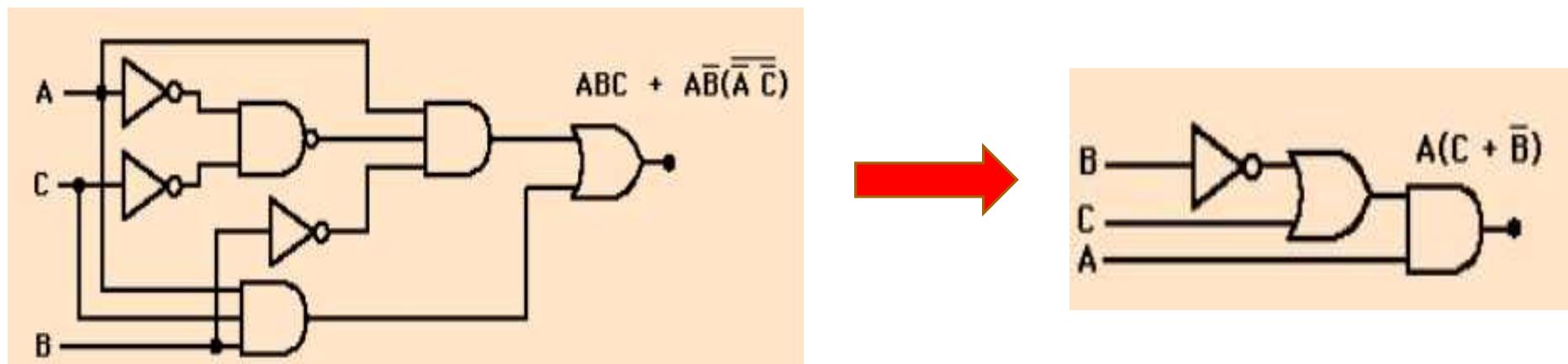
# Simplification of Combinational Circuits Using Mathematical Methods

---

There are two methods to simplify the combinational Circuits

- ❖ Boolean Algebraic method
- ❖ K-map

# Simplification of Combinational Circuits



$ABC + AB(\bar{A} + \bar{C})$  DeMorgan's theorem

$ABC + A\bar{B}A + A\bar{B}\bar{C}$  sum of products form

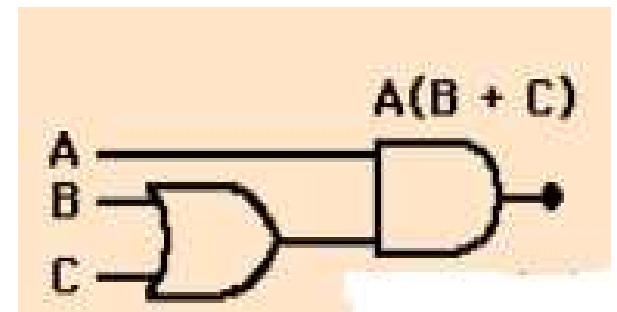
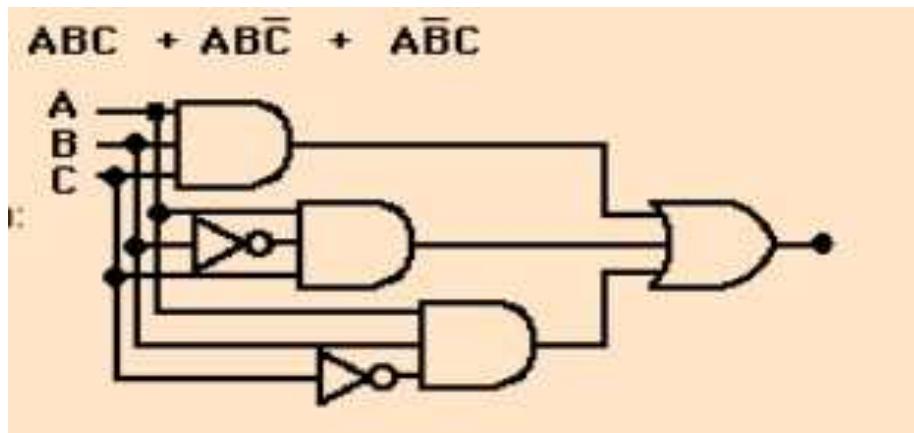
$ABC + A\bar{B} + A\bar{B}\bar{C}$   $BA = AB$  and  $AA = A$

$AC(B + \bar{B}) + A\bar{B}$

$AC + A\bar{B}$   $B + \bar{B} = 1$

$A(C + \bar{B})$

# Simplification of Combinational Circuits



$$\begin{aligned} &ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} \\ &AB(C + \bar{C}) + AC(\bar{B} + B) \\ &AB + AC = A(B + C) \end{aligned}$$

$$\begin{aligned} ABC &= ABC + A\bar{B}\bar{C} \\ \bar{C} + C &= 1 \end{aligned}$$

---

# K-Map to Simplify the Digital Logic Circuits



# Introduction : K-Map

---

- ❖ The Karnaugh map was invented in 1952 by Edward W. Veitch.
- ❖ It was further developed in 1953 by Maurice Karnaugh, a physicist at Bell Labs, to help simplify digital electronic circuits.
- ❖ While exploring the new field of digital logic and its application to the design of telephone circuits, he invented a graphical way of visualizing and then simplifying Boolean expressions.
- ❖ This graphical representation, now known as a Karnaugh map, or Kmap, is named in his honor.

# Description of K-maps and Terminology

---

A Kmap is a matrix consisting of rows and columns that represent the output values of a Boolean function.

The output values placed in each cell are derived from the *minterms* of a Boolean function.

A ***minterm*** is a product term that contains all of the function's variables exactly once, either complemented or not complemented.

# Description of Kmaps and Terminology

- ❖ For example, the minterms for a function having the inputs  $x$  and  $y$  are:
- ❖ Consider the Boolean function,
- ❖ Its minterms are:

$$F(x, y) = xy + \bar{x}y$$

A two-variable function, such as  $f(x,y)$ , has  $2^2 = 4$  minterms:

Minterm	x	y
$\bar{x}\bar{y}$	0	0
$\bar{x}y$	0	1
$x\bar{y}$	1	0
$xy$	1	1

# Description of Kmaps and Terminology

---

- ❖ Similarly, a function having three inputs, has the minterms that are shown in this diagram.
- ❖  $2^3 = 8$  miniterms

Minterm	x	y	z
$\bar{x}\bar{y}\bar{z}$	0	0	0
$\bar{x}\bar{y}z$	0	0	1
$\bar{x}y\bar{z}$	0	1	0
$\bar{x}yz$	0	1	1
$x\bar{y}\bar{z}$	1	0	0
$x\bar{y}z$	1	0	1
$xy\bar{z}$	1	1	0
$xyz$	1	1	1

# Description of Kmaps and Terminology

---

- ❖ A Kmap has a cell for each minterm.
- ❖ This means that it has a cell for each line for the truth table of a function.
- ❖ The truth table for the function  $F(x,y) = xy$  is shown at the right along with its corresponding Kmap.

$F(x,y) = xy$		
x	y	$xy$
0	0	0
0	1	0
1	0	0
1	1	1

x	y	0	1
0	0	0	0
1	0	0	1

# Description of Kmaps and Terminology

- ❖ As another example, we give the truth table and KMap for the function,  $F(x,y) = x + y$  at the right.
- ❖ This function is equivalent to the OR of all of the minterms that have a value of 1. Thus:

$$F(x, y) = x + y = \bar{x}y + x\bar{y} + xy$$

$F(x, y) = x + y$		
x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

x	y	0	1
0	0	0	1
1	1	1	1

# Kmap Simplification for Two Variables

---

- ❖ Of course, the minterm function that we derived from our Kmap was not in simplest terms.
  - ❖ That's what we started with in this example.
- ❖ We can, however, reduce our complicated expression to its simplest terms by finding adjacent 1s in the Kmap that can be collected into groups that are powers of two.

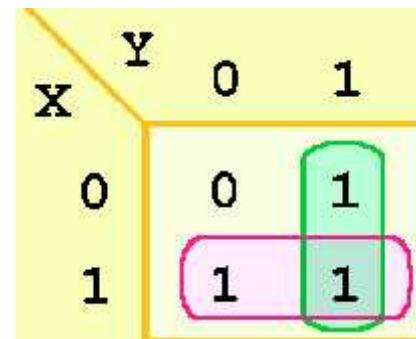
- ❖ In our example, we have two such groups.
  - ❖ Can you find them?

	$y$	0	1
$x$	0	0	1
1	1	1	1

# Kmap Simplification for Two Variables

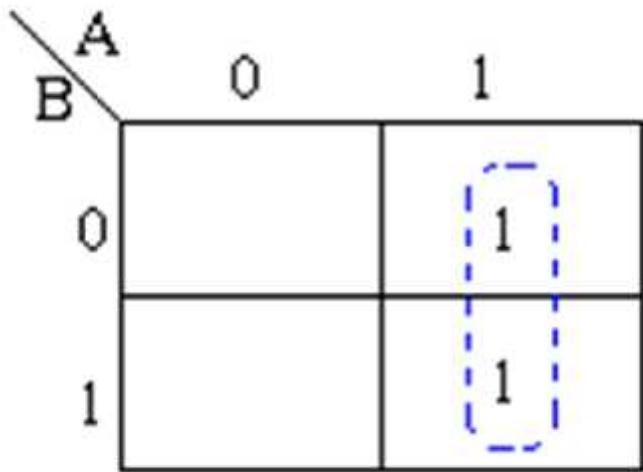
---

- ❖ The best way of selecting two groups of 1s from our simple Kmap is shown below.
- ❖ We see that both groups are powers of two and that the groups overlap.



# Example -1

Consider the following map.



The function plotted is:

$$Z = f(A,B) = AB + \overline{A}\overline{B}$$

Using algebraic simplification,

$$Z = AB + \overline{A}\overline{B}$$

$$Z = A(B + \overline{B})$$

$$Z = A$$

B becomes redundant

Referring to the map, the two adjacent 1's are grouped together. Through inspection it can be seen that variable B has its true and false form within the group. This eliminates variable B leaving only variable A which only has its true form.

The minimized answer therefore is  $Z = A$ .

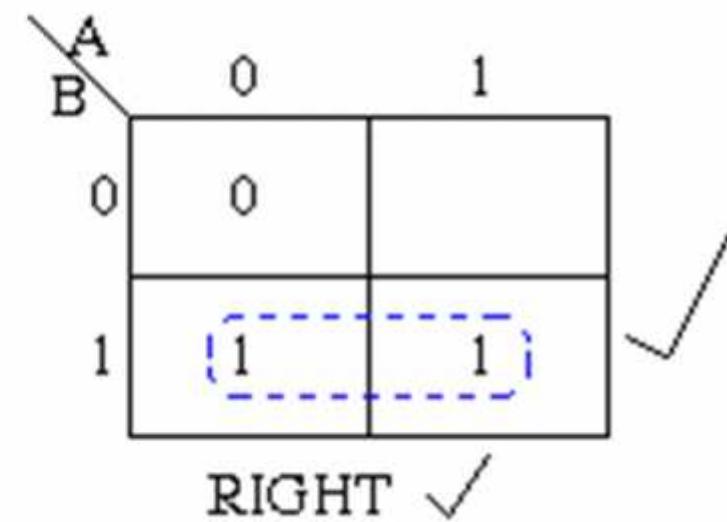
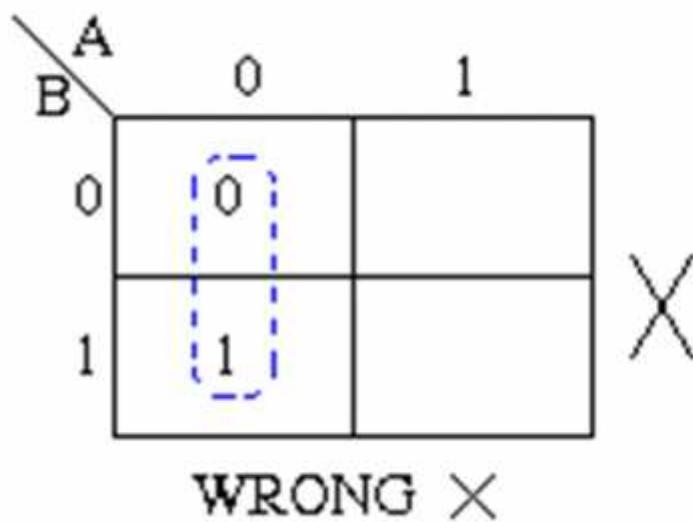
# Kmap Simplification for Two Variables

---

- ❖ The rules of K-map simplification are:
  - ✓ Groupings can contain only 1s; no 0s.
  - ✓ Groups can be formed only at right angles; diagonal groups are not allowed.
  - ✓ The number of 1s in a group must be a power of 2 – even if it contains a single 1.
  - ✓ The groups must be made as large as possible.
  - ✓ Groups can overlap and wrap around the sides of the Kmap.

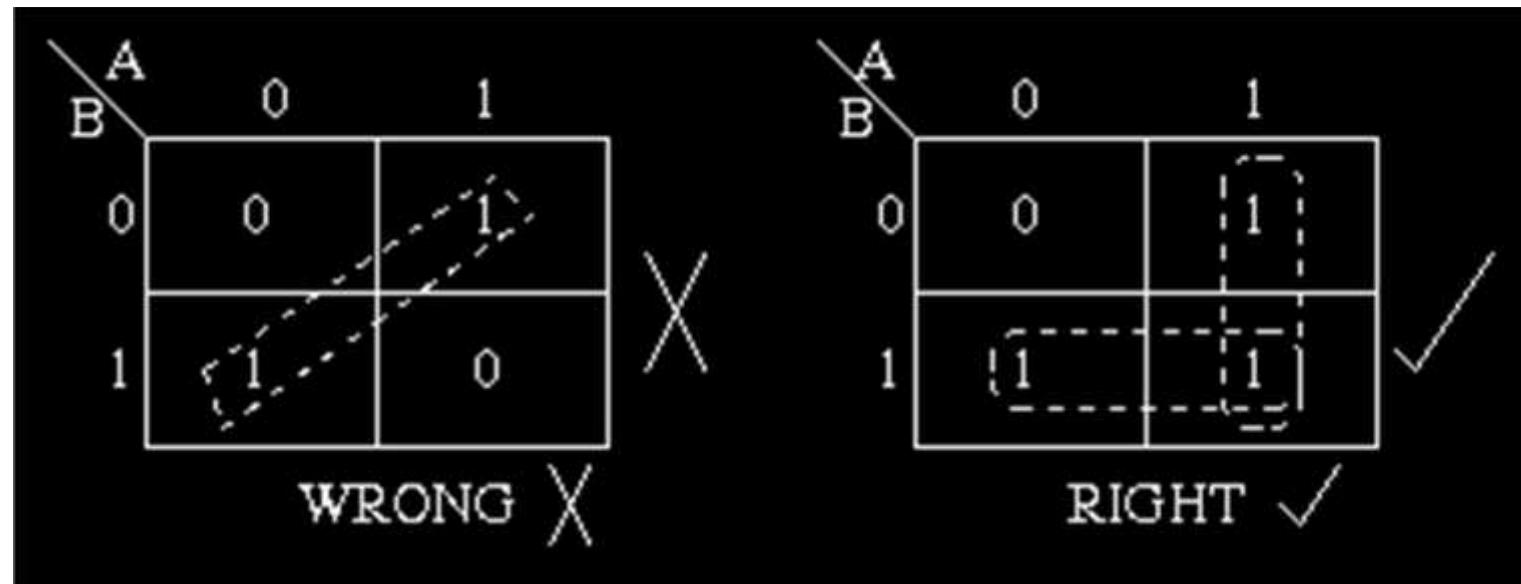
# Karnaugh Maps – Rules

---



Group only elements containing 1

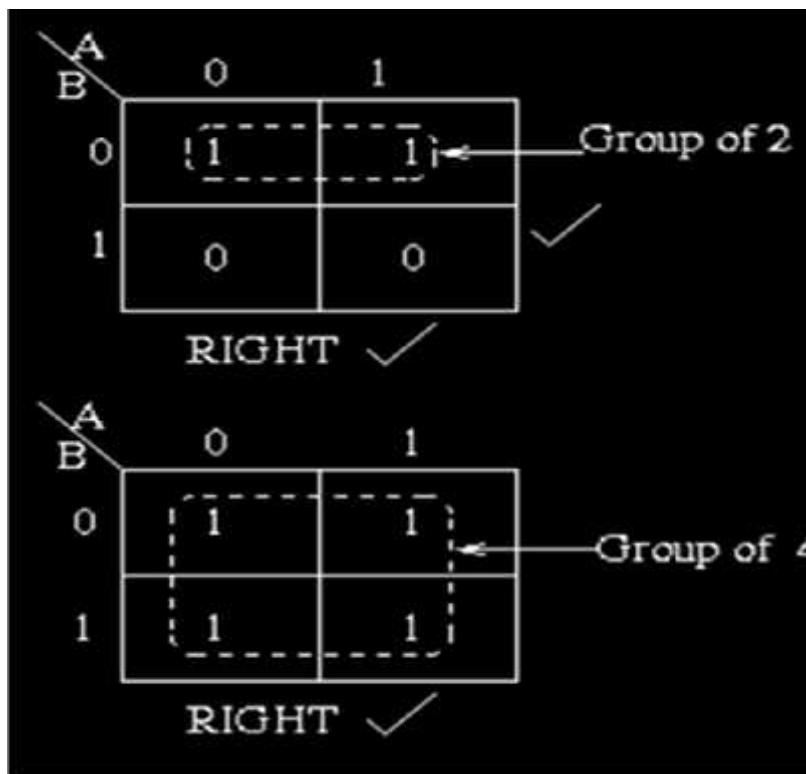
# Karnaugh Maps – Rules



Only Horizontal and Vertical Grouping  
Diagonal not allowed

# Karnaugh Maps – Rules

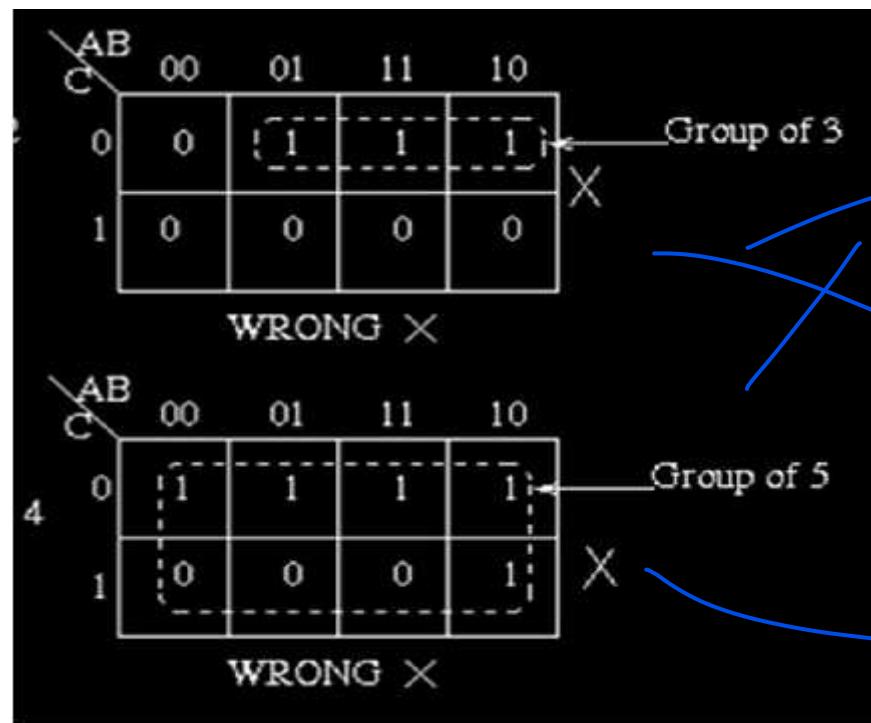
---



Groups Powers of 2



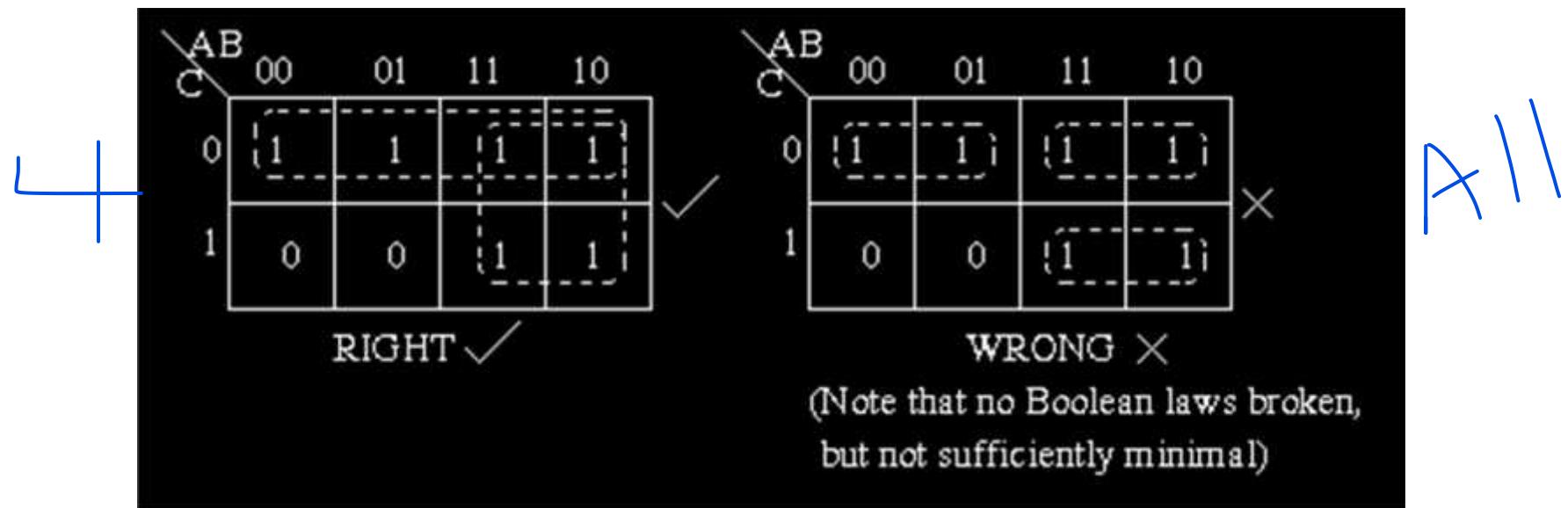
# Karnaugh Maps – Rules



Groups Power of 2  
3

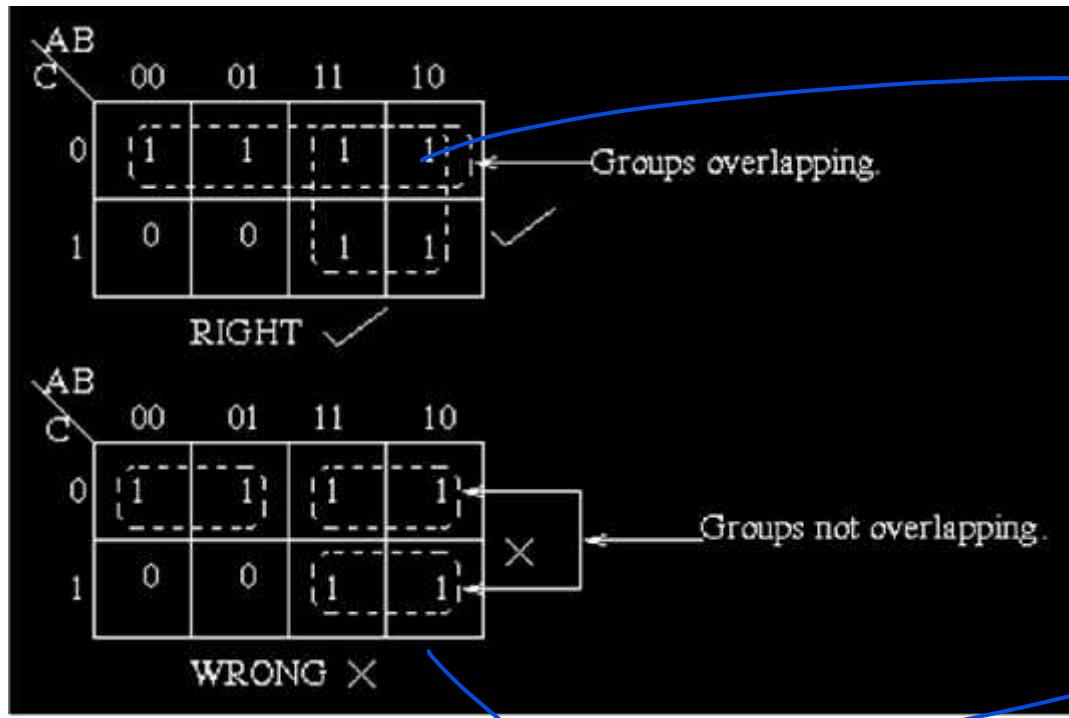
D

# Karnaugh Maps – Rules



Each group should be large as possible

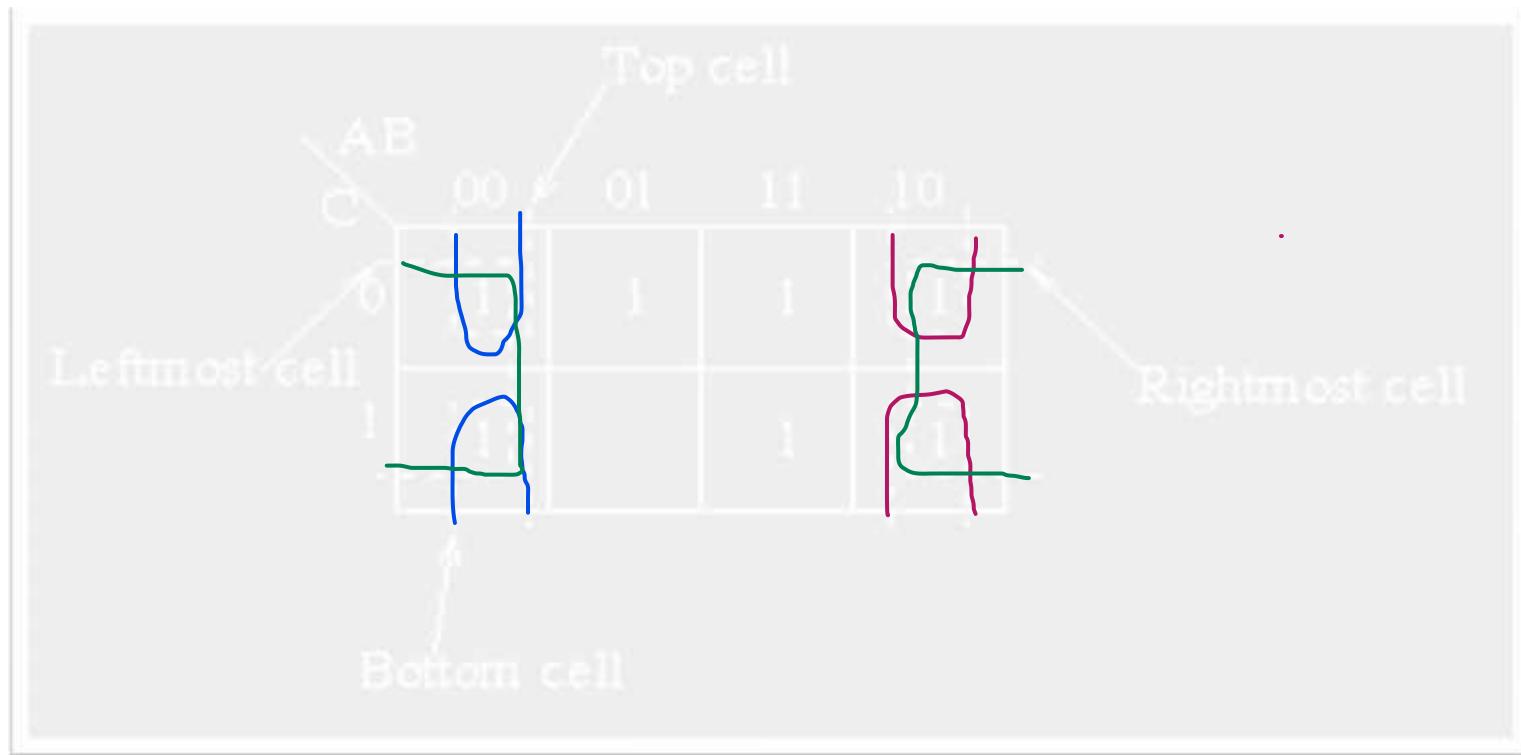
# Karnaugh Maps – Rules



4  
Groups may overlap

ot A //

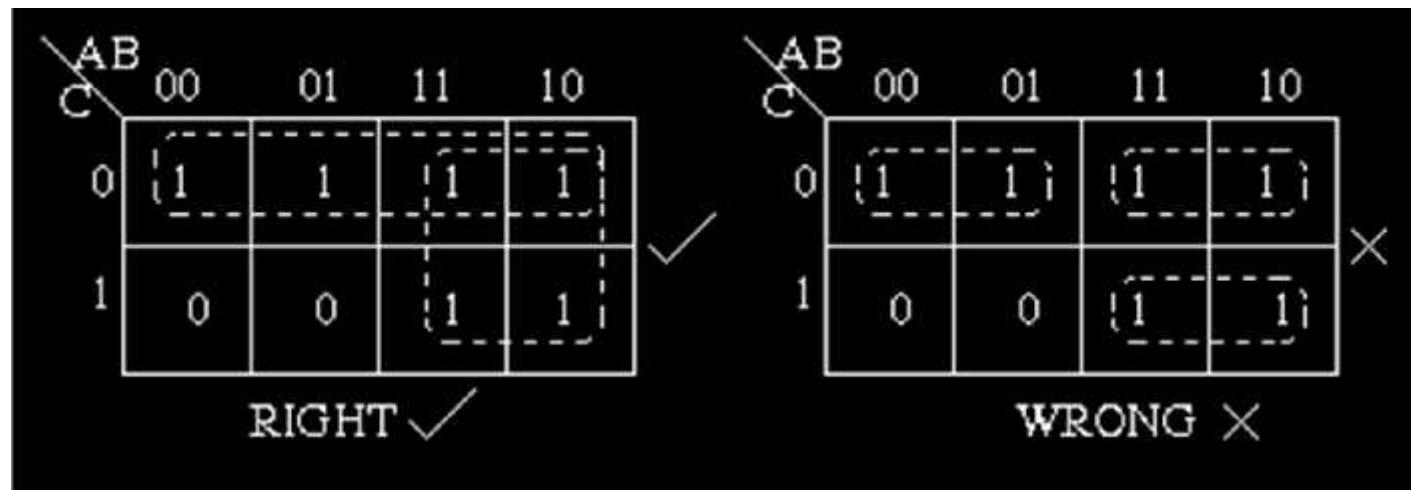
# Karnaugh Maps – Rules



Groups may wrap around the table

# Karnaugh Maps – Rules

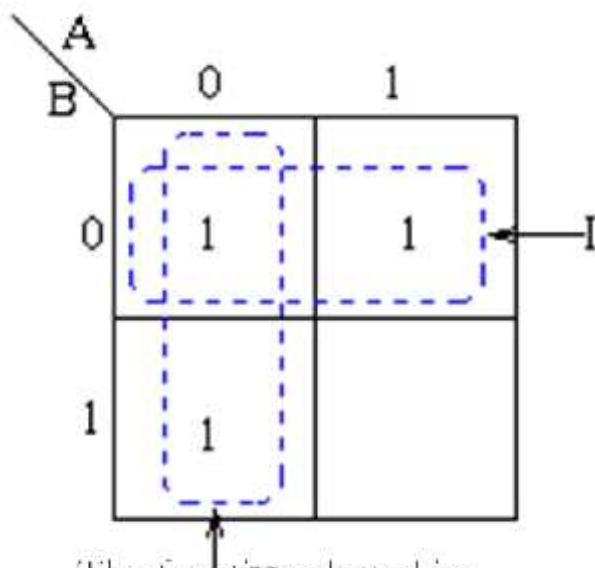
---



Should have few groups as possible

## Example-2

Consider the following map.



The function plotted is:

$$Z = f(A, B) = AB + \overline{AB} + A\overline{B}$$

Pairs of 1's are *grouped* as shown above, and the simplified answer is obtained by using the following steps:

- Note that two groups can be formed for the example given above, bearing in mind that the largest rectangular clusters that can be made consist of two 1s.
- Notice that a 1 can belong to more than one group.
- The first group labelled I, consists of two 1s which correspond to  $A = 0, B = 0$  and  $A = 1, B = 0$ .
- Put in another way, all squares in this example that correspond to the area of the map where  $B = 0$  contains 1s, independent of the value of A.
- So when  $B = 0$  the output is 1.
- The expression of the output will contain the term  $B$

For group labeled II corresponds to the area of the map where  $\overline{A} = 0$ .

The group can therefore be defined as  $A$

This implies that when  $A = 0$  the output is 1.

The output is therefore 1 whenever  $\overline{B} = 0$  and  $\overline{A} = 0$

Hence the simplified answer is  $Z = \overline{A} + \overline{B}$

# Kmap Simplification for Three Variables

- ❖ A Kmap for three variables is constructed as shown in the diagram below.
- ❖ We have placed each minterm in the cell that will hold its value.
  - Notice that the values for the  $yz$  combination at the top of the matrix form a pattern that is not a normal binary sequence.

		yz	00	01	11	10
		x	00	01	11	10
x	0	$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}z$	$\bar{x}yz$	$\bar{x}y\bar{z}$	
	1	$x\bar{y}\bar{z}$	$x\bar{y}z$	$xyz$	$x\bar{y}\bar{z}$	

# Kmap Simplification for Three Variables

---

- ❖ Thus, the first row of the Kmap contains all minterms where  $x$  has a value of zero.
- ❖ The first column contains all minterms where  $y$  and  $z$  both have a value of zero.

		yz	00	01	11	10
		x	00	01	11	10
y	0	xyz	$\bar{x}\bar{y}z$	$\bar{x}yz$	$\bar{x}\bar{y}\bar{z}$	$\bar{x}yz$
	1	$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}z$	$\bar{x}yz$	$\bar{x}\bar{y}\bar{z}$	$\bar{x}y\bar{z}$

# Kmap Simplification for Three Variables

---

- ❖ Consider the function:

$$F(X, Y, Z) = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z + XYZ$$

- ❖ Its Kmap is given below.
  - What is the largest group of 1s that is a power of 2?

		YZ				
		00	01	11	10	
X		0	0	1	1	0
		1	0	1	1	0

# Kmap Simplification for Three Variables

- ❖ This grouping tells us that changes in the variables  $x$  and  $y$  have no influence upon the value of the function: They are irrelevant.
- ❖ This means that the function,

$$F(x, y, z) = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xy\bar{z}$$

reduces to  $F(x) = z$ .

You could verify this reduction with identities or a truth table.

		YZ	00	01	11	10	
		x	0	0	1	1	0
x	0	0	1	1	0		
	1	0	1	1	0		

# Kmap Simplification for Three Variables

---

- ❖ Now for a more complicated Kmap. Consider the function:

$$F(X, Y, Z) = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XY\bar{Z}$$

- ❖ Its Kmap is shown below. There are (only) two groupings of 1s.
  - ✓ Can you find them?

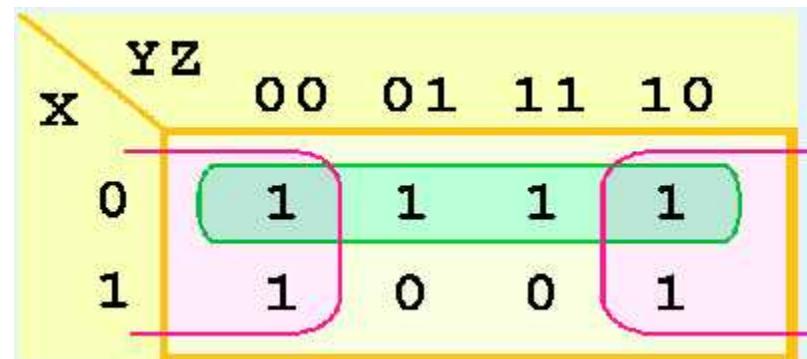
		YZ	00	01	11	10
		X	0	1	1	1
X	0	1	1	1	1	
	1	1	0	0	1	

# Kmap Simplification for Three Variables

---

- ❖ In this Kmap, we see an example of a group that wraps around the sides of a Kmap.
- ❖ This group tells us that the values of  $x$  and  $y$  are not relevant to the term of the function that is encompassed by the group.
  - What does this tell us about this term of the function?

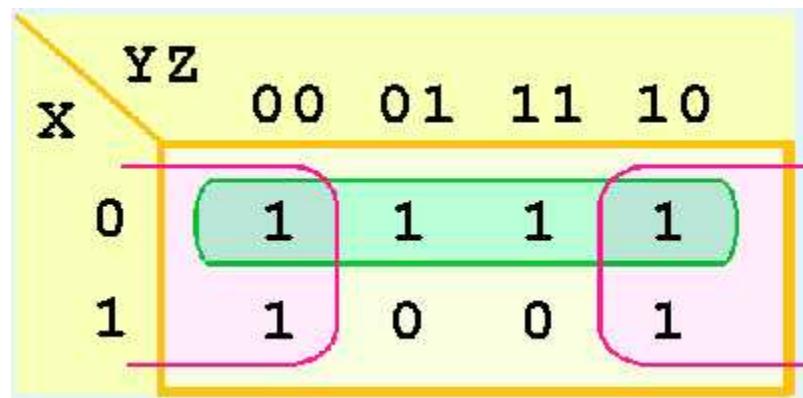
**What about the green group in the top row?**



# Kmap Simplification for Three Variables

- ❖ The green group in the top row tells us that only the value of  $x$  is significant in that group.
- ❖ We see that it is complemented in that row, so the other term of the reduced function is  $\bar{z}$ .
- ❖ Our reduced function is:  $F(x,y,z) = \bar{x} + \bar{z}$

❖ Recall that we had six minterms in our original function!



# Kmap Simplification for Four Variables

- ❖ Our model can be extended to accommodate the 16 minterms that are produced by a four-input function.
- ❖ This is the format for a 16-minterm Kmap.

		YZ	00	01	11	10
		WX	00	01	11	10
00	00	$\bar{W}\bar{X}Y\bar{Z}$	$\bar{W}X\bar{Y}Z$	$\bar{W}\bar{X}YZ$	$\bar{W}XY\bar{Z}$	
	01	$\bar{W}X\bar{Y}\bar{Z}$	$\bar{W}X\bar{Y}Z$	$\bar{W}XY\bar{Z}$	$\bar{W}XYZ$	
	11	$W\bar{X}\bar{Y}\bar{Z}$	$W\bar{X}\bar{Y}Z$	$WXY\bar{Z}$	$WXY\bar{Z}$	
	10	$W\bar{X}\bar{Y}\bar{Z}$	$W\bar{X}\bar{Y}Z$	$W\bar{X}YZ$	$W\bar{X}YZ$	

# Kmap Simplification for Four Variables

- We have populated the Kmap shown below with the nonzero minterms from the function:

$$F(W, X, Y, Z) = \bar{W}\bar{X}\bar{Y}\bar{Z} + \bar{W}\bar{X}\bar{Y}Z + \bar{W}\bar{X}Y\bar{Z} \\ + \bar{W}XY\bar{Z} + W\bar{X}\bar{Y}\bar{Z} + W\bar{X}\bar{Y}Z + W\bar{X}Y\bar{Z} + WXY\bar{Z}$$

- Can you identify (only) three groups in this Kmap?

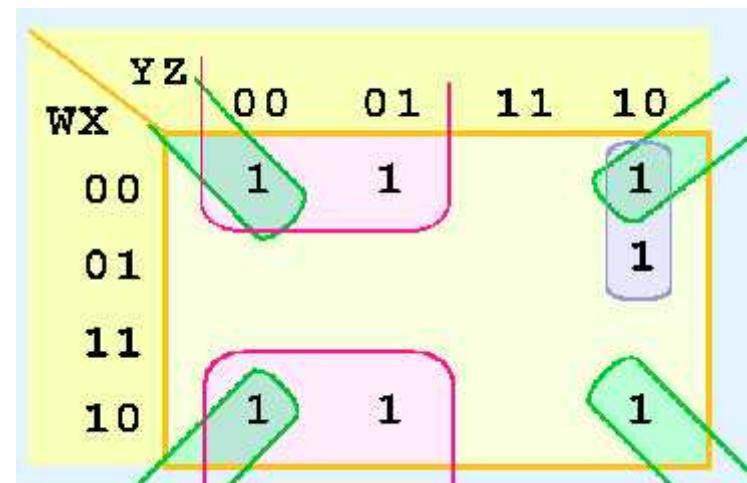
Recall that groups can overlap.

		YZ	00	01	11	10
		wx	00	01	11	10
w	0	1	1		1	
	1				1	
	0	1	1		1	

# Kmap Simplification for Four Variables

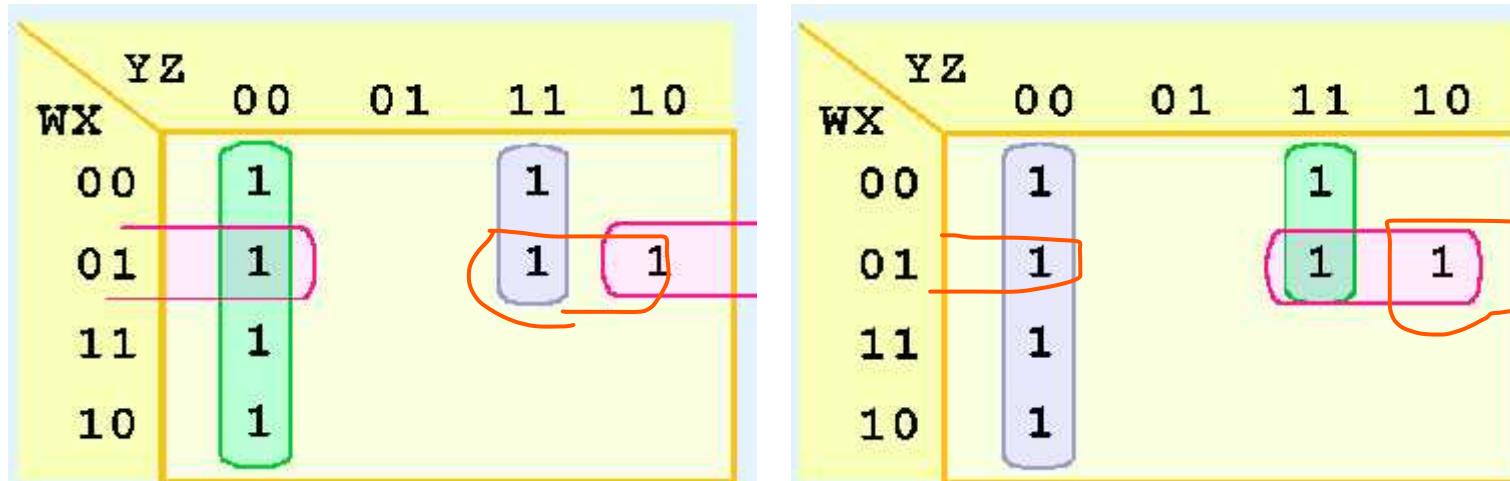
- ❖ Our three groups consist of:
  - ✓ A purple group entirely within the Kmap at the right.
  - ✓ A pink group that wraps the top and bottom.
  - ✓ A green group that spans the corners.
- ❖ Thus we have three terms in our final function:

$$F(W, X, Y, Z) = \\ \bar{X}\bar{Y} + \bar{X}\bar{Z} + \bar{W}YZ$$



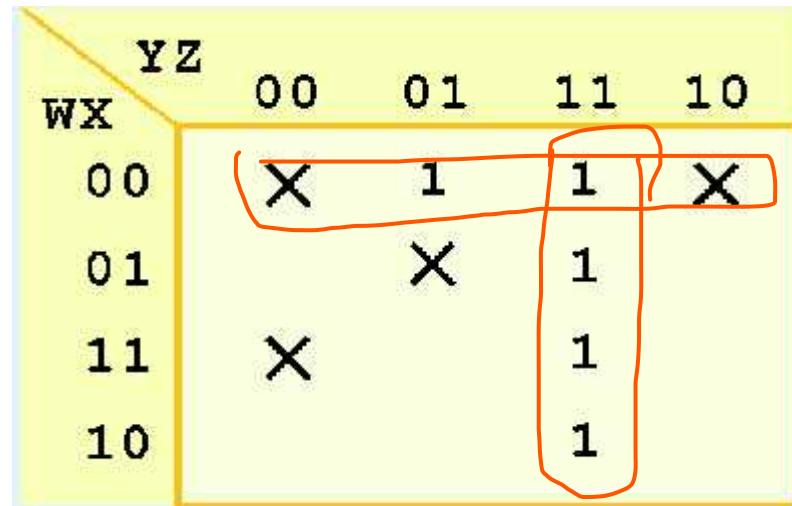
# Kmap Simplification for Four Variables

- ❖ It is possible to have a choice as to how to pick groups within a Kmap, while keeping the groups as large as possible.
- ❖ The (different) functions that result from the groupings below are logically equivalent.



# Don't Care Conditions

- ❖ In a Kmap, a don't care condition is identified by an  $X$  in the cell of the minterm(s) for the don't care inputs, as shown below.
- ❖ In performing the simplification, we are free to include or ignore the  $X$ 's when creating our groups.

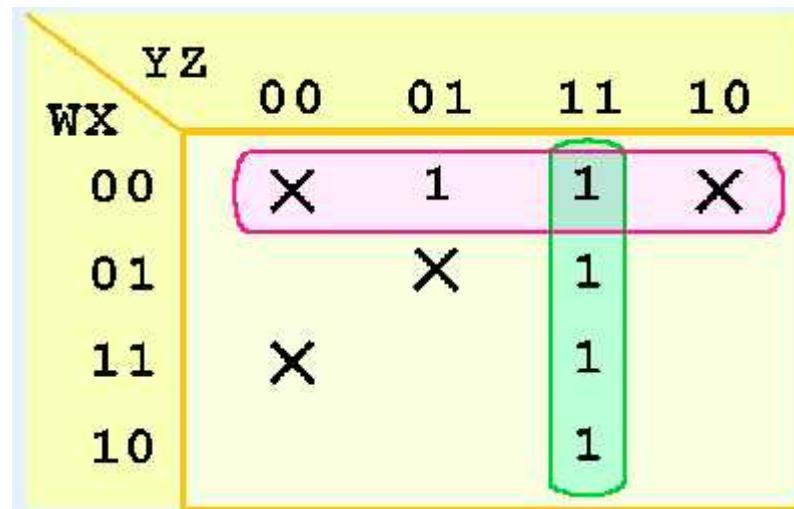


# Don't Care Conditions

---

- ❖ In one grouping in the Kmap below, we have the function:

$$F(W, X, Y, Z) = \bar{W}\bar{X} + YZ$$



# Don't Care Conditions

---

- ❖ A different grouping gives us the function:

$$F(W, X, Y, Z) = \bar{W}Z + YZ$$

WX \ YZ	00	01	11	10
00	X	1	1	X
01	X		1	
11	X		1	
10			1	

# Don't Care Conditions

- ❖ The truth table of:

$$F(W, X, Y, Z) = \overline{W}\overline{X} + YZ$$

- ❖ is different from the truth table of:

$$F(W, X, Y, Z) = \overline{W}Z + YZ$$

- ❖ However, the values for which they differ, are the inputs for which we have don't care conditions.

Y Z	00	01	11	10
W X	X	1	1	X
Y Z	00	X	1	1
Y Z	01	X	1	1
Y Z	11	X	1	1
Y Z	10		1	1

Y Z	00	01	11	10
W X	X	1	1	X
Y Z	00	X	1	1
Y Z	01	X	1	1
Y Z	11	X	1	1
Y Z	10		1	1

# Summery

---

Recapping the rules of Kmap simplification:

- ❖ Groupings can contain only 1s; no 0s.
- ❖ Groups can be formed only at right angles; diagonal groups are not allowed.
- ❖ The number of 1s in a group must be a power of 2 – even if it contains a single 1.
- ❖ The groups must be made as large as possible.
- ❖ Groups can overlap and wrap around the sides of the Kmap.
- ❖ Use don't care conditions when you can.

# Thank You

---

End of Lecture 5.1

Lecture 5.2 : Combinational and  
Sequential Circuits



# Introduction to Computer Systems

---

LECTURE 6: COMBINATIONAL AND SEQUENTIAL  
LOGIC CIRCUITS

# Objectives

---

After completing this lesson you will be able to:

- ❖ Understand different types of Digital Logic Circuits.
- ❖ Understand the characteristics of the DLC.
- ❖ Understand the usage of each type.



# Digital Logic Circuits

---

# Digital Logic Circuits

---

Logic circuits can be categorized as

- ❖ Combinational Circuits
- ❖ Sequential circuits.

A combinational circuit consists of input variables, logic gates and output variables.

- ❖ Output depends only on current input
- ❖ Has no memory

# Combinational Circuits

---

A combinational circuit consists of input variables, logic gates, and output variables.

