Computing Systems

S.L.O. # 1
Sub Topics: 9 Total SLO: 29
MCQ: (8) 8 Marks CRQ: (3) 9 Marks ERQ: (0) 0 Marks

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

1.1 Data Representation in a Digital Computer

SLO	Students should be able to	Cognitive Level
1.1.1	differentiate between analog and digital signals;	U
1.1.2	explain the binary data representation using binary pulses, i.e., 0/ low/ off and 1/ high/ on;	U