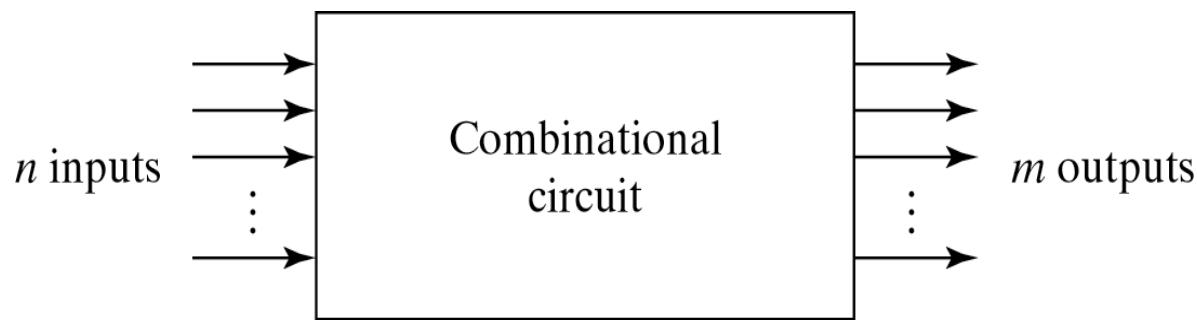
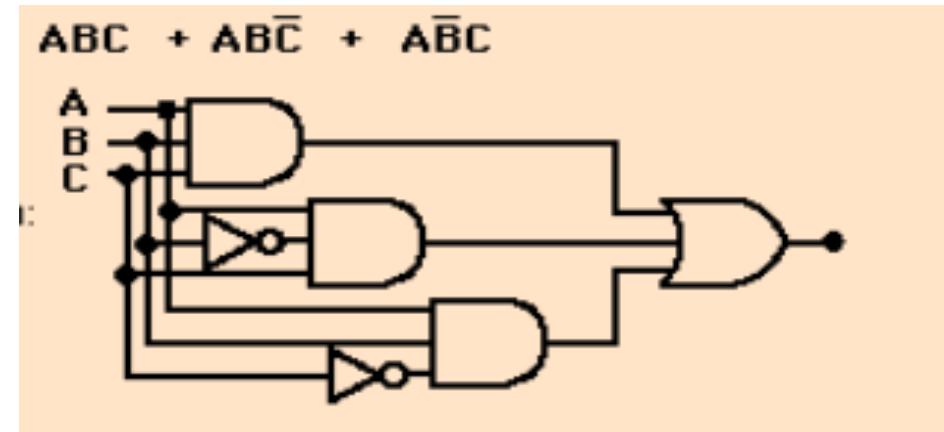
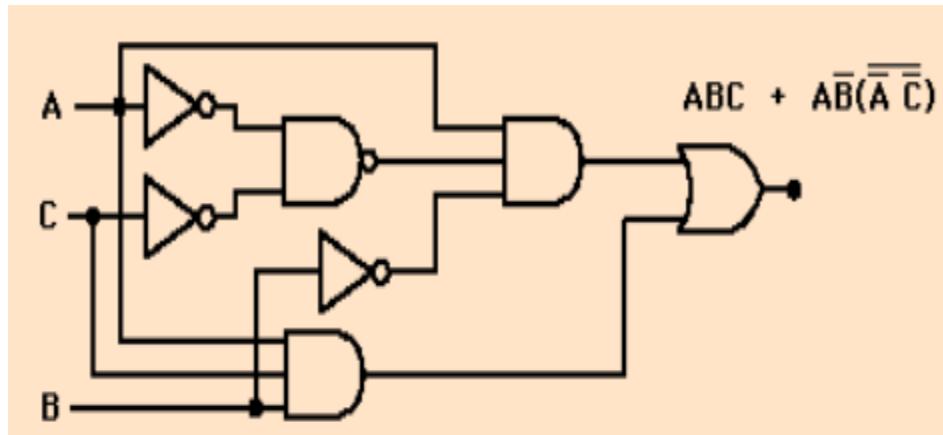


Fig. 4-1 Block Diagram of Combinational Circuit

# Combinational Circuits (Examples)

---



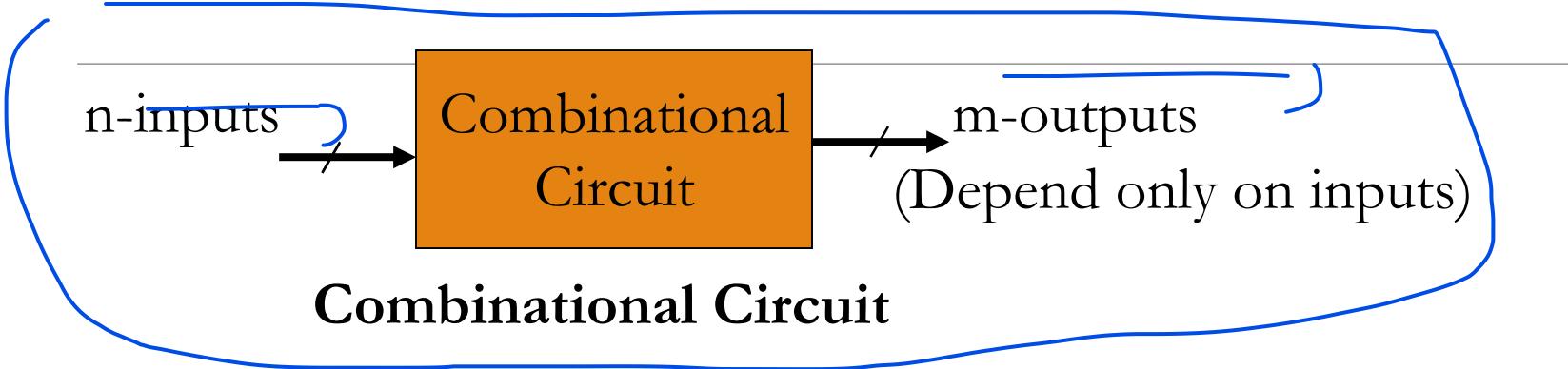
# Sequential Logic

---

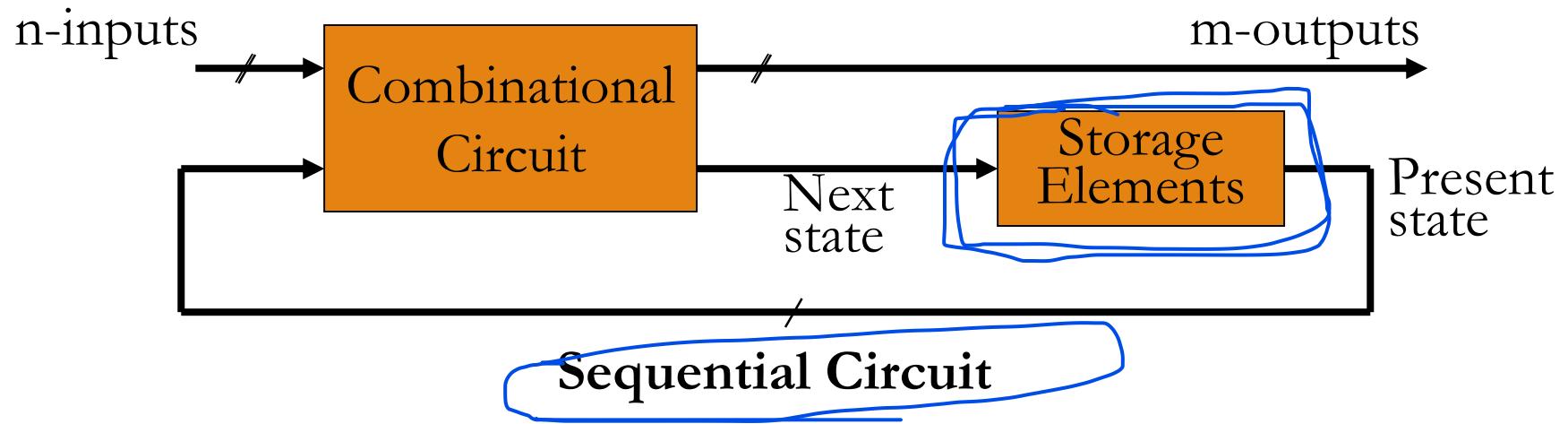
## Sequential Logic:

- ❖ Output depends not only on current input but also on past input values, e.g., design a counter
- ❖ Need some type of memory to remember the past input values

# Combinational vs. Sequential Circuits



Combinational Circuit



Sequential Circuit

---

# Combinational Circuits

# Combinational Circuits

## Half Adder

Half Adder : The sum is **XOR** operation and the carry an **AND**

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

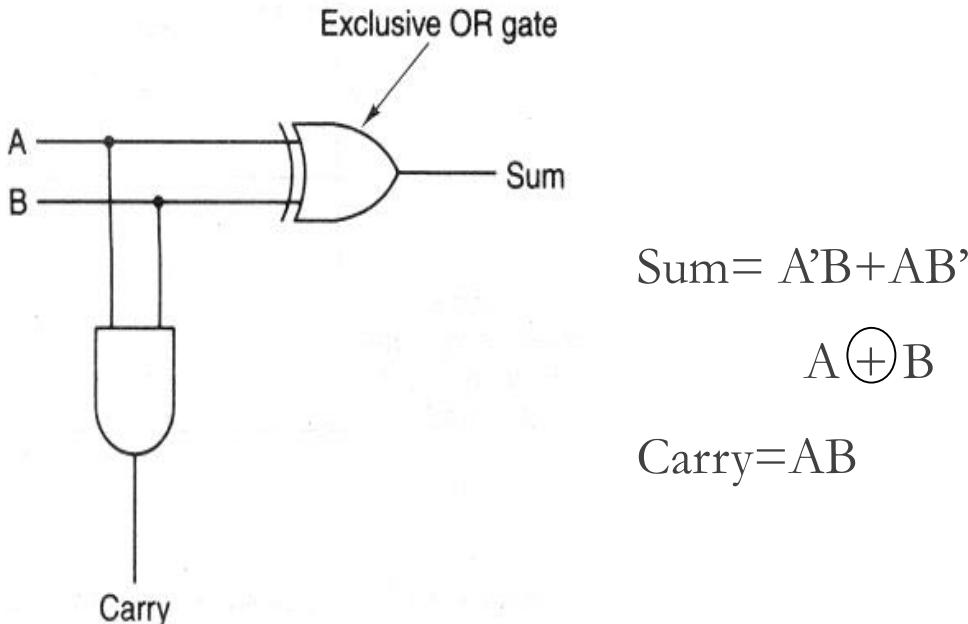


Figure 4-23 (a) Truth table for 1 bit addition. (b) A circuit for a half adder

# Combinational Circuits – Full Adder

5. Draw a diagram

Input			Output	
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

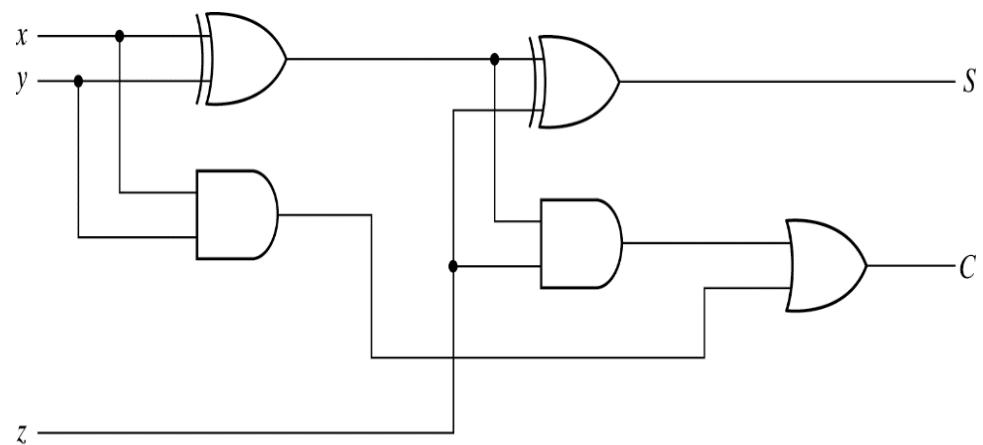
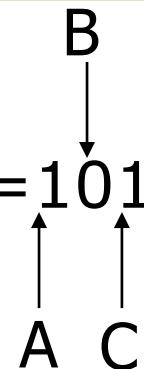


Fig. 4-8 Implementation of Full Adder with Two Half Adders and an OR Gate

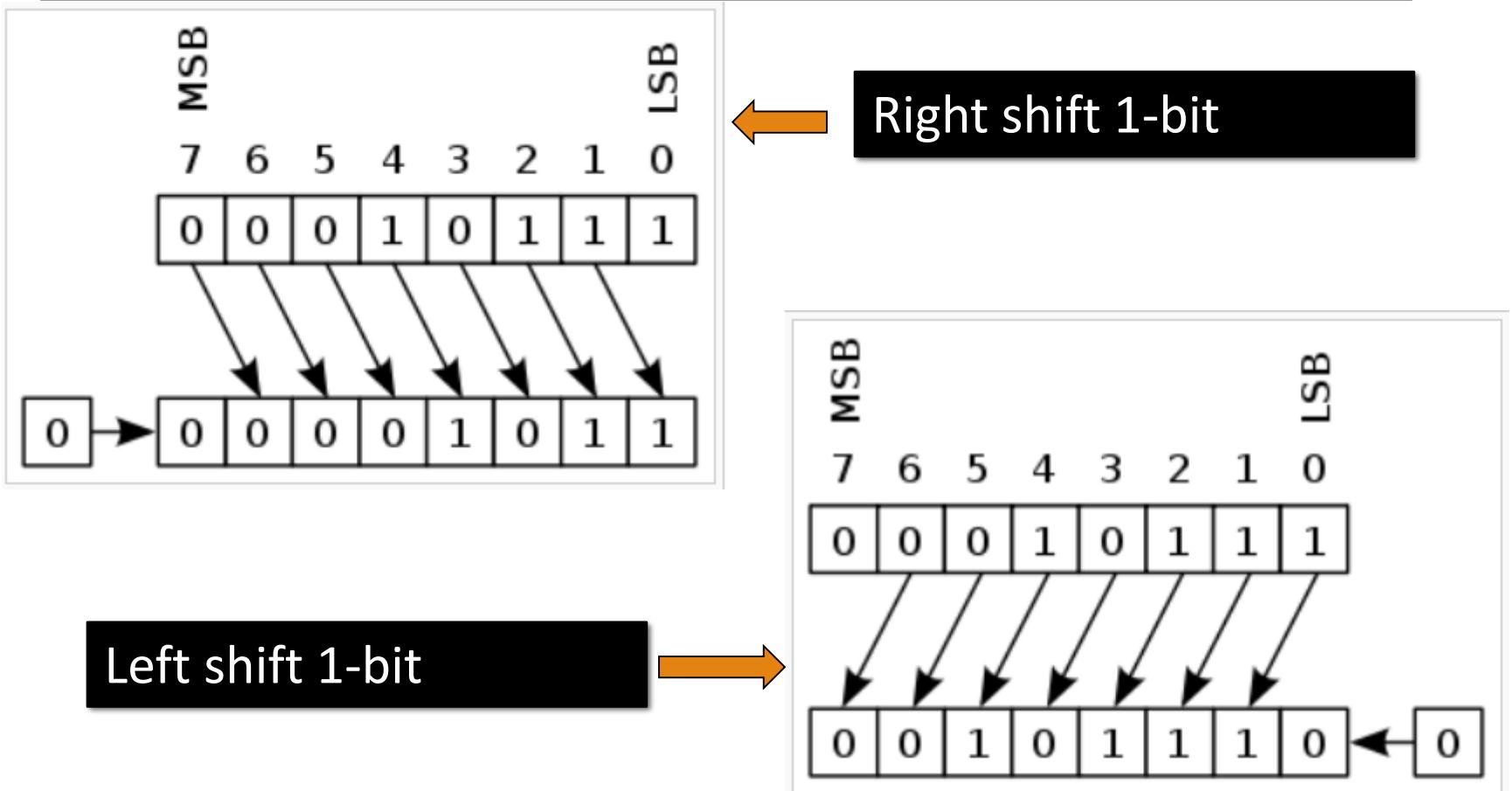
# Binary Multiplication

□ E.g:  $15 * 5 = 75$ ,  $15=1111, 5=101$  <-multiplier

- 0000 0000 initialize to 0
- 0000 1111  $A = 1$
- 0000 1111 sum
- 0001 1110 shift  $\leftarrow$  (left)
- 0000 0000  $B = 0$
- 0001 1110 sum
- 0011 1100 shift  $\leftarrow$  (left)
- 0000 1111  $C = 1$
- 0100 1011 sum, no shift



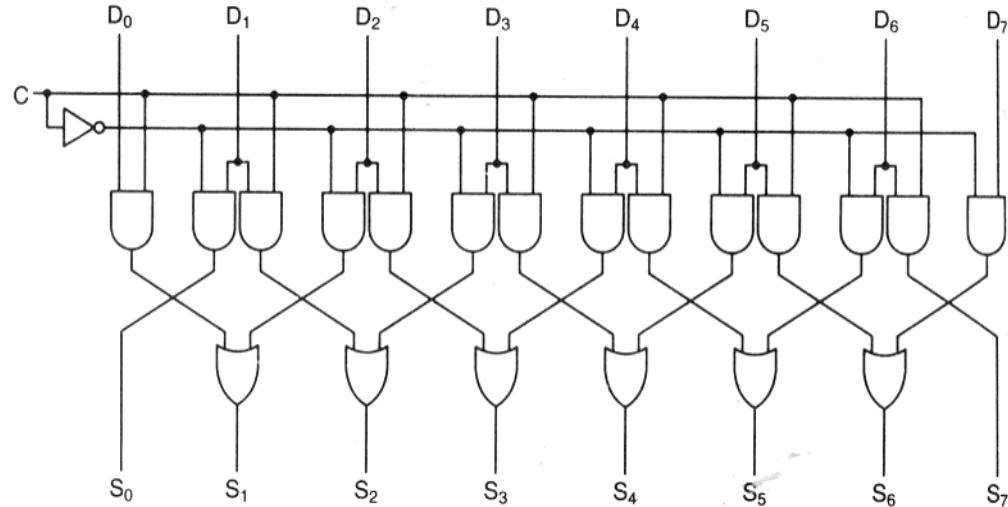
# Shifter



# Shifter

Figure shows an eight-input, eight-output, 1-bit left/right shifter.

- An eight-bit input is applied onto D<sub>0</sub>, D<sub>1</sub>...D<sub>7</sub>, and the shifted output is taken out of S<sub>0</sub>, S<sub>1</sub>...S<sub>7</sub>.
- The shift direction is controlled with the control signal C.



**Figure 4-22.** A 1-bit left/right shifter.

- A logic one on C enables the right hand side AND gates and pass each input bit to the OR gate to its right and hence, a one-bit right shift is performed.
- Similarly, a logic zero on C performs a left shift.

# Comparator

---

A comparator circuit compares two input words.

The simplest comparator circuit is an exclusive NOR gate, which compares two individual bits:

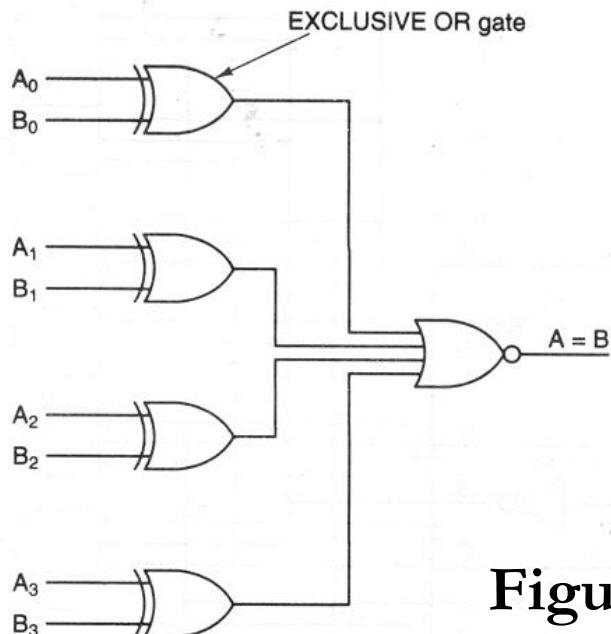
- ❖ produces a zero if the two bits are unequal
- ❖ produces a logic one if they are equal

# Comparator

---

When two words of length  $n$  are to be compared,

- $n$  XOR gates and
- a NOR gate can be combined



**Figure:** A simple 4-bit comparator

# Decoder/Encoder

---

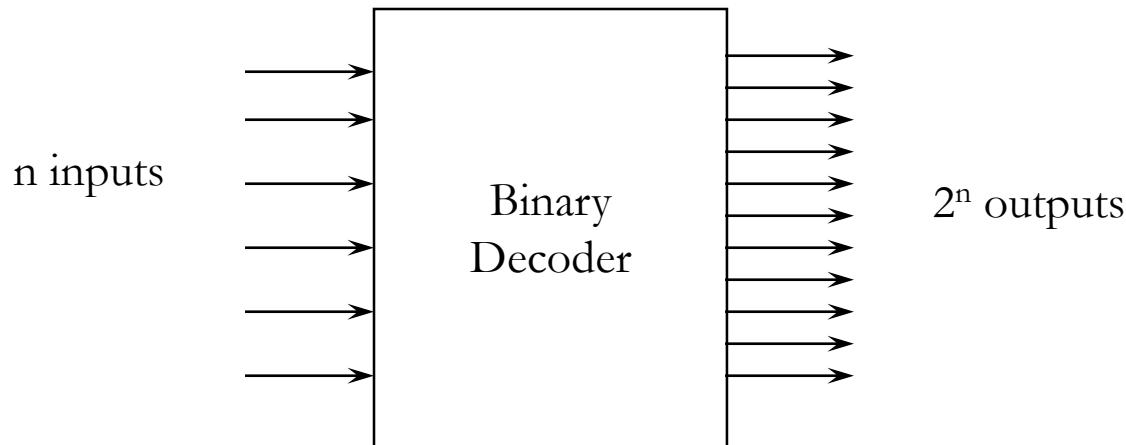
## Decoder

- A combinational circuit that converts binary information from the  $n$  coded inputs to a maximum of  $2^n$  unique outputs
- **$n$ -to- $m$**  line decoder =  **$n \times m$**  decoder Ex: Octal to Binary line decoder
  - **$n$**  inputs,  **$m$**  outputs (3 inputs, 8 outputs)
  - If the  $n$ -bit coded information has unused bit combinations, the decoder may have less than  $2^n$  outputs
  - $m \leq 2^n$

# Binary Decoder

---

- Black box with  $n$  input lines and  $2^n$  output lines

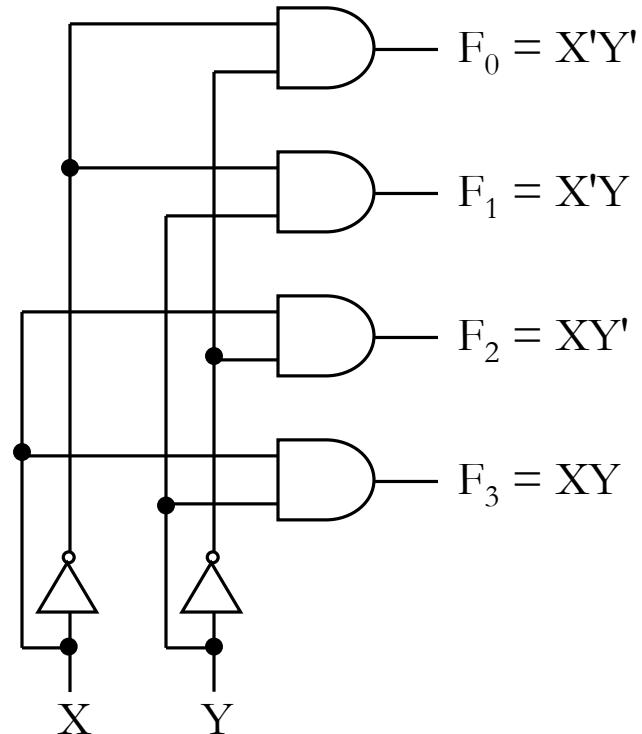
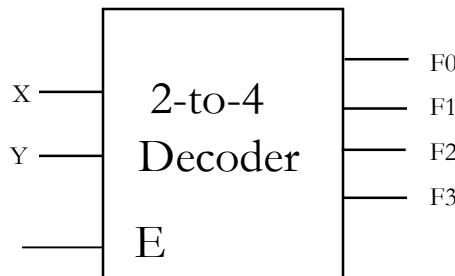


# 2-to-4 Binary Decoders

Truth Table:

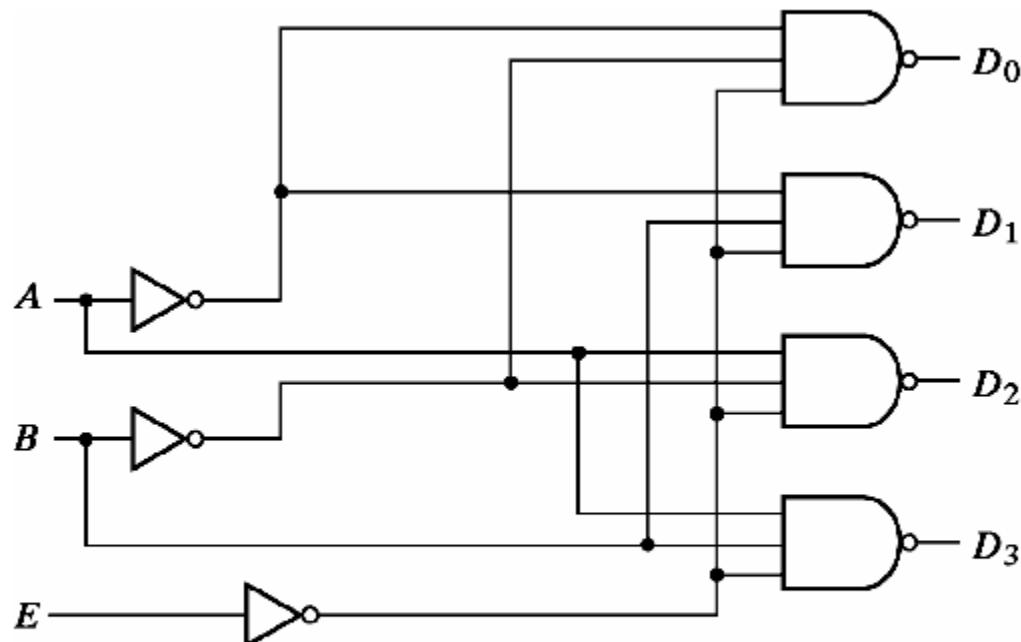
X	Y	F <sub>0</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- From truth table, circuit for 2x4 decoder is:
- Note: Each output is a 2-variable minterm ( $X'Y'$ ,  $X'Y$ ,  $XY'$  or  $XY$ )



# 2-to-4 Decoders : NAND Implementation

Decoder is enabled when  $E=0$  and an output is active if it is 0



(a) Logic diagram

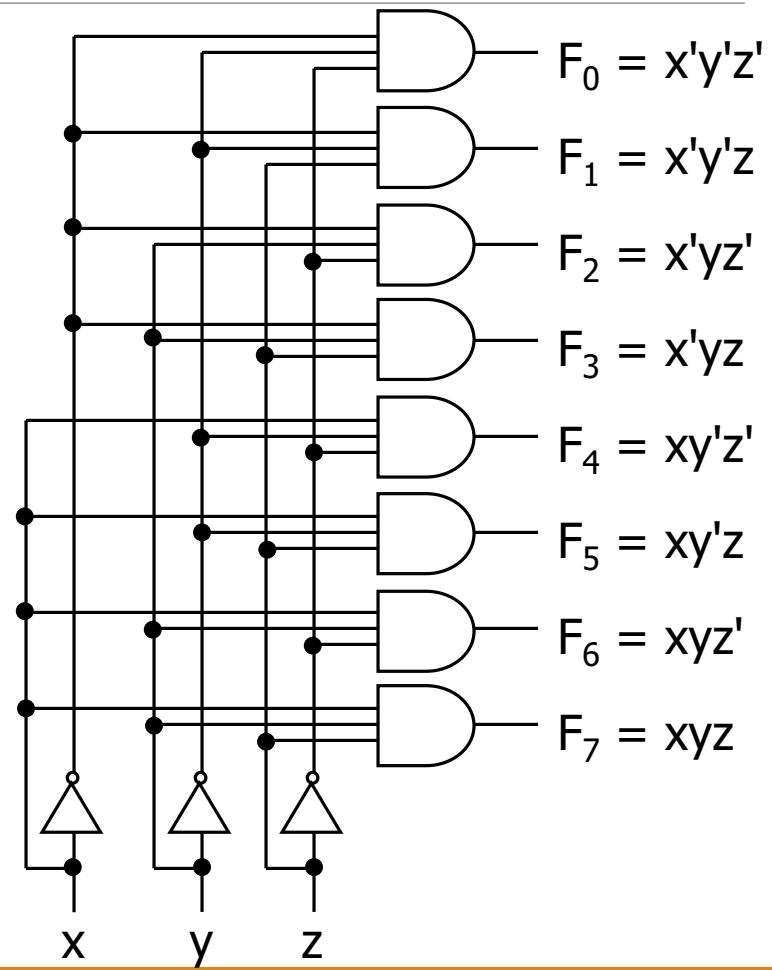
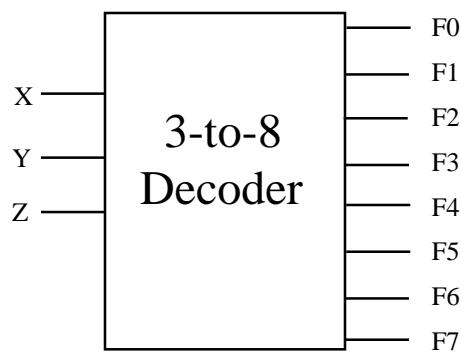
$E$	$A$	$B$	$D_0$	$D_1$	$D_2$	$D_3$
1	$X$	$X$	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

# 3-to-8 Binary Decoder

Truth Table:

x	y	z	F <sub>0</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>4</sub>	F <sub>5</sub>	F <sub>6</sub>	F <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



# Decoder Expansion

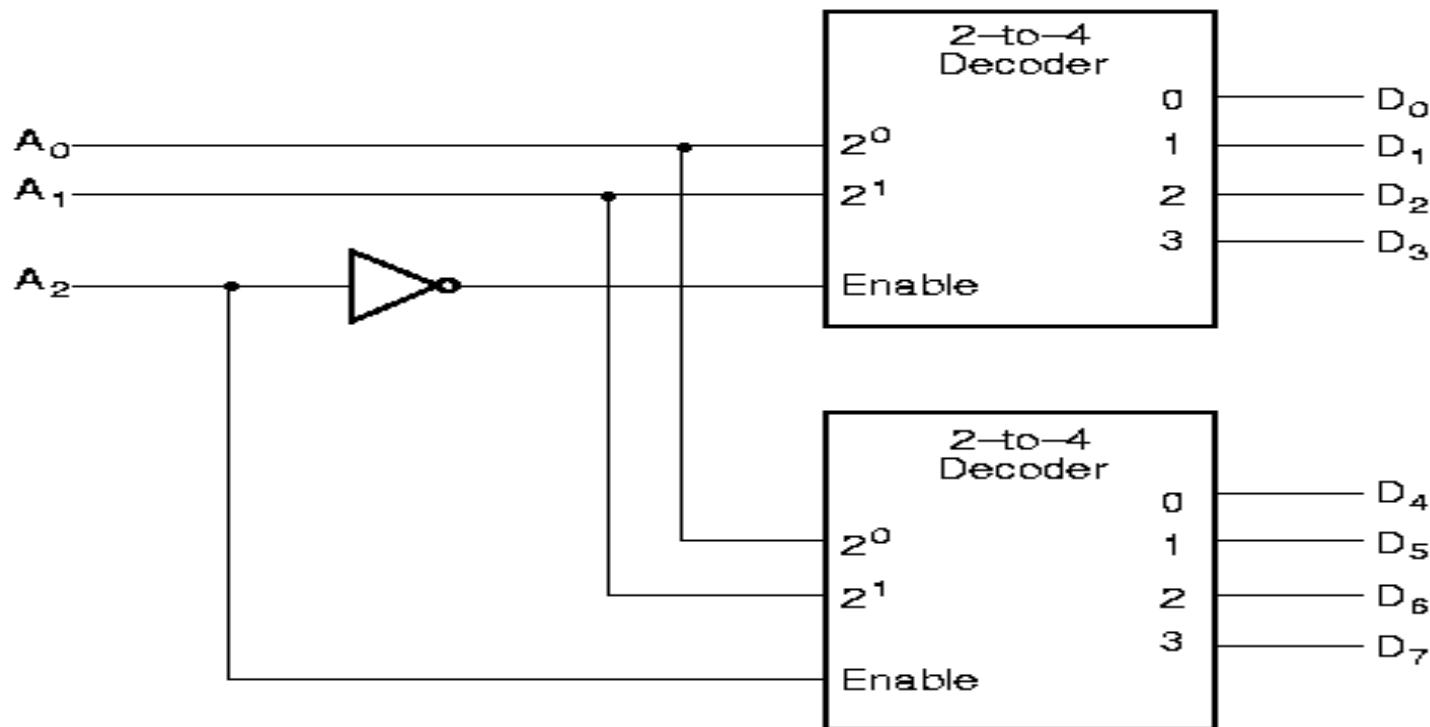
---

## Decoder expansion

- Combine two or more small decoders with enable inputs to form a larger decoder
- 3-to-8-line decoder constructed from two 2-to-4-line decoders
  - The **MSB** is connected to the enable inputs
  - if  $A_2=0$ , upper is enabled; if  $A_2=1$ , lower is enabled.

# Decoder Expansion(Contd.)

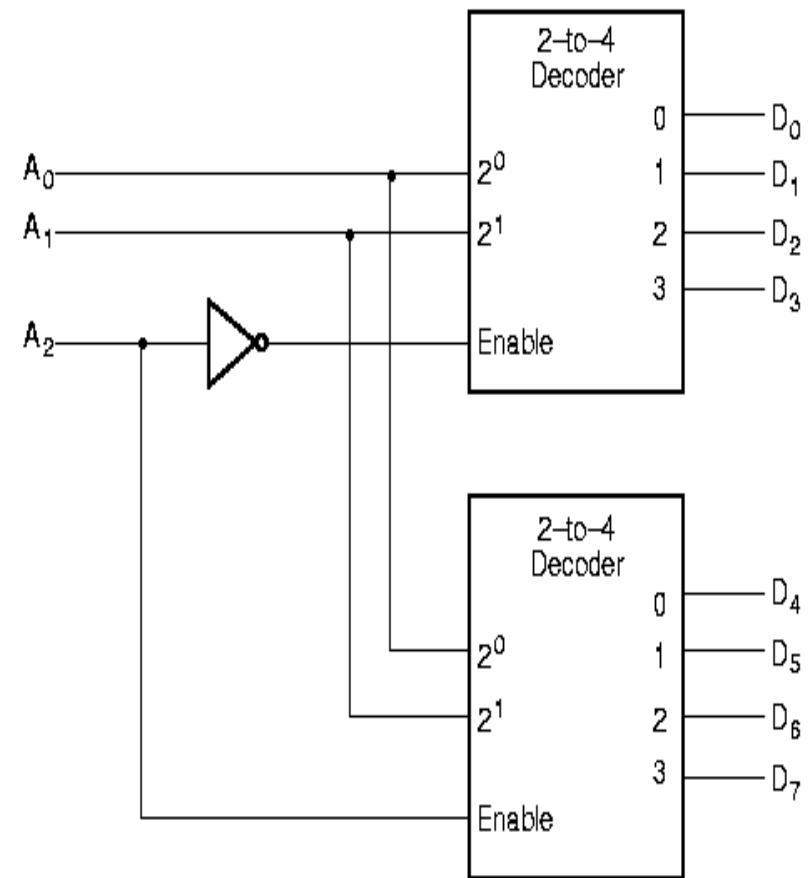
---



# Combining two 2-4 decoders to form one 3-8 decoder using enable switch

A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

The highest bit is used for the enables



# Combinational Circuit Design with Decoders

---

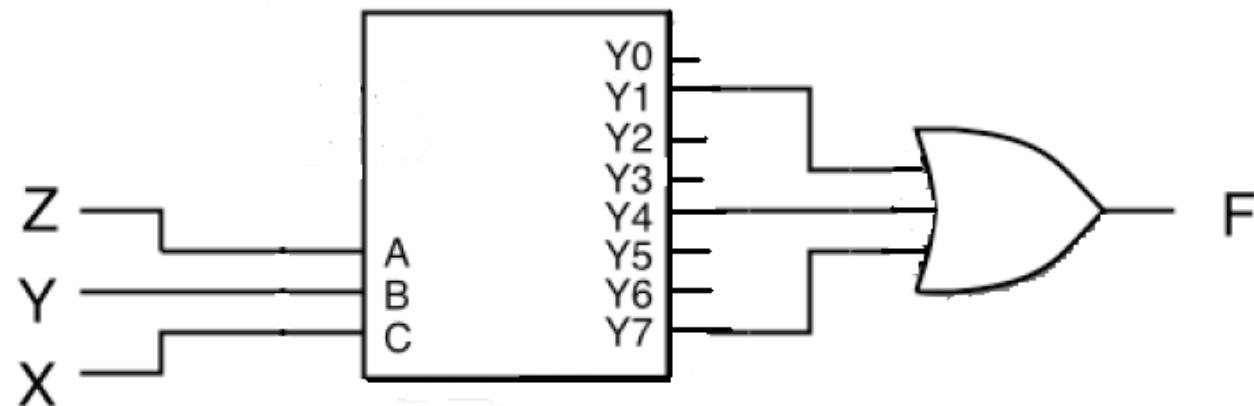
## Combinational circuit implementation with decoders

- ❖ A decoder provide  $2^n$  minterms of  $n$  input variables
- ❖ Since any Boolean function can be expressed as a sum of minterms, one can use a decoder and external OR gates to implement any combinational function.

# Combinational Circuit Design with Decoders

---

Example Realize  $F(X,Y,Z) = \Sigma(1, 4, 7)$  with a decoder:



# Encoder

Inverse Operation of a decoder

$2^n$  input,  $n$  output

Truth Table

3 OR Gates Implementation

- $A_0 = D_1 + D_3 + D_5 + D_7$
- $A_1 = D_2 + D_3 + D_6 + D_7$
- $A_2 = D_4 + D_5 + D_6 + D_7$

Inputs								Outputs		
D7	D6	D5	D4	D3	D2	D1	D0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

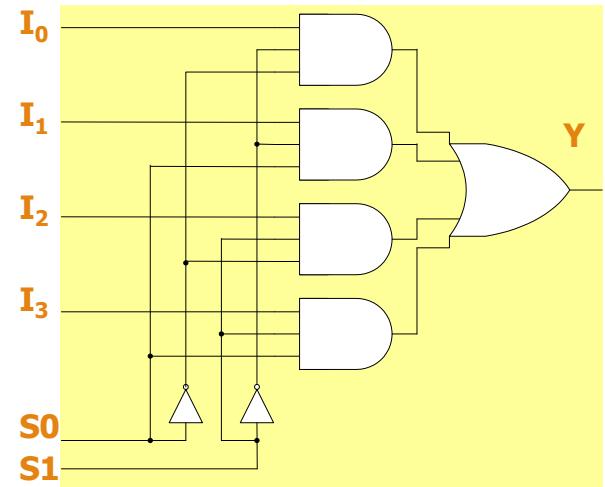
*Truth Table for Encoder*

# Multiplexer (MUX)

- A combinational circuit that receives binary information from one of  $2^n$  input data lines and directs it to a single output line
- A  $2^n$  -to 1 multiplexer has  $2^n$  *input data lines* and *n input selection lines*
- 4-to-1 multiplexer Diagram
- 4-to-1 multiplexer Function Table

**Function Table for  
4-to-1 line Multiplexer**

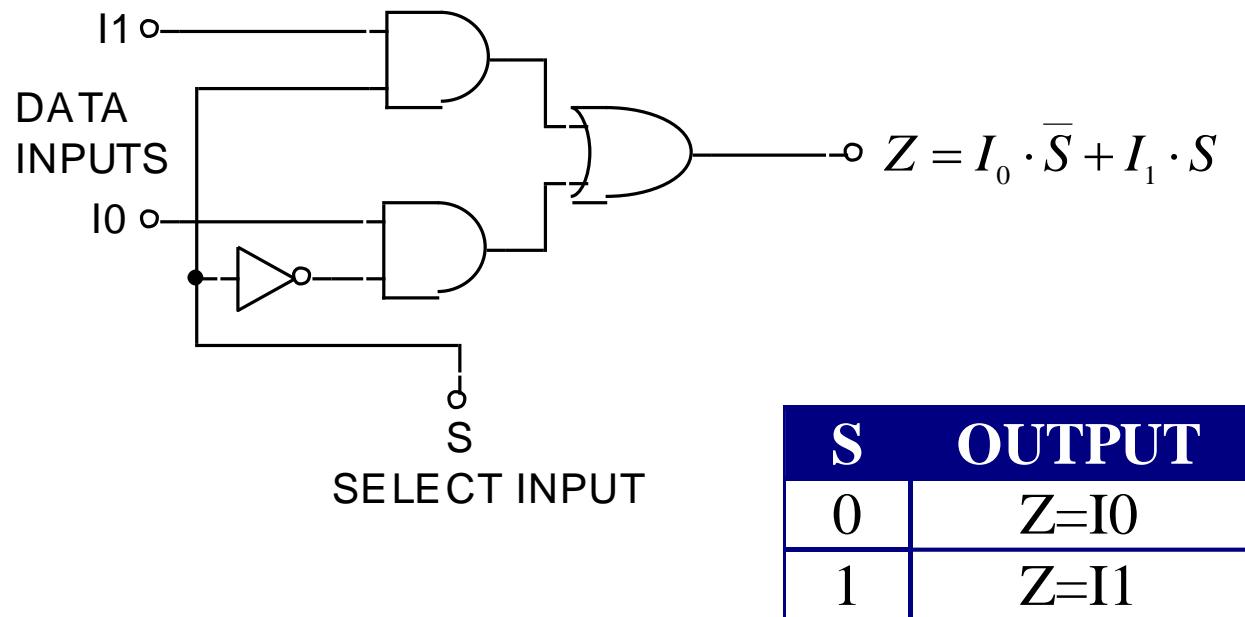
Select		Output
S1	S0	Y
0	0	I <sub>0</sub>
0	1	I <sub>1</sub>
1	0	I <sub>2</sub>
1	1	I <sub>3</sub>



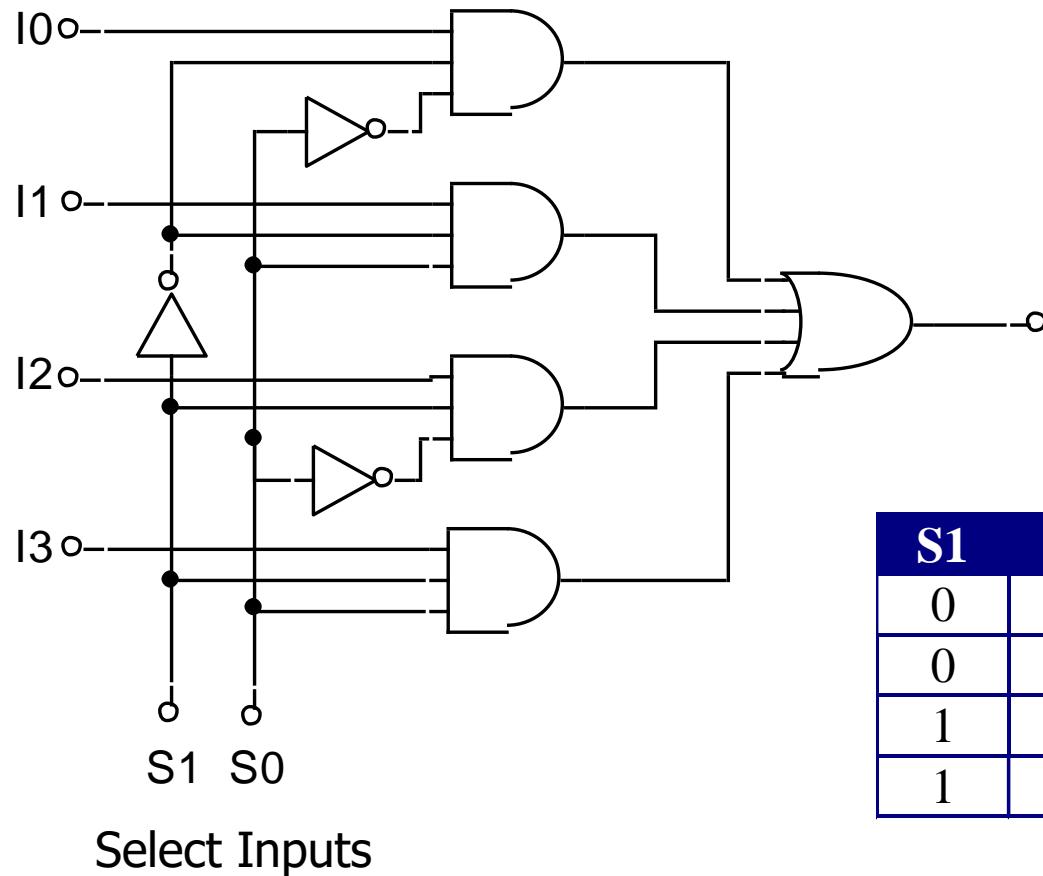
**Fig. 4-to-1 Line Multiplexer**

Usage: Data routing, Parallel-to-serial conversion, Operation sequencing, Implement logic function of a truth table

# Basic 2-Input Multiplexer

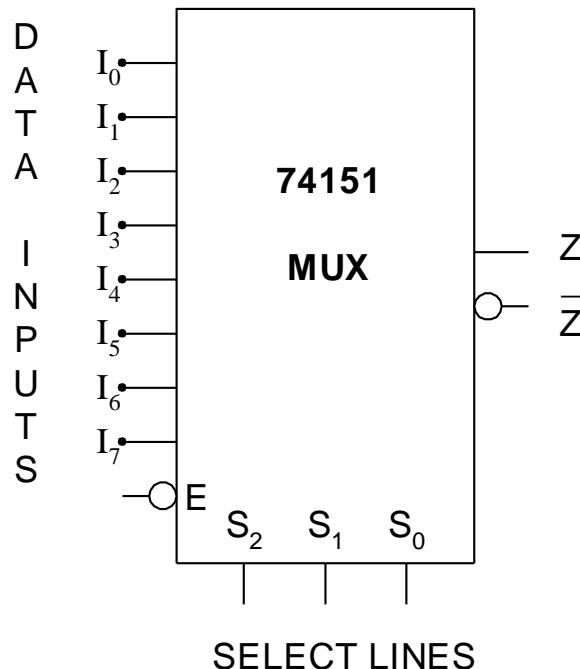


# 4-Input Multiplexer



# Multiplexer Logic Diagram

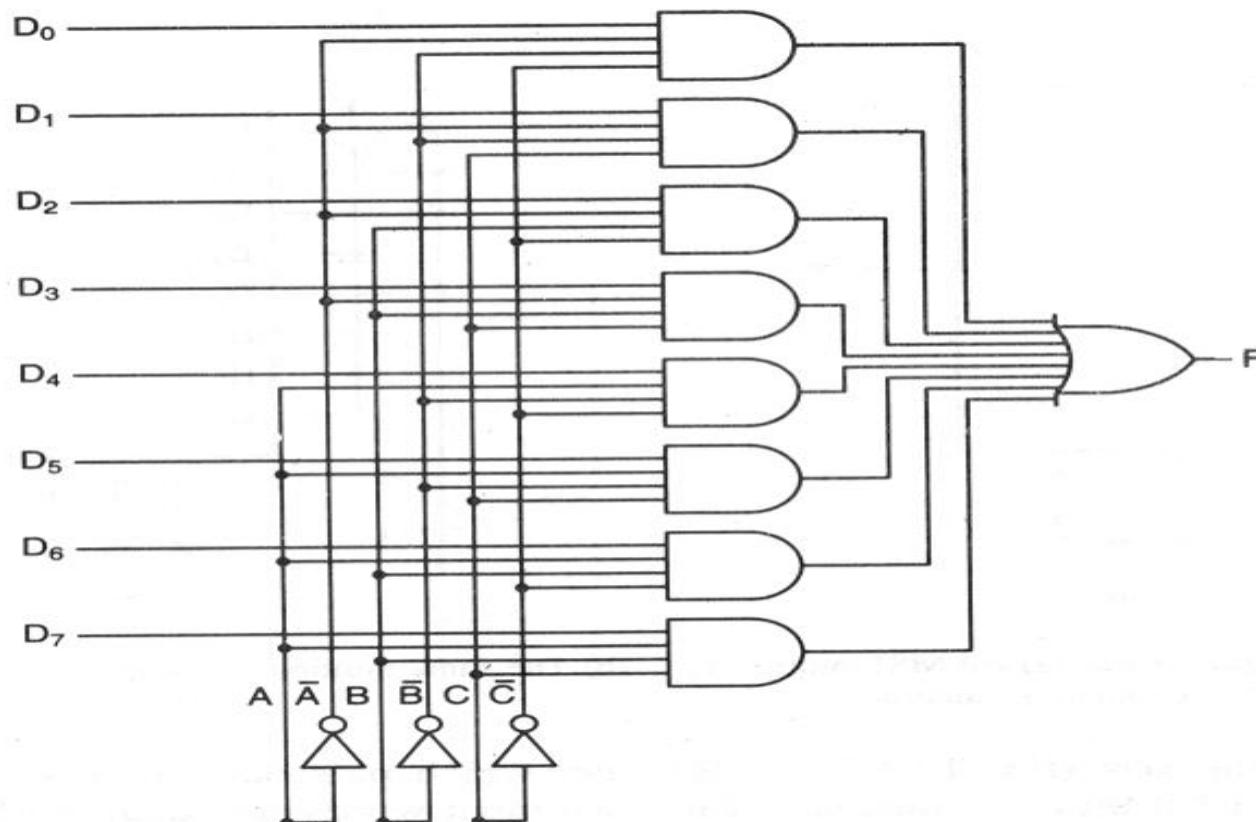
- ❖ Takes one of many inputs and funnels it to an output Z.
- ❖ Take the selector lines convert to a decimal number and this is the input funneled to the output.
- ❖ Strobe is active low enable



S2	S1	S0	E	Z
0	0	0	0	$I_0$
0	0	1	0	$I_1$
0	1	0	0	$I_2$
0	1	1	0	$I_3$
1	0	0	0	$I_4$
1	0	1	0	$I_5$
1	1	0	0	$I_6$
1	1	1	0	$I_7$

# Multiplexer

---



**Figure :**A 8-line to 1-line multiplexer

# Programmable Logic Arrays (PLA)

---

Programmable logic arrays (PLA) are very general chips that provide a quick and easy way to implement Boolean functions expressed in the sum-of-products form.

- A PLA comprises of an array of AND gates and another array of OR gates.
  - Each one of the AND gate outputs is tapped through tiny fuses to the inputs of each OR gate.
  - Similarly, the input lines of the PLA chip are tapped to the inputs of each AND gate.
- Applying a high voltage to the chip a designer can blow selected fuses leaving only the desired circuit connections intact.
- Therefore, by choosing the proper set of fuses to blow, a designer can implement or “program” the desired Boolean functions rapidly.

# Programmable Logic Arrays (PLA)

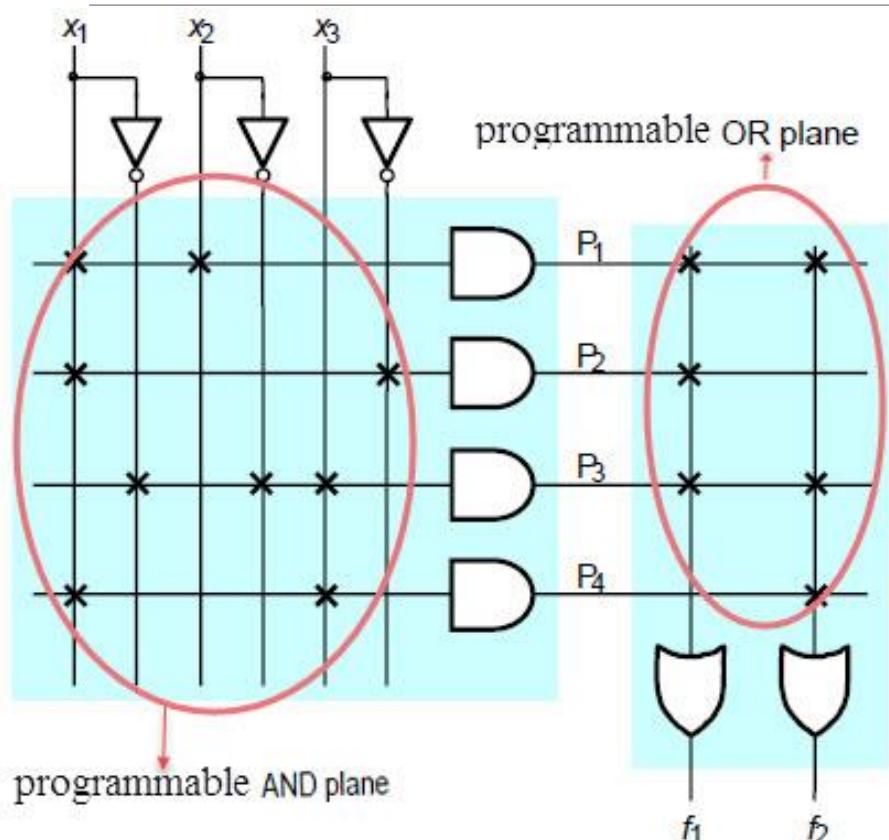


Figure: Customary schematic for the PLA

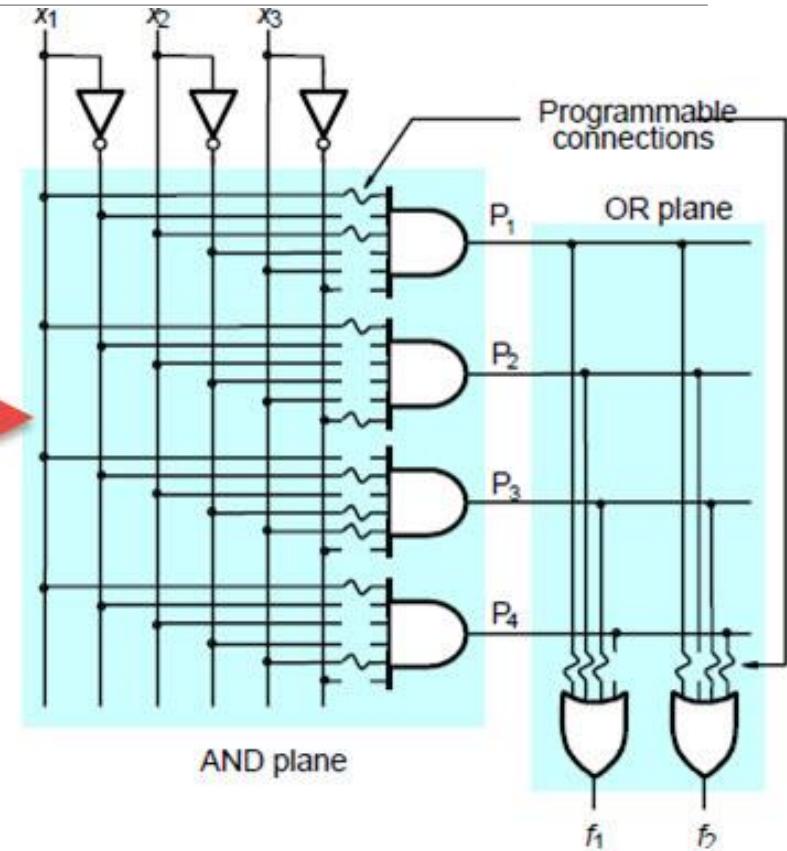


Figure: Gate-level diagram of a PLA

# Arithmetic Logic Unit (ALU)

---

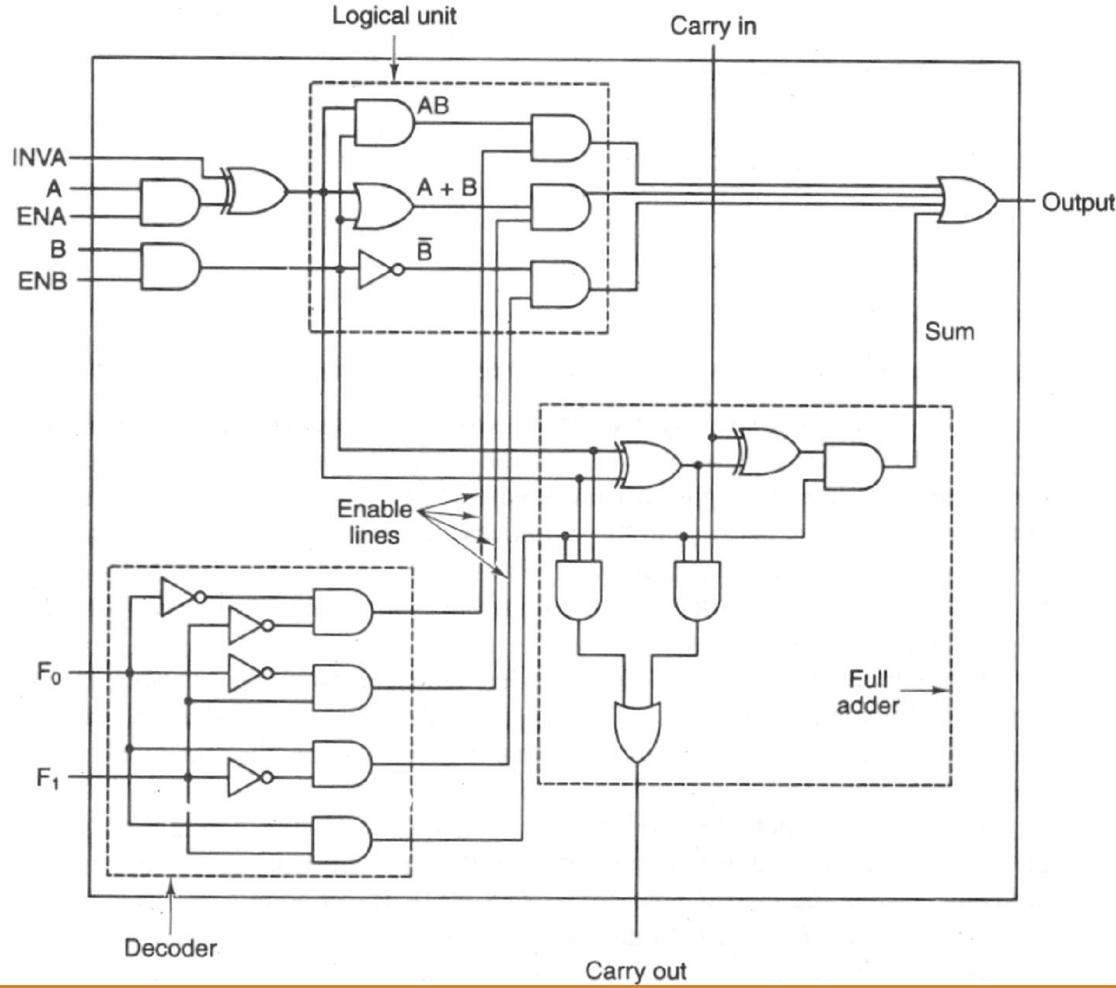
At the heart of any computer CPU is the arithmetic logic unit or the ALU.

It is a circuit that performs AND, OR, and sum operations of two input words.

AND and OR logical operations are easily implemented using the corresponding logic gates.

The sum operation is achieved using the full adder circuit described earlier.

# Arithmetic Logic Unit (ALU)



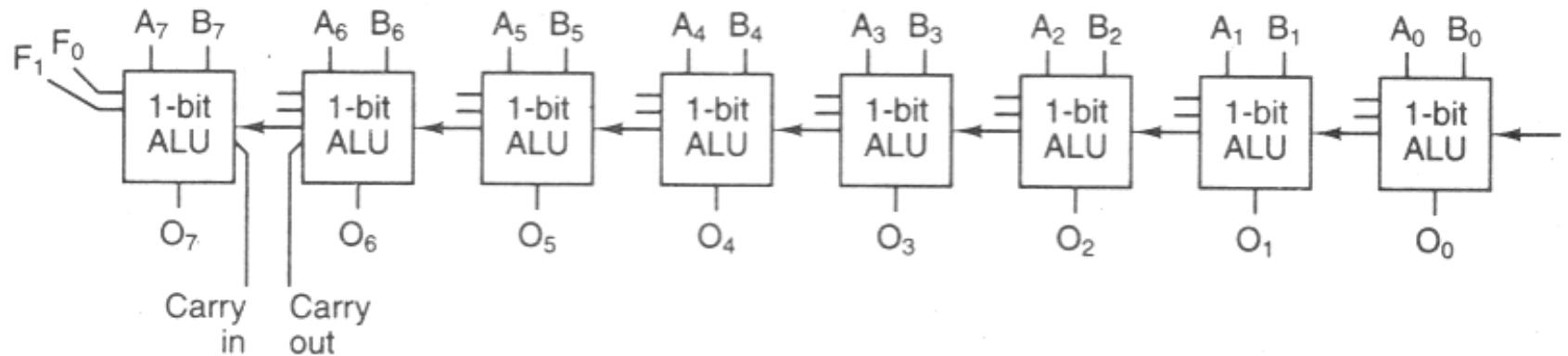
# Arithmetic Logic Unit (ALU)

---

If the CPU uses  $n$ -bit machine words, then  $n$  such ALU blocks must be connected together.

While the AND and OR operations only need the  $n$  ALUs to be placed in the  $n$  bit positions, the SUM operation requires the Carry-in and Carry-out signals of adjacent units cascaded

# Arithmetic Logic Unit (ALU)



**Figure 4-27.** Cascading eight 1-bit ALUs to form an 8-bit ALU

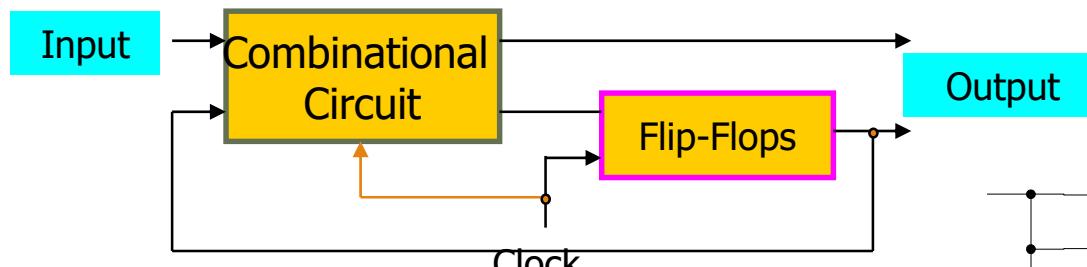
---

# Sequential Circuits

# Sequential Circuits

A sequential circuit is an interconnection of F/F and Gate

Clocked synchronous sequential circuit

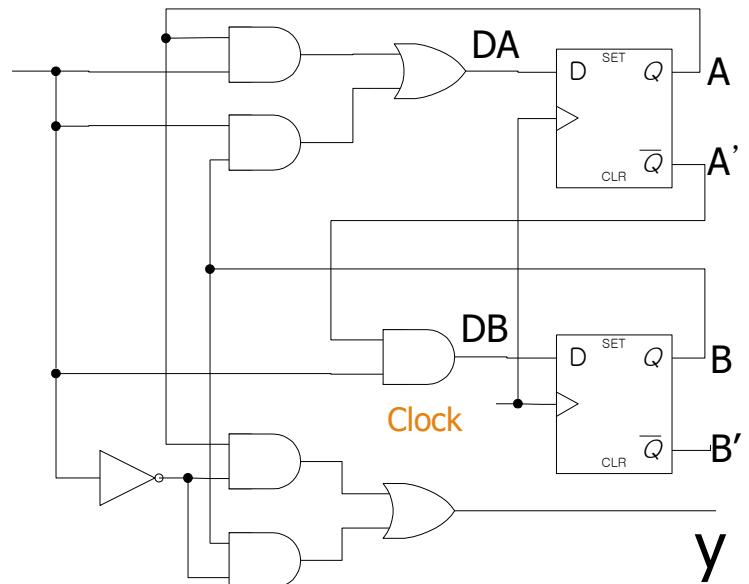


Flip-Flop Input Equation

- Boolean expression for F/F input
- Input Equation
  - $D_A = Ax + Bx, D_B = A'x$
- Output Equation
  - $y = Ax' + Bx'$
- Example of a sequential circuit

Combinational Circuit = Gate  
Sequential Circuit = Gate + F/F

X



Y

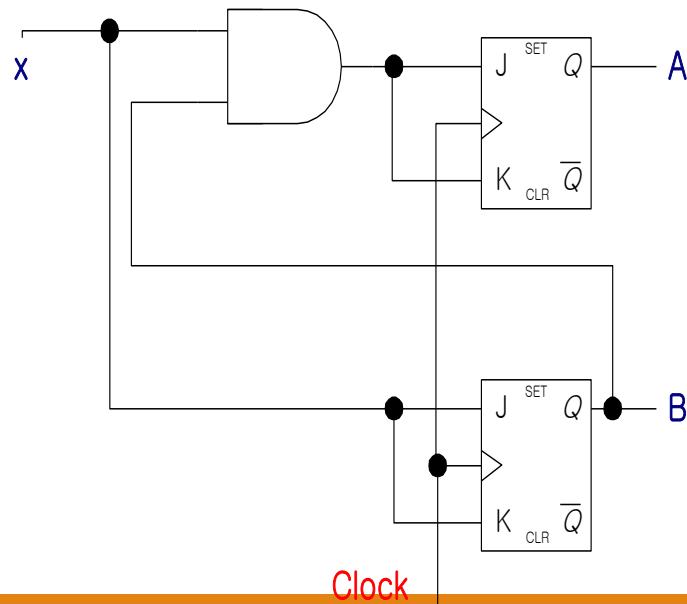
# Sequential Circuits

## Sequential Circuit Design Procedure

Recall Combinational Circuit Design

Sequential Circuit: State diagram, State table

F/F :  $2^{m+n}$  (m - State , n - Input )



◆ Logic Diagram

1. The Problem is stated
2. I/O variables are assigned
3. Truth table(I/O relation)
4. Simplified Boolean Function
5. Logic circuit diagram

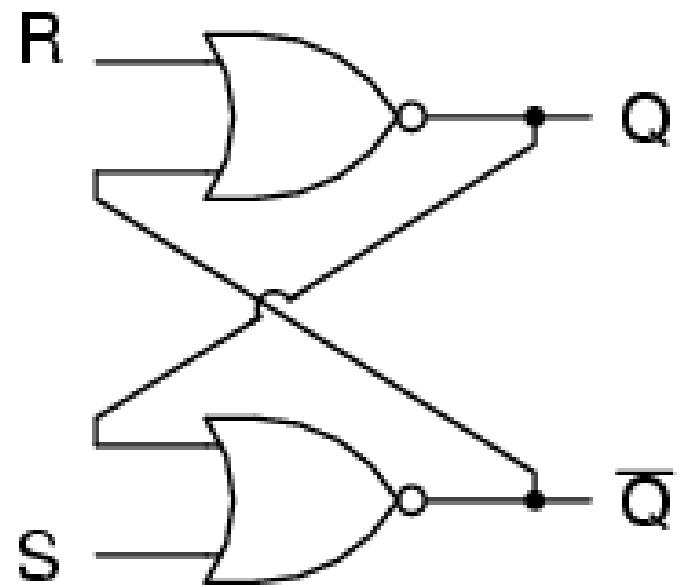
# Latches

---

## SR Latch

S	R	Action
0	0	Keep state
0	1	$Q=0$
1	0	$Q=1$
1	1	Restricted Combination

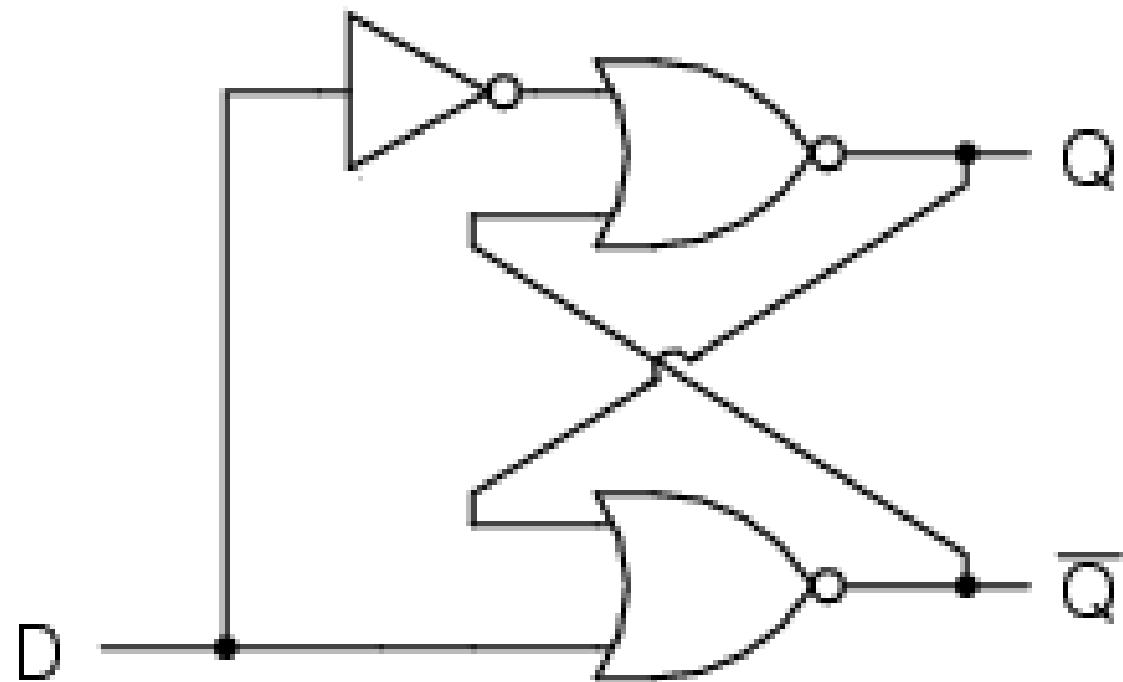
Characteristic Table



# Latches

---

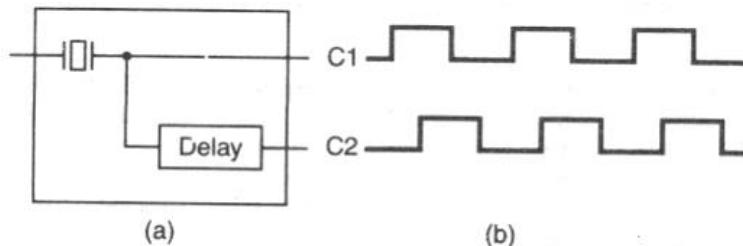
## D Latch



# Clock Signals

Clock signals are used to maintain the desired timing in the circuits.

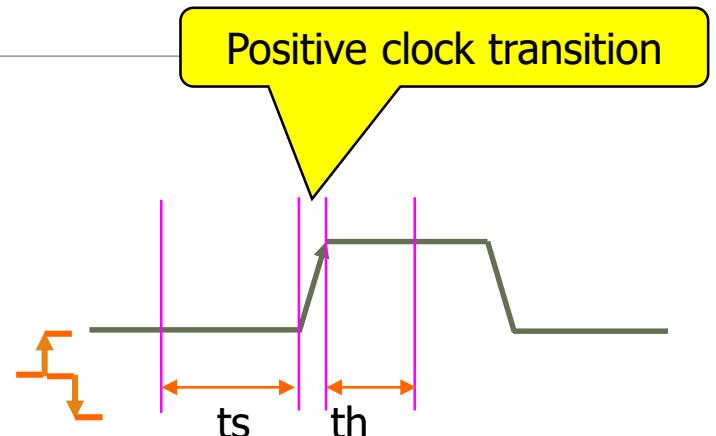
- Clock circuits emit pulse trains of precise repetition interval and width.
- Sometimes it is necessary to have one clock pulse train trail another by a fixed time.
- A circuit with the appropriate delay may be inserted to achieve the desired phase shift



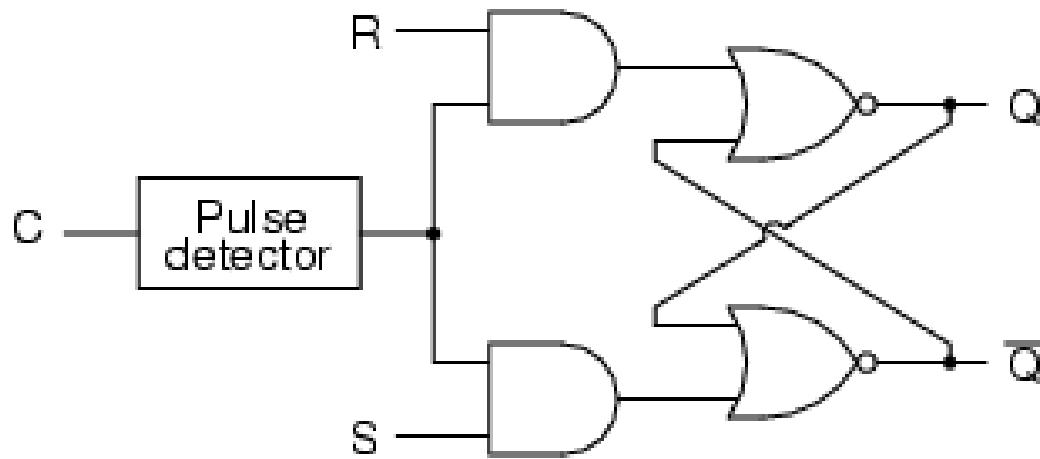
**Figure 4-28.** Clock circuit and the clock waveforms

# Edge-Triggered Flip Flops

- State Change : *Clock Pulse*
  - Rising Edge(positive-edge transition)
  - Falling Edge(negative-edge transition)
- Setup time(**20ns**)
  - Minimum time that D input must remain at constant value before the transition.
- Hold time(**5ns**)
  - Minimum time that D input must not change after the positive transition.
- Propagation delay(**max 50ns**)
  - time between the clock input and the response in Q



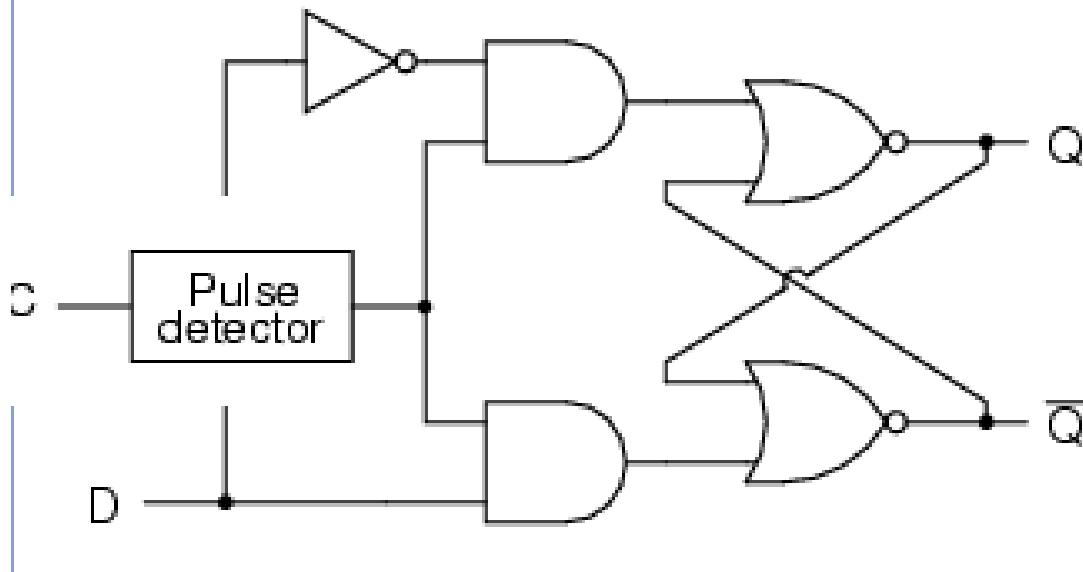
# SR Flip-flop



C	S	R	Q	$\bar{Q}$
—	0	0	latch	latch
—	0	1	0	1
—	1	0	1	0
—	1	1	0	0
x	0	0	latch	latch
x	0	1	latch	latch
x	1	0	latch	latch
x	1	1	latch	latch

Characteristic  
Table

# D Flip-flop



E	D	Q	$\bar{Q}$
0	0	latch	latch
0	1	latch	latch
1	0	0	1
1	1	1	0

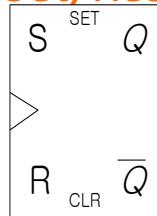
# Flip-flops

The *storage elements* employed in clocked *sequential circuit*

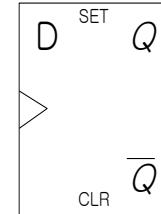
A binary cell capable of storing one bit of information

n D(*Data*) F/F

SR(*Set/Reset*) F/F

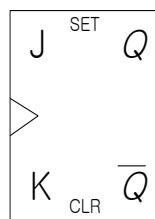


S	R	Q(t+1)	
0	0	Q(t)	no change
0	1	0	clear to 0
1	0	1	set to 1
1	1	?	Indeterminate



D	Q(t+1)	
0	0	clear to 0
1	1	set to 1

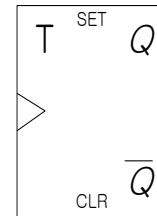
n JK(*Jack/King*) F/F



J	K	Q(t+1)	
0	0	Q(t)	no change
0	1	0	clear to 0
1	0	1	set to 1
1	1	Q(t)'	Complement

- ◆ JK F/F is a refinement of the SR F/F
- ◆ The indeterminate condition of the SR type is defined in complement

1) Disable Clock 2) Feedback output into input  
n T(*Toggle*) F/F



T	Q(t+1)	
0	Q(t)	no change
1	Q'(t)	Complement

- ◆  $T=1(J=K=1)$ ,  $T=0(J=K=0)$  JK F/F
- ◆  $Q(t+1) = Q(t) \oplus T$

# Flip-flops

## Excitation Table

- Required input combinations for a given change of state
- Present State Next State

SR F/F			
Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	0	1
1	0	1	0
1	1	X	1

JK F/F			
Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

D F/F			
Q(t)	Q(t+1)	D	
0	0	0	
0	1	1	
1	0	0	
1	1	1	

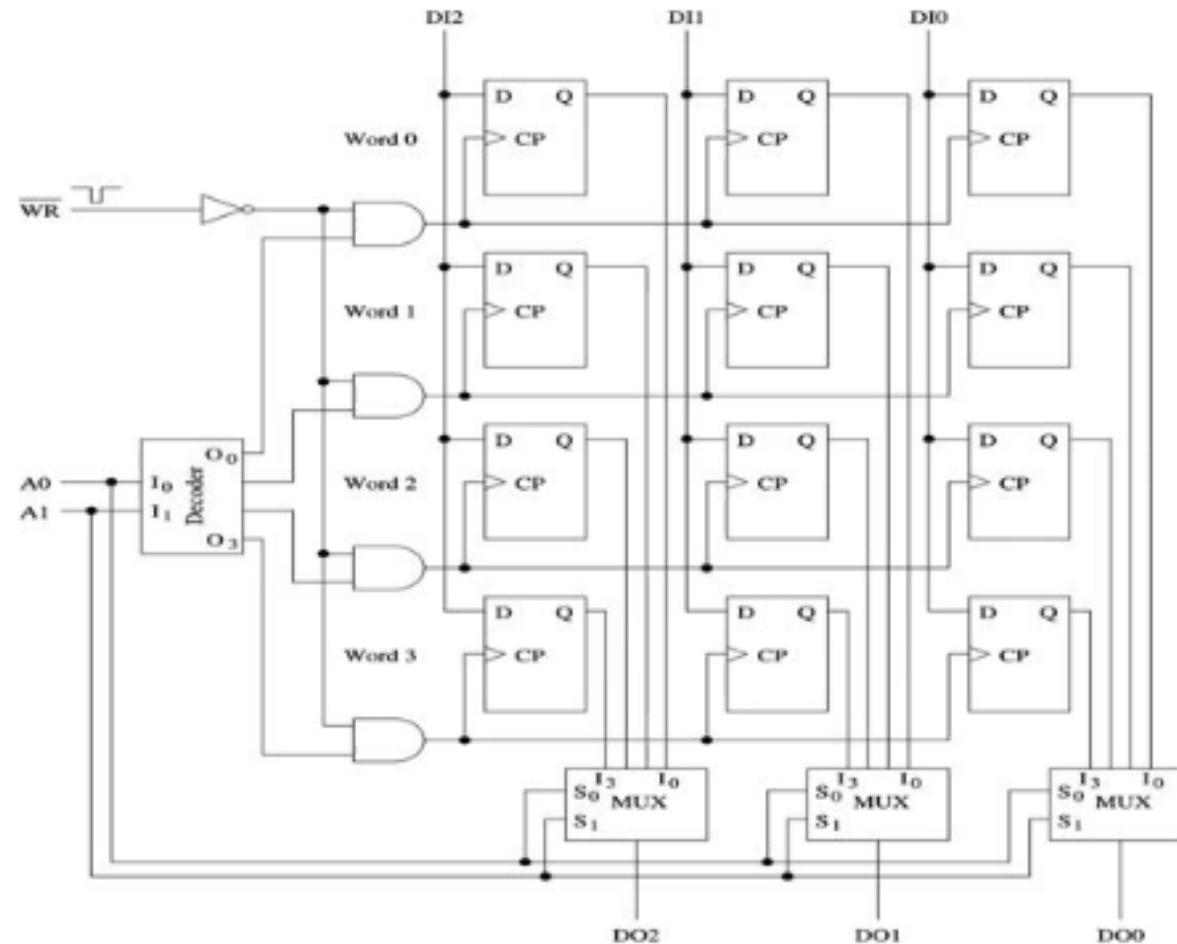
T F/F			
Q(t)	Q(t+1)	T	
0	0	0	
0	1	1	
1	0	1	
1	1	0	

Don't Care

1 : Set to 1  
0 : Complement

1 : Clear to 0  
0 : No change

# Use of Circuits: Memory



---

Thank You  
End of CF from ICS

# TEACHING COMPUTER NETWORKING

## NAVIGATION

[HOME](#)[LECTURES](#)[LESSON 1 -  
INTRODUCTION TO  
TEACHING COMPUT...  
NETWORKING](#)[LESSON 2 -  
IDENTIFYING THE  
BENEFITS OF  
NETWORKS](#)[LESSON 3 - USE OF  
COMPUTER  
COMMUNICATIONS](#)[LESSON 4 - NETWORKS](#)[LESSON 5 -  
COMMUNICATIONS  
OVER THE TELEPHONE  
NETWORK](#)[LESSON 6 -  
COMMUNICATION  
DEVICES](#)[LESSON 7 - HOME  
NETWORKS](#)[LESSON 8 - PHYSICAL  
TRANSMISSION MEDIA](#)[LESSON 9 - WIRELESS  
TRANSMISSION MEDIA](#)

## Lesson 1 - Introduction to TEACHING COMPUTER NETWORKING

# INTRODUCTION

Networks are everywhere—or so it seems. You can hardly do anything with data that does not involve a network. Like the human networks that we are all part of, computer networks let us share information and resources. In business, the reliance on networks is even more pervasive than in homes or schools. Networks help individuals and businesses alike save money, but they also help create income. Without a doubt, networking within the home will catch on over the next few years as it has in business. Soon, nearly all individuals in even moderately developed nations will have networked components throughout their homes. Those that don't will be ***netologically*** disadvantaged because they will not be able to learn or to function at the same level as those who are networked.

In this chapter, you'll begin by relating networks to situations and concepts you already know. Once you have a basic

[IP ADDRESSING](#)

understanding of what networks are and what they can do, it helps if you can actually begin working with them. In fact, it is so helpful to learn the ropes of networking through hands-on guided practice that that's what is planned for you here. You will play the role of an employee in a fictional company, and you'll have to learn on the job. The more you *become* the person, the more you will learn about the need for and operation of computer networks.

[MAKING ETHERNET CABLES](#)[LABORATORIES](#)[ACTIVITIES](#)[ATTENDANCE](#)[CONTACT ME](#)

# UNDERSTANDING NETWORKS

## Human Networks

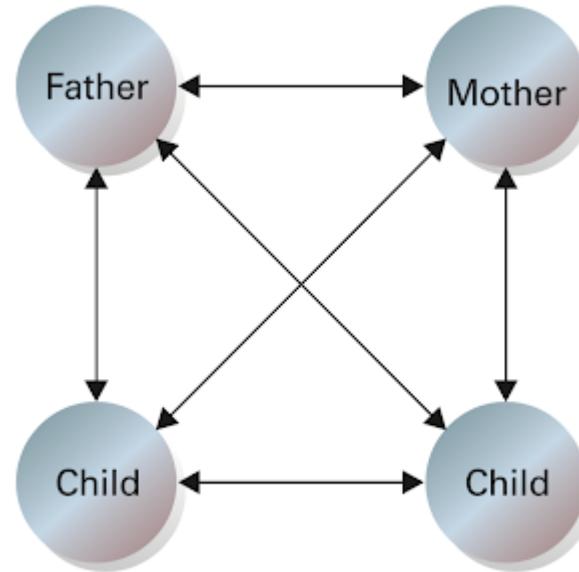
In its broadest sense, a network consists of two or more entities, or objects, sharing resources and information. Although this book is about computer networks, there are networks that don't involve computers, and those networks are everywhere. You have grown accustomed to working with them, possibly without even knowing it.

It may not matter to you that, in a basic sense, sharing (giving or getting) is a fundamental aspect of networking. You just know that you do it.

## Family Network

Most people belong to a family network in which related people share their resources and information. This sharing is bi-directional because even the youngest family members share

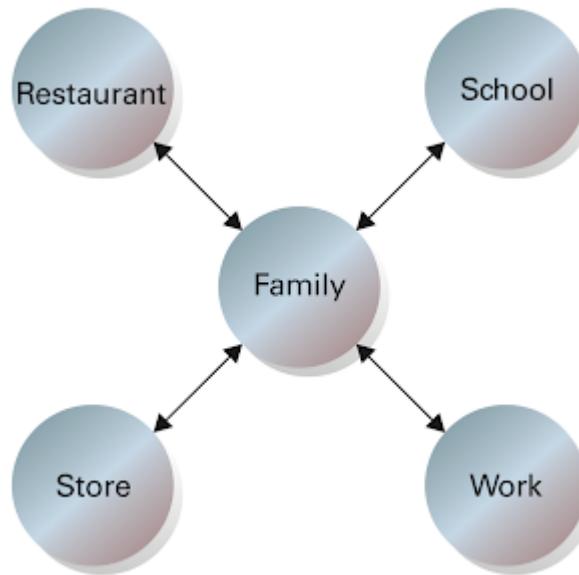
information of some sort. As the family grows, so does the network.



A network connects members of a family together.

### Peer Network

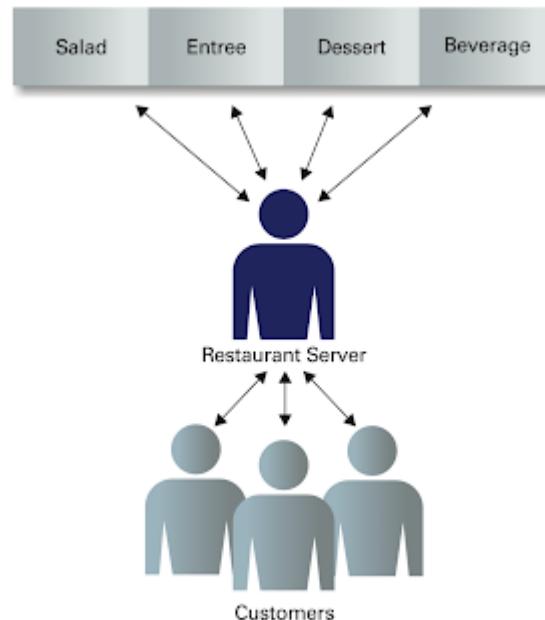
Outside the family, there is a community that offers a wider array of resources than the typical family can provide. Naturally, it makes sense to connect the family to this community to take advantage of the wealth of resources available around town. This type of information/resource sharing can be as simple as loaning a hammer to a neighbor, car-pooling with work associates, or helping a friend with his or her homework. All of these activities involve sharing, or trading, resources. This kind of network is represented by a two-way relationship, a give and take among equals or peers.



The family network connects with the greater community.

### **Restaurant Network: The Client and the Server**

So, in any type of human network, there's a lot of giving and taking. You're already more accustomed to the client/server perspective in networking than you realize. For instance, when you go to dinner at a restaurant, you become a customer, or client, enjoying the food and drink prepared and served to you by the restaurant. On the other hand, the waiter works as a server, controlling and providing his customers with access to resources in the form of placing orders for and delivering food items. The server knows that requests will be made of him (access is sought when an order is placed) and that he will service those making the requests (access is granted when the order is delivered).



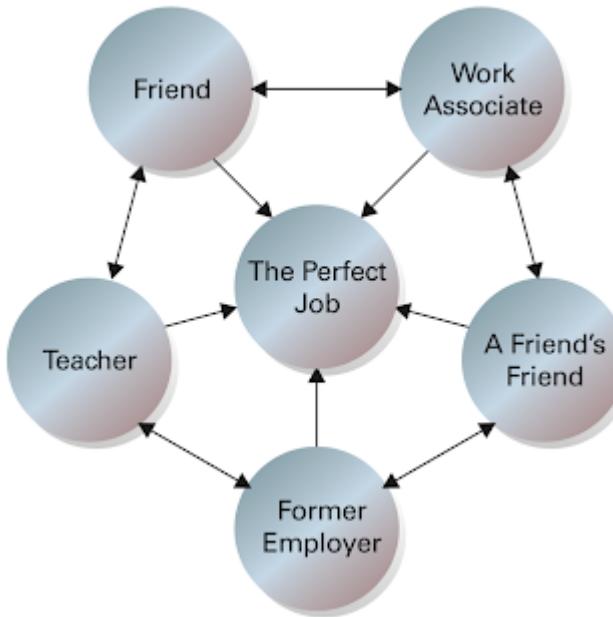
In a dining situation, it is easy to know whether you are supposed to be serving or being served.

### Contact Network

Anyone who has looked for a job knows that one of the best ways to find a job is to network. That is, create a list of friends and associates who will help you find the perfect job. The more people you meet and get to know, the better your chances of obtaining work. As you develop and nurture your career, this contact network will serve you best because your role in it will change as you gain more experience. Soon, you may be able to help the people who helped you. And as your personal and professional networks grow, so do your opportunities.

These examples of human networks should help you under-

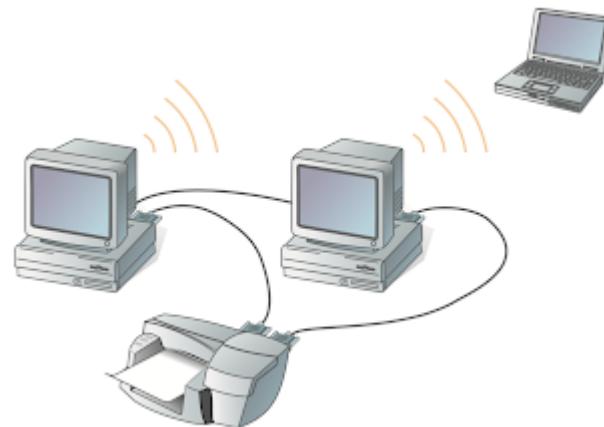
stand that networking is common between people and is not just an activity restricted to computers. However, this book will focus on computer networks—connecting computers and having them communicate with each other.



The more people in your network, the better your chances of finding that perfect job.

## Computer Networks

A **computer network** consists of two or more computing devices that are connected in order to share the components of your network (its resources) and the information you store there, as shown in Figure 1.1. The most basic computer network (which consists of just two connected computers) can expand and become more usable when additional computers join and add their resources to those being shared.



A computer network can be as simple as two or more computers communicating.

The first computer, yours, is commonly referred to as your **local computer**. It is more likely to be used as a location where you do work, a **workstation**, than as a storage or controlling location, a server. As more and more computers are connected to a network and share their resources, the net-work becomes a more powerful tool, because employees using a network with more information and more capability are able to accomplish more through those added computers or additional resources.

The real power of networking computers becomes apparent if you envi- sion your own network growing and then connecting it with other distinct networks, enabling communication and resource sharing across both net- works. That is, one network can be connected to another network and be- come a more powerful tool because of the greater resources. For example, you could connect the network you and your

classmates develop for this course to similarly constructed networks from other introductory networking classes if you wanted them to share your information and networked resources. Those classes could be within your own school, or they could be anywhere in the world. Wherever that newly joined network is, the communication and resource sharing activities in that new network could then be shared with anyone connected to your network. All you have to do is join that new network's community or allow its members to join yours.

In addition, a company's cost of doing business can be reduced as a result of sharing data (defined as a piece or pieces of information) and resources. Instead of having individual copies of the data at several locations around the company, and needing to keep all of them similarly updated, a company using a network can have just one shared copy of that data and share it, needing to keep only that one set of data updated.

Furthermore, sharing networked resources (like printers) means that more people can use a particular resource and a wider variety of resources (like different printers) can be used by each network user. Any time a company can do more with less, or buy fewer items to do the same job, its total costs are reduced, and it is able to make more money per dollar spent.

## Network Plan

Networking computers first and tracking the connections later can quickly become confusing and unmanageable as you try to find which computer communicates with and shares resources with which other computers. In your human network, do you

share everything with your friends? In your family network, would you want your parents or guardians to know your every thought? You have your information-sharing plan in your head, and it is important to keep track of it so you don't make a mistake and share something where it was not intended.

Similar concerns must be considered while designing a computer network. Before you even connect your first computers together, you should have a plan. A network plan, therefore, is a formally created product that shows all the network's components and the planned connections between them. Such a plan is also used to manage the various types of information. Your plan should show what types of information are stored where, and who is allowed to use each type.

### **Information Management**

Your network plan should help you manage the information gathered, stored, and shared between your users. If you were given an empty three-drawer filing cabinet and told to use it to organize your company's information, you would have an excellent (although manual) example of a filing system that needs a plan. Having an overall guide that tells you who will be allowed access to the three drawers will help determine what you store in each one. Once you have that part of the plan, you could put the least-used information in the bottom drawer, the more-used in the middle drawer, and the most-used in the top drawer so that it is easier for your users to access their information. Knowing who needs to know what, and its corollary — who does not need to know what—lets you determine whether to lock a particular drawer, too.

Even when we discuss implementing a three-drawer manual filing system, the importance of having a network plan ahead of time becomes evident. If you put the limited-access material in a drawer open to all employees, how do you keep it secure? Additional security measures (like adding a lock to a drawer, or moving the secure information somewhere else) may be required later.

A networking plan could tell you that as specific types of sensitive data (like medical, personal, or payroll information) are gathered or grouped, they should be stored higher in the hierarchical structure (ranked from most sensitive to least sensitive), and this can save you time in the end. That plan should specify that the access requirements are stricter for sensitive data and reduce the number of people able to use specific types of information.

The distribution side of the networking plan, as opposed to the accumulation side of the plan discussed above, should spell out that the more an individual has access to the data in storage, the less they should be able to share groups of information entrusted to them. For example, you may not mind sharing your first name, but you would probably object to an instructor openly distributing all information in your school records to anyone requesting it.

### **Information's Importance**

If you think about the manual filing system we discussed using

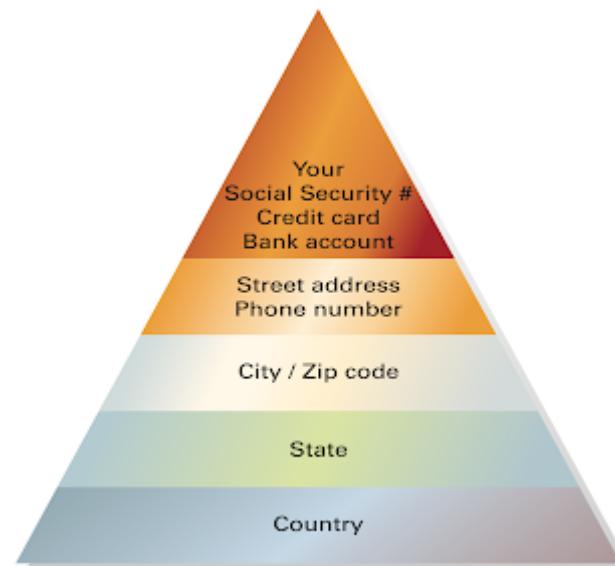
a filing cabinet, an important computing concept is easy to recognize. Some information is more important or more sensitive than the rest. It is usually obvious in real filing cabinet systems, because the top drawer is usually where the most sensitive information is stored, and it is locked. Few people in an organization have access to that information. For example, credit card or Social Security numbers are information that should be given the highest level of security—access to that information is given only to a limited number of people in a company. On the other hand, some information, such as Webpages, newsletters, and product information, is created for everyone to see, even outside a company. Figure 1.2 shows how this kind of information is organized into a hierarchy of information, where the most detailed information is found at the top and the more general, less secure information is located at the bottom. How much information would you be willing to provide about yourself to a perfect stranger? Country of birth? Sure. State of residence? Why not? But you might have second thoughts about advertising your street address or phone number to a stranger.

The collection and proper manipulation of many seemingly unimportant pieces of information, and the effective tracking of them, makes information management on networks so important, just as when you are maintaining a manual filing system. A single piece of information in a data field, such as your first name, can seem unimportant. However, by combining your first name with other pieces of related information, like your last name, address, age, gender, and phone number (stored in other data fields), the pieces can be

put together to create a data record, which can accurately describe something (or someone) that is important—like you.

Finally, combining similar records (such as records describing all your classmates) creates a file that, because it contains sensitive information from more than one source, is more sensitive than a single record.

Information sharing, therefore, has serious security issues to be considered, and network access to data must be evaluated carefully so that only those who need it can access it.



The hierarchy of information

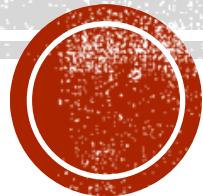
The more specific the information becomes, the more restricted it should be.

What kind of data would you be willing to give to a stranger?



# **INTRODUCTION TO COMPUTER SYSTEMS**

## **COMPUTER NETWORKS**



## **We: The Malabe campus team**

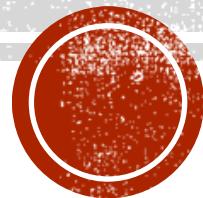
**Dr. Anuradha Jayakody (PhD, Curtin - Australia):** [anuradha.j@sliit.lk](mailto:anuradha.j@sliit.lk)

**Dr. Janaka Wijekoon (PhD, Keio Univ. - Japan):** [janaka.w@sliit.lk](mailto:janaka.w@sliit.lk)

**Ms. Shashika Lokuliyana (MSc (IT) - SLIIT):** [nuwan.ku@sliit.lk](mailto:nuwan.ku@sliit.lk)

**Ms. Hansika Mahaadhikara (MSc (IT) - SLIIT):** [hansika.m@sliit.lk](mailto:hansika.m@sliit.lk)

**Mr. Kavinga Yapa Abeywardena (MSc - UK):** [kavinga.y@sliit.lk](mailto:kavinga.y@sliit.lk)



# COURSE OUTLINE

- Week 9: Living in a Network Centric World
- Week 10: Communicating over the Network
- Week 11: ISO – OSI Reference Model
- Week 12: Addressing in a Network
- Week 13: Design a Network



## **LECTURE 07**

# **LIVING IN A NETWORK CENTRIC WORLD**



# LECTURE OUTLINE

- Networking
- Computer Networks
- Classification of Computer Networks
  - Technologies
  - Topologies
  - Devices
- Internet
- Private vs Public Networks
- Intranets vs Extranets



# NETWORKING TODAY

- Network has no boundary and supports the way we:
  - ✓ Communicate
  - ✓ Work
  - ✓ Share
  - ✓ Learn
  - ✓ Play



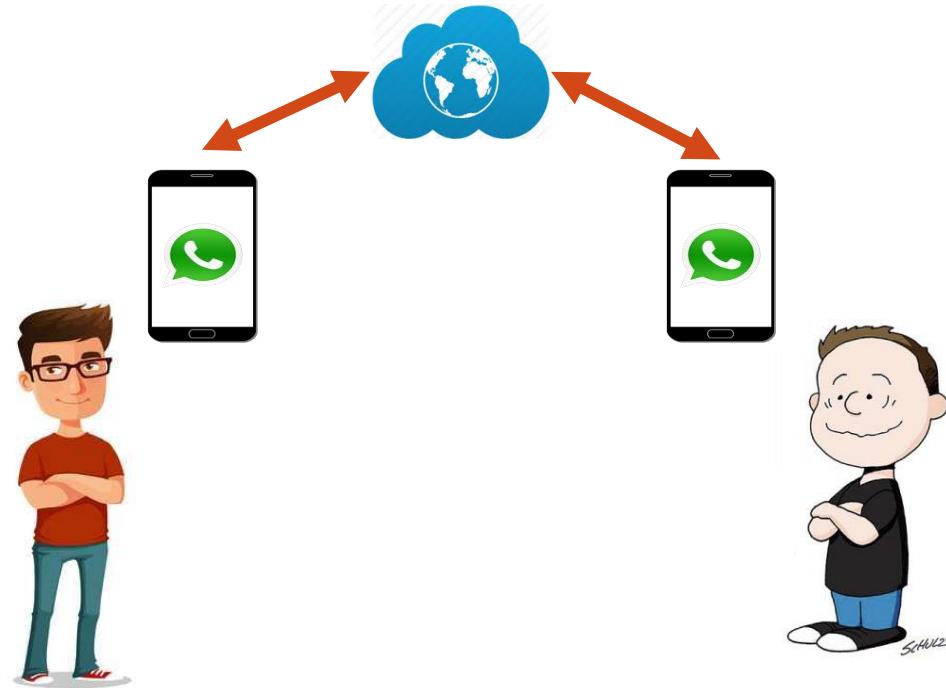
# GENERAL COMMUNICATION

- ✓ Rules/agreements are established
- ✓ Information may need to be repeated
- ✓ Communication modes may impact the effectiveness of getting the message across



# DATA COMMUNICATION NETWORKS NEEDS

- Devices
  - To communicate with one another
- Medium
  - Connects devices together
- Messages
  - Information over media
- Rules
  - Govern how messages flow across networks



# COMPUTER NETWORKS

- ✓ A collection of computers and other associated devices  
(printers, fax machines, or any devices capable of sending/receiving data)
- ✓ interconnected by intermediate networking devices  
(hubs, routers and switches)
- ✓ using a transmission media  
(copper, fiber, free space)



# TODAYS NETWORKS

## Converged networks

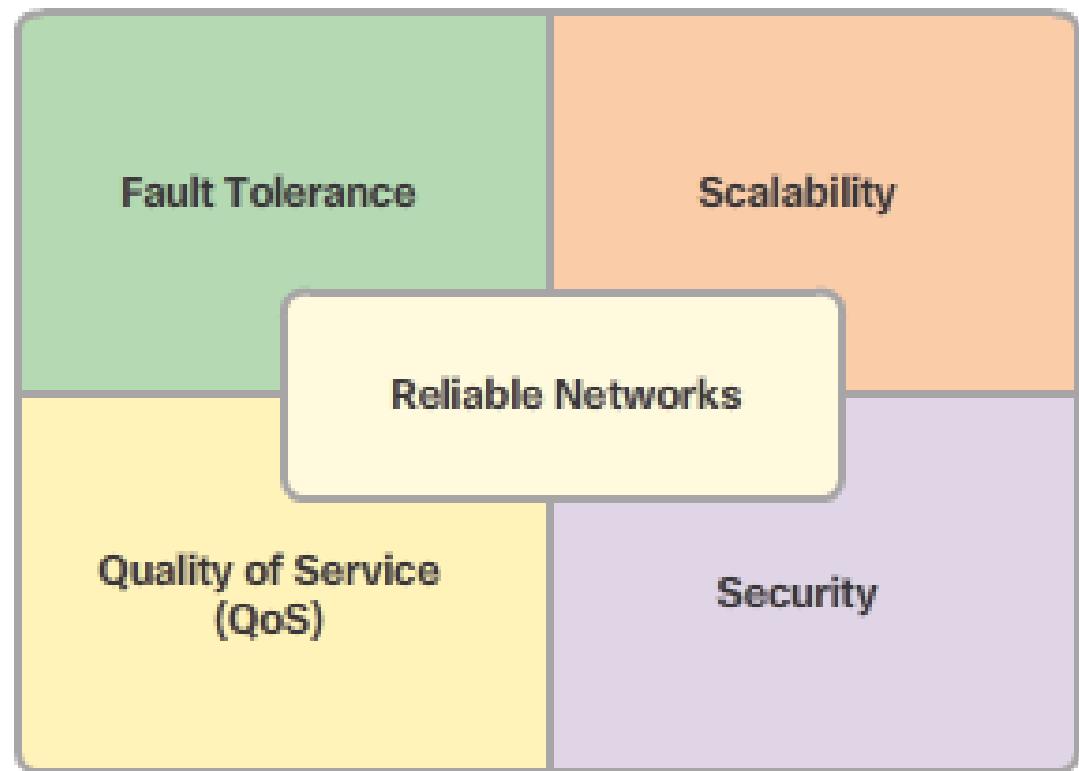
- Carry data, voice, video & images over the same network



# RELIABLE NETWORKS

- Four Basic Characteristics of Network Architecture

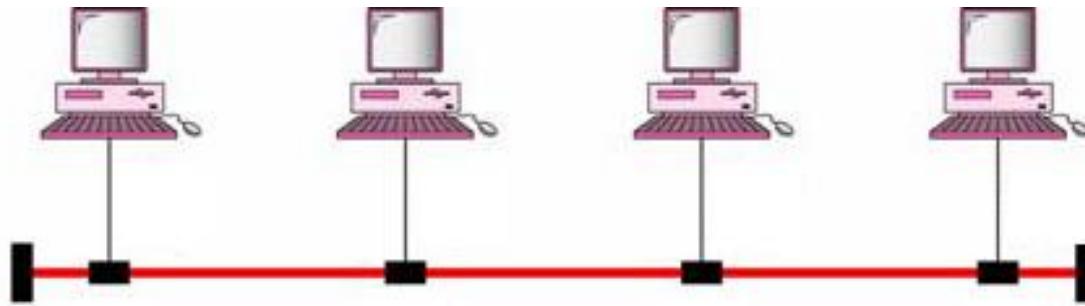
- ✓ Fault Tolerance
- ✓ Scalability
- ✓ Quality of Service (QoS)
- ✓ Security



# CONNECTING DEVICES IN A NETWORK



- **Point-to-Point:** Provides a dedicated link between the two devices.

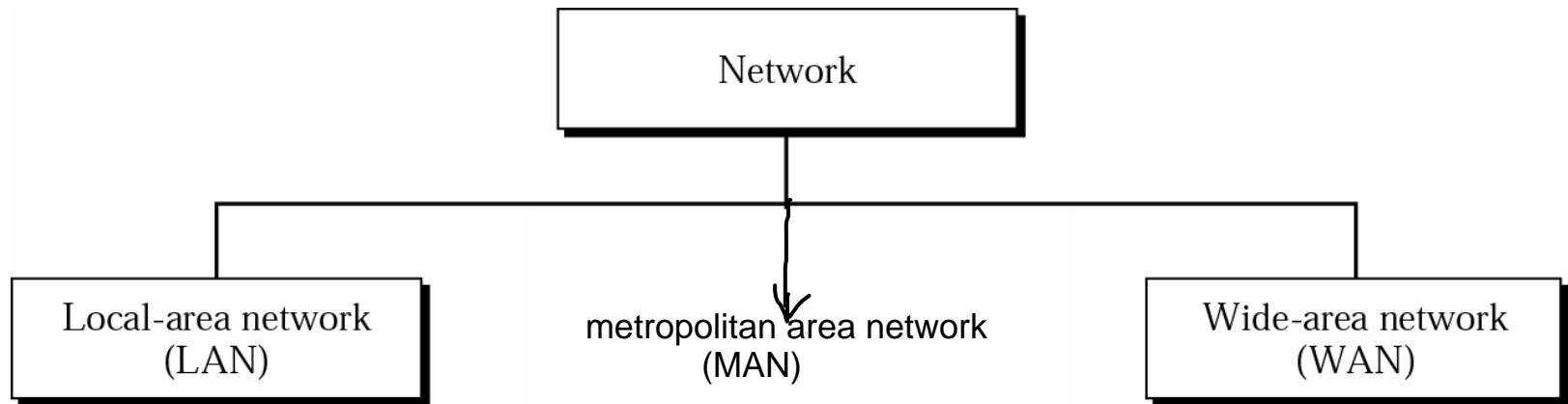


- **Multipoint:** More than two devices (i.e. several) share a single link.



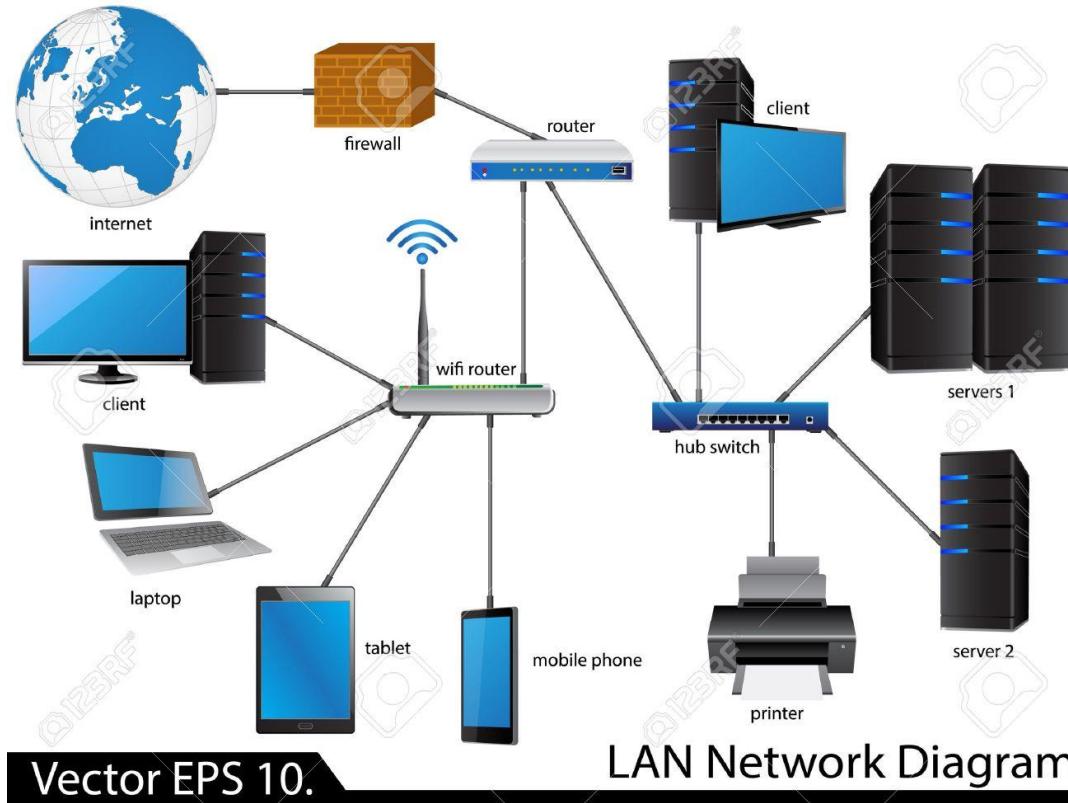
# CLASSIFICATION OF COMPUTER NETWORKS

- Based on the covered geographical area, computer networks can be divided into categories



# LOCAL AREA NETWORK (LAN)

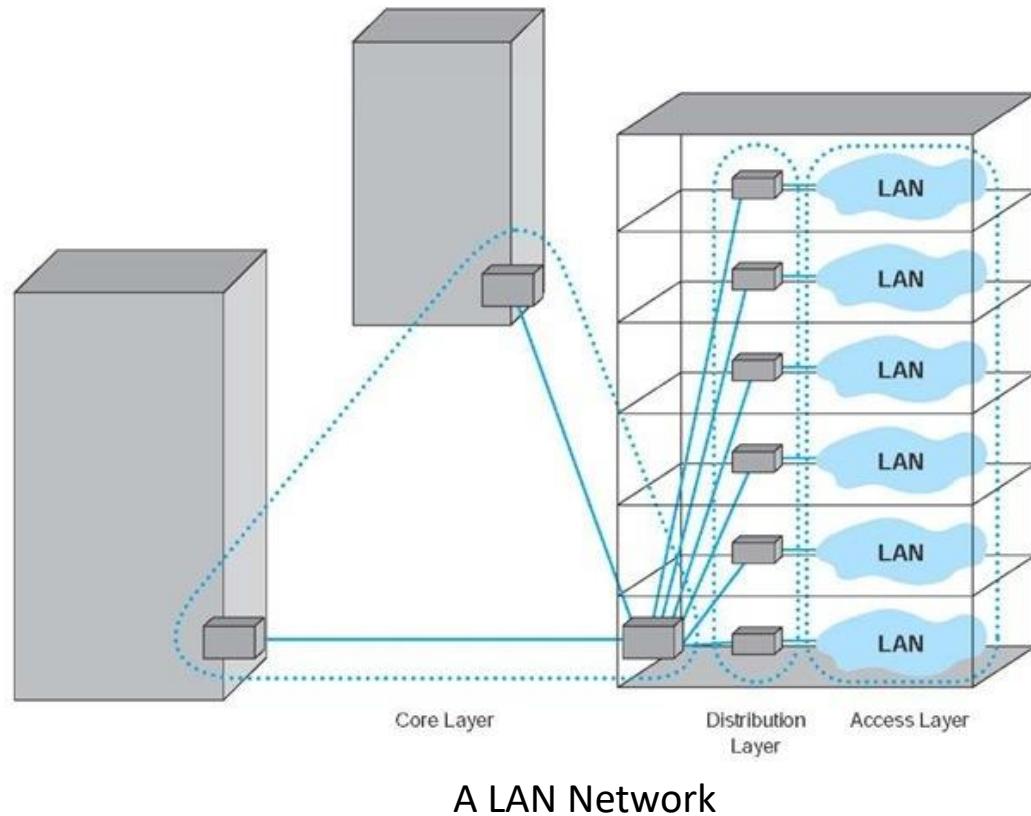
- A LAN connects network devices over a relatively small geographical area.
- A networked office building, school, campus or home usually contains a single LAN.



# FEATURES OF A LAN

- Smaller scope (Building or Small Campus)
- Usually owned by same organization
- Data rates are much higher
- Usually a single broadcast systems
- Easy to manage

**Any LAN environment  
you are familiar with?**



# NETWORK DEVICES USED TO BUILD A LAN

- **Switches**
- **Wi-Fi Routers**
- Repeaters
- Hubs
- Bridges



Switch



Repeater



Hub

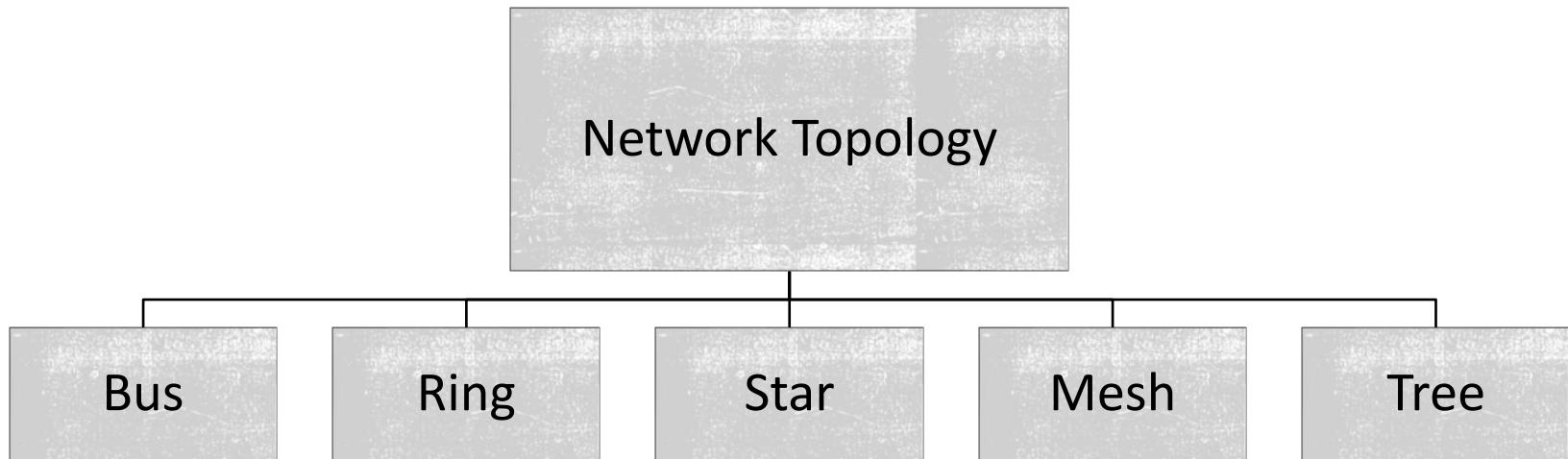
# LAN CONNECTIVITY METHODS

- There are several standards (or specifications) to setup a LAN,
  - ✓ **Ethernet**
  - ✓ **Fast Ethernet**
  - ✓ **Gigabit Ethernet**
  - ✓ Token Ring
  - ✓ Token Bus
- Each standard specifies a certain topology and a media access technique for the network



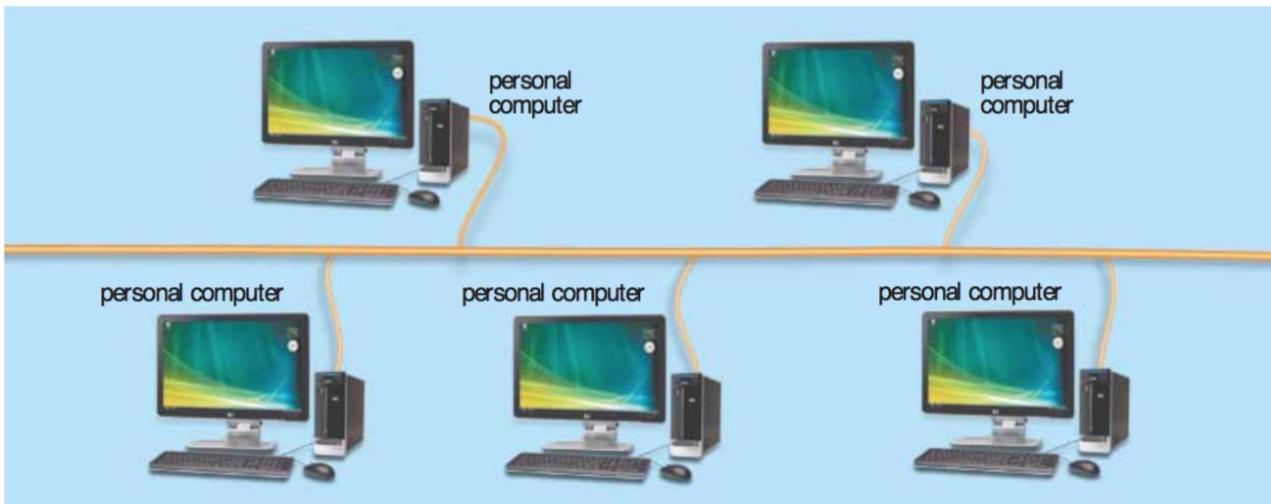
# TOPOLOGIES

- Topology (or network topology) refers to the **physical arrangement** of the **computers** in the network



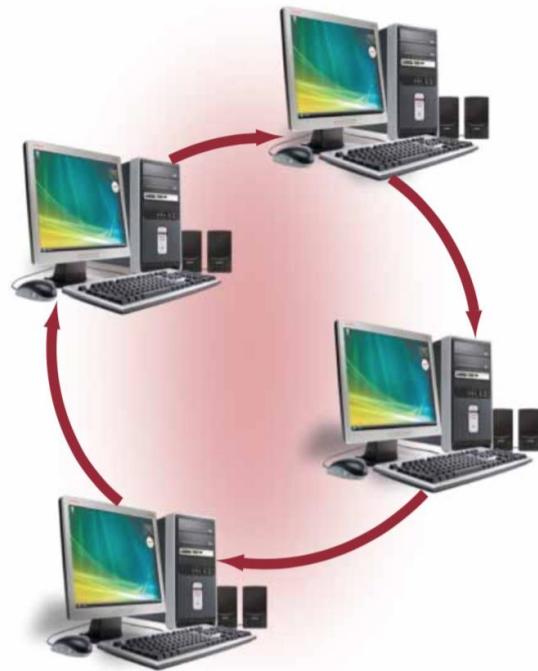
# BUS TOPOLOGY

- ✓ A common media connecting all the computers



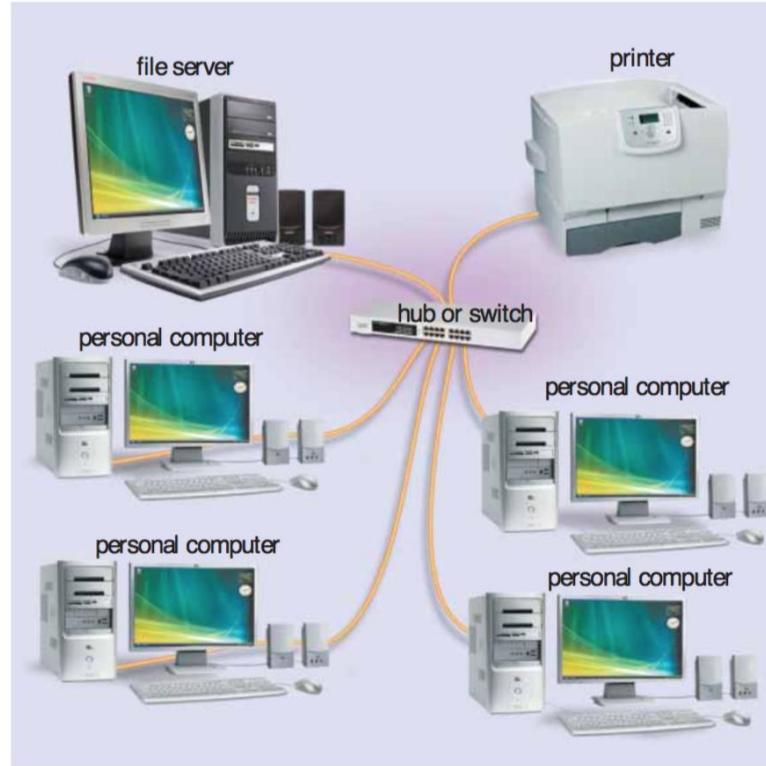
# RING TOPOLOGY

- ✓ Also shares a common media, but computers are arranged in a ring



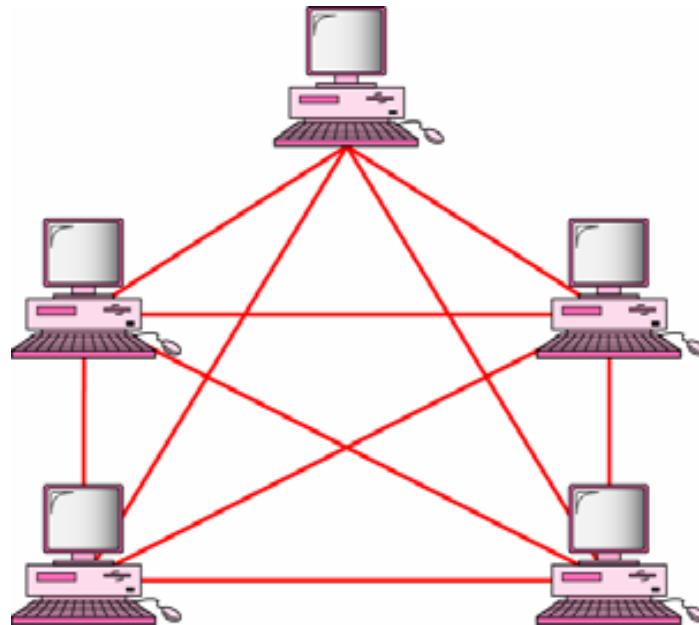
# STAR TOPOLOGY

- ✓ Computers are connected to a central device (typically a hub) and forms a star like arrangement.



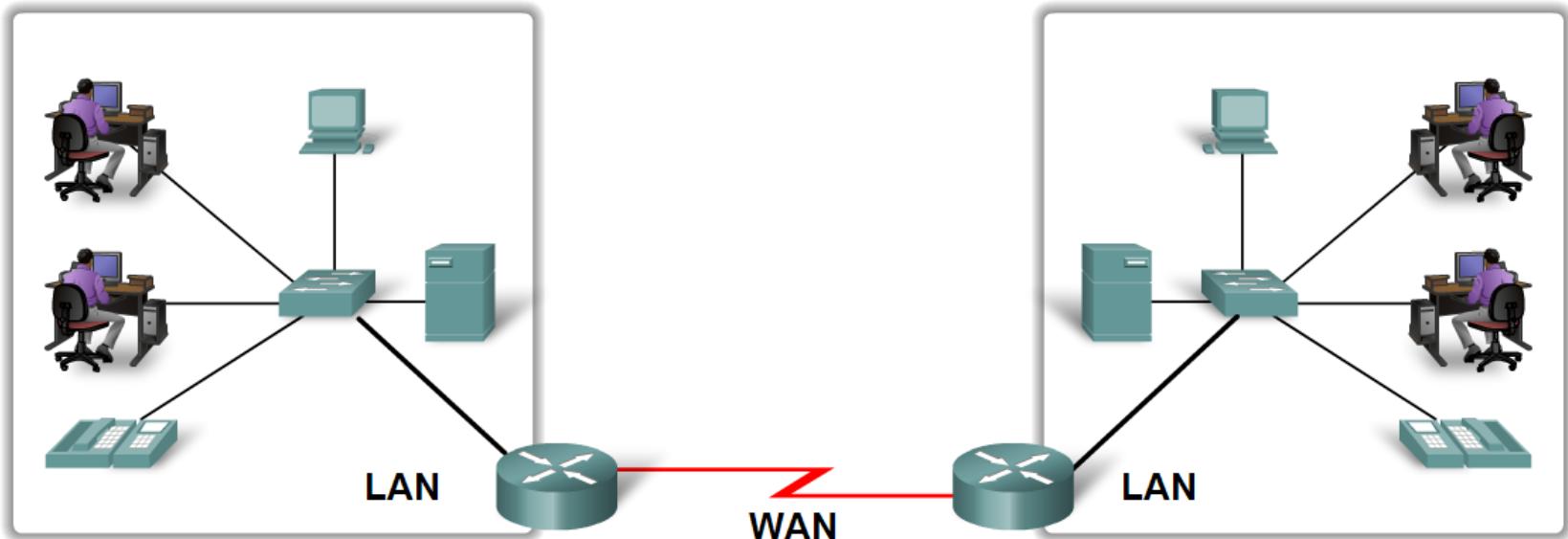
# MESH TOPOLOGY

- ✓ Each computer has a dedicated point-to-point connection to every other computer in the network



# WIDE AREA NETWORK (WAN)

- A WAN spans across a large geographical area.
- It is a geographically-dispersed collection of LANs.
- Internet is the largest WAN spanning the entire globe.



# NETWORK DEVICES USED TO BUILD A WAN

Name	Device	Icon
Routers		
WAN switches ( L3 switches) <a href="https://www.conceptdraw.com/How-To-Guide/picture/Cisco-switches-and-hubs-Design-elements.png">https://www.conceptdraw.com/How-To-Guide/picture/Cisco-switches-and-hubs-Design-elements.png</a>		 MGX 8000 Multiservice switch
ADSL Routers		 IP DSL Switch
Servers (e.g. Web Servers, Proxy Servers)		 File server
Multiplexers		 Cisco 15800

# WAN TECHNOLOGIES

Over the years there have been many WAN technologies

- ✓ Dial up connections
- ✓ Leased line
- ✓ Frame relay
- ✓ ISDN
- ✓ ADSL
- ✓ VPN



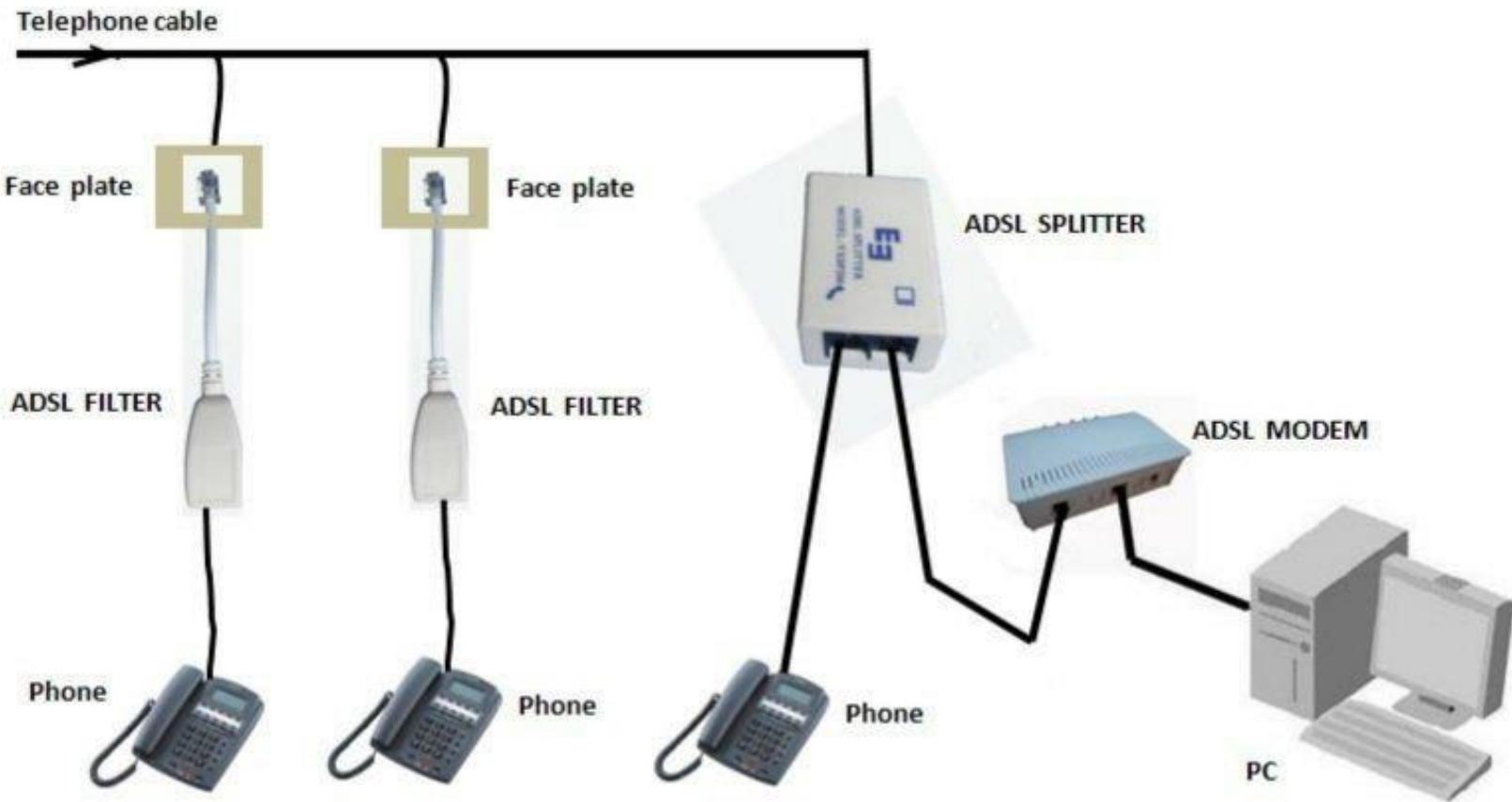
# DIAL UP CONNECTIONS



- If the computers are a long distance away from the WAN, PSTN network can be used to connect them to the WAN.

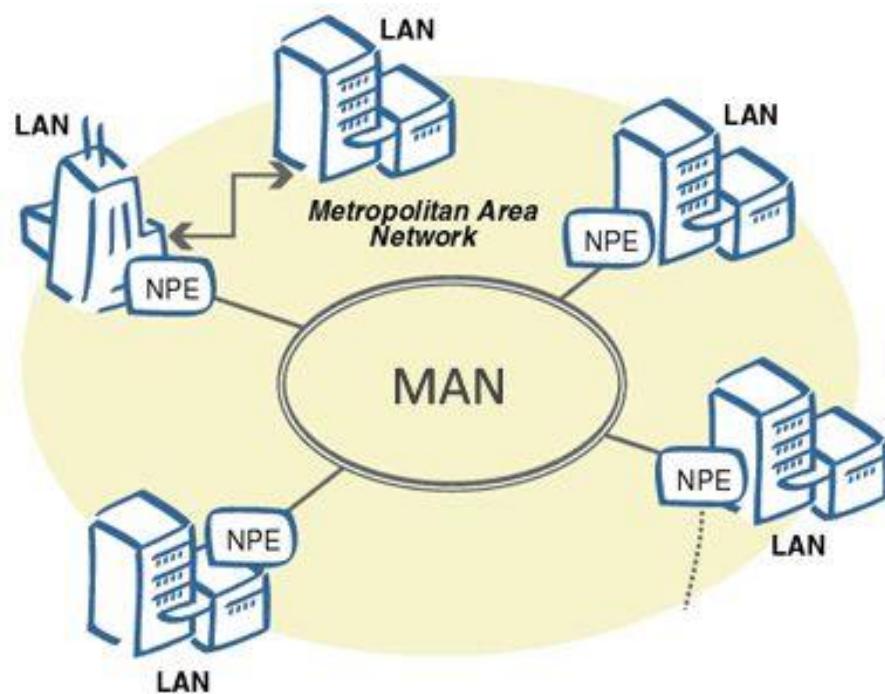


# ASYMMETRIC DIGITAL SUBSCRIBER LINE (ADSL)



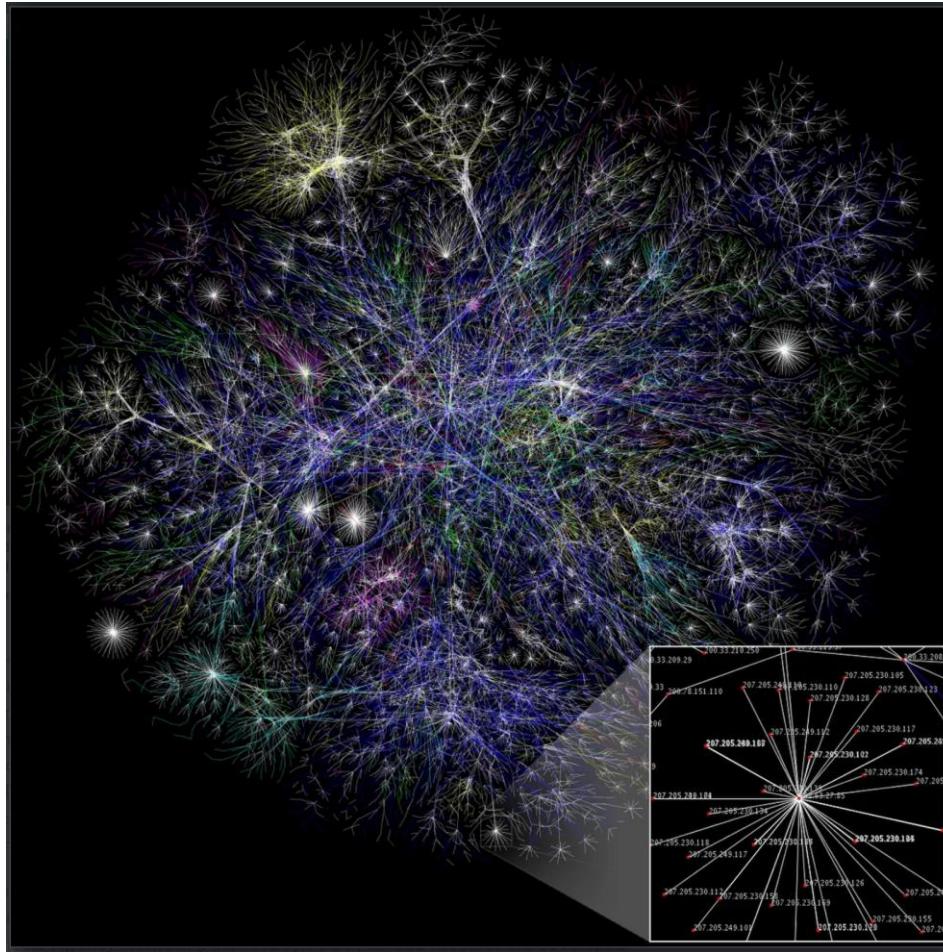
# METROPOLITAN AREA NETWORK (MAN)

- A network spanning a geographical area larger than a LAN but smaller than a WAN, such as a city.
- A MAN is typically owned and operated by a single entity such as a government body or large corporation.

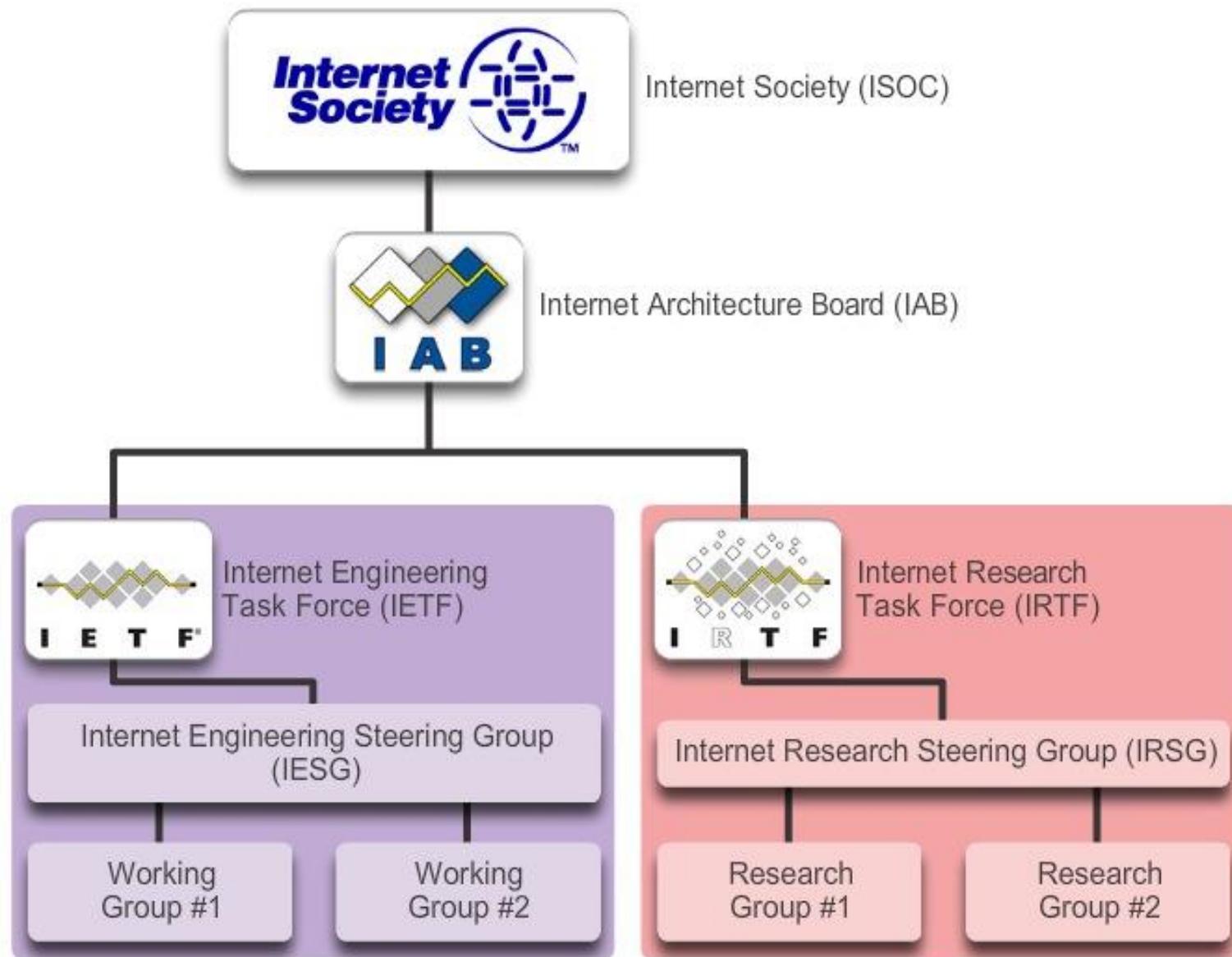


# INTERNET

- The **internet** is defined as a global mesh of interconnected networks



# WHO MANAGES INTERNET



# INTERNET SERVICES

Internet offers various services which includes,

- ✓ Web Services
- ✓ Email Services
- ✓ News Services
- ✓ Online Gaming
- ✓ Social Networking
- ✓ Knowledge Services (e.g. Wikipedia)
- ✓ Audio/Video Streaming



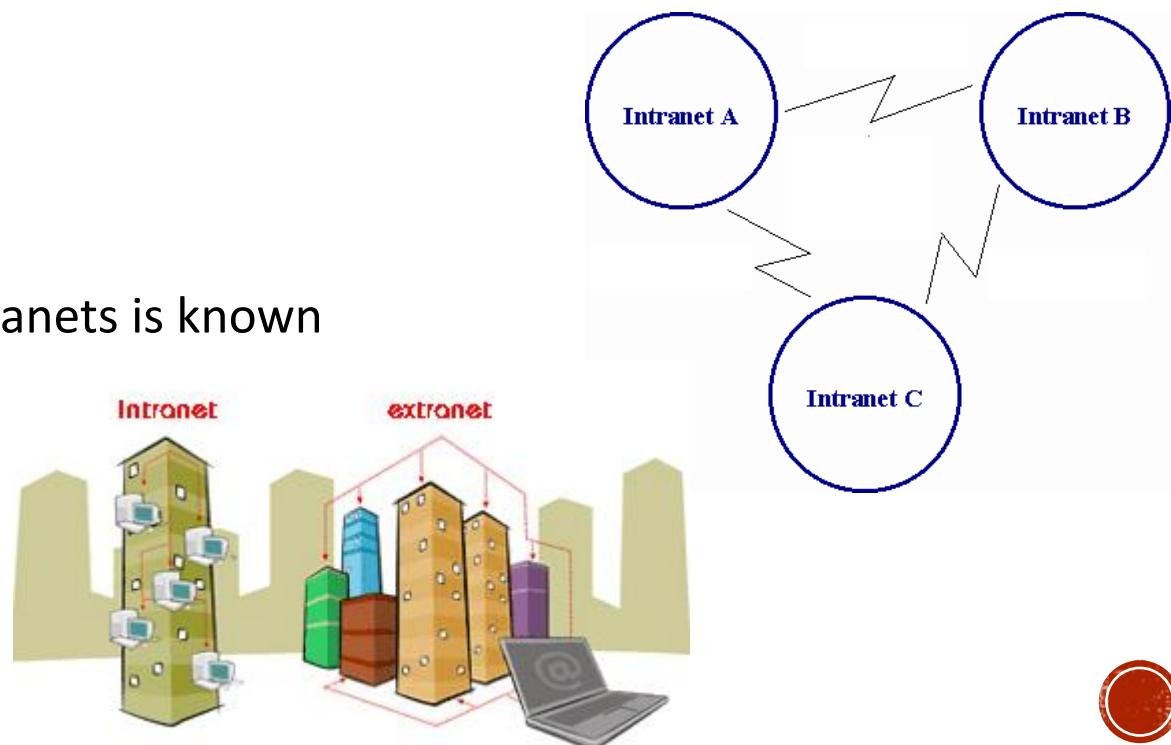
# INTRANET VS. EXTRANET

## Intranet

- A private network maintained by a company or particular organization.
- It is used to exchange internal information and it is restricted to the general public.

## Extranet

- A set of interconnected Intranets is known as an extranet.





*That's all Folks!*

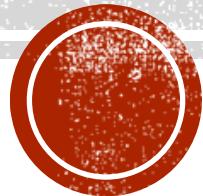


# Questions



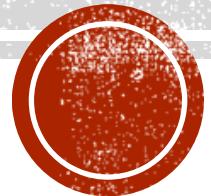
# **INTRODUCTION TO COMPUTER SYSTEMS**

## **COMPUTER NETWORKS**



# **LECTURE 09**

## **ISO – OSI REFERENCE MODEL**



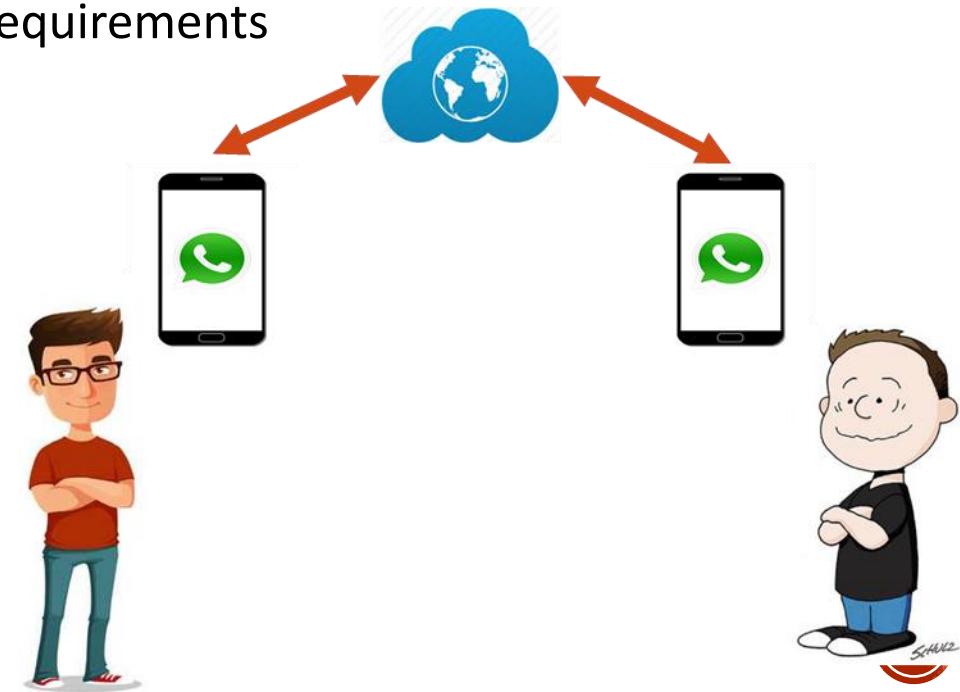
# LECTURE OUTLINE

- Rules of Communication
- Layered Architecture
- IOS – OSI Model
- TCP / IP Model



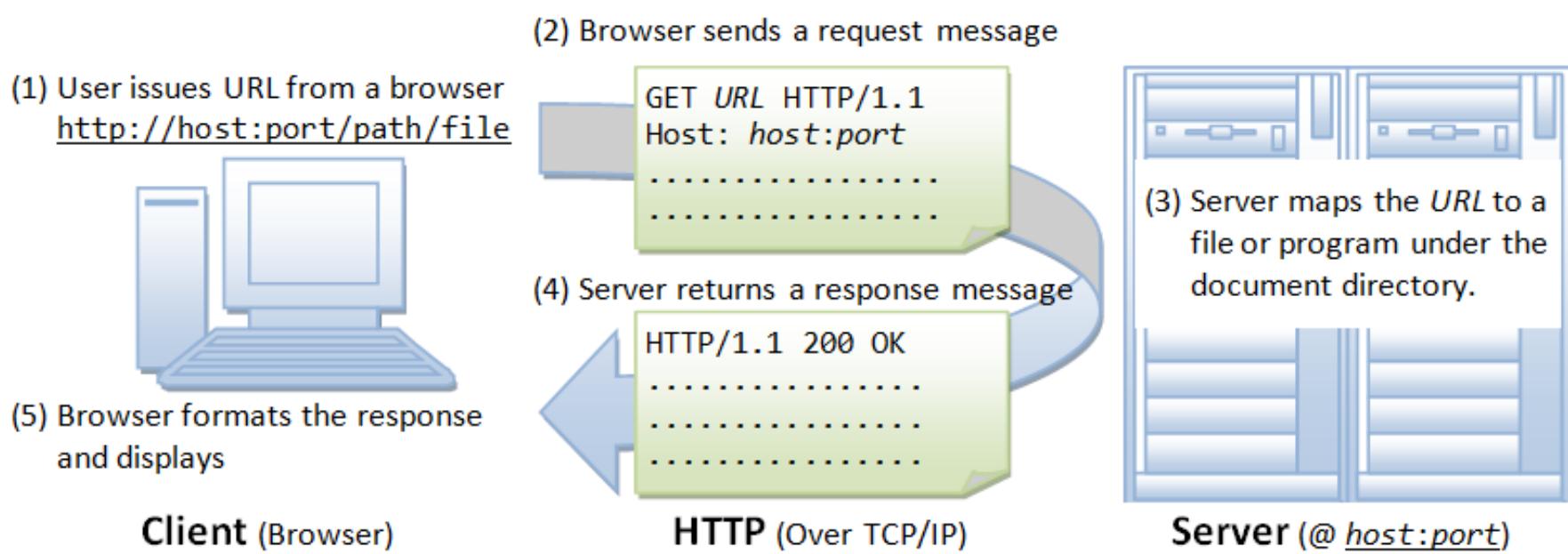
# RULES OF COMMUNICATION

- ✓ Identified sender and receiver
- ✓ Communication language and grammar
- ✓ Speed and timing of delivery
- ✓ Confirmation or acknowledgment requirements



# NETWORK PROTOCOLS

- Networking protocols define a **common format** and a **set of rules** for exchanging messages between devices

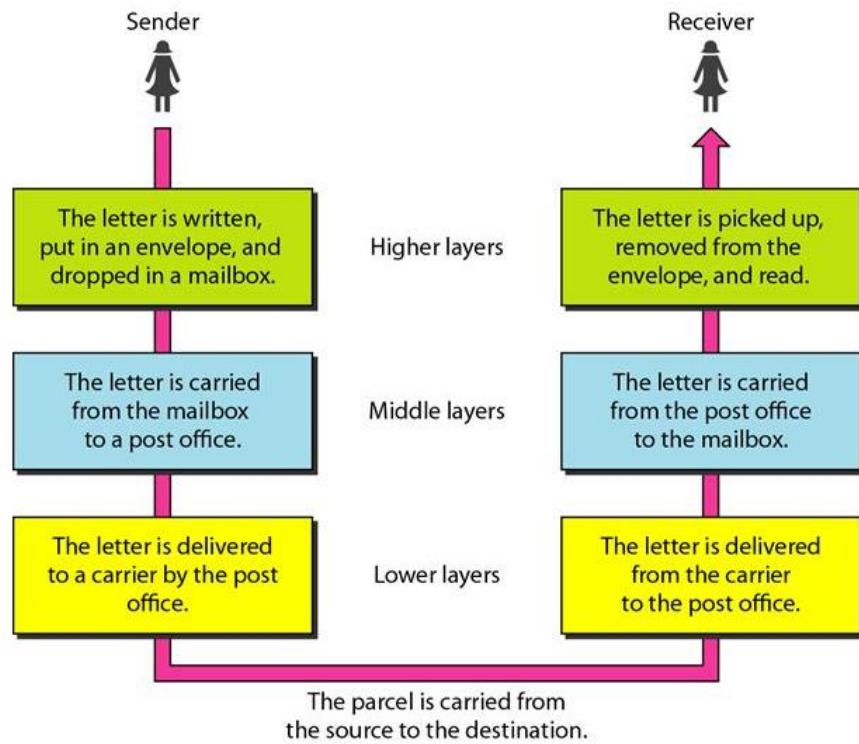


# REFERENCE MODEL FOR NETWORK COMMUNICATION

- Defines how applications can communicate over a network: the full process
- Guide vendors and developers, so the digital communication products and software programs they create will interoperate
- A layered reference model divides the full process into specific related groups of actions at each layer

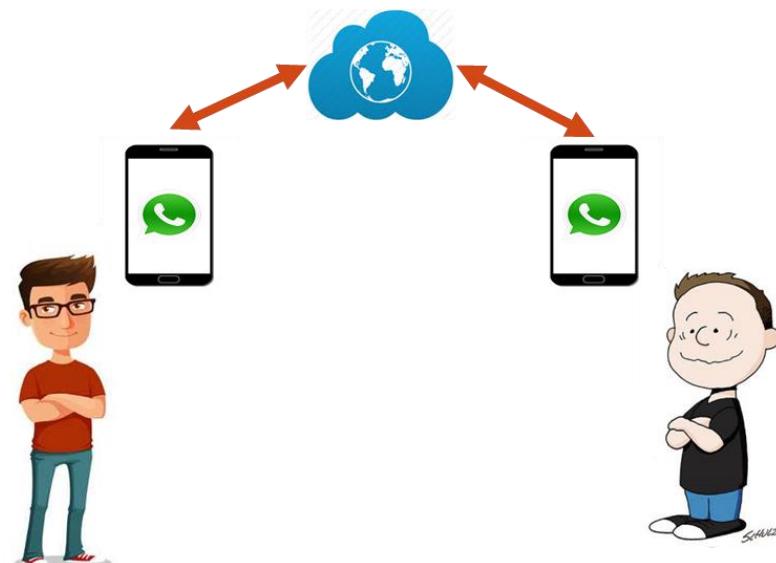


# LAYERED TASKS



## OLD SCHOOL METHOD

## THE METHOD YOU ALL KNOW



# BENEFITS OF USING A LAYERED MODEL

- Provides a common language
- Create a competition between vendors
- Changes in one layer do not affect other layers
- Assists in protocol design



# ISO – OSI REFERENCE MODEL

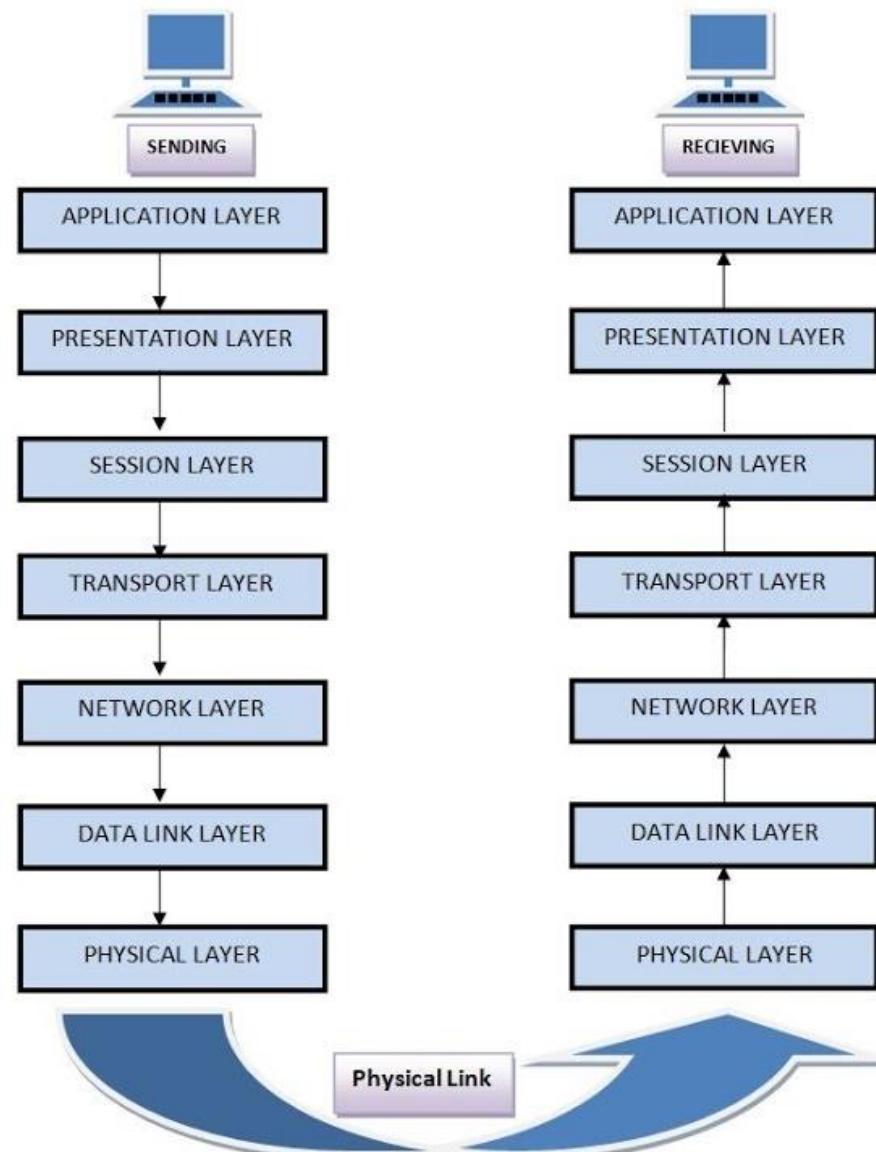
- ISO – International Organization for Standardization
  - OSI – Open Systems Interconnection
  - The International Organization for Standardization (ISO)
    - is a multinational body dedicated to worldwide agreement on international standards
  - An ISO standard that covers all aspects of network communications
    - is the Open Systems Interconnection (OSI) model
  - It was first introduced in the late 1970s
- ✓ Note: ISO is the organization, OSI is the model



# ISO – OSI REFERENCE CONT.

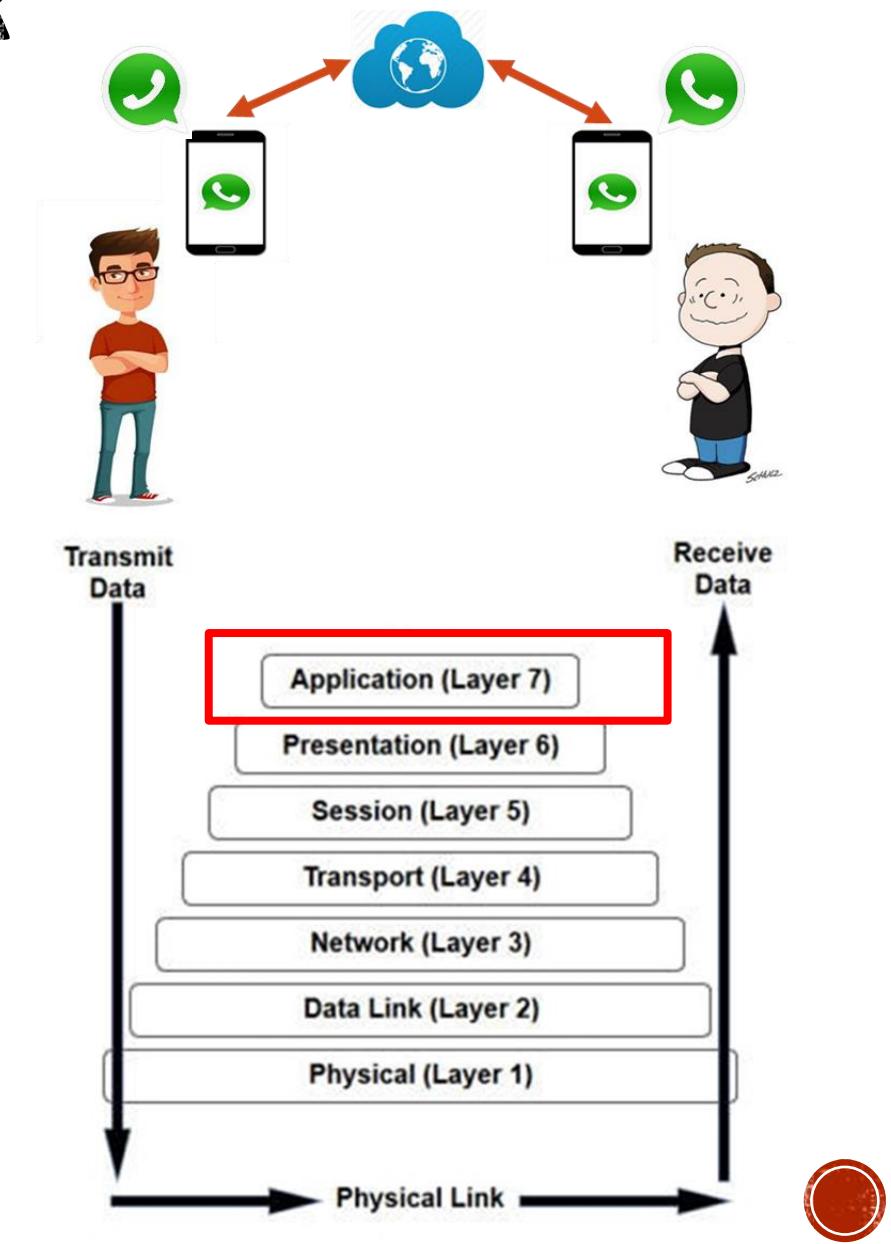


# ISO – OSI REFERENCE CONT.



# APPLICATION LAYER

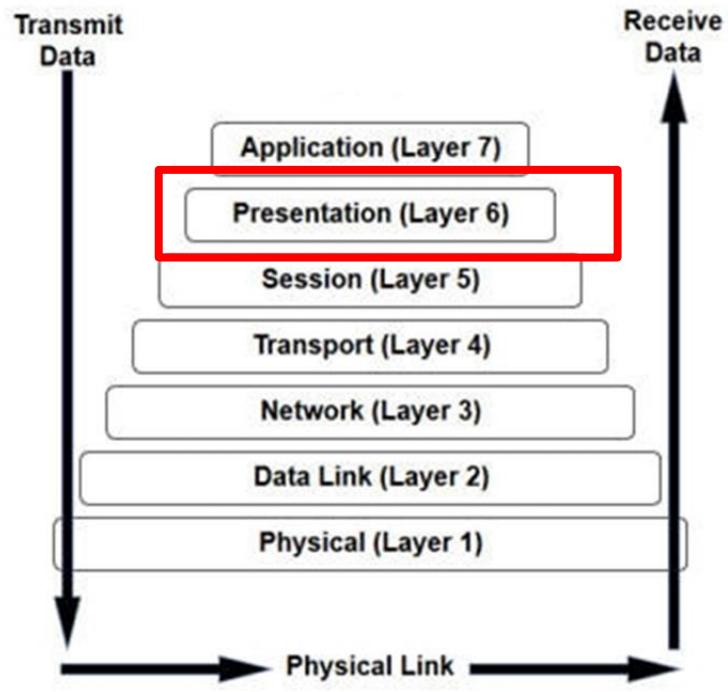
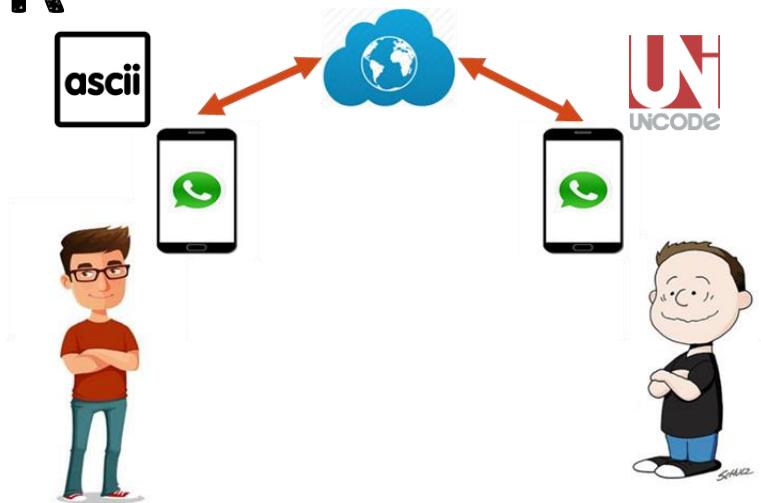
- Application layer is our (i.e. users) interface to the network
- Application layer provides many useful applications to facilitate our day-to-day tasks,
  - ✓ Email services
  - ✓ File transferring
  - ✓ Remote terminals
  - ✓ Web services



# PRESENTATION LAYER

There are three major functions performed by the Presentation Layer,

1. Translation
2. Encryption
3. Compression



# PRESENTATION LAYER CONT.

There are three major functions performed by the Presentation Layer,

## Translation

- Sender and Receiver may be using two different formats to represent data (e.g. ASCII vs. UNICODE).
- Translation is the process of converting data,
  - From **sender-dependent** format to a **common** format at the **sender**
  - From **common format** to the **receiver-dependent** format at the **receiver**.



# **PRESENTATION LAYER CONT.**

## **Encryption**

- Translate the data into an not recognizable format
- To prevent unauthorized people from viewing the data

## **Compression**

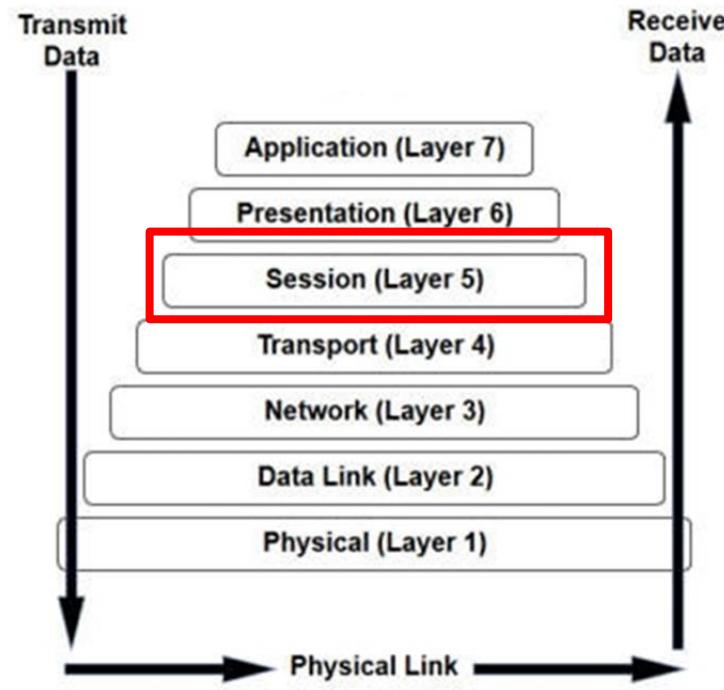
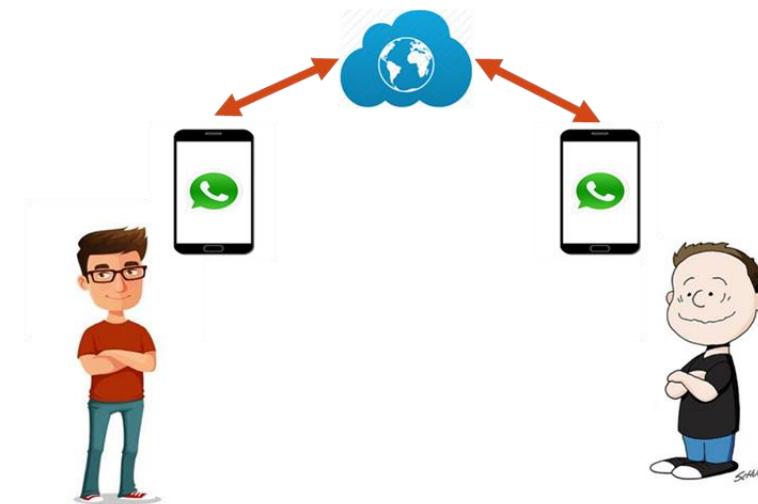
- To reduce the number of bits to be transmitted (e.g., reduce the size)
- Important when sending multimedia content such as audio and video



# SESSION LAYER

There are **two major functions** performed by the Session Layer,

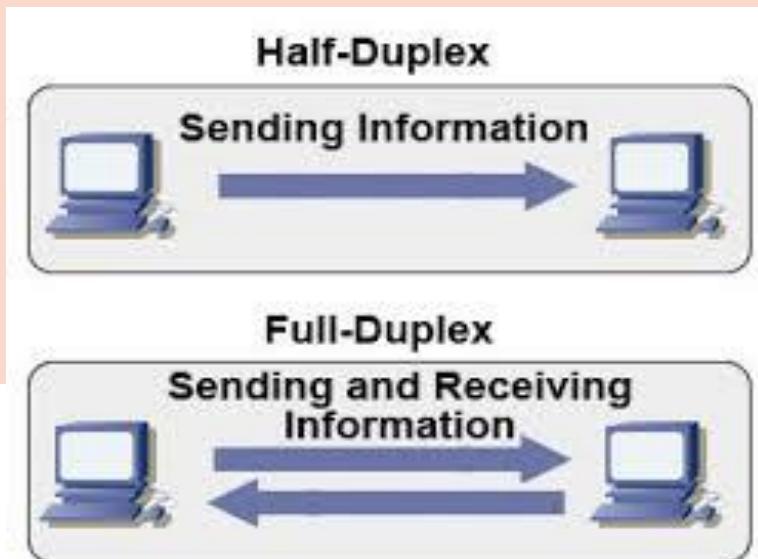
1. **Dialog Control**
2. **Synchronization**



# SESSION LAYER CONT.

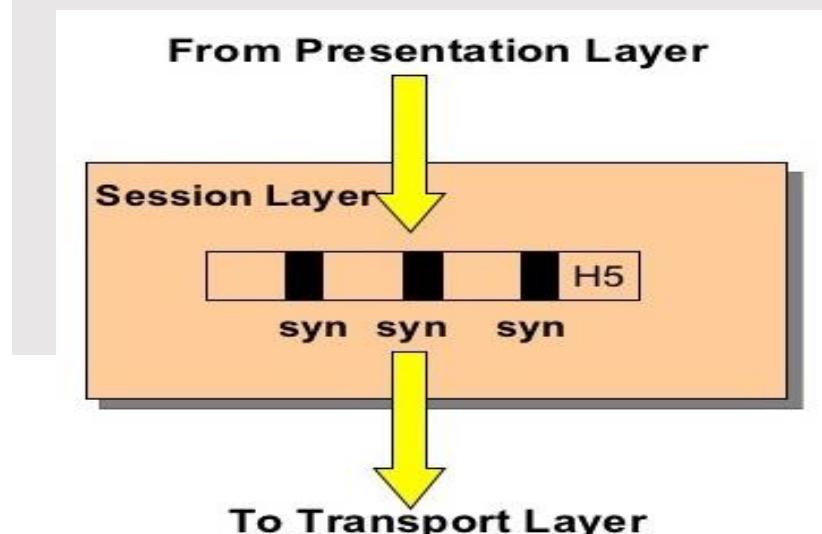
## ▪ Dialog Control

- ✓ Allows the **two** communicating processes to enter into a dialog (i.e. a two way communication)
- ✓ This dialog can be either half duplex or full duplex



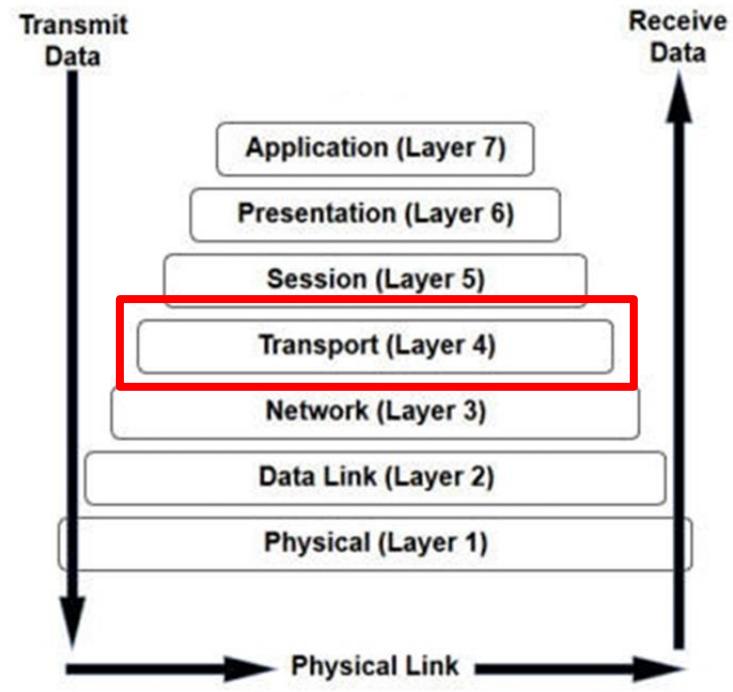
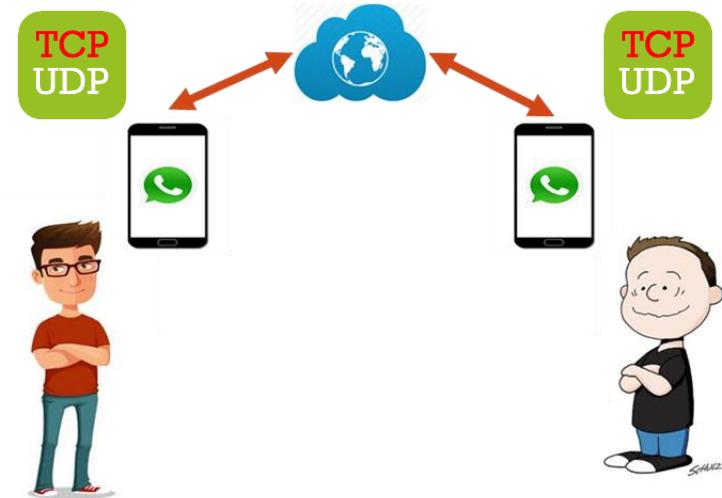
## ▪ Synchronization

- ✓ Session layer **adds checkpoints** in **between** the long data stream passed by the layer above (i.e. presentation layer)
- ✓ Adds an additional level of verification to the receiver



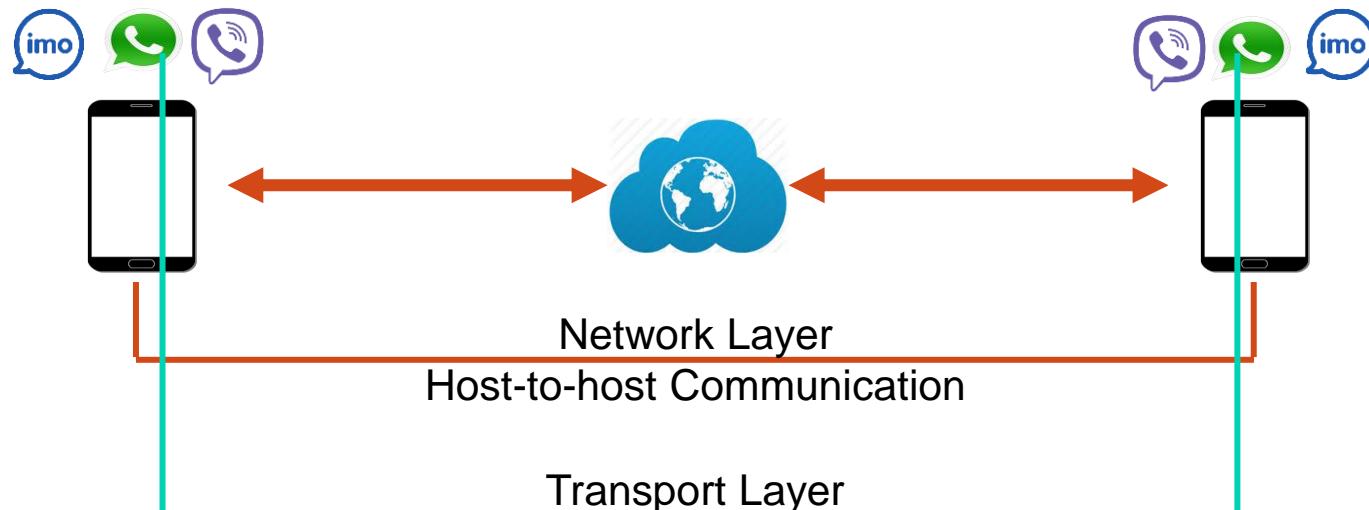
# TRANSPORT LAYER

- Several very important functionalities,
  - ✓ **PORT Numbers:** an address to each application
  - ✓ **Segmentation** and reassembly
  - ✓ **Connection control**
  - ✓ **Flow control**
  - ✓ **Error control**



# TRANSPORT LAYER CONT.

## TRANSPORT LAYER: PROCESS TO PROCESS DELIVERY

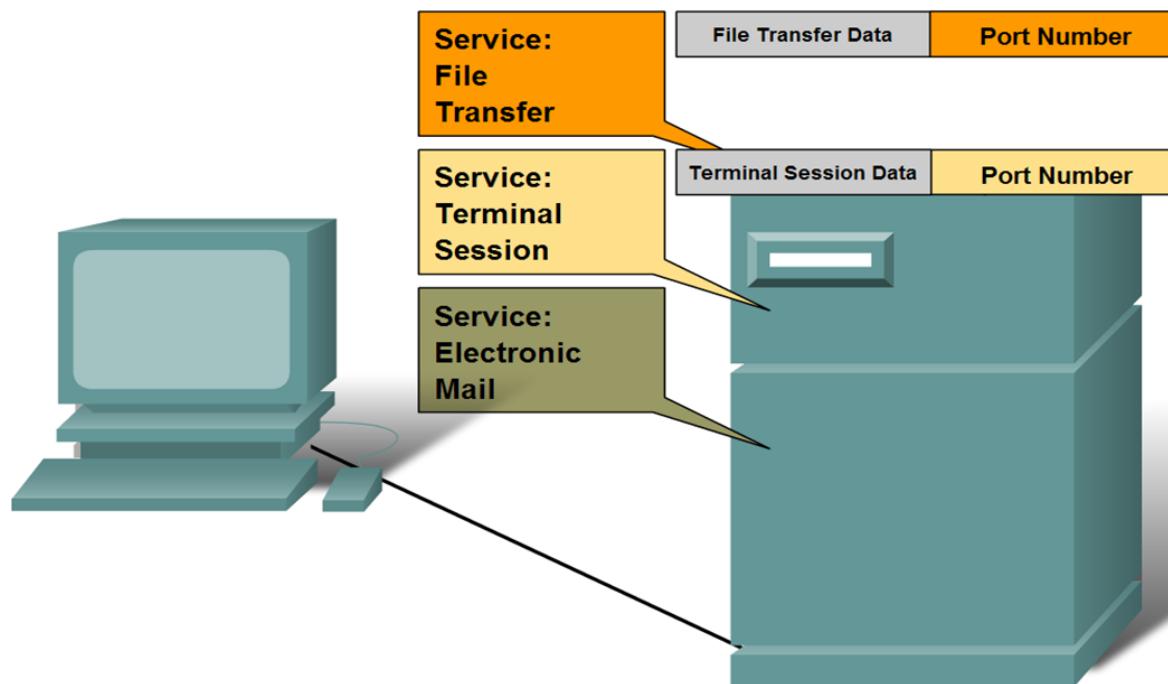


# TRANSPORT LAYER CONT.

## TRANSPORT LAYER ADDRESS: PORT ADDRESS

- Uses to identify the source and destination processes for communication

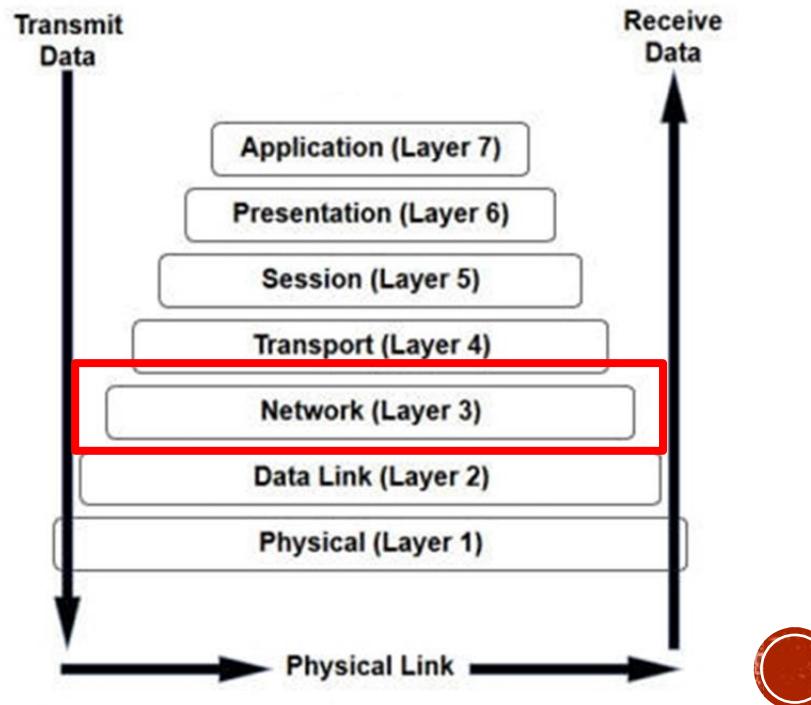
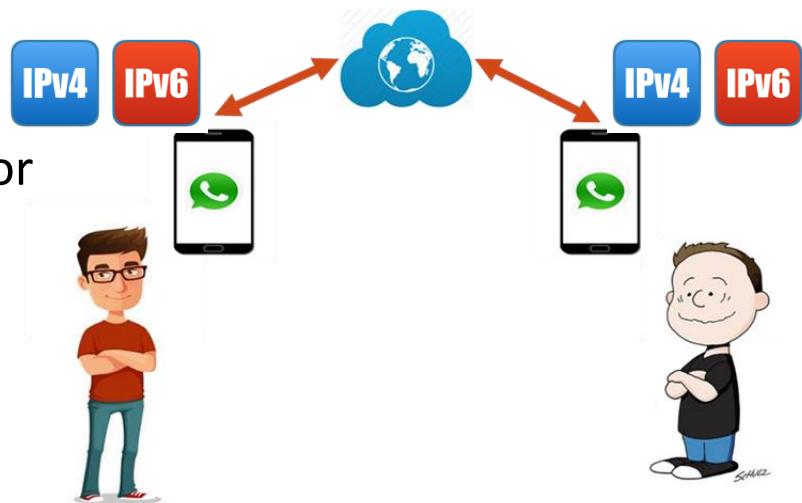
At the end device, the service port number directs the data to the correct conversation.



# NETWORK LAYER

- Network layer is responsible for two major functionalities:

1. **Routing**
2. **Logical addressing (IP address)**



# NETWORK LAYER CONT.

## Logical addressing (IP address)

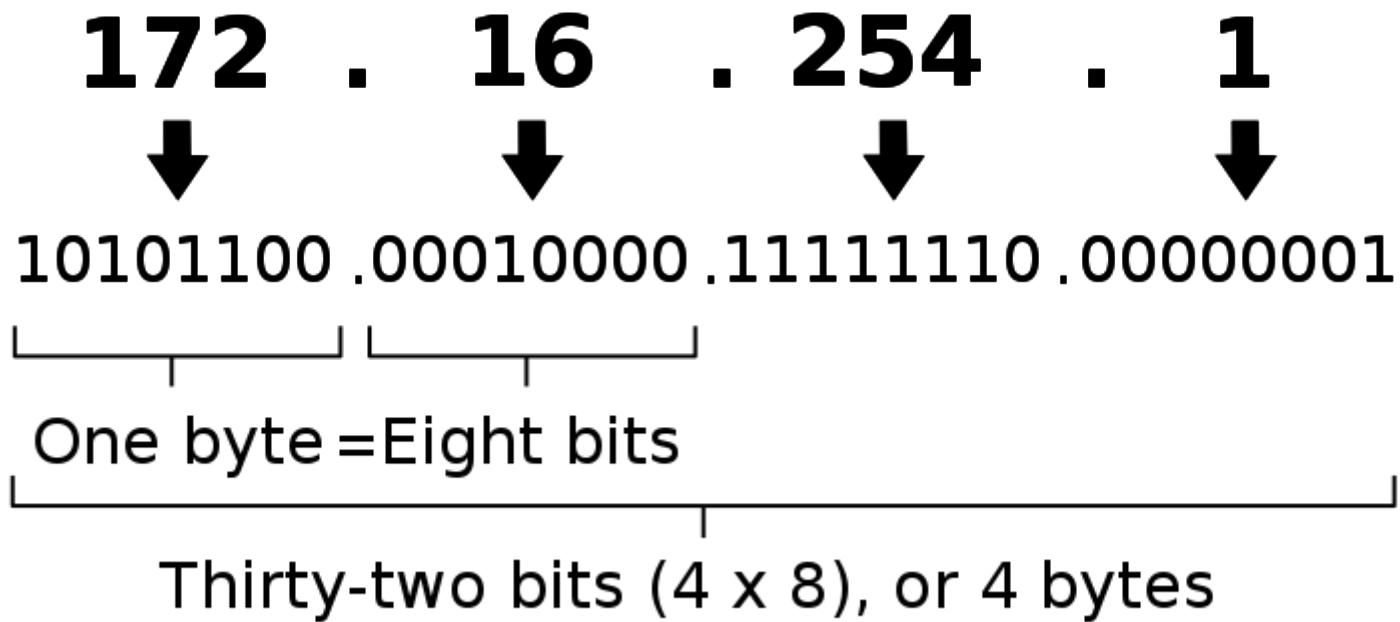
- Each **device** should **have a address**
- Network layer issues addresses to the devices called the **logical addresses: IP addresses**

## Routing

- Applications are in **different devices**
- Different devices are **connected to same/different networks**
- So there should be a mechanism to **travers the data** between the devices **through the networks**
- That mechanism is known as **routing**

# NETWORK LAYER CONT.

## NETWORK LAYER ADDRESS: IP ADDRESS



# NETWORK LAYER CONT.

## NETWORK LAYER ADDRESS: IP ADDRESS

- IP address has two parts: the network prefix and the host part

192	.	168	.	10	.	1
11000000	10101000	00001010		00000001		

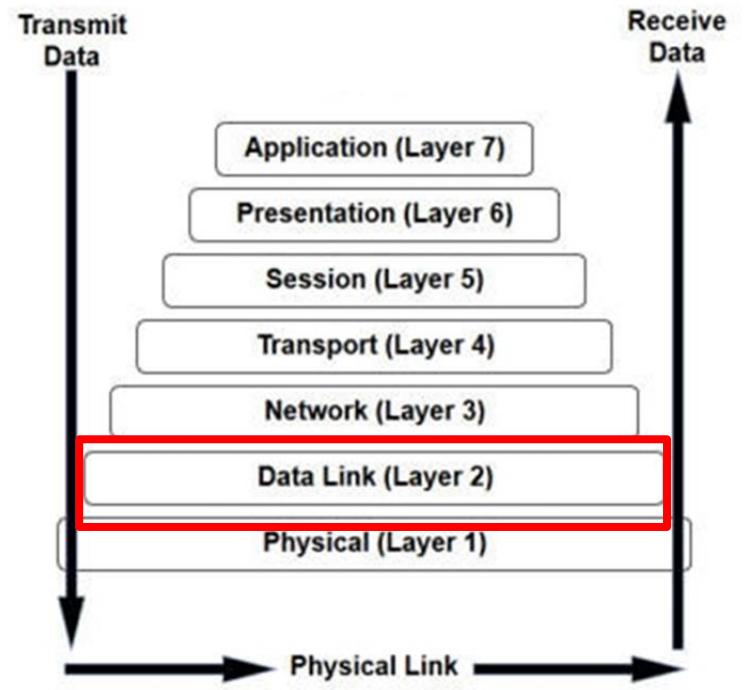
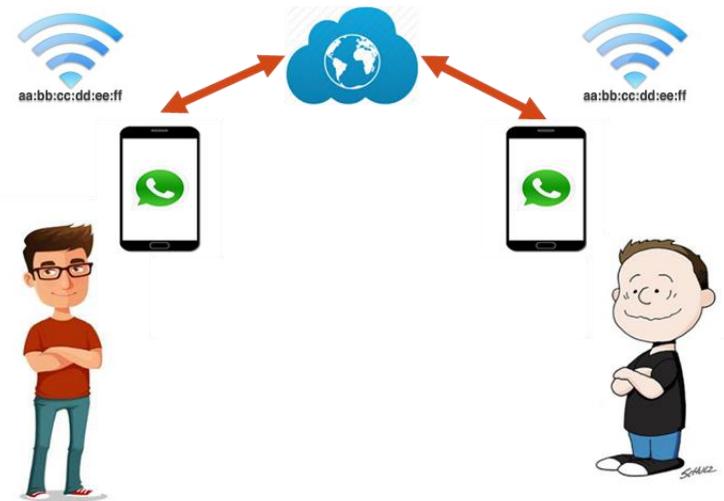
The computer using this IP address is on network 192.168.10.0.



# DATA LINK LAYER

- Responsible to guarantee the successful transmission of frames from one hop to the next

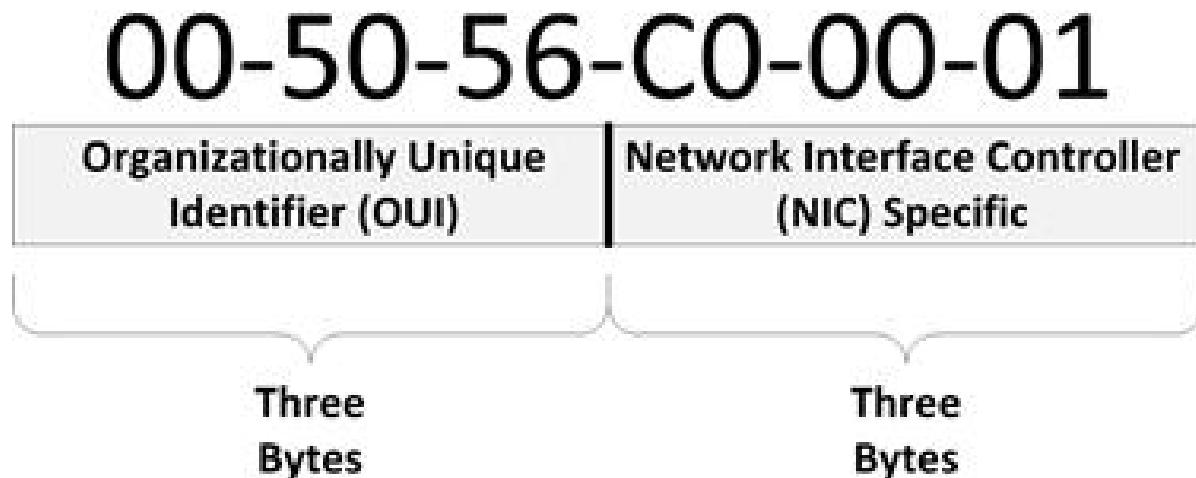
1. Framing – Ethernet frame
2. Physical addressing (MAC address)
3. Flow control
4. Error control
5. Access control



# DATA LINK LAYER CONT.

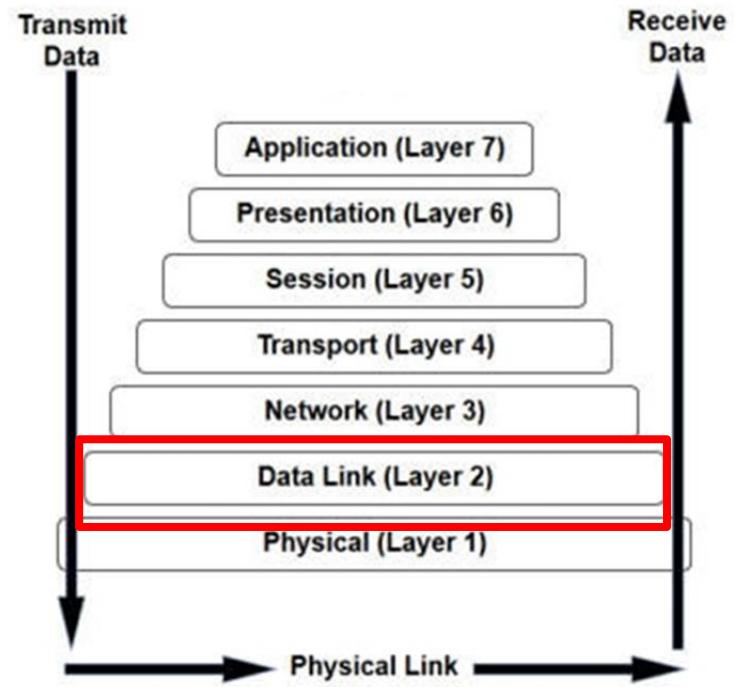
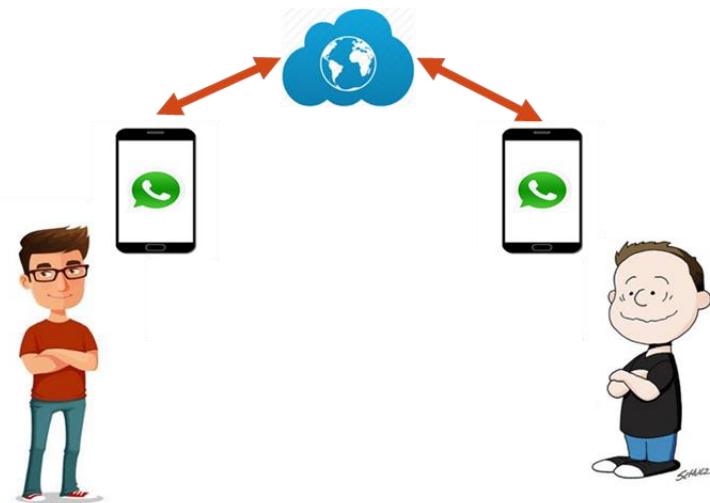
## DATA LINK LAYER ADDRESS: MAC ADDRESS

- Physical address
- Universally Unique address
- 48 bit long address

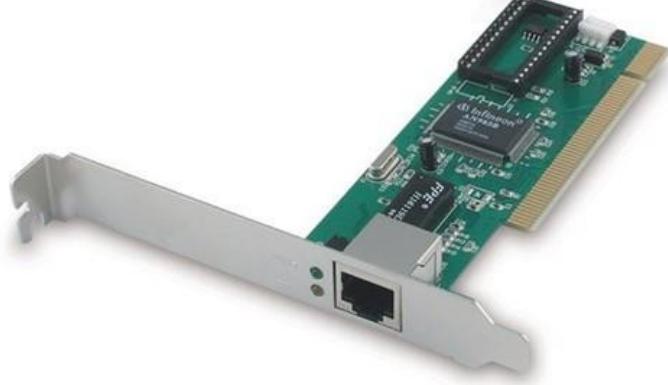


# PHYSICAL LAYER

- Physical Interfaces transmit and receive unstructured raw data over the transmission media
- Converts the digital/analog bits into electrical or optical signals



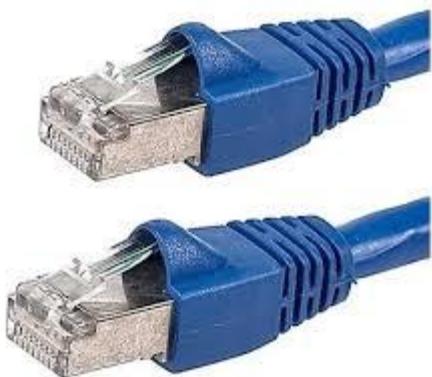
# PHYSICAL LAYER CONT.



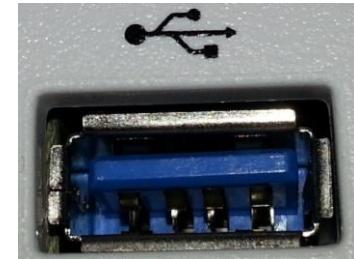
??



??



??



??



??



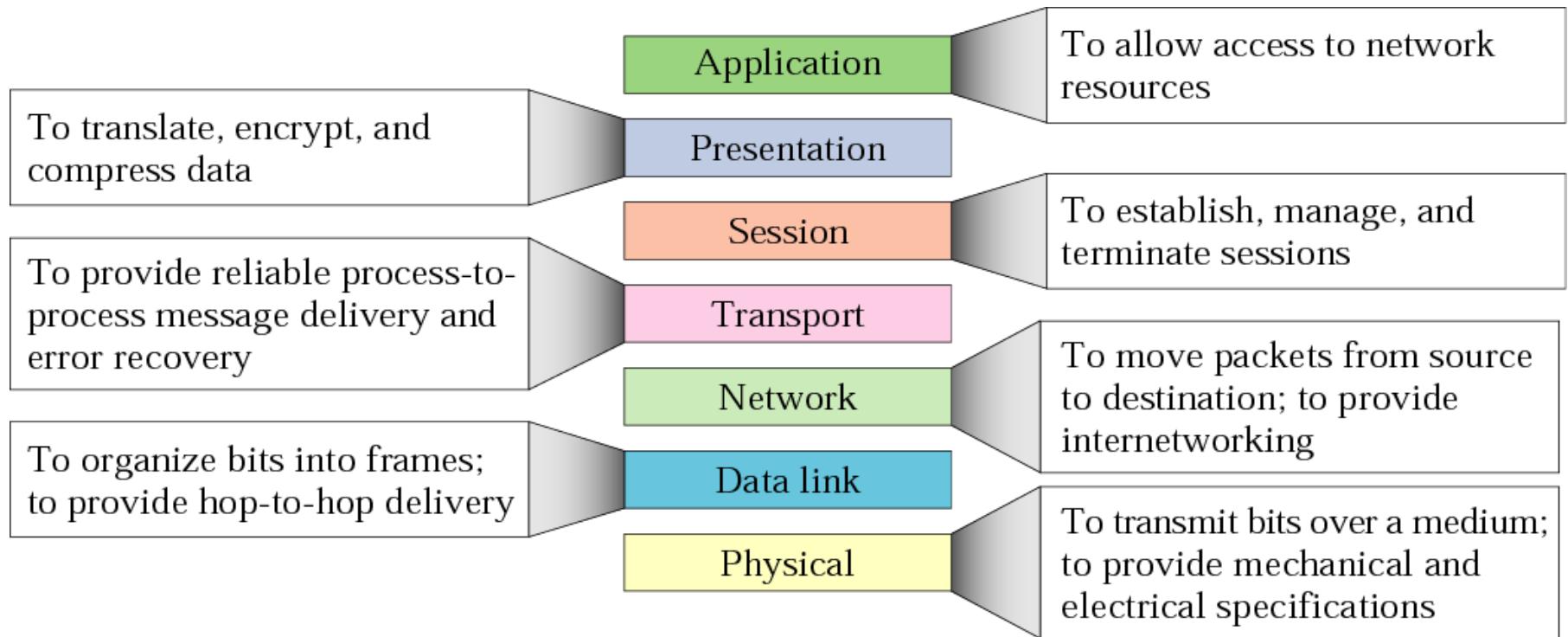
# PHYSICAL LAYER CONT.

## STANDARD BODIES THAT DEFINE PHYSICAL LAYER SPECIFICATIONS

- EIA: Electronic Industries Alliance
- ITU-T: International Telecommunication Union  
Telecommunication Standardization Sector
- ANSI: American National Standards Institute

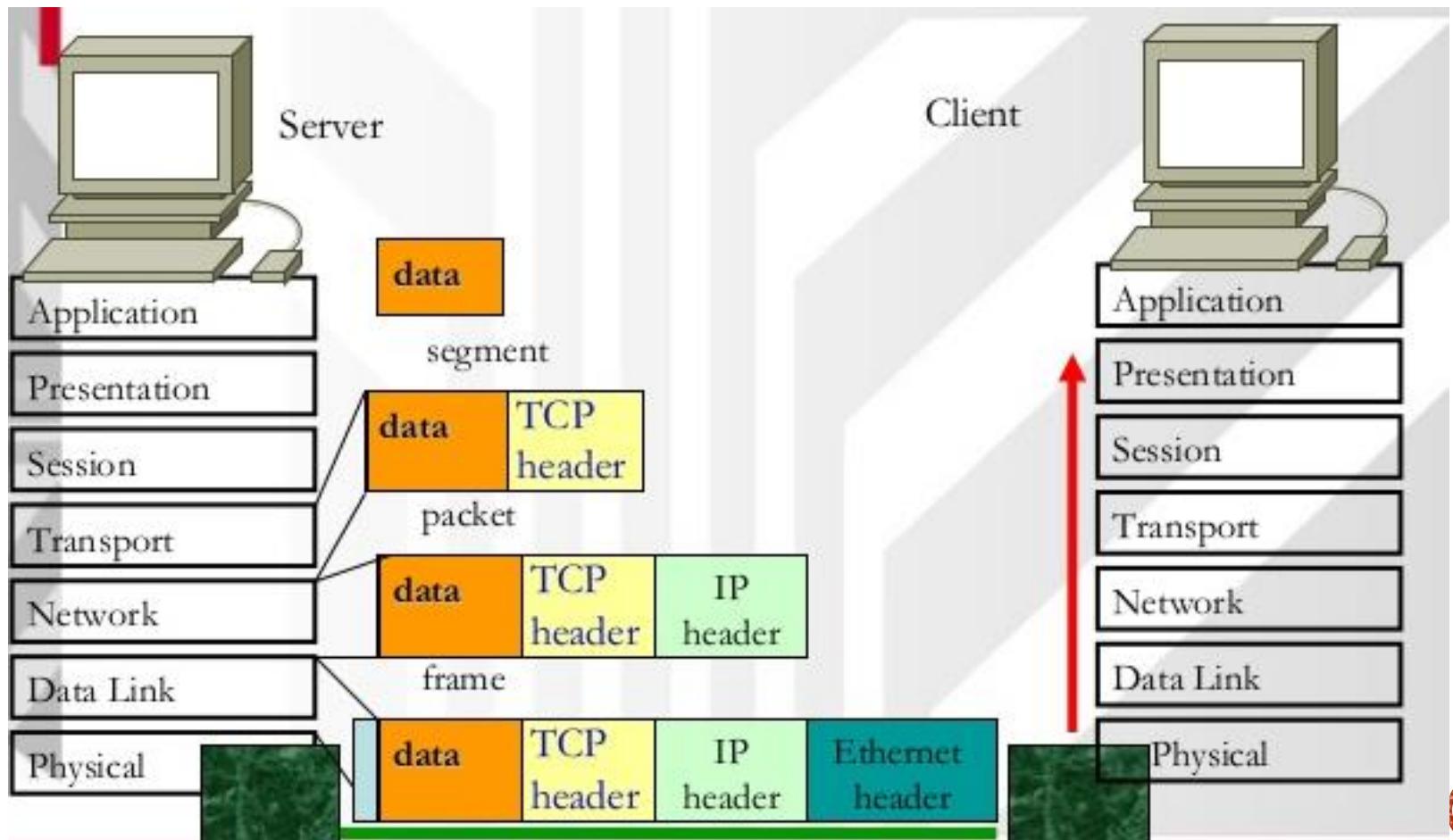


# SUMMARY OF THE ISO – OSI LAYERS

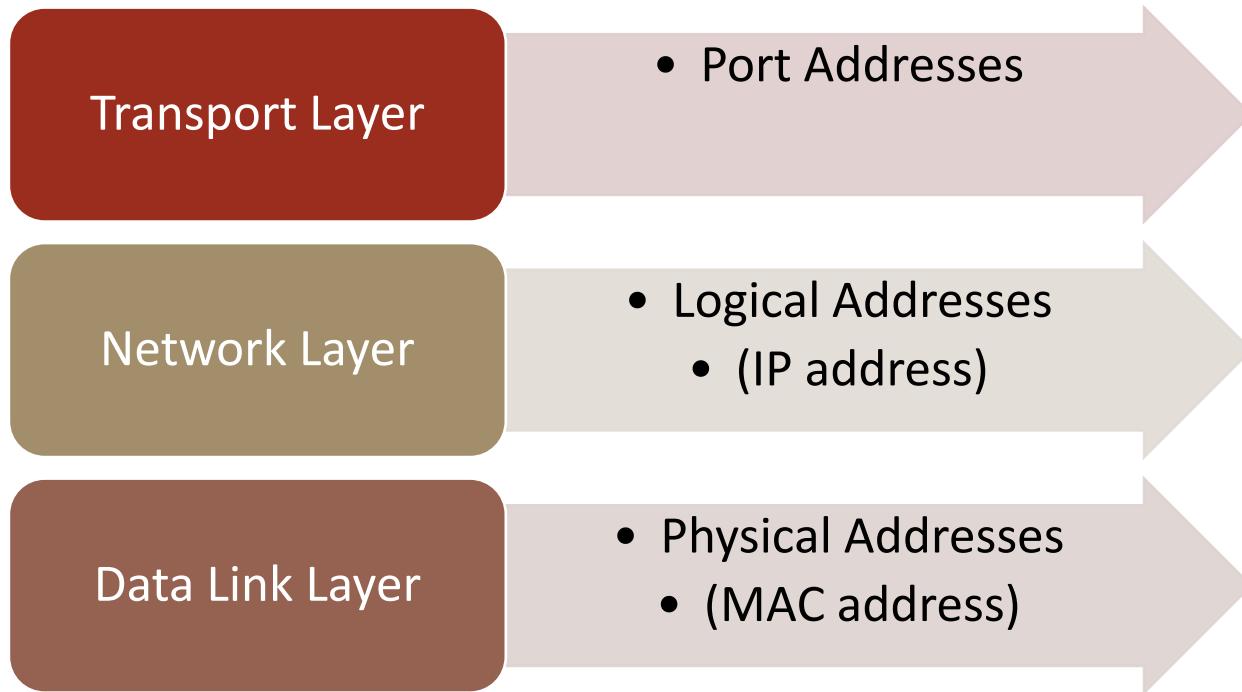


# ISO - OSI REFERENCE MODEL CONT.

## HEADERS CONCEPT: EMBED CONTROL INFORMATION



# WHAT ARE THE ADDRESS USE IN NETWORK COMMUNICATION ?



# WHAT ARE THE CONNECTIONS USED AT DIFFERENT LAYERS?

Transport Layer

- **Process-to-Process**
- i.e. Source Application to Destination Application

Network Layer

- **Host-to-Host**
- i.e. Source Computer to Destination Computer

Data Link Layer

- **Hop-to-Hop**
- i.e. One hop to the next hop



# NAMES USED FOR DATA AT EACH LAYER

Application Layer	• Data
Presentation Layer	• Data
Session Layer	• Data
Transport Layer	• Segments
Network Layer	• Packets
Data Link Layer	• Frames
Physical Layer	• Bits

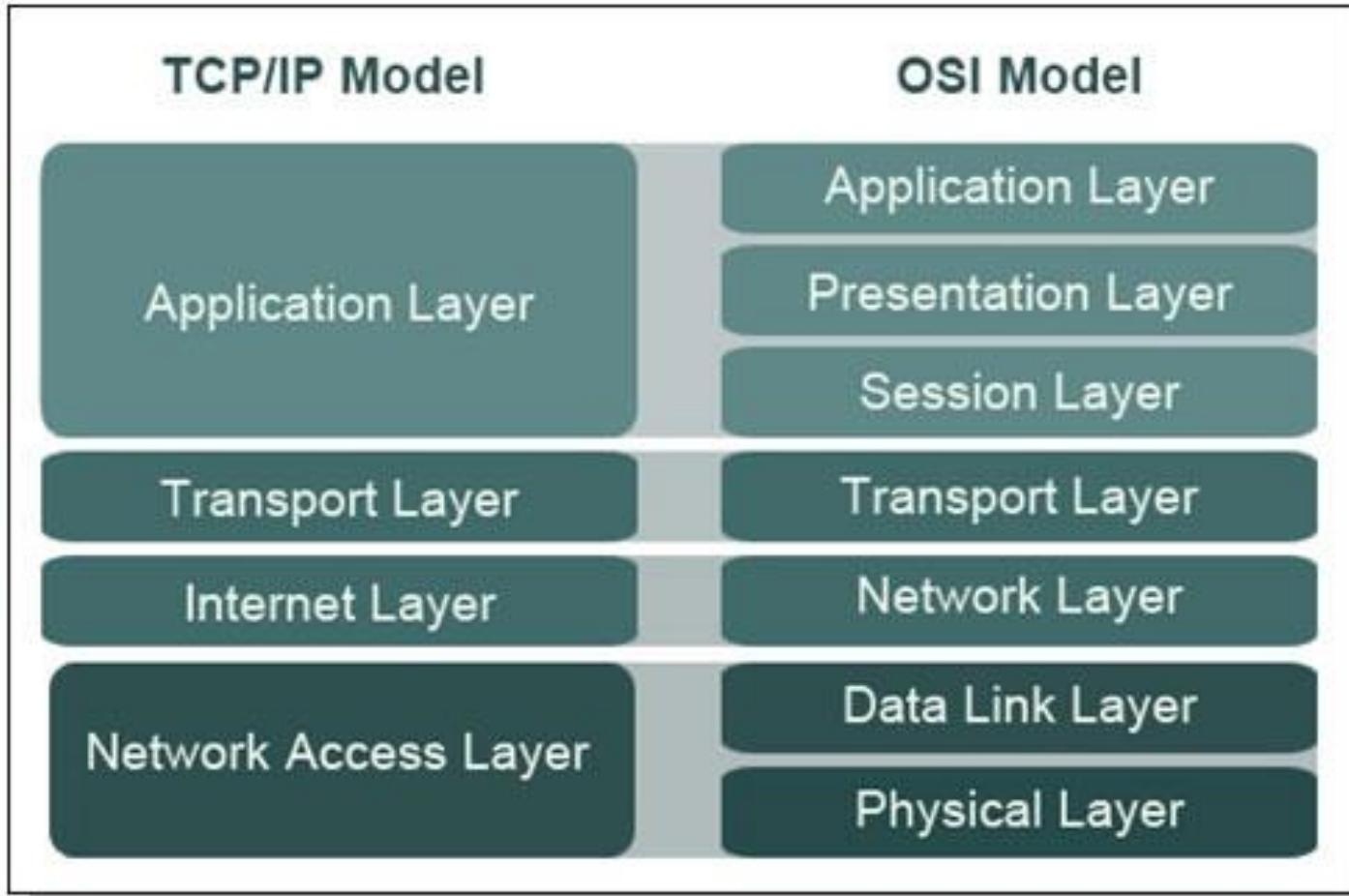


# PROTOCOLS USED AT DIFFERENT LAYERS

Layer	Name	Example protocols
7	Application Layer	HTTP, FTP, DNS, SNMP, Telnet
6	Presentation Layer	SSL, TLS
5	Session Layer	NetBIOS, PPTP
4	Transport Layer	TCP, UDP
3	Network Layer	IP, ARP, ICMP, IPSec
2	Data Link Layer	PPP, ATM, Ethernet
1	Physical Layer	Ethernet, USB, Bluetooth, IEEE802.11



# TCP/IP STACK





*That's all Folks!*

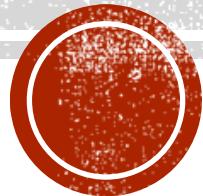


# Questions



# **INTRODUCTION TO COMPUTER SYSTEMS**

## **COMPUTER NETWORKS**



# BRACE YOURSELVES

ONLINE

# EXAMS ARE COMING

memegenerator.net

# **LECTURE 10**

## **COMMUNICATING OVER THE NETWORK**



# LECTURE OUTLINE

- Network Devices
- Broadcast Domain
- Collision Domain
- Transmission Media
  - Guided
  - Un-Guided



# DATA COMMUNICATION NETWORKS NEEDS

- Devices
  - To communicate with one another
- Medium
  - Connects devices together
- Messages
  - Information over media
- Rules
  - Govern how messages flow across networks



# NETWORK DEVICES

LAN

- Switch
- Hub
- Bridge
- Repeater

WAN

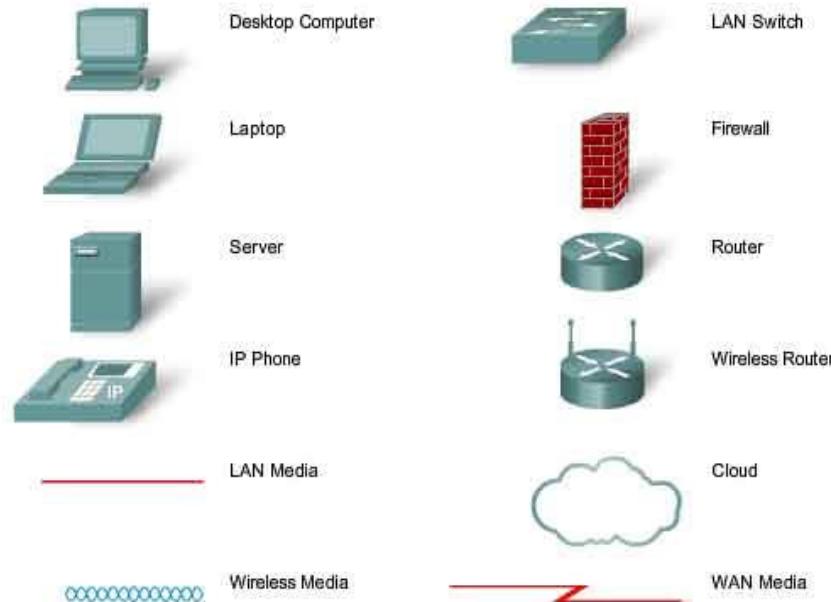
- Router
- Modem

Utility

- Servers
- Firewall
- Gateway
- IPS/IDS

- The major devices used to create networks:

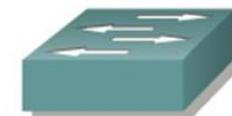
Common Data Network Symbols



Router



Switch



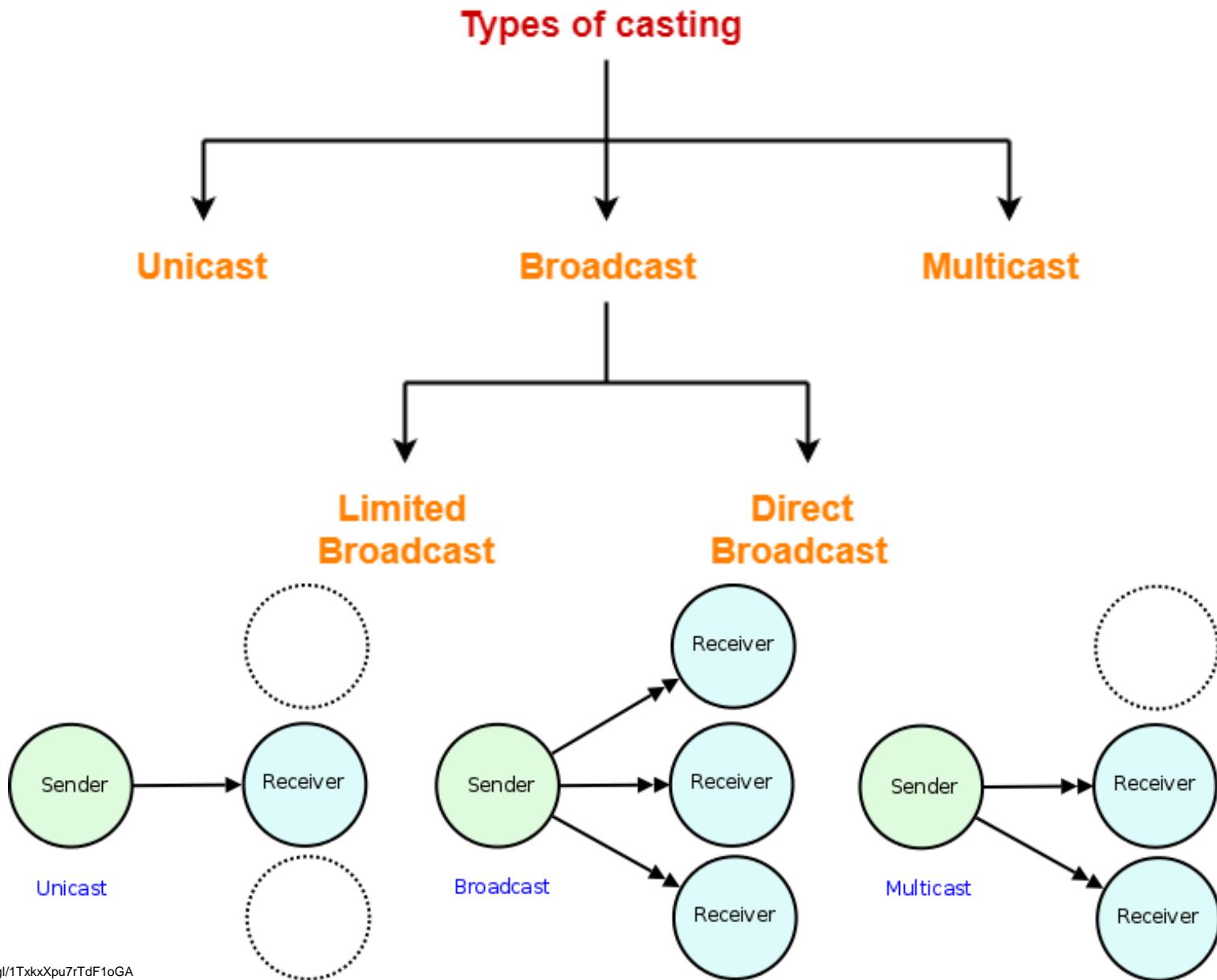
Modem (CSU/DSU)



Communication Server

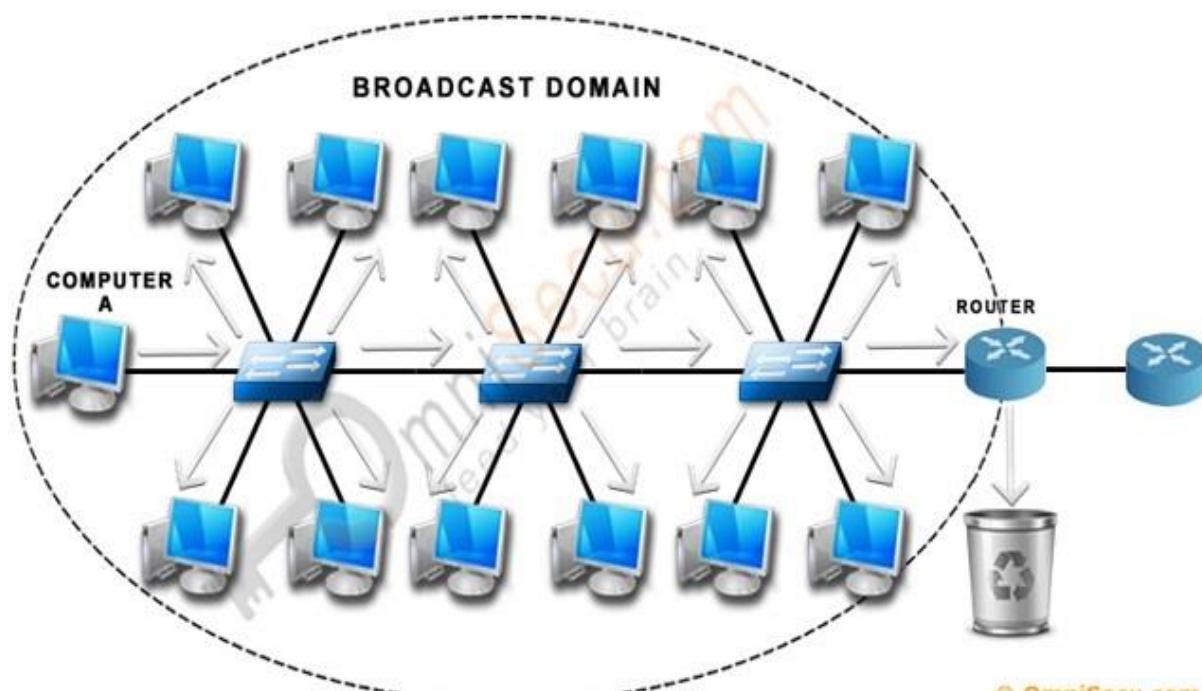


# MESSAGE DELIVERY MODES



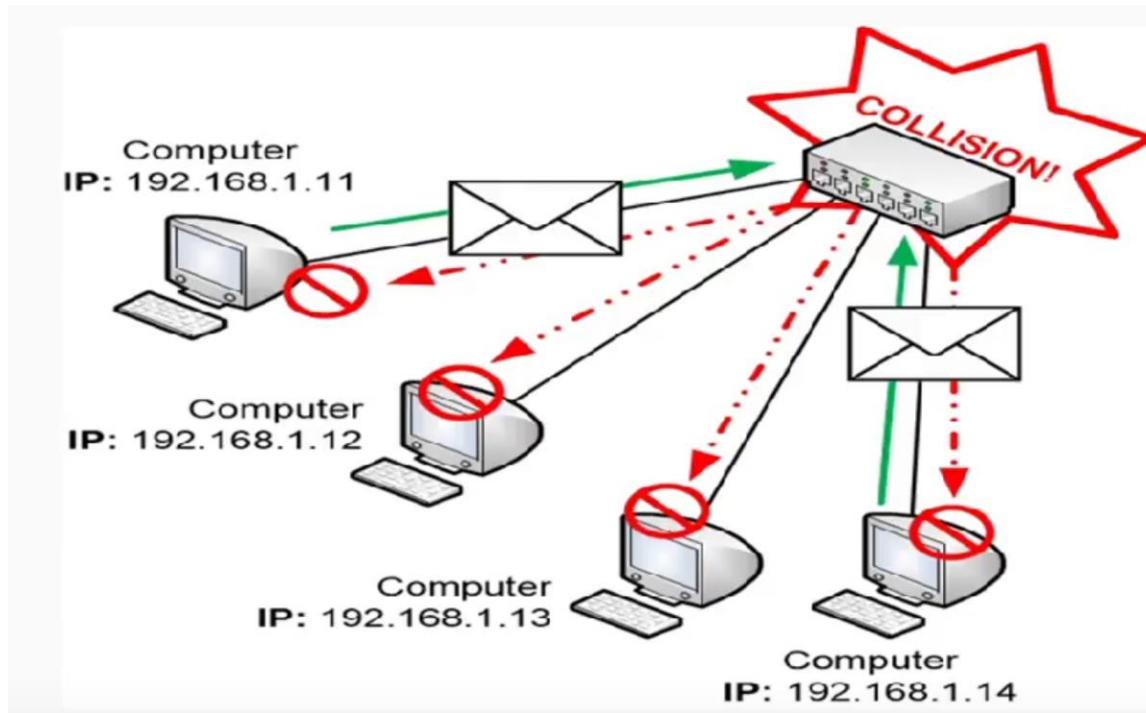
# BROADCAST DOMAIN

- A **broadcast domain** is a collection of computers in which, when **one sender sends** a message, the message is delivered to **all others** in the **same domain**.

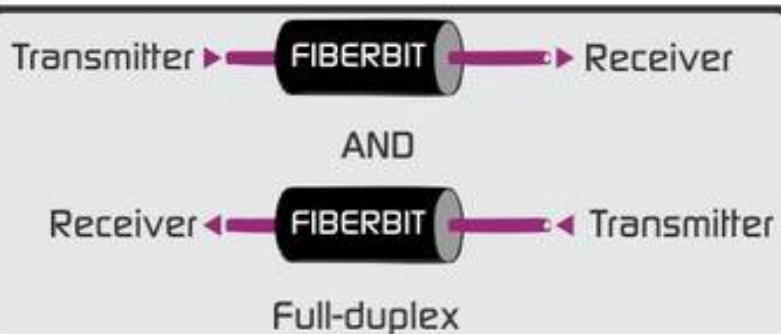
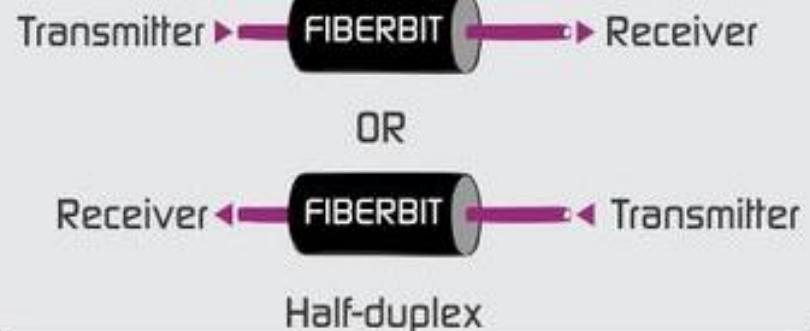


# COLLISION DOMAIN

- A **collision domain** is a collection of **computers** in which, if **more than one sender** tries to **send some data simultaneously**, the **signal will collide** in the transmission media and make all the sent information unusable.



# FULL DUPLEX VS HALF DUPLEX VS SIMPLEX



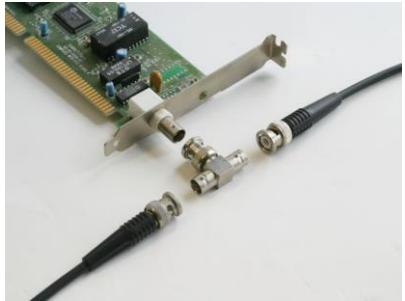
# NETWORK INTERFACE CARD (NIC)

## AKA NETWORK INTERFACE CONTROLLER

- NIC is the **hardware interface** between a **computer** and a **network**
- NIC can operate in **half duplex or full duplex** modes
- NIC performs,
  - ✓ Carrier Sense
  - ✓ Converting the binary data into encoded signals and vice versa
  - ✓ Media Access Control

<b>Application Layer</b>
✓ Message format, Human-Machine Interfaces
<b>Presentation Layer</b>
✓ Coding into 1s and 0s; encryption, compression
<b>Session Layer</b>
✓ Authentication, permissions, session restoration
<b>Transport Layer</b>
✓ End-to-end error control
<b>Network Layer</b>
✓ Network addressing; routing or switching
<b>Data Link Layer</b>
✓ Error detection, flow control on physical link
<b>Physical Layer</b>
✓ Bit stream: physical medium, method of representing bits

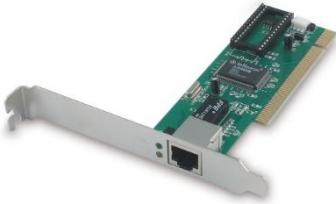
# TYPES OF NICS



Old NIC with BNC connection



A transitional NIC with BNC and RJ45 connections



A typical modern NIC with RJ45 connection



Wireless NIC



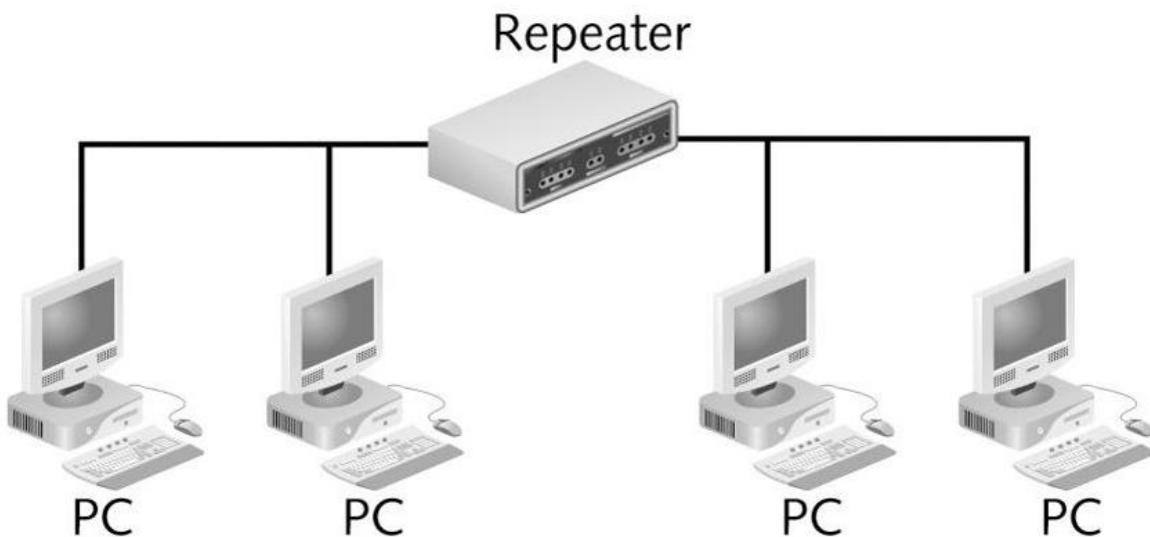
A multiport NIC typically used in servers



# REPEATER



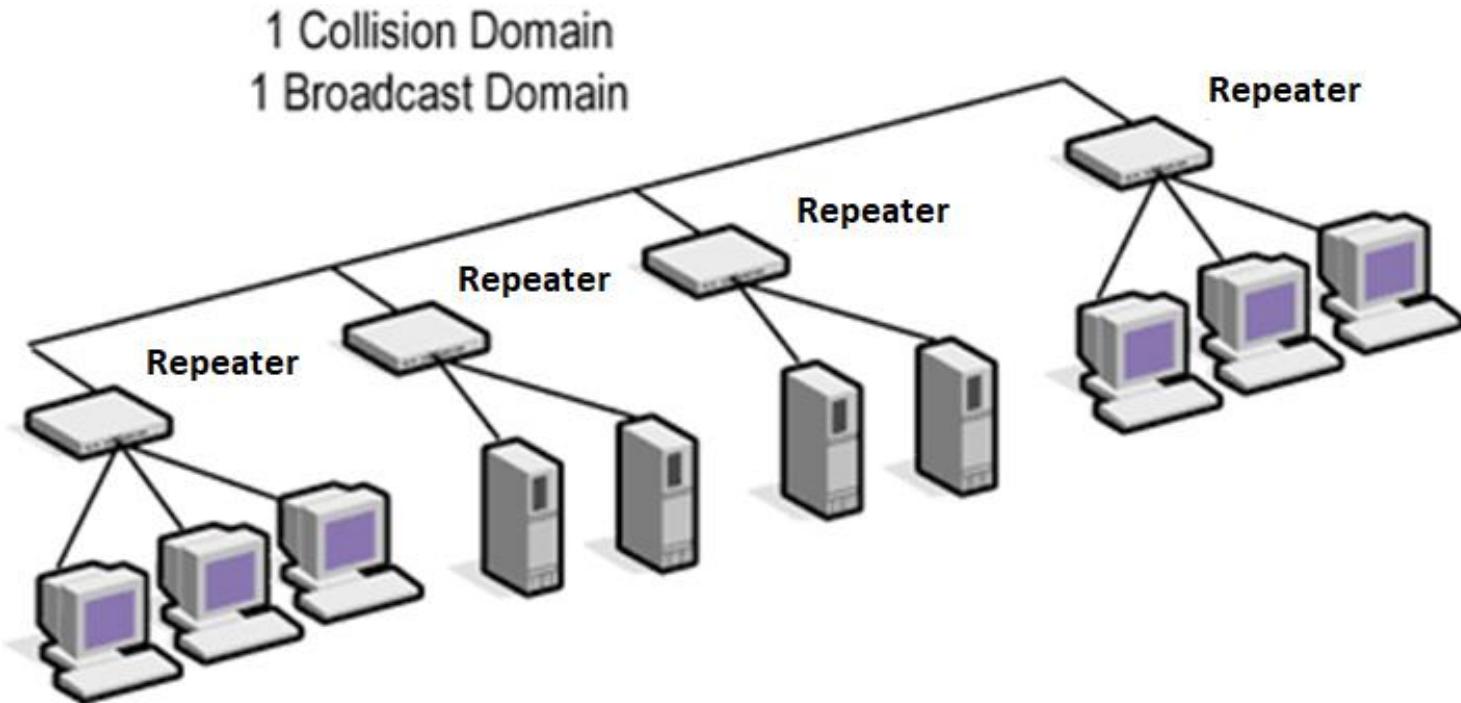
- Connects two network segments
- **Regenerates the signals to proper amplitudes** and sends them to the next segment



<b>Application Layer</b>
✓ Message format, Human-Machine Interfaces
<b>Presentation Layer</b>
✓ Coding into 1s and 0s; encryption, compression
<b>Session Layer</b>
✓ Authentication, permissions, session restoration
<b>Transport Layer</b>
✓ End-to-end error control
<b>Network Layer</b>
✓ Network addressing; routing or switching
<b>Data Link Layer</b>
✓ Error detection, flow control on physical link
<b>Physical Layer</b>
✓ Bit stream: physical medium, method of representing bits

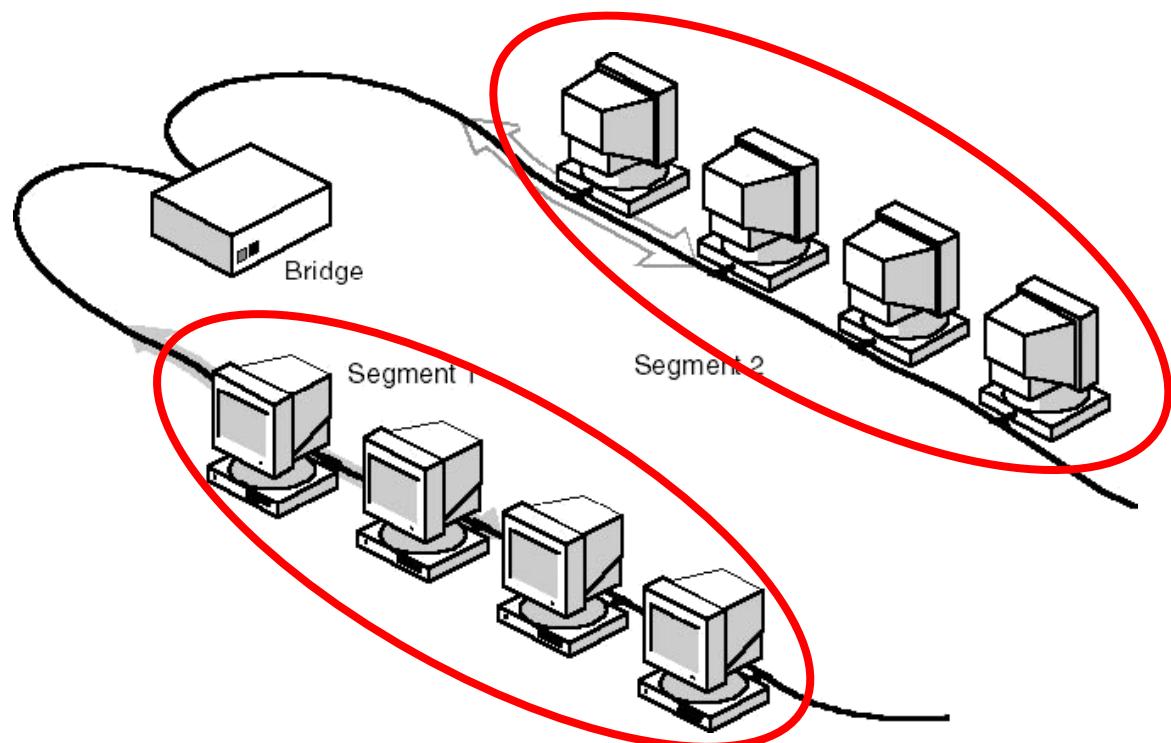
# REPEATER CONT.

- One broadcast domain
- One collision domain



# BRIDGE

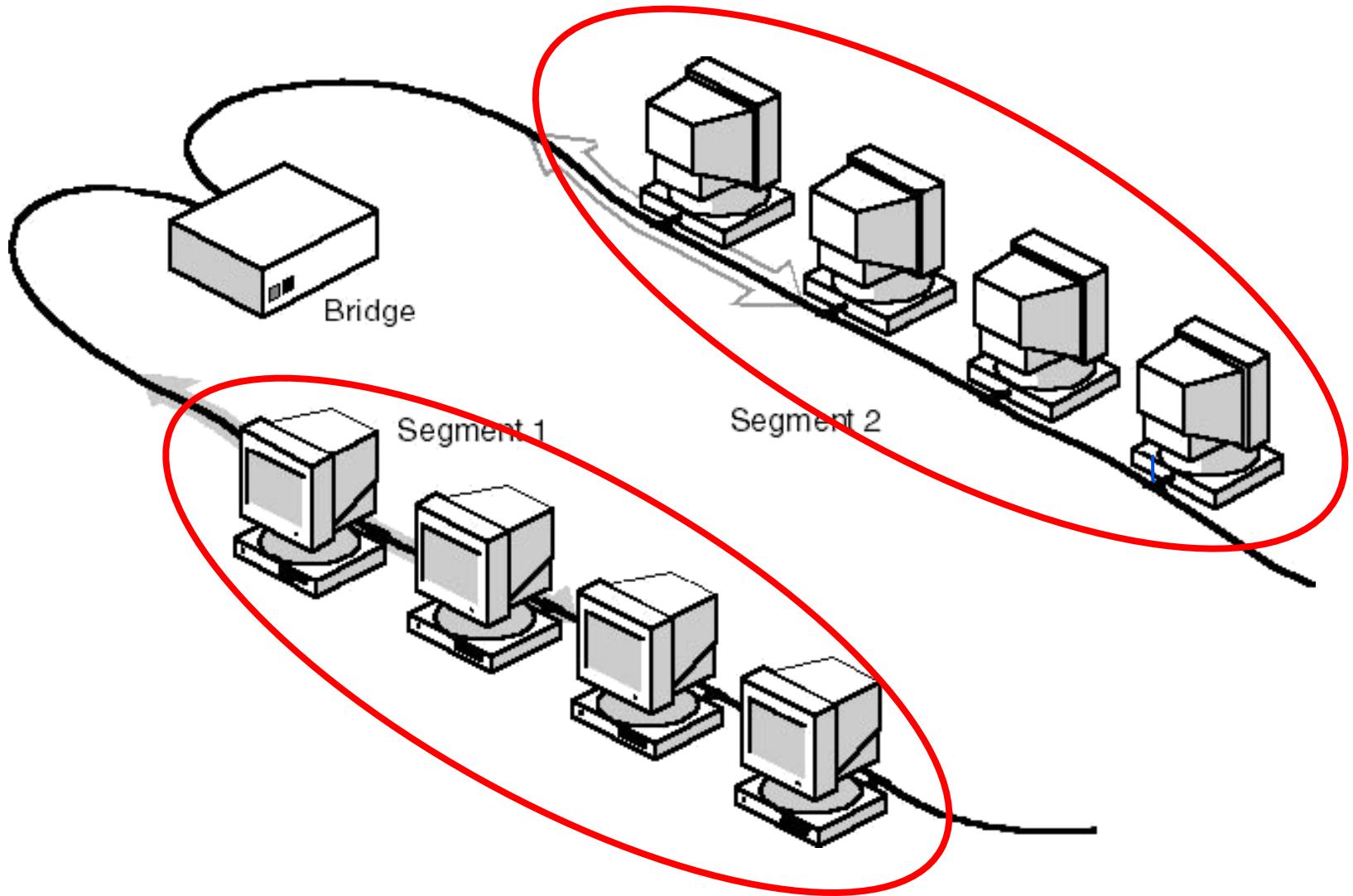
- Used to connect **two or more** separate **LAN** segments



<b>Application Layer</b>
✓ Message format, Human-Machine Interfaces
<b>Presentation Layer</b>
✓ Coding into 1s and 0s; encryption, compression
<b>Session Layer</b>
✓ Authentication, permissions, session restoration
<b>Transport Layer</b>
✓ End-to-end error control
<b>Network Layer</b>
✓ Network addressing; routing or switching
<b>Data Link Layer</b>
✓ Error detection, flow control on physical link
<b>Physical Layer</b>
✓ Bit stream: physical medium, method of representing bits

# BRIDGE CONT.

2 Collision Domains  
1 Large Broadcast Domain



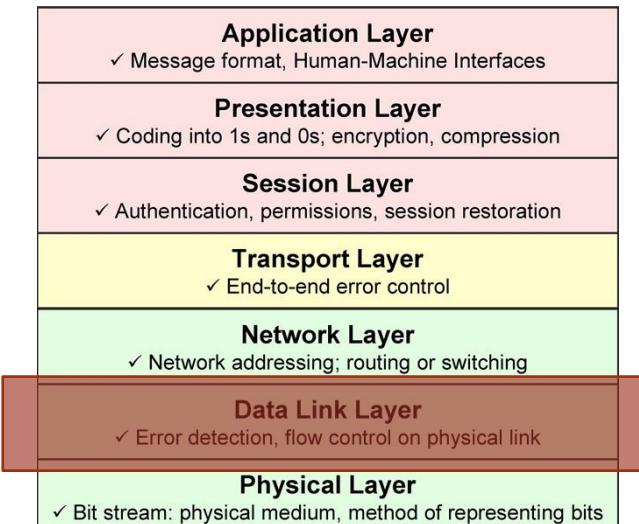
# SWITCH



- A switch is **similar in functionality to a bridge**  
(it is also used to connect multiple LAN segments together)

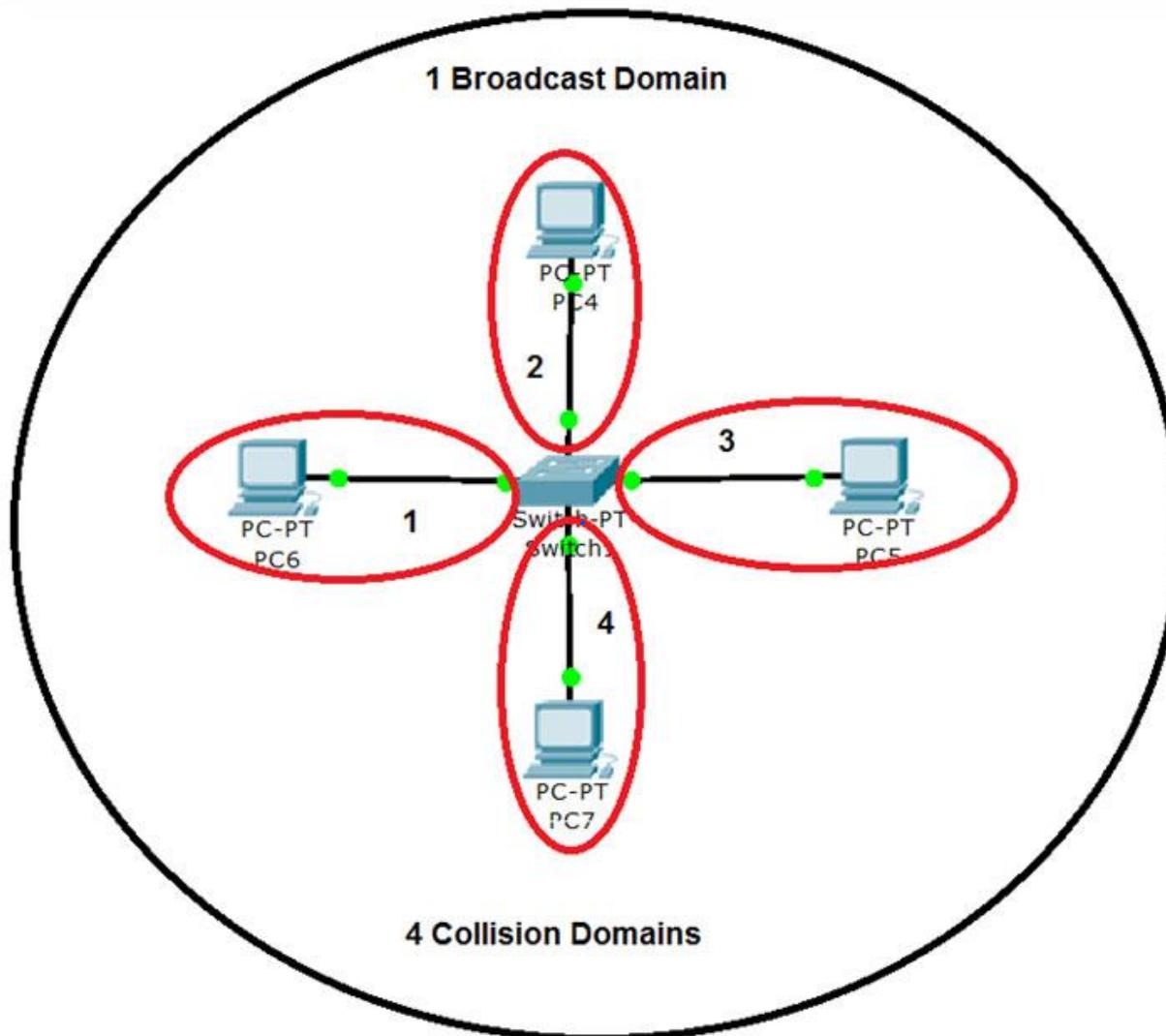
- However,

- ✓ A switch has a **large number of ports** compared to a bridge  
(a switch is commonly referred to as a multiport bridge)
- ✓ A switch **uses ASICs** for its processing and **switches are faster**  
(bridges **use** software running, using a **normal processor**)
- ✓ **Switches are widely used in today's LANs**  
**(bridges are no longer used)**



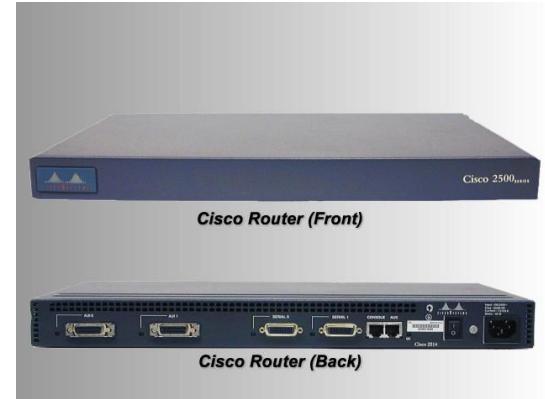
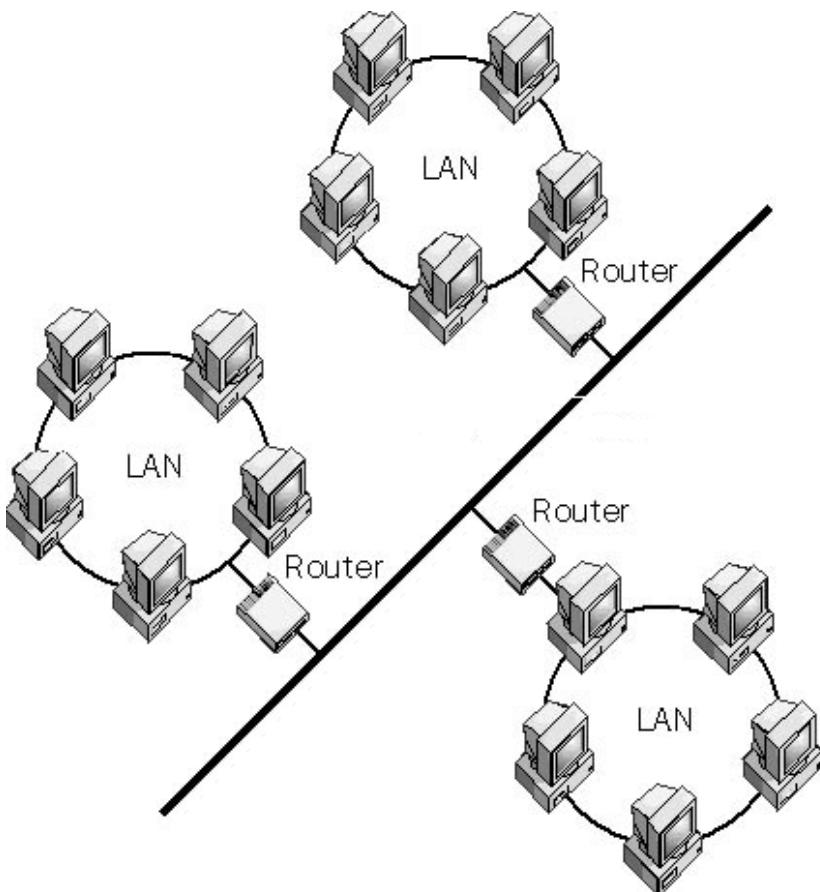
# SWITCH CONT.

4 Collision Domains  
1 Large Broadcast Domain



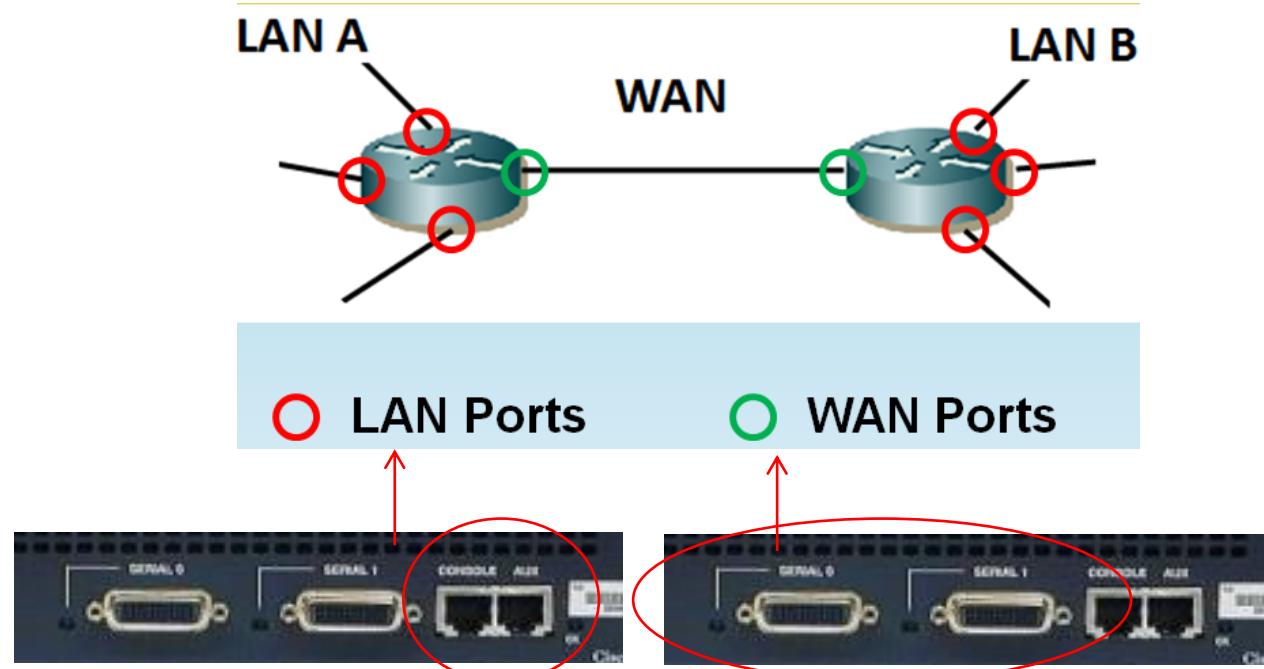
# ROUTER

- A router is used to **interconnect two or more LANs**



<b>Application Layer</b>
✓ Message format, Human-Machine Interfaces
<b>Presentation Layer</b>
✓ Coding into 1s and 0s; encryption, compression
<b>Session Layer</b>
✓ Authentication, permissions, session restoration
<b>Transport Layer</b>
✓ End-to-end error control
<b>Network Layer</b>
✓ Network addressing; routing or switching
<b>Data Link Layer</b>
✓ Error detection, flow control on physical link
<b>Physical Layer</b>
✓ Bit stream: physical medium, method of representing bits

# ROUTER CONT.

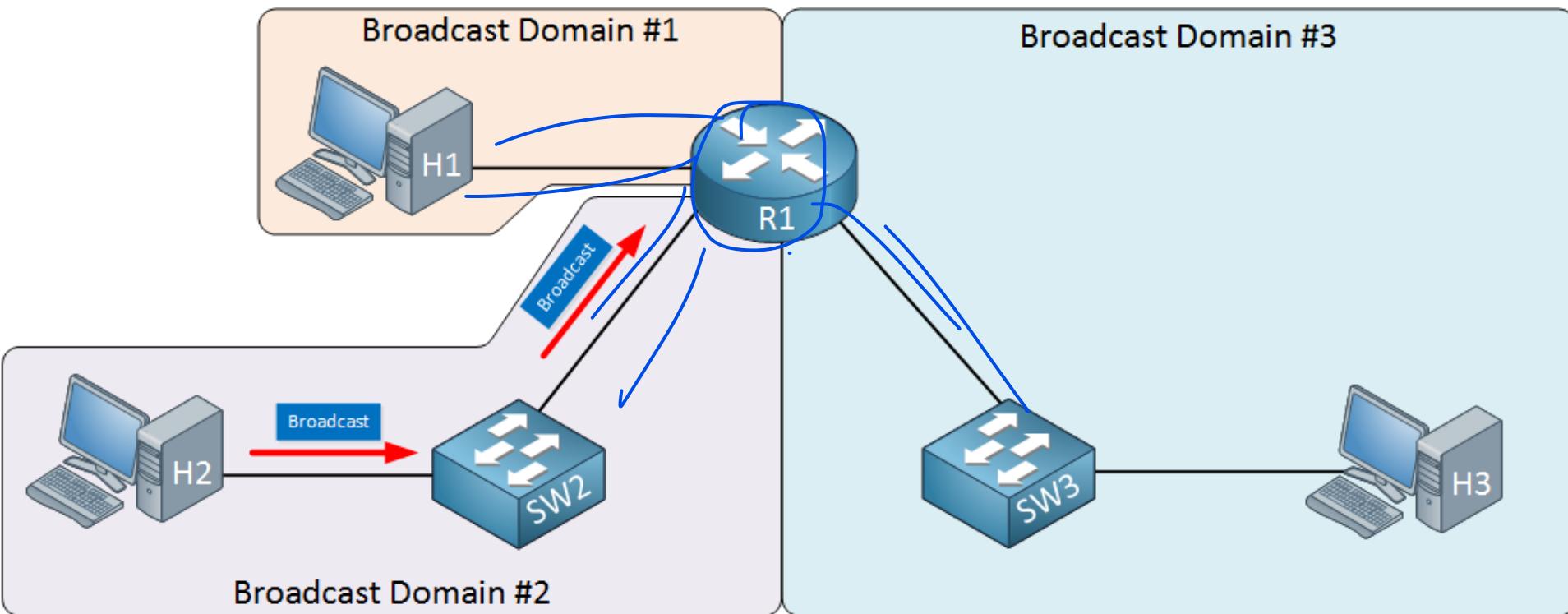


- LAN ports are RJ45 connections while WAN ports are generally serial connections



# ROUTER CONT.

5 Collision Domains  
3 Broadcast Domains



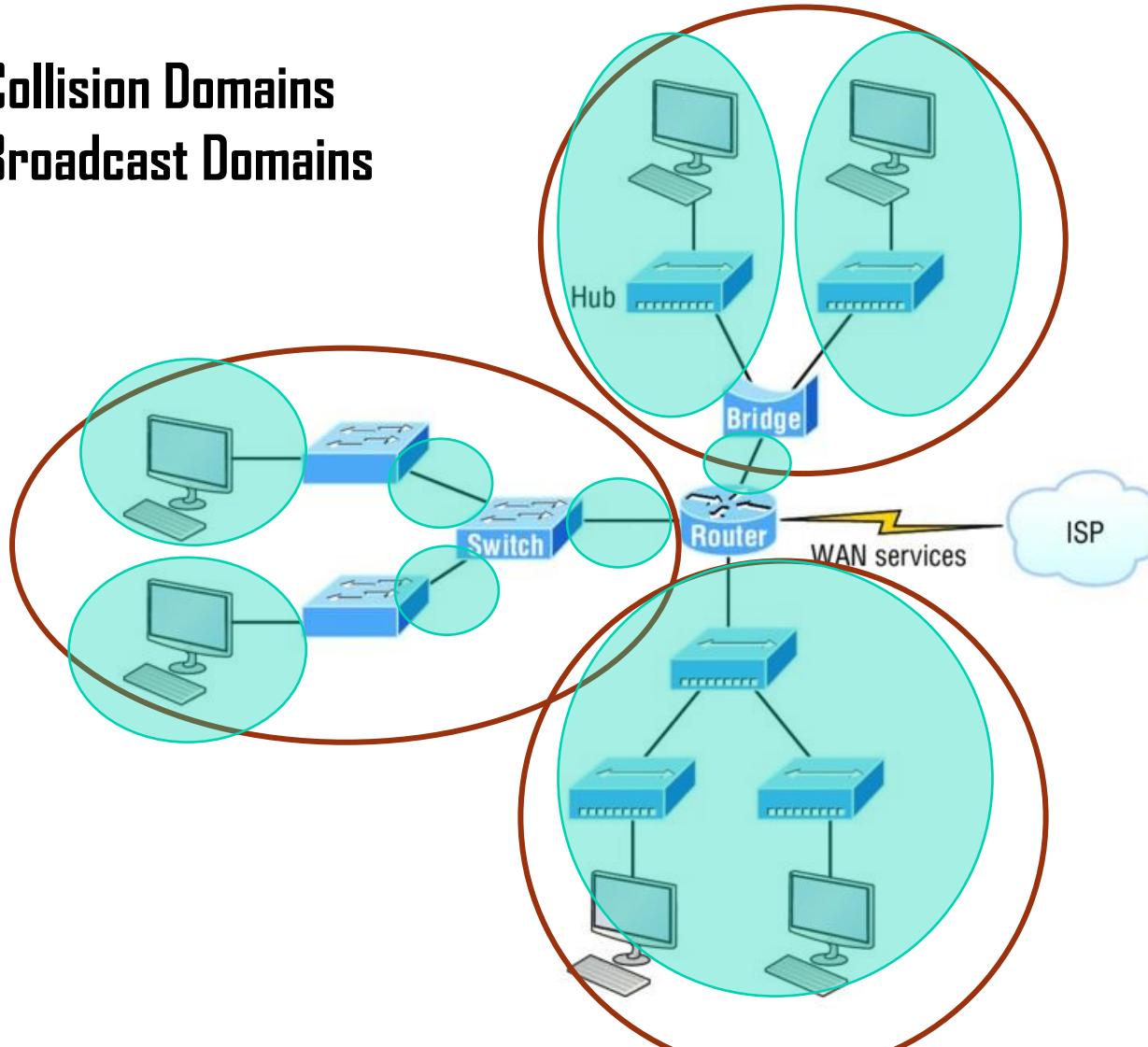
# LET'S FIND NUMBER OF BROADCAST DOMAINS AND COLLISIONS DOMAINS



9 Collision Domains



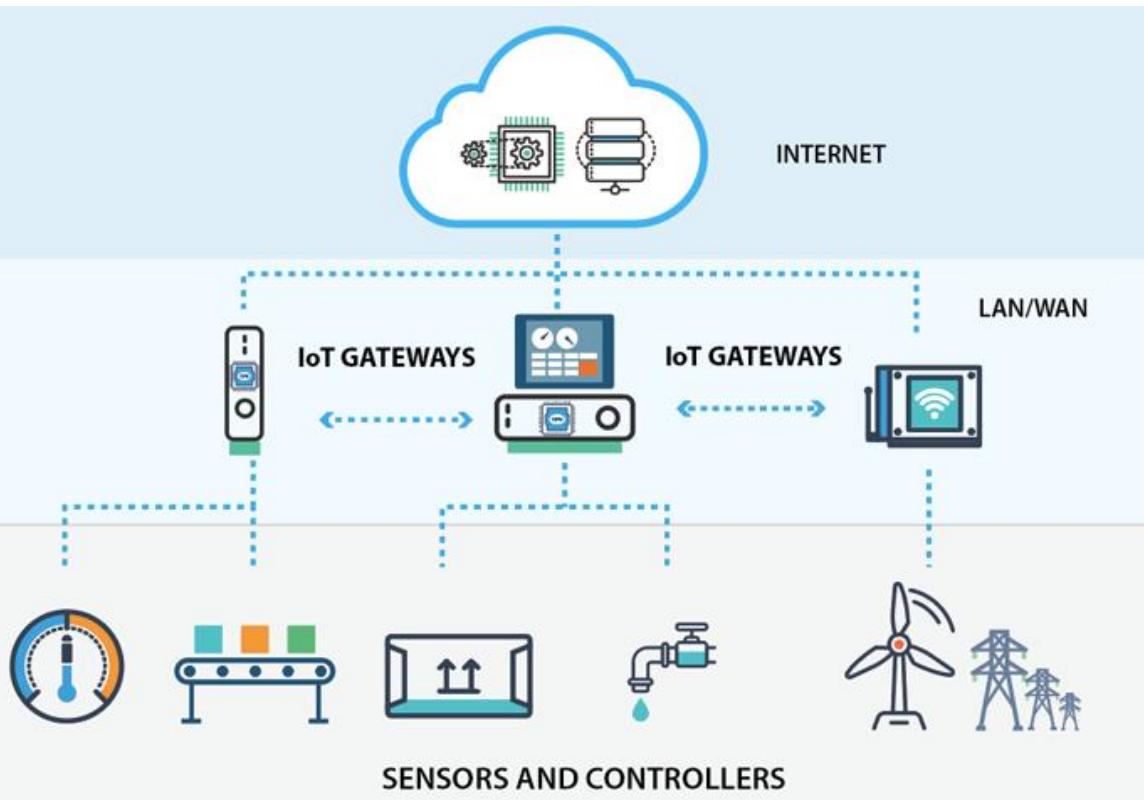
3 Broadcast Domains



# GATEWAY



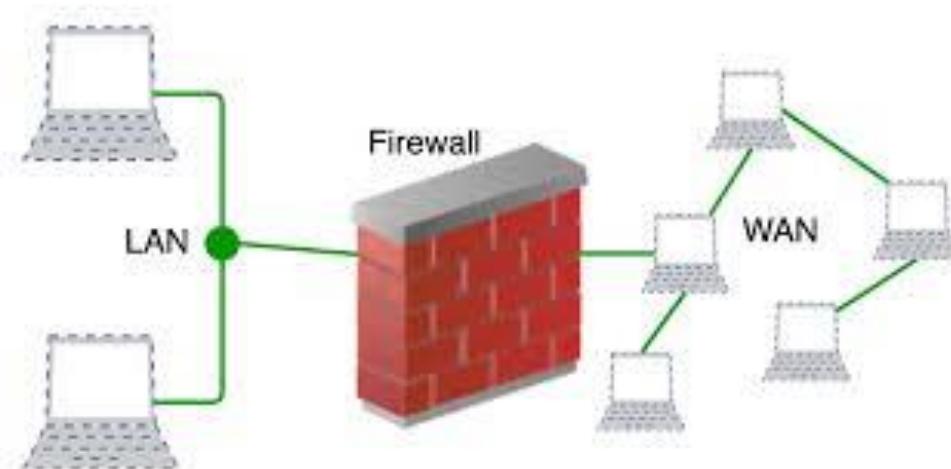
- A network device which can interconnect two networks that are using different protocols to communicate



<b>Application Layer</b>
✓ Message format, Human-Machine Interfaces
<b>Presentation Layer</b>
✓ Coding into 1s and 0s; encryption, compression
<b>Session Layer</b>
✓ Authentication, permissions, session restoration
<b>Transport Layer</b>
✓ End-to-end error control
<b>Network Layer</b>
✓ Network addressing; routing or switching
<b>Data Link Layer</b>
✓ Error detection, flow control on physical link
<b>Physical Layer</b>
✓ Bit stream: physical medium, method of representing bits

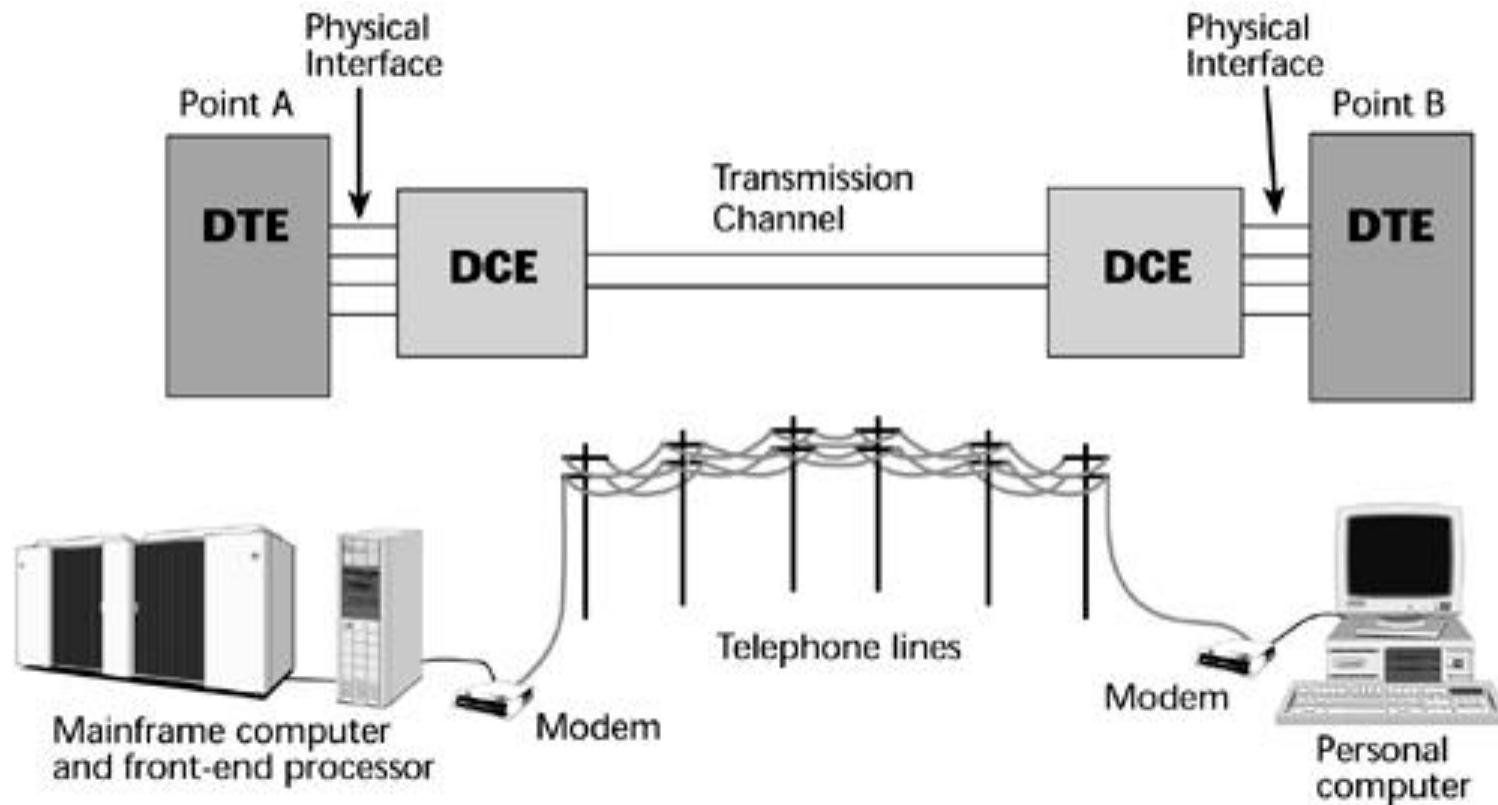
# OTHER DEVICES

- Firewall
- Servers
- IoT devices



<b>Application Layer</b>
✓ Message format, Human-Machine Interfaces
<b>Presentation Layer</b>
✓ Coding into 1s and 0s; encryption, compression
<b>Session Layer</b>
✓ Authentication, permissions, session restoration
<b>Transport Layer</b>
✓ End-to-end error control
<b>Network Layer</b>
✓ Network addressing; routing or switching
<b>Data Link Layer</b>
✓ Error detection, flow control on physical link
<b>Physical Layer</b>
✓ Bit stream: physical medium, method of representing bits

# DATA COMMUNICATION MODEL



- DTE – Data Terminal Equipment (Ex: Computer, Printer, Fax machine)
- DCE – Data Circuit-Terminating Equipment (Ex: Modem)



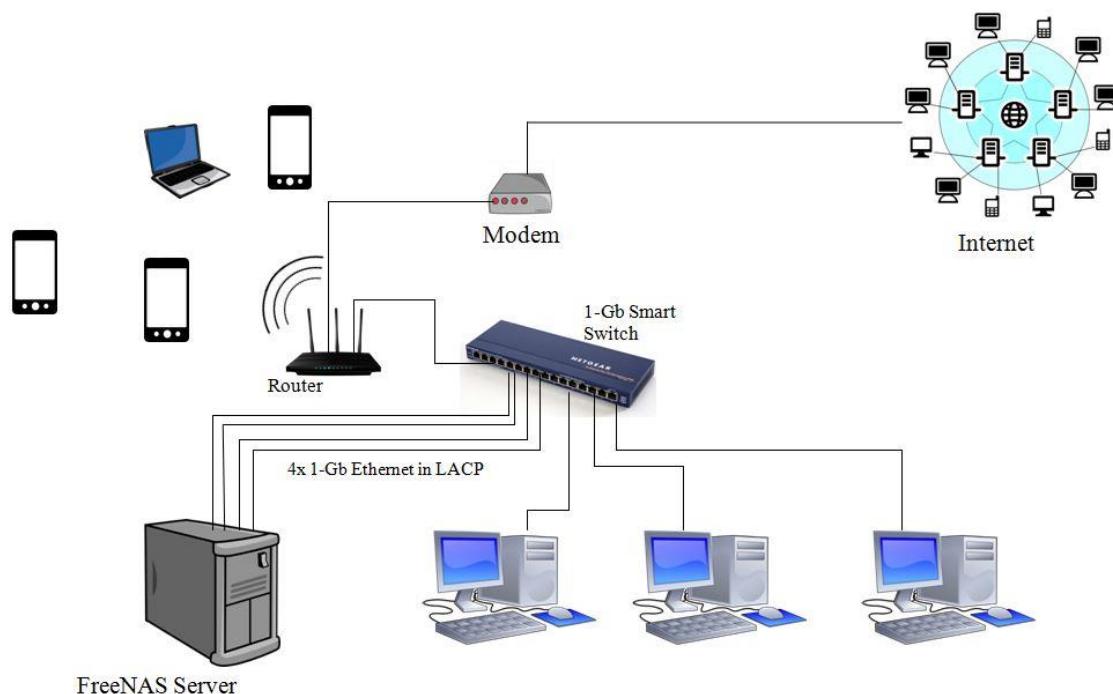
# DATA COMMUNICATION NETWORKS NEEDS

- Devices
  - To communicate with one another
- Medium
  - Connects devices together
- Messages
  - Information over media
- Rules
  - Govern how messages flow across networks

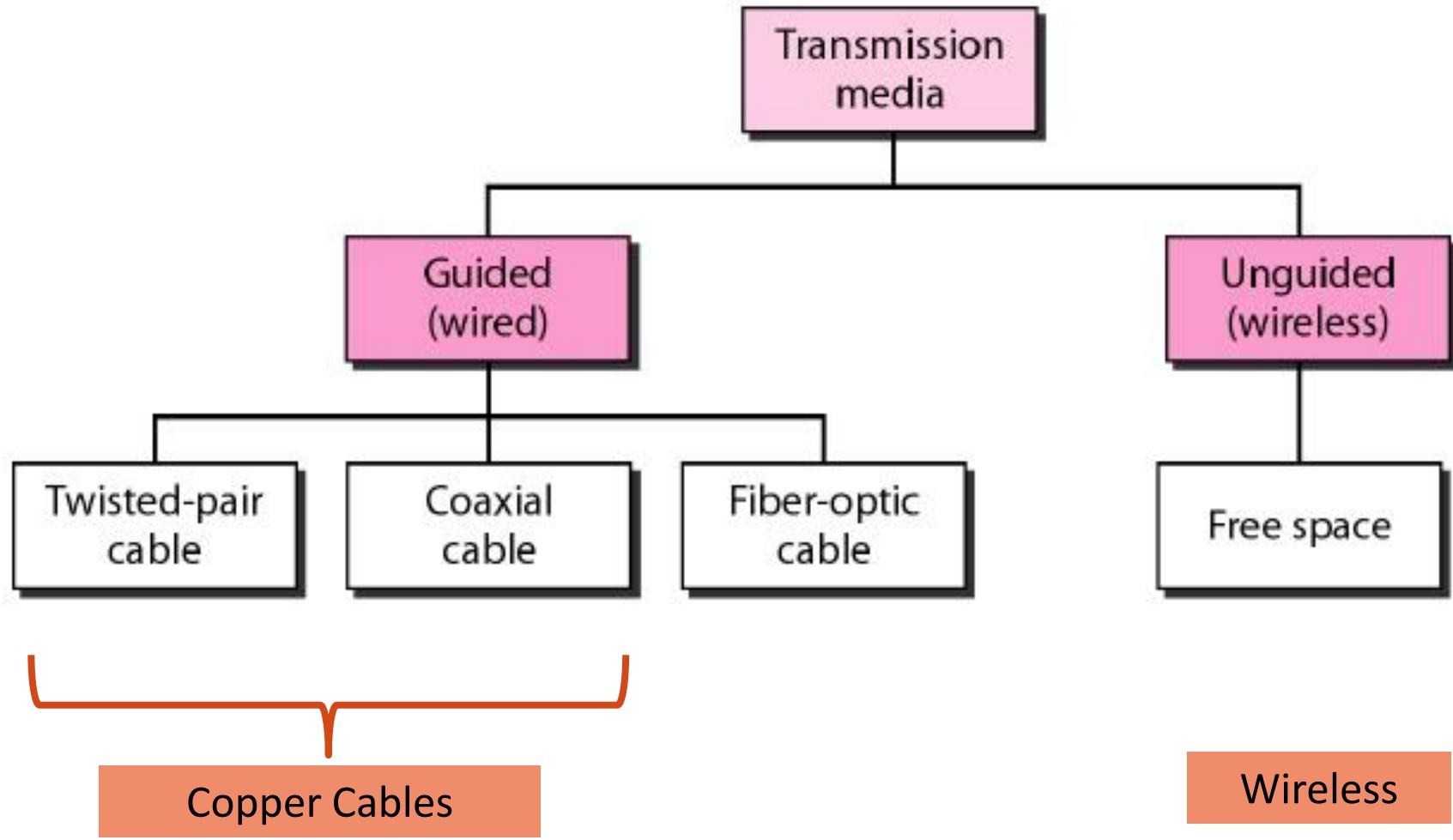


# TRANSMISSION MEDIA

- A **transmission medium** (plural transmission media) is a **material substance** (solid, liquid, gas, or plasma) **that can propagate energy waves**  
– *from Wikipedia.*
- We use transmission media to propagate the signals carrying some information (data) from a sender to a receiver



# TRANSMISSION MEDIA CONT.



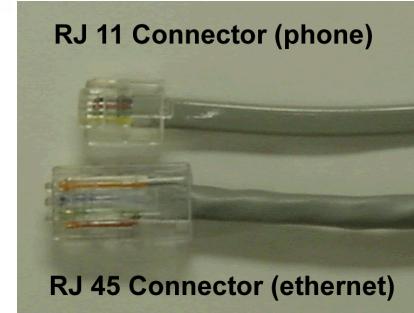
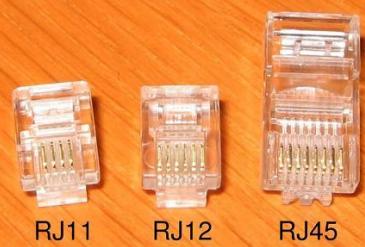
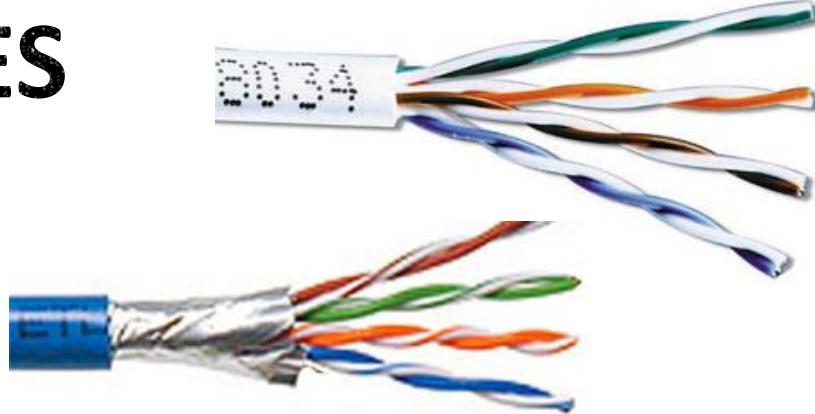
# TWISTED-PAIR CABLES

- The least expensive and most widely used copper based guided media

- A twisted-pair consist of two insulated copper wires arranged in a regular spiral pattern

- Twisted-pair cables come in **two variants**,

- ✓ Unshielded Twisted Pair (UTP)
  - ✓ Shielded Twisted Pair (STP)



- Most widely used connectors for twisted-pair cables are Registered Jack (RJ) connectors

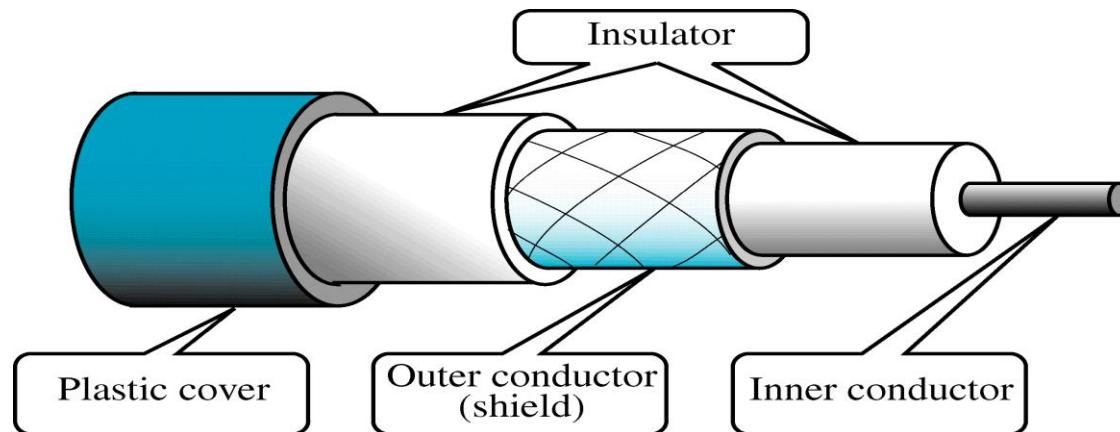


# COAXIAL CABLES

- Coaxial cables were **widely used in past, but now obsolete**, to bus topology local area networks



- A coaxial cable consist of
  - two conductors as same as twisted-pair
  - but these two conductors are arranged in a different manner



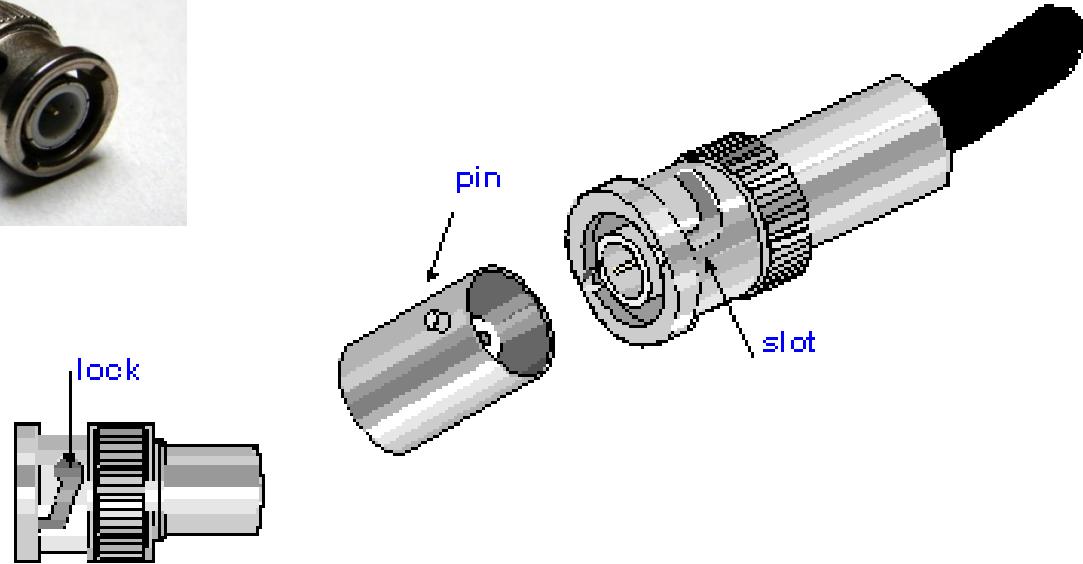
# COAXIAL CABLES CONT.

- Coaxial cables are widely used in other applications as well,
  - ✓ Television distribution (i.e. cable TV)
  - ✓ Long-distance telephone transmission
  - ✓ Antenna cables used with Televisions



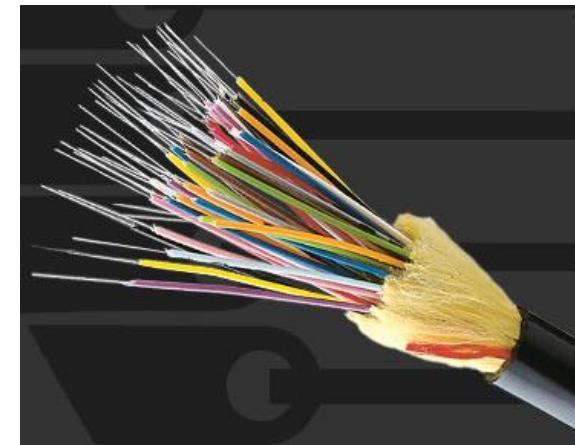
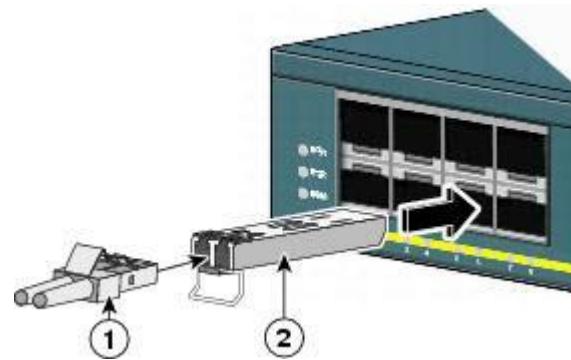
# COAXIAL CABLES CONT.

- Most widely used connectors for coaxial cables were Bayonet Network Connectors (BNC)

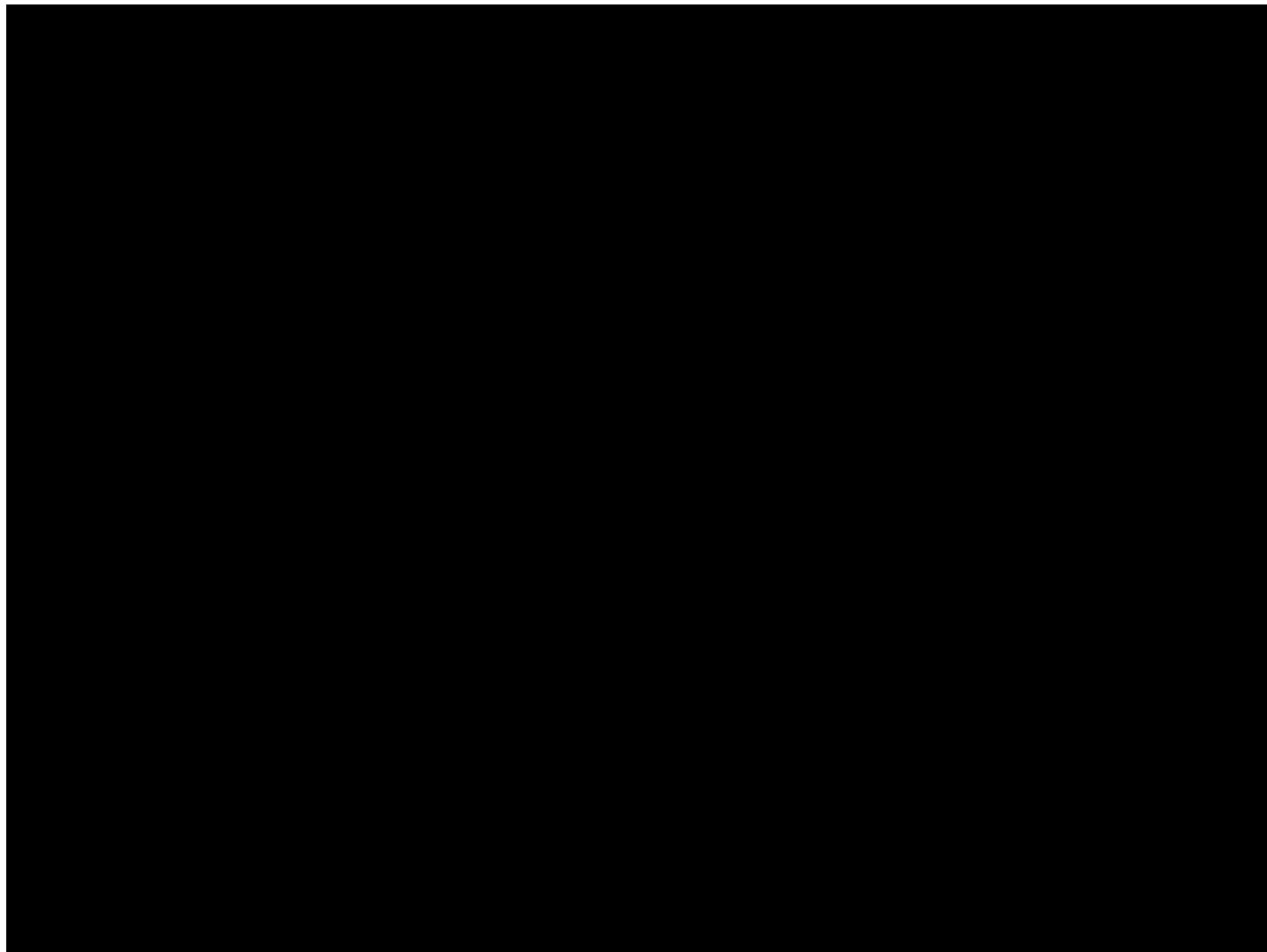


# FIBER OPTIC CABLES

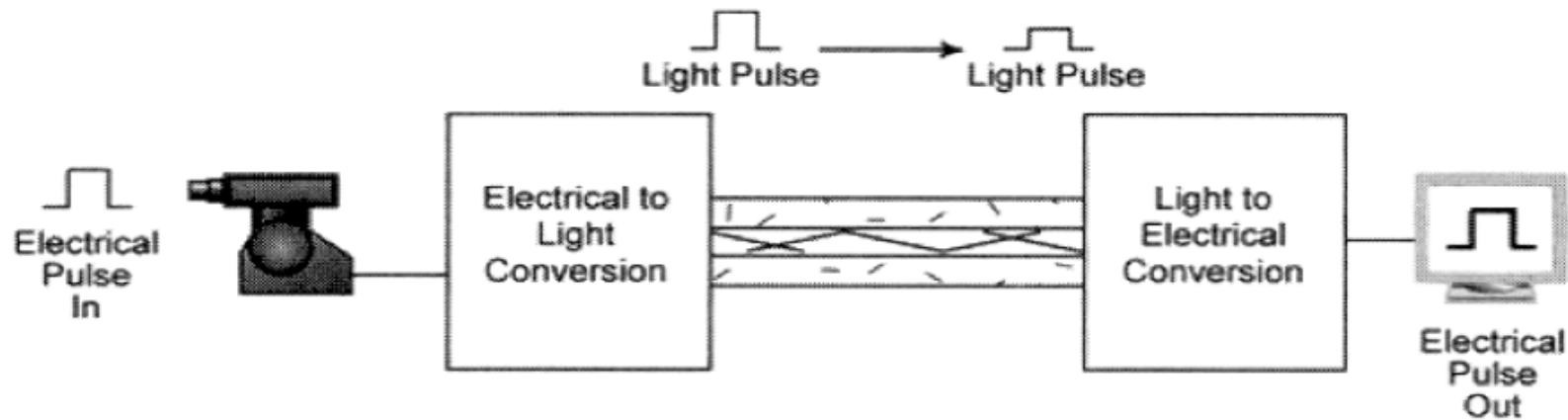
- Optical fibers are made of glass or plastic material and the signals are transmitted as light rays
- **Optical fibers operate** based on an optical phenomena known as **total internal reflection**



# **OPTICAL TRANSMISSION OF DATA**



# OPTICAL TRANSMISSION OF DATA



- Electrical signal generated by the transmitter is converted to a light pulse and this light pulse is transmitted through the optical fiber
- At the receiver, the light pulse is converted back to an electrical signal



# SUMMERY OF GUIDED MEDIA

- Three Types of Guided Media

- 1. -----

- Advantages
    - -----
    - Disadvantages
    - -----

- 2. -----

- Advantages
    - -----
    - Disadvantages
    - -----

- 3. -----

- Advantages
    - -----
    - Disadvantages
    - -----



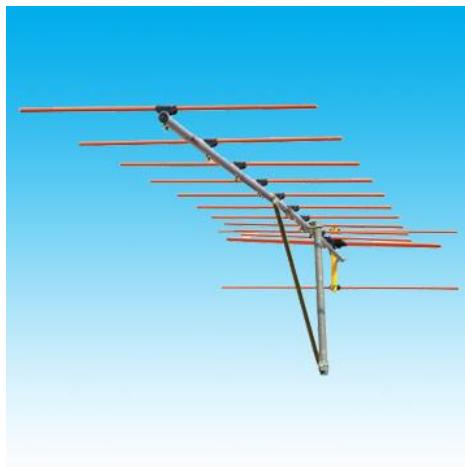
# UNGUIDED MEDIA

- **Unguided media**, or more commonly referred to as wireless communication, requires **transportation of electromagnetic waves** without the usage of a physical conductor
- The signals are **transmitted into free space**
- In order to transmit signals into free space a special equipment called **antenna is required**



# ANTENNA TYPES

Yagi Antenna



Parabolic Antenna



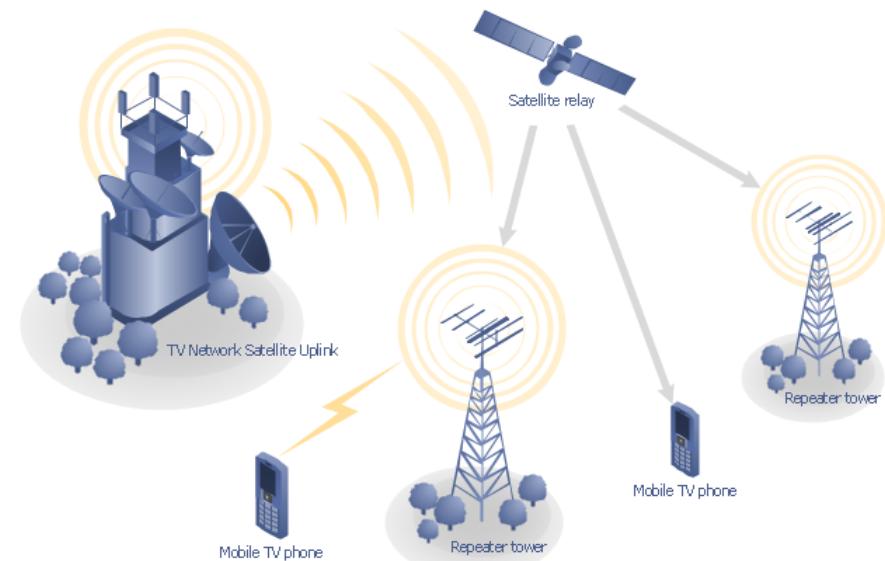
Omnidirectional  
Antenna



# TYPES OF WIRELESS COMMUNICATION

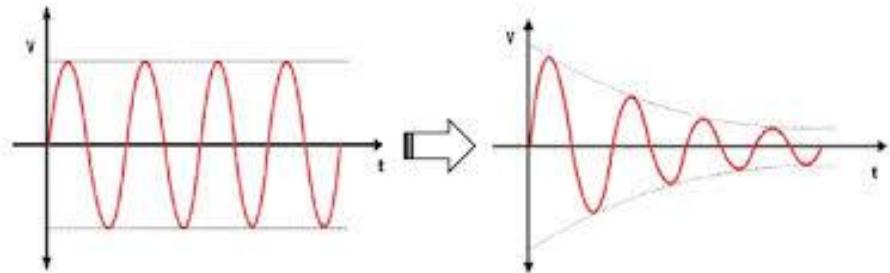
- Wireless transmission of signals can be divided into several categories based on the range of frequencies used by each type,

- ✓ Broadcast radio
- ✓ E.g. \_\_\_\_\_
- ✓ Terrestrial microwave
- ✓ E.g. \_\_\_\_\_
- ✓ Satellite microwave
- ✓ E.g. \_\_\_\_\_

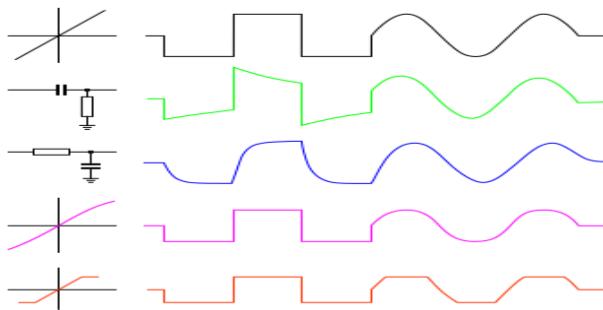


# IMPORTANT FACTORS

- Attenuation: Attenuation or, in some contexts, extinction is the gradual loss of flux intensity through a medium (Wiki)



- Distortion: Distortion, in acoustics and electronics, any change in a signal that alters the basic waveform or the relationship between various frequency components (Britanica)



- Noise: Noise is an unwanted disturbance in an electrical signal.





*That's all Folks!*

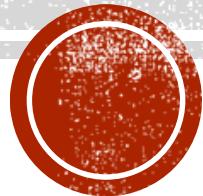


# Questions



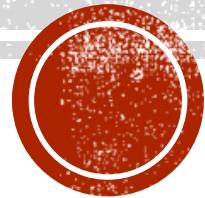
# **INTRODUCTION TO COMPUTER SYSTEMS**

## **COMPUTER NETWORKS**



# **LECTURE 10**

# **ADDRESSING IN A NETWORK**

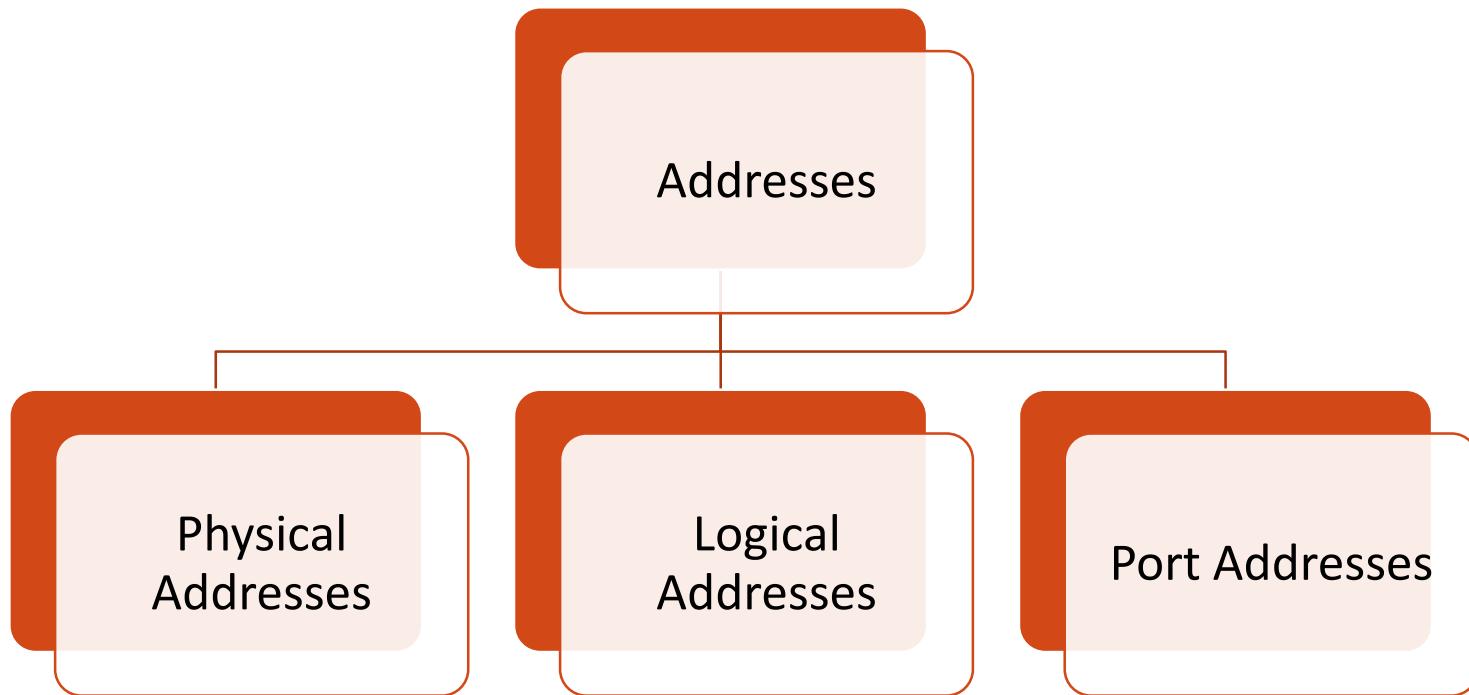


# LECTURE OUTLINE

- Addresses used in a Network
- IPv4 Addressing
- Classful Addresses
- Public vs Private Addresses



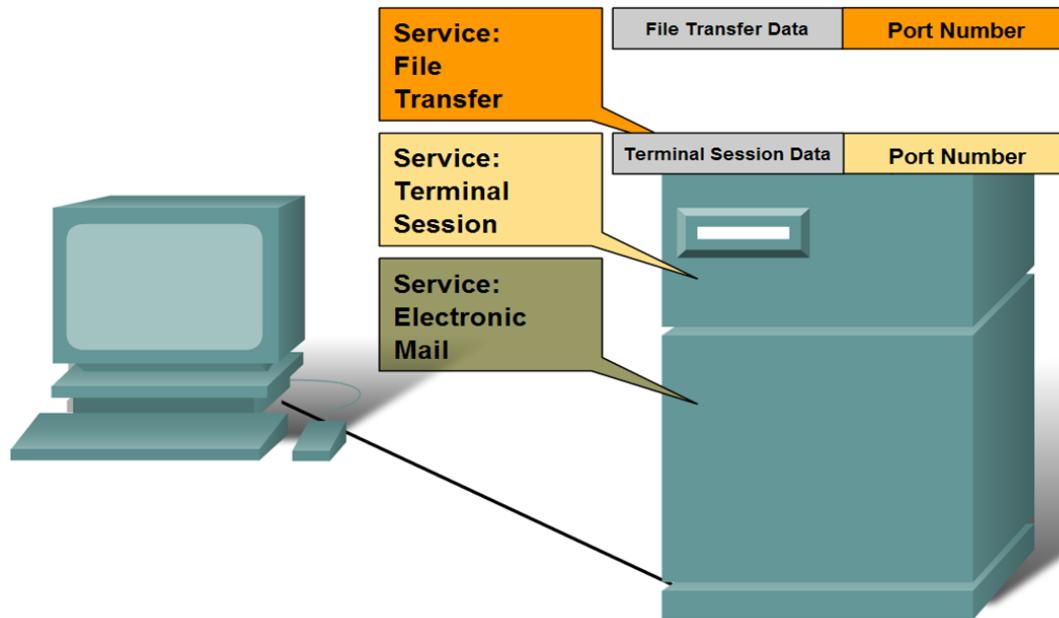
# ADDRESSES USED IN COMPUTER NETWORKS



# TRANSPORT LAYER ADDRESS: PORT ADDRESS

- Used to identify the **source** and **destination processes** for communication
- Ex: Port address

At the end device, the service port number directs the data to the correct conversation.



What is the layer of IOS-OSI Ref. model the PORT addresses are introduced?



# **PHYSICAL ADDRESS**

## **DATA LINK LAYER ADDRESS: MAC ADDRESS**

**00-50-56-C0-00-01**

Organizationally Unique  
Identifier (OUI)

Network Interface Controller  
(NIC) Specific

Three  
Bytes

Three  
Bytes

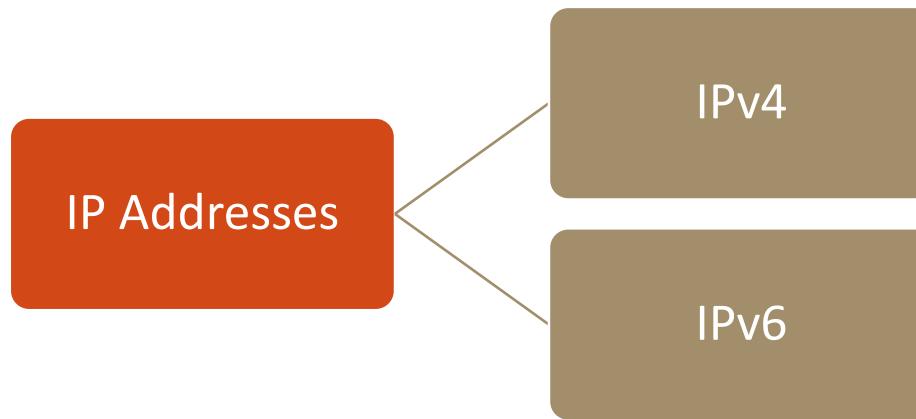
**What is the layer of IOS-OSI Ref. model the PORT addresses are introduced?**



# LOGICAL ADDRESS

## NETWORK LAYER ADDRESS: IP ADDRESS

- There are **two major versions** of IP addresses



- IP version 4 (**IPv4**) address is **32 bits long** (i.e. **4 bytes**)
- IP version 6 (**IPv6**) address is **128 bits long** (i.e. **16 bytes**)



# IP VERSION 4 (IPV4)

**172 . 16 . 254 . 1**



10101100 .00010000 .11111110 .00000001



One byte = Eight bits



Thirty-two bits ( $4 \times 8$ ), or 4 bytes



# EXERCISES

- Change the following IP addresses from binary notation to dotted decimal notation.

10000001 00001011 00001011 11101111

- Change the following IP addresses from dotted decimal notation to binary notation.

111.56.45.78

- Find the error, if any, in the following IP address:

111.56.045.78



# CLASSFUL ADDRESSING

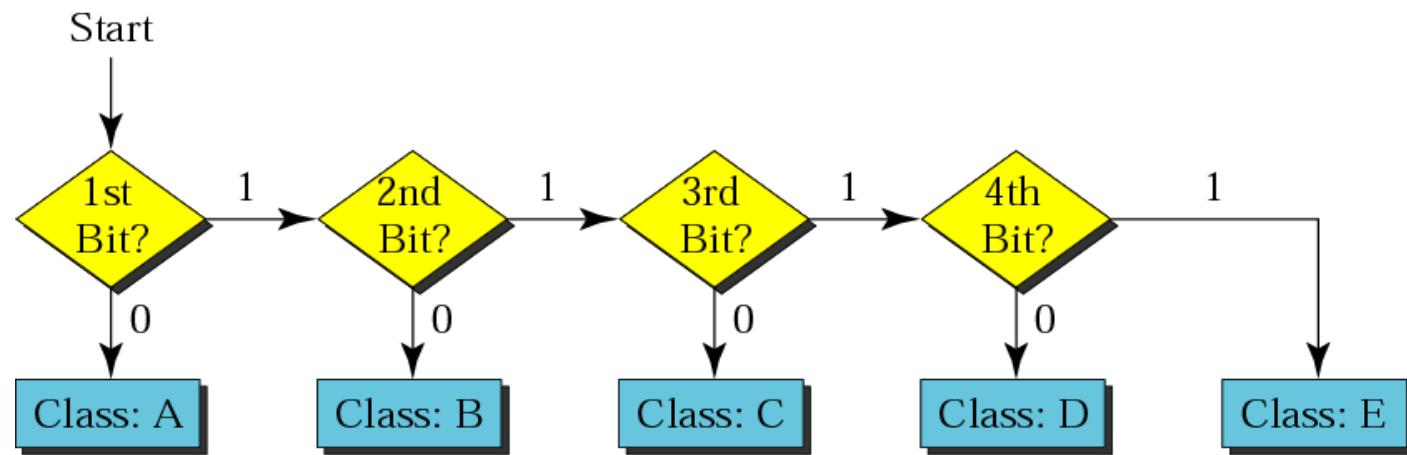
- When IP addressing was first introduced, all **IPv4 addresses** were divided into **5 classes**.

Class	Usage
Class A	General purpose
Class B	General purpose
Class C	General purpose
Class D	Multicasting
Class E	Reserved for future use



# FINDING THE CLASS IN BINARY NOTATION

	First byte	Second byte	Third byte	Fourth byte
Class A	<b>0</b>			
Class B	<b>10</b>			
Class C	<b>110</b>			
Class D	<b>1110</b>			
Class E	<b>1111</b>			



# EXERCISES

Find the class of each address:

- 00000001 00001011 00001011 11101111 A
- 11000001 10000011 00011011 11111111 C
- 11011111 10110000 00011111 01011101 C
- 11110111 11110011 10000111 11011101 E
- 10101111 11000000 11110000 00011101 B



# FINDING THE CLASS IN DECIMAL NOTATION

	First byte	Second byte	Third byte	Fourth byte
Class A	<b>0 to 127</b>			
Class B	<b>128 to 191</b>			
Class C	<b>192 to 223</b>			
Class D	<b>224 to 239</b>			
Class E	<b>240 to 255</b>			



# EXERCISES

Find the class of each address:

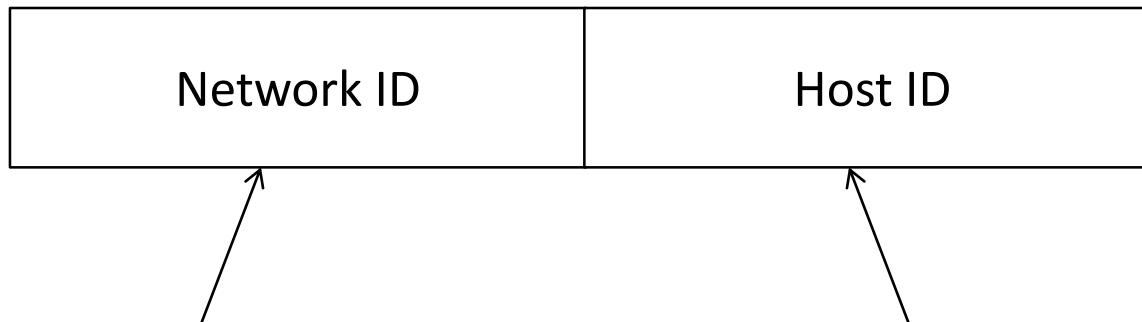
- ✓ 227.12.14.87 D
- ✓ 193.14.56.22 C
- ✓ 14.23.120.8 A
- ✓ 252.5.15.111 E
- ✓ 134.11.78.56 B



# **NETWORK ID (NET ID) AND HOST ID**

- When an organization reserves a set of IP addresses from their ISP (Internet Service Provider) to use for the hosts used within that organization,

**each IP address consist of two parts**

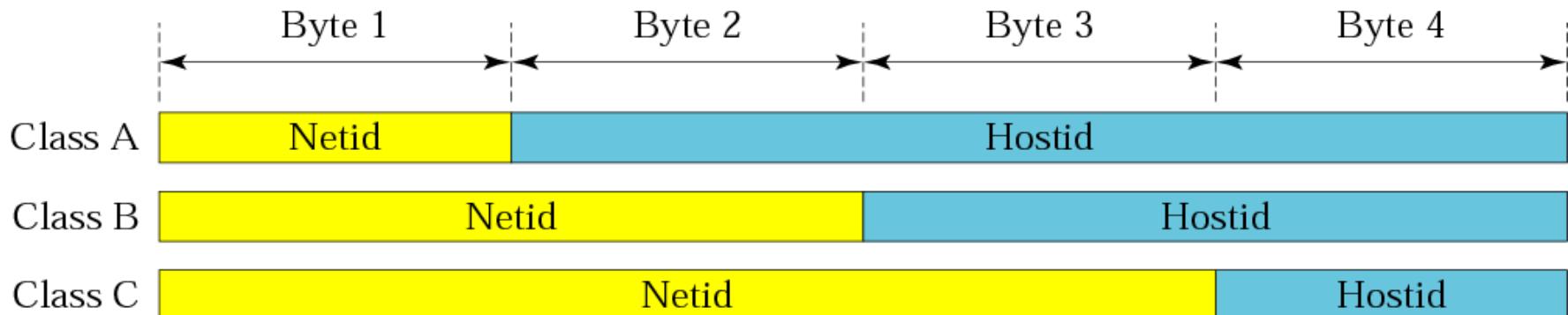


**Common in all the hosts within that organization**

**Unique to each host**



# NET ID AND HOST ID CONT.



Class	1 <sup>st</sup> Octet Decimal Range	1 <sup>st</sup> Octet High Order Bits	Network/Host ID (N=Network, H=Host)	Default Subnet Mask	Number of Networks	Hosts per Network (Usable Addresses)
A	1 – 126*	0	N.H.H.H	255.0.0.0	126 ( $2^7 - 2$ )	16,777,214 ( $2^{24} - 2$ )
B	128 – 191	10	N.N.H.H	255.255.0.0	16,382 ( $2^{14} - 2$ )	65,534 ( $2^{16} - 2$ )
C	192 – 223	110	N.N.N.H	255.255.255.0	2,097,150 ( $2^{21} - 2$ )	254 ( $2^8 - 2$ )
D	224 – 239	1110		Reserved for Multicasting		
E	240 – 254	1111		Experimental; used for research		

**Note:** Class A addresses 127.0.0.0 to 127.255.255.255 cannot be used and is reserved for loopback and diagnostic functions.



# MASKING CONCEPT

- Each LAN is owned by a particular organization, the net ID is what differentiates one LAN from another in Internet terms
- Finding the net ID is extremely important because net ID is used by routers to route the packets from one LAN to another LAN over the Internet

```
Connection-specific DNS Suffix . . . . . slit
Description . . . . . : Marvell AVASTAR Wireless-AC Network Controller
Physical Address. . . . . : 98-5F-D3-
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::803f:d5c1:30d4:6018%21(Preferred)
IPv4 Address. . . . . : 172.29.0.62(Preferred)
Subnet Mask . . . . . : 255.255.248.0
Lease Obtained. . . . . :
Lease Expires . . . . . :
Default Gateway . . . . . :
DHCP Server . . . . . :
DHCPv6 IAID . . . . . : 345530323
DHCPv6 Client DUID. . . . . : 00-01-00-01-23-E2-A7-10-9C-EB-E8-0A-E3-F5
DNS Servers . . . . . : 172.16.
                           172.16.
```

NET1 ID  
172.16.0.0

NET2 ID  
192.168.0.0

```
Connection-specific DNS Suffix . . . . . :
Description . . . . . :
Physical Address. . . . . : 98-5F-D3-
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::803f:d5c1:30d4:6018%21(Preferred)
IPv4 Address. . . . . : 192.168.8.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Thursday, May 9, 2019 10:27:25 AM
Lease Expires . . . . . : Friday, May 10, 2019 10:27:25 AM
Default Gateway . . . . . : fe80::a257:e3ff:fe0a:73c2%21

DHCP Server . . . . . :
DHCPv6 IAID . . . . . : 345530323
DHCPv6 Client DUID. . . . . : 00-01-00-01-23-E2-A7-10-9C-EB-E8-0A-E3-F5
DNS Servers . . . . . : fe80::a257:e3ff:fe0a:73c2%21

NetBIOS over Tcpip. . . . . : Enabled
```

# MASKING CONCEPT CONT.

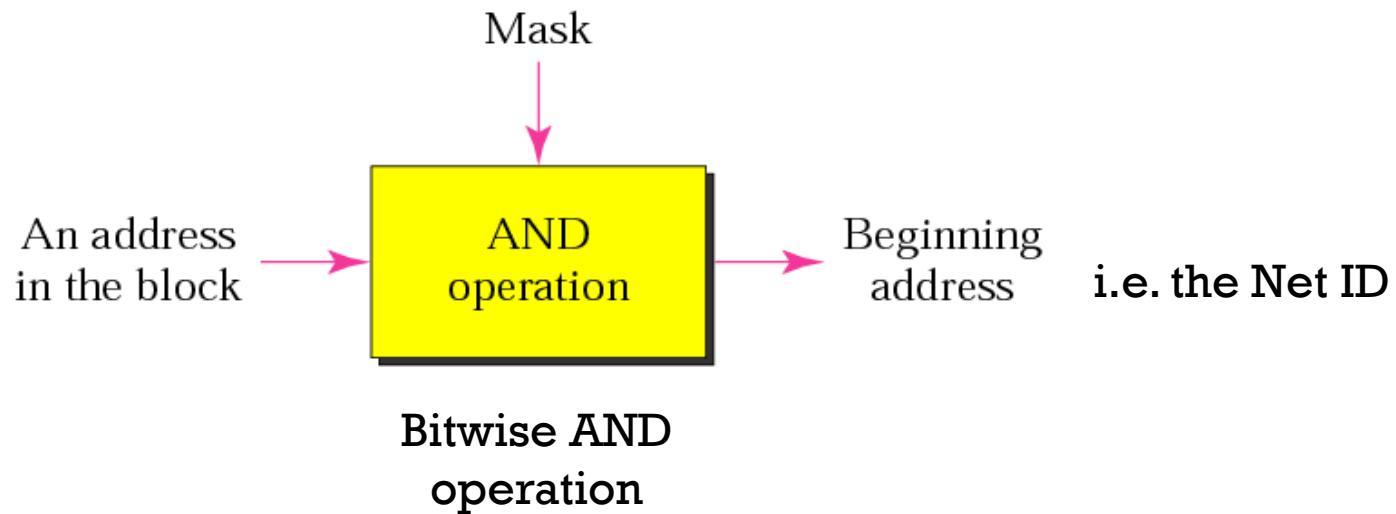
## DEFAULT MASKS

Class	Mask in binary	Mask in dotted decimal	Mask in slash (/) notation
A	<b>11111111 00000000 00000000 00000000</b>	<b>255.0.0.0</b>	/8
B	<b>11111111 11111111 00000000 00000000</b>	<b>255.255.0.0</b>	/16
C	<b>11111111 11111111 11111111 00000000</b>	<b>255.255.255.0</b>	/24



# MASKING CONCEPT CONT.

- Although we humans can easily interpret the net ID of a given classful IP address, how does a router calculate the net ID?
- For this we use the concept of masking



# EXERCISES

- Find the subnet mask for following addresses

- ✓ 35.56.7.91 → 255.0.0.0
  - ✓ 68.87.34.68 255.0.0.0
  - ✓ 140.30.1.21 255.255.0.0
  - ✓ 210.50.60.1 255.255.255.0
  - ✓ 188.1.1.1 255.255.0.0
- ✓ Then Find the Network address for above addresses

Address: 35.56.7.91 00100011.0011000.00000111.01011011

Netmask: 255.0.0.0 = 8 1111111.00000000.00000000.00000000

=> Network: 35.0.0.0/8 00100011.00000000.00000000.00000000 (Class A)

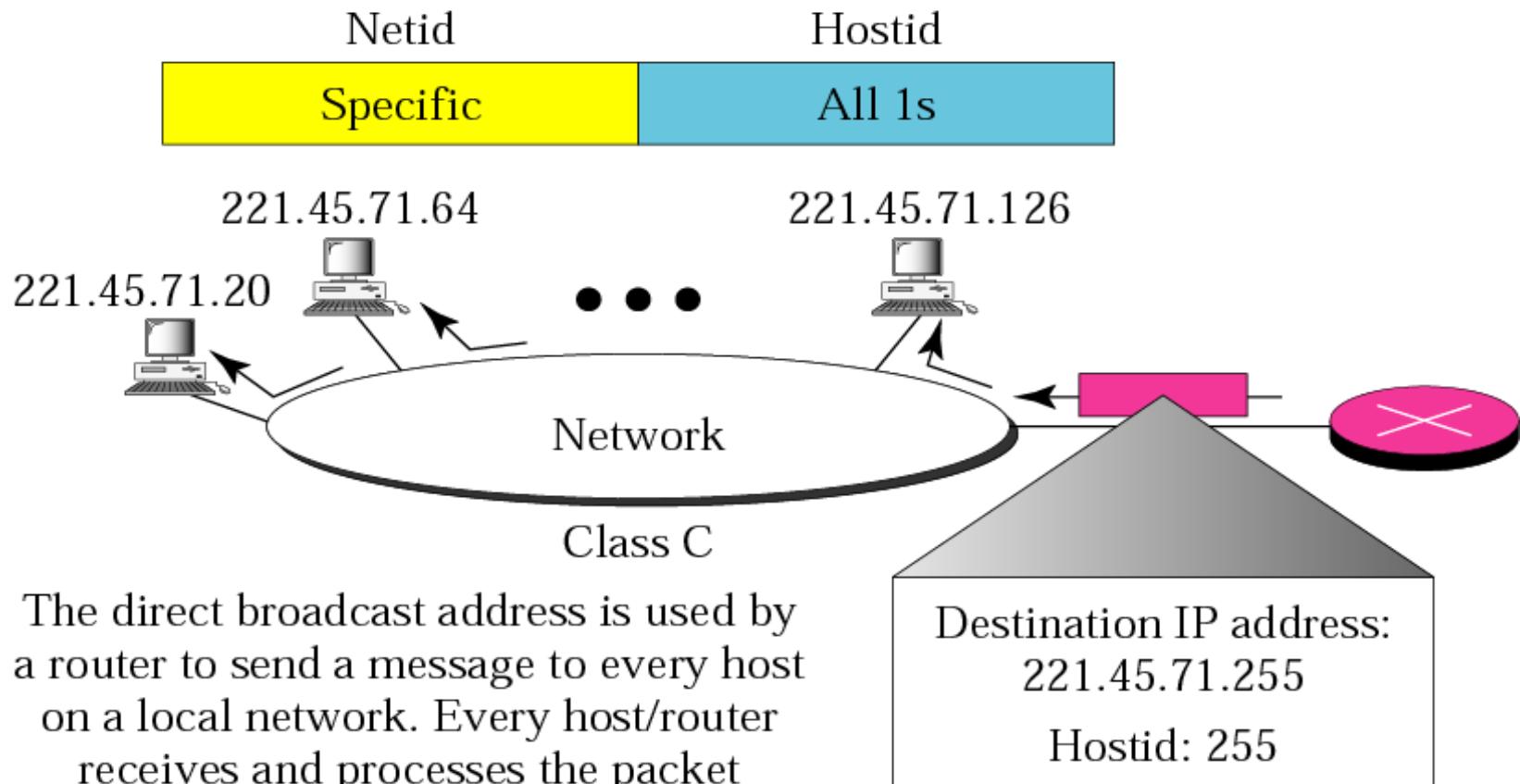


# SPECIAL IPV4 ADDRESSES

<i>Special Address</i>	<i>Netid</i>	<i>Hostid</i>	<i>Source or Destination</i>
Network address	Specific	All 0s	None
Direct broadcast address	Specific	All 1s	Destination
<b>Limited broadcast address</b>	All 1s	All 1s	Destination
<b>This host on this network</b>	All 0s	All 0s	Source
<b>Specific host on this network</b>	All 0s	Specific	Destination
Loopback address	127	Any	Destination



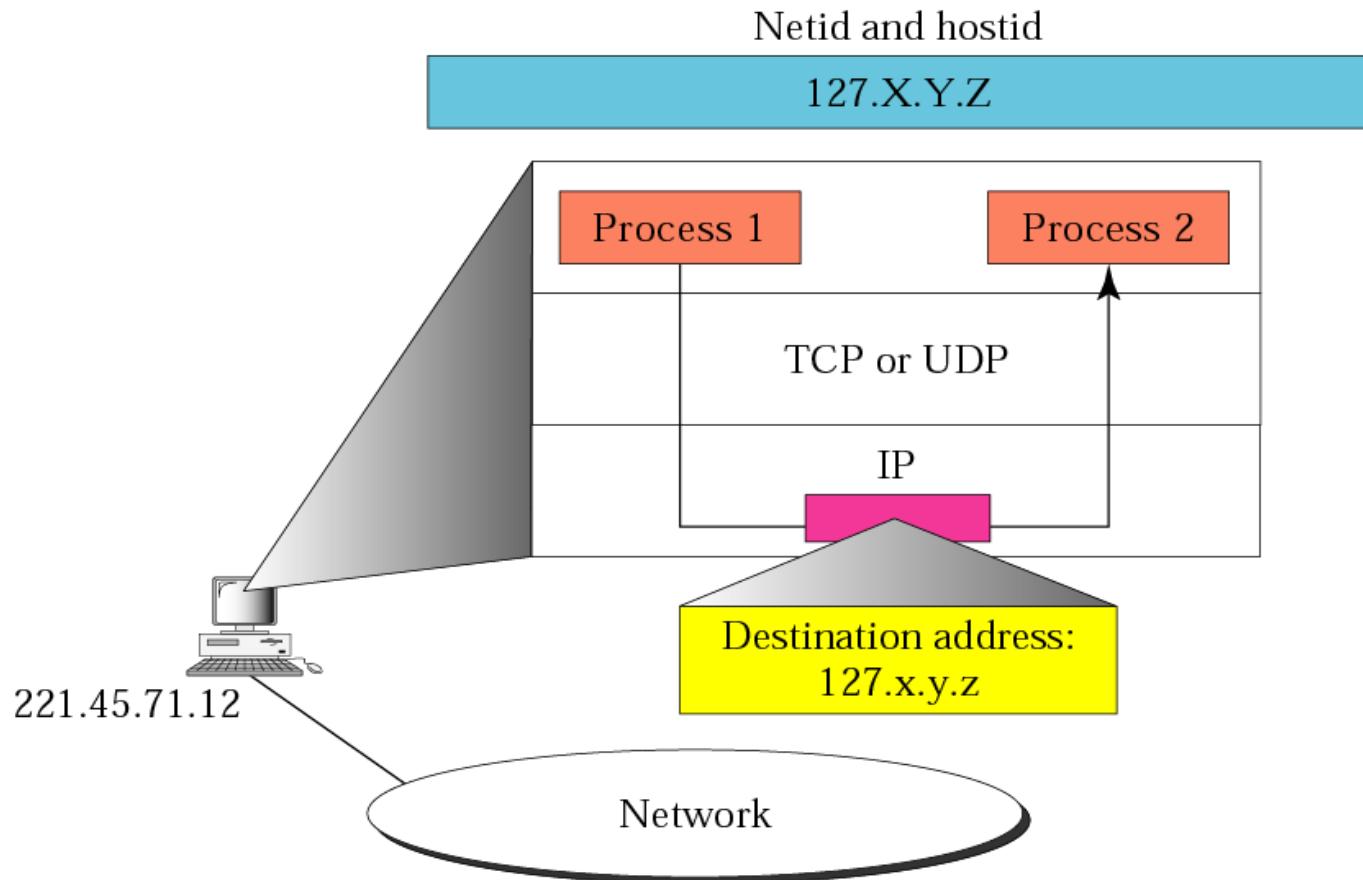
# DIRECT BROADCAST ADDRESS (BROADCAST ADDRESS)



The direct broadcast address is used by a router to send a message to every host on a local network. Every host/router receives and processes the packet with a direct broadcast address.



# LOOPBACK ADDRESS



A packet with a loopback address will not reach the network.

- The most widely used loopback address is **127.0.0.1**



# EXERCISES

- For following address, find the
  - network address
  - Subnet mask
  - Broadcast address
  - 1<sup>st</sup> usable host ip address
  - Last usable host ip address

✓ 23.56.7.91

✓ 72.87.34.10

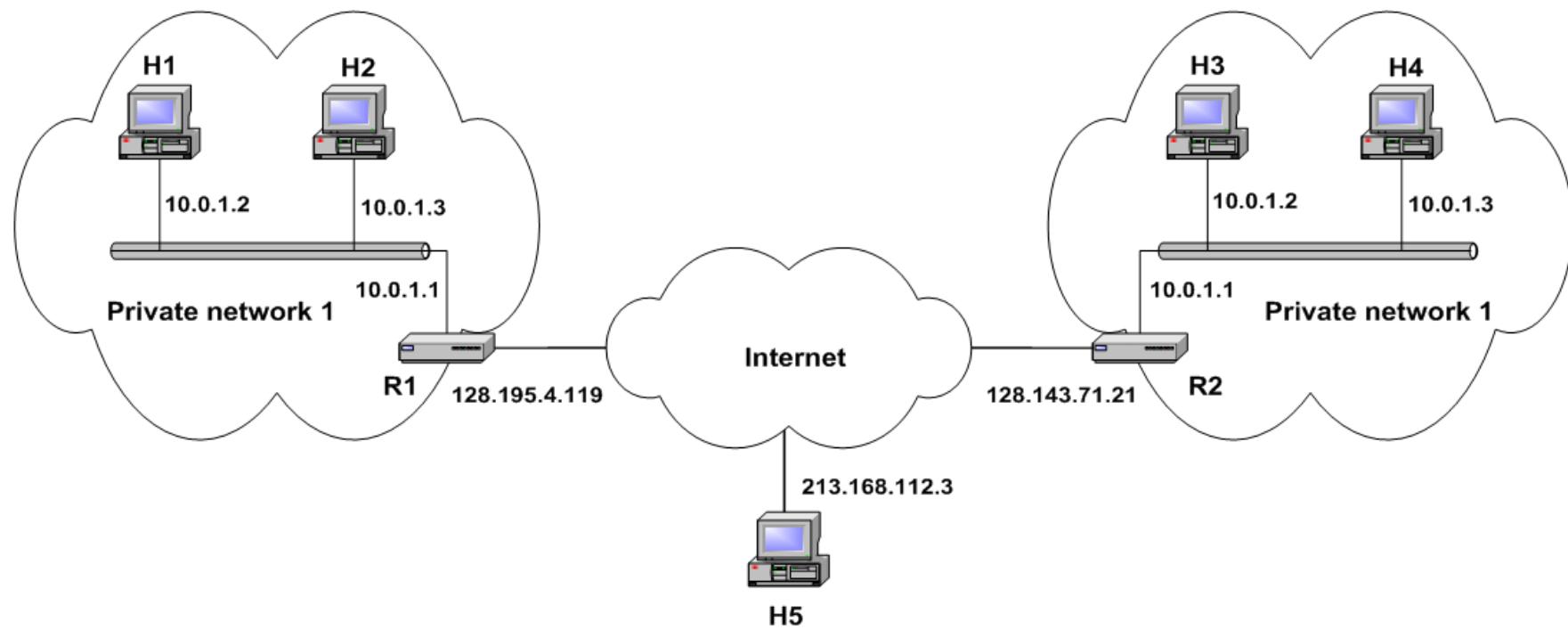
✓ 130.10.1.21

✓ 200.50.60.1

✓ 198.1.1.1



# PUBLIC ADDRESSES VS PRIVATE ADDRESSES



# IPV4 PRIVATE ADDRESS RANGES

- Following ranges are reserved to be used in Local Area Networks for private addresses.

<i>Class</i>	<i>Netids</i>	<i>Blocks</i>
A	10.0.0	1
B	172.16 to 172.31	16
C	192.168.0 to 192.168.255	256

- Remember: You cannot use these ranges for machines/interfaces that are directly connected to Internet.



# PROBLEMS WITH CLASSFUL ADDRESSING

- Class A and B are too large for typical organizations and many IP addresses will not be used and wasted.
- Class C is not enough for most organizations resulting the reservation of at least a Class B address range for the organization.
- The end result is that, the available IP addresses are depleting at an alarming rate and soon there will be no more IP addresses.



# SOLUTIONS

- ✓ **Short Term:**

- Subnetting
- Classless Addressing

- ✓ **Long Term: IPv6**



# ADDRESS ALLOCATION

- Who has the authority to provide you an IP address?
  - ✓ Internet Service Provider (e.g. Sri Lanka Telecom).
- Who has the authority to reserve IP addresses to each ISP?
  - ✓ A global authority called the Internet Corporation for Assigned Names and Addresses (ICANN).





*That's all Folks!*

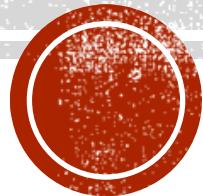


# Questions



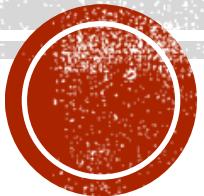
# **INTRODUCTION TO COMPUTER SYSTEMS**

## **COMPUTER NETWORKS**



# **LECTURE 11**

## **DESIGN A NETWORK**



# LECTURE OUTLINE

- Design Considerations for a Small Network
- Scale to a Large Network
- Threats to the Network
- Keeping the Network Safe



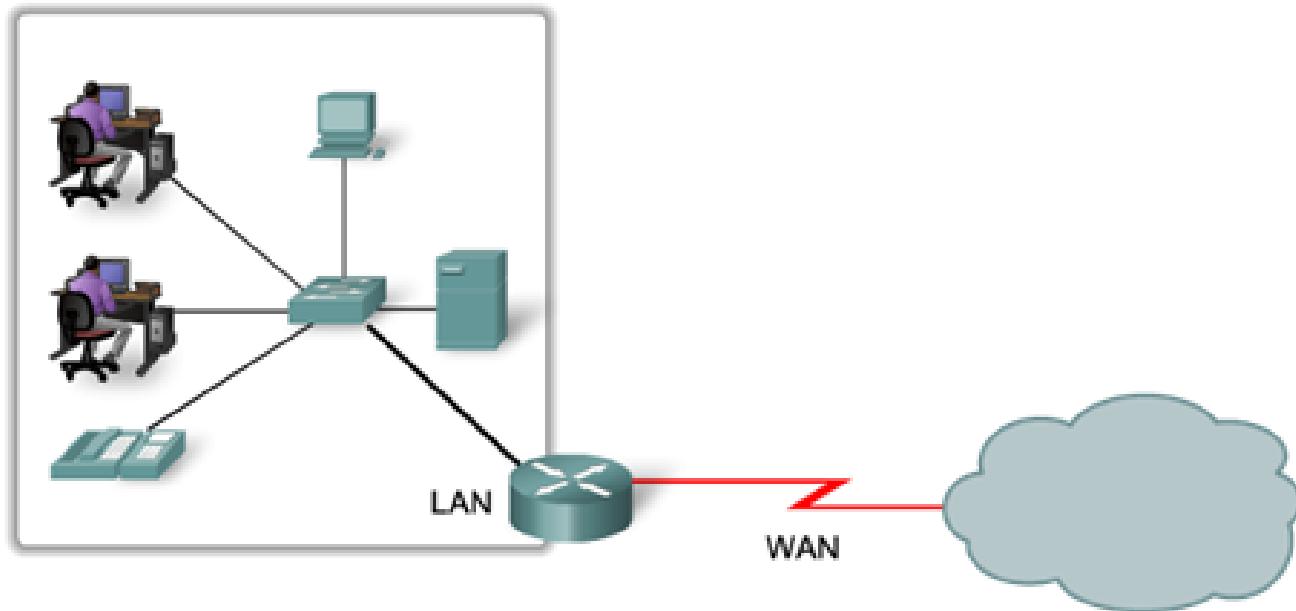
# DESIGN AND CREATE A SMALL NETWORK

- ✓ Topologies
- ✓ Device selection
- ✓ IP addressing
- ✓ Redundancy
- ✓ Design considerations
- ✓ Common applications
- ✓ Common protocols
- ✓ Real time applications
- ✓ Scaling the small network
- ✓ Protocol analysis



# TOPOLOGIES

Typical Small Network Topology



# DEVICE SELECTION

Factors to be considered when selecting intermediate devices.



COST



PORTS



SPEED



EXPANDABLE/ MODULAR



MANAGEABLE



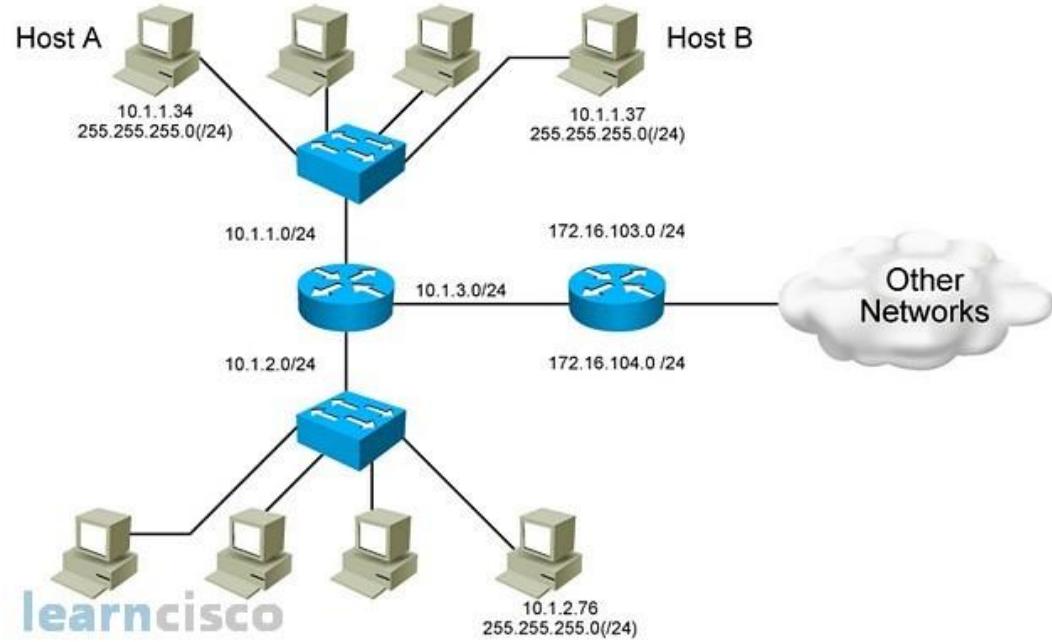
# IP ADDRESSING

- IP addressing scheme should be planned, documented and maintained based on the type of devices receiving the address.
- Examples of devices that will be part of the IP design:
  - End devices for users
  - Servers and peripherals
  - Hosts that are accessible from the Internet
  - Intermediary devices



# IP ADDRESSING CONT.

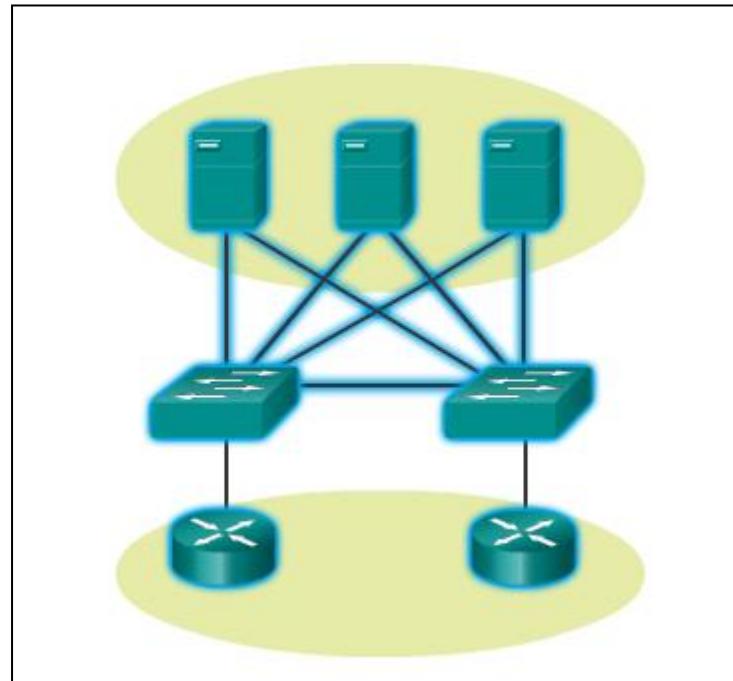
- Planned IP schemes help the administrator:
  - Track devices and troubleshoot
  - Control access to resources



# REDUNDANCY

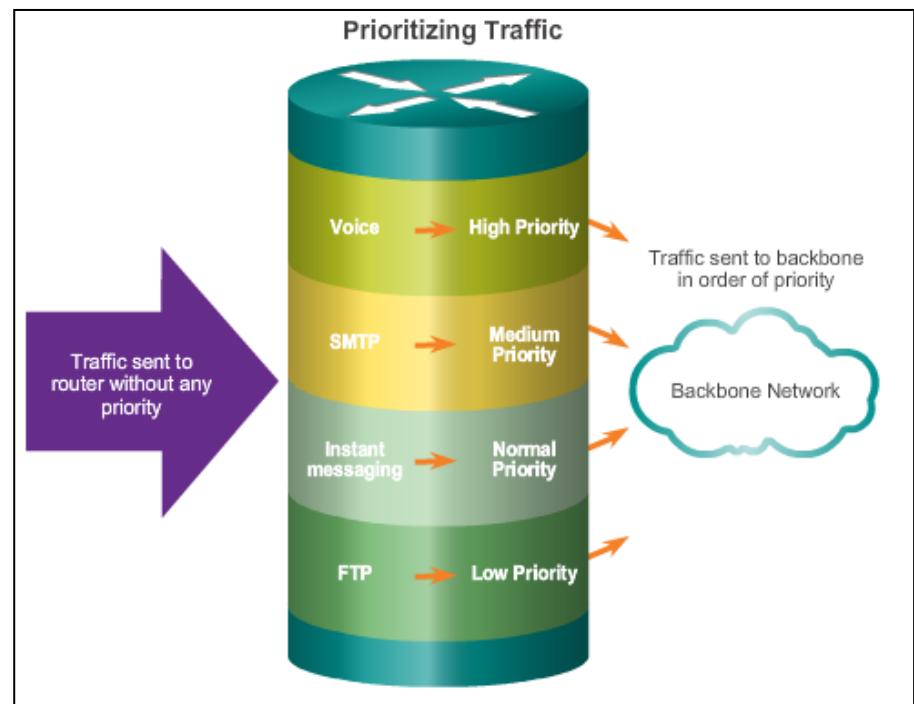
- Redundancy helps to eliminate single points of failure.
- Improves the reliability of the network.

Redundancy to a Server Farm



# DESIGN CONSIDERATIONS

- The following should be included in the network design:
  - Secure file and mail servers in a centralized location.
  - Protect the location by physical and logical security measures.
  - Create redundancy in the server farm.



# COMMON APPLICATIONS

- ✓ **Network-Aware Applications**

Software programs that are used to communicate over the network.

- ✓ **Application Layer Services**

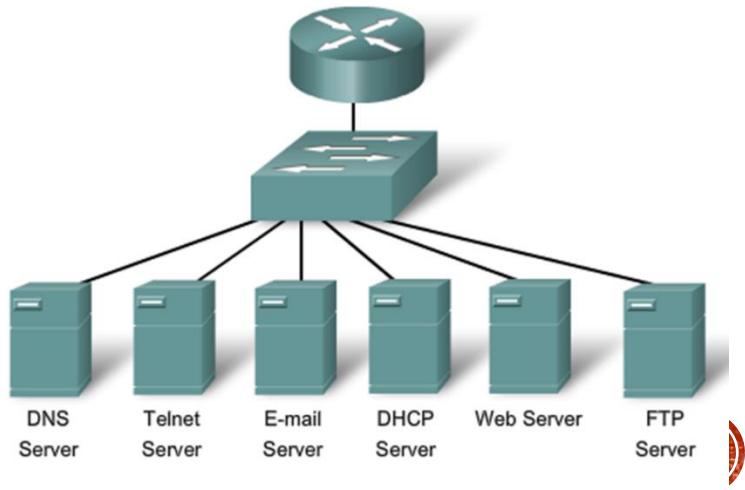
Programs that interface with the network and prepare the data for transfer.



# COMMON PROTOCOLS

## Network Protocols Define:

- Processes on either end of a communication session.
- Types of messages.
- Syntax of the messages.
- Meaning of informational fields.
- How messages are sent and the expected response.
- Interaction with the next lower layer.



# REAL-TIME APPLICATIONS



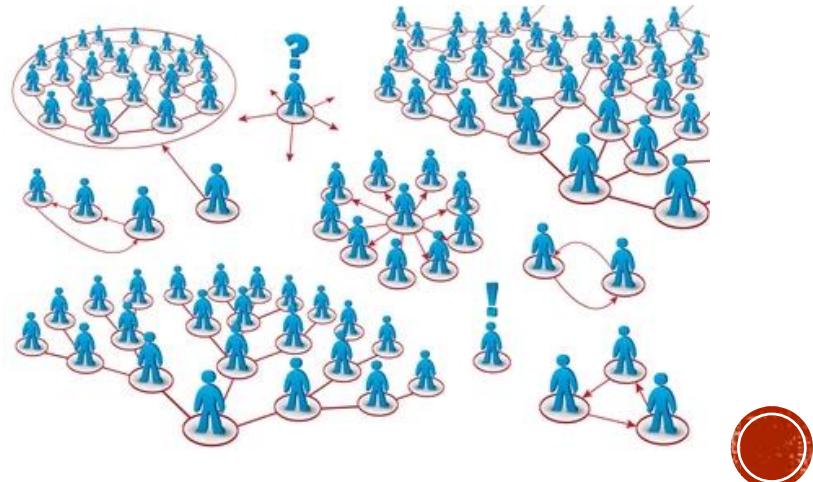
- Real-time applications require planning and dedicated services to ensure priority delivery of voice and video traffic.
  - ✓ **Infrastructure** – Needs to be evaluated to ensure it will support proposed real time applications.
  - ✓ **VoIP** – Is implemented in organizations that still use traditional telephones.
  - ✓ **IP telephony** – The IP phone itself performs voice-to-IP conversion.
  - ✓ **Real-time Video Protocols** – Use Time Transport Protocol (RTP) and Real-Time Transport Control Protocol (RTCP).



# SCALING A SMALL NETWORK

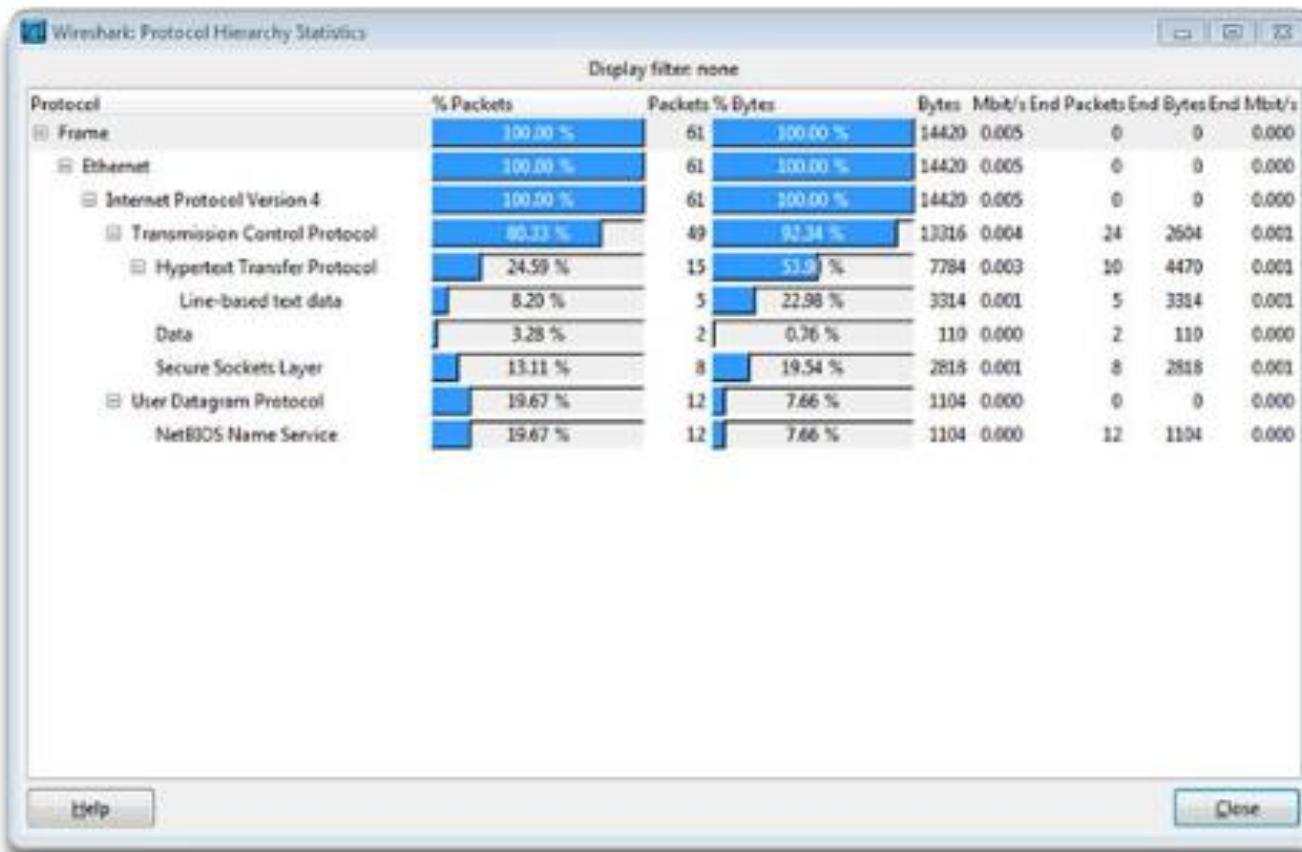
**Important considerations when growing to a larger network:**

- **Documentation** –Physical and logical topology.
- **Device inventory** – List of devices that use or comprise the network.
- **Budget** – Itemized IT expense items, including the amount of money allocated to equipment purchase for that fiscal year.
- **Traffic Analysis** – Protocols, applications, and services and their respective traffic requirements should be documented.



# PROTOCOL ANALYSIS

- Information gathered by protocol analysis can be used to make decisions on how to manage traffic more efficiently.



# KEEPING THE NETWORK SAFE

- Threats to network security
- Physical security
- Security vulnerabilities
- Malware
- Network Attacks
- Mitigating Network Attacks



# THREATS TO NETWORK SECURITY

- Categories of Threats to Network Security



Information Theft

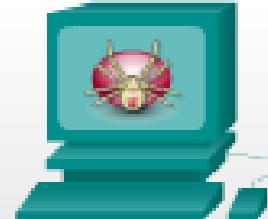


Data Loss and Manipulation



Identity Theft

404  
page not  
found



Disruption of Service



# PHYSICAL SECURITY



**Four classes of physical threats are:**

- **Hardware threats**

- ✓ Physical damage to servers, routers, switches, cabling plant, and workstations

- **Environmental threats**

- ✓ Temperature extremes (too hot or too cold)
  - ✓ humidity extremes (too wet or too dry)



# **PHYSICAL SECURITY CONT.**

- **Electrical threats**

- ✓ voltage spikes
- ✓ insufficient supply voltage (brownouts)
- ✓ unconditioned power (noise)
- ✓ total power loss

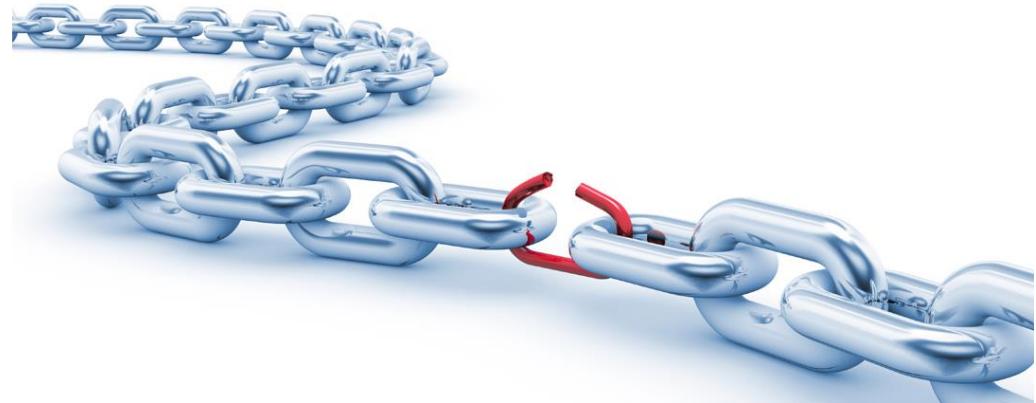
- **Maintenance threats**

- ✓ Poor handling of key electrical components (electrostatic discharge)
- ✓ lack of critical spare parts
- ✓ poor cabling
- ✓ poor labeling



# TYPES OF SECURITY VULNERABILITIES

- ✓ Technological
- ✓ Configuration
- ✓ Security policy



# MALWARE

## Virus

- ✓ Malicious software that is attached to another program to execute a particular unwanted function on a workstation.

## Trojan horse

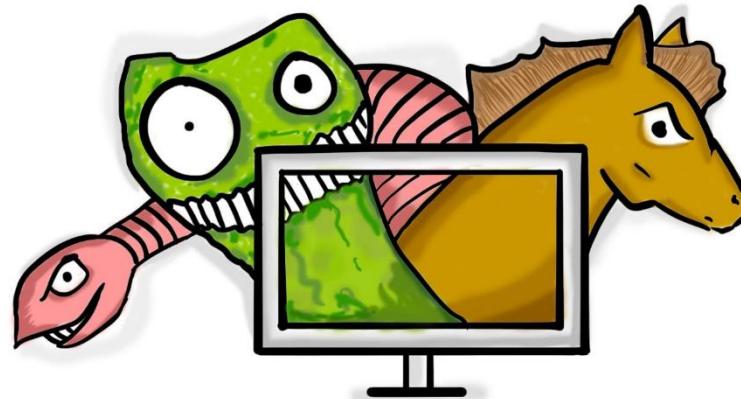
- ✓ An entire application written to look like something else, when in fact it is an attack tool.



# MALWARE CONT.

## Worms

- ✓ Worms are self-contained programs that attack a system and try to exploit a specific vulnerability in the target.
- ✓ The worm copies its program from the attacking host to the newly exploited system to begin the cycle again.



# NETWORK ATTACKS

- Malware is a means to get a payload delivered.
- When it is delivered and installed, the payload can be used to cause a variety of network related attacks.
- **Categories of network attacks:**
  - Reconnaissance Attacks
  - Access Attacks
  - Social Engineering Attacks
  - DoS Attacks



# RECONNAISSANCE ATTACKS

- Known as information gathering.
- Analogous to a thief surveying a neighbourhood by going door-to-door pretending to sell something.
- Ex:
  - Perform an information query of a target
  - Initiate a port scan of active IP addresses
  - Run Vulnerability Scanners



# ACCESS ATTACKS

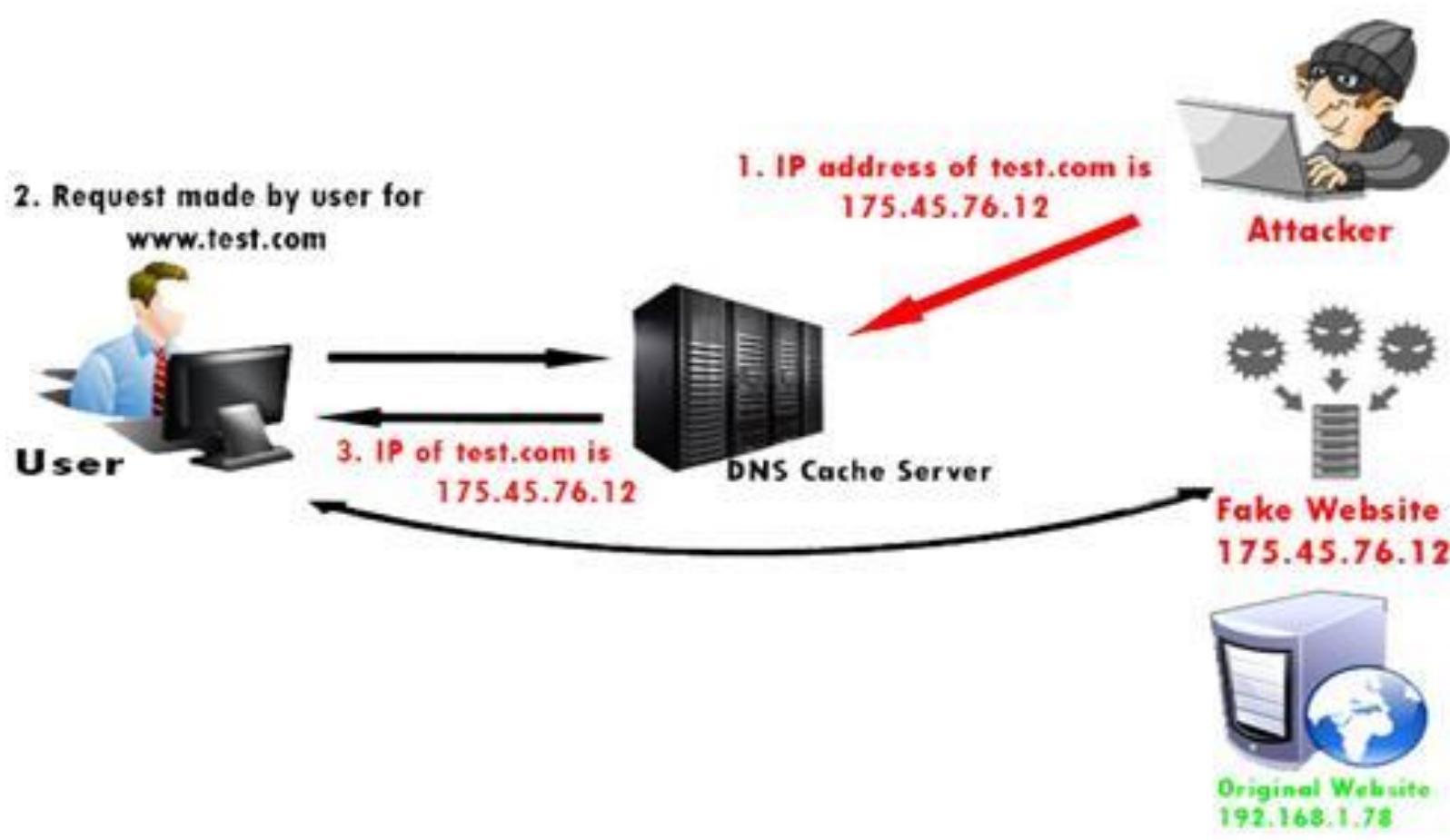
## Password Attacks

- ✓ Brute-force attack
- ✓ Dictionary attack



# ACCESS ATTACKS CONT.

## PORT RE-DIRECTION



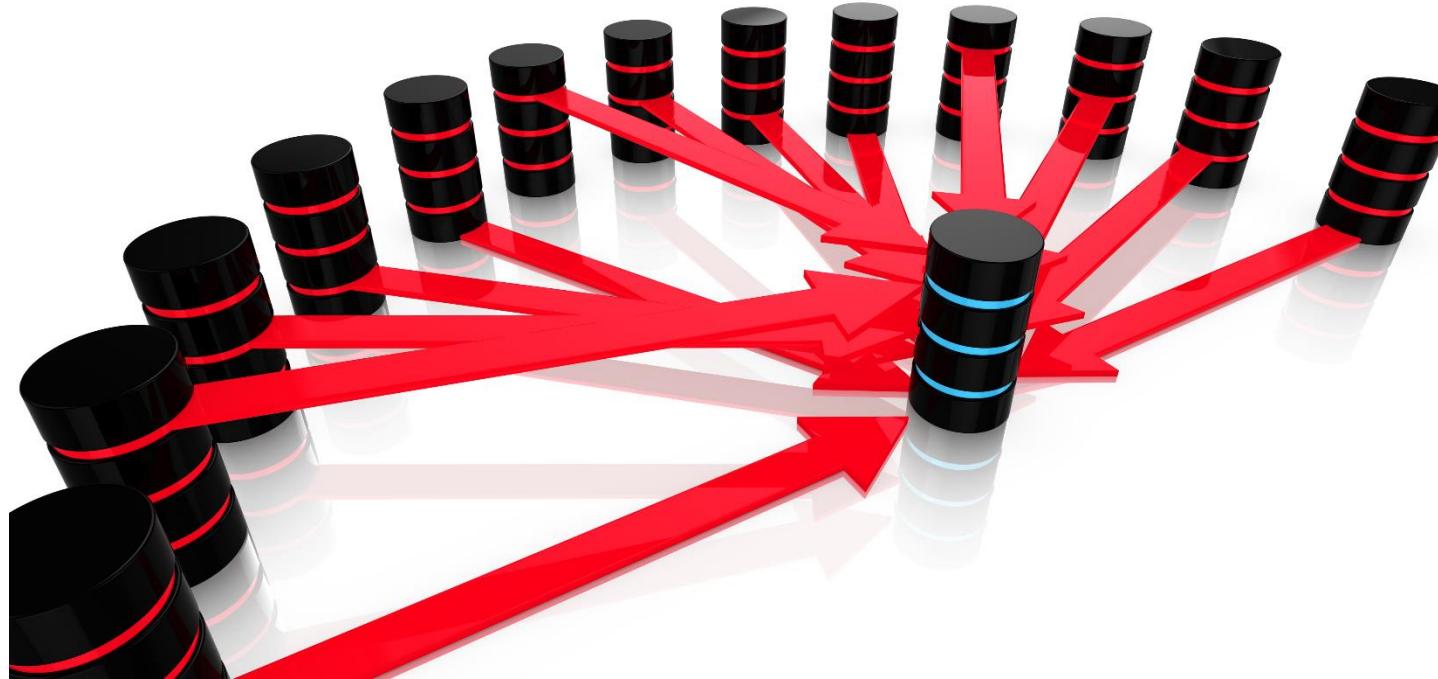
# SOCIAL ENGINEERING

- Attempts to manipulate individuals into performing actions or divulging confidential information.



# DENIAL OF SERVICE ATTACKS (DOS)

- Attacker floods servers, systems or networks with traffic in order to overwhelm the victim resources and make it difficult or impossible for legitimate users to use them.



# MITIGATING NETWORK ATTACKS

## Backup, Upgrade, Update, and Patch

- Antivirus software can detect most viruses and many Trojan horse applications and prevent them from spreading in the network.
- Keep current with the latest versions of antivirus software.
- Install updated security patches.



# MITIGATING NETWORK ATTACKS CONT.

## AUTHENTICATION, AUTHORIZATION, AND ACCOUNTING (AAA)

### Authentication

- Users and administrators must prove their identity.
  - ✓ username and password combinations
  - ✓ challenge and response questions
  - ✓ token cards

A Authentication  
A Authorization  
A Accounting



# MITIGATING NETWORK ATTACKS CONT.

## AUTHENTICATION, AUTHORIZATION, AND ACCOUNTING (AAA)

### Authorization

- Determines which resources the user can access and the operations that the user is allowed to perform.

A Authentication  
A Authorization  
A Accounting



# MITIGATING NETWORK ATTACKS CONT.

## AUTHENTICATION, AUTHORIZATION, AND ACCOUNTING (AAA)

### Accounting

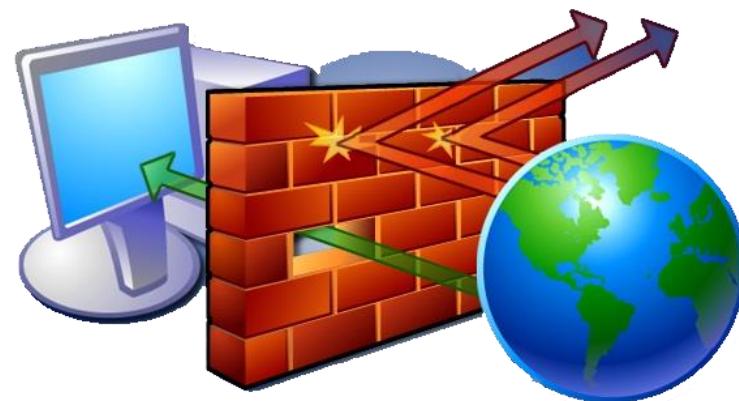
- Records what the user accessed,  
the amount of time the resource is accessed  
and any changes made.



# MITIGATING NETWORK ATTACKS CONT.

## FIREWALLS

- A Firewall resides between two or more networks.
- It controls traffic and helps prevent unauthorized access.



# MITIGATING NETWORK ATTACKS CONT.

## FIREWALLS CONT.

- Methods used are:
  - Packet Filtering
  - Application Filtering
  - URL Filtering
  - Stateful Packet Inspection (SPI) – Incoming packets must be legitimate responses to requests from internal hosts.



# MITIGATING NETWORK ATTACKS CONT.

## ENDPOINT SECURITY

- Common endpoints: laptops, desktops, servers, smart phones, and tablets.
- Employees must follow the companies documented security policies to secure their devices.
- Policies include the use of anti-virus software and host intrusion prevention.





*That's all Folks!*



# Questions

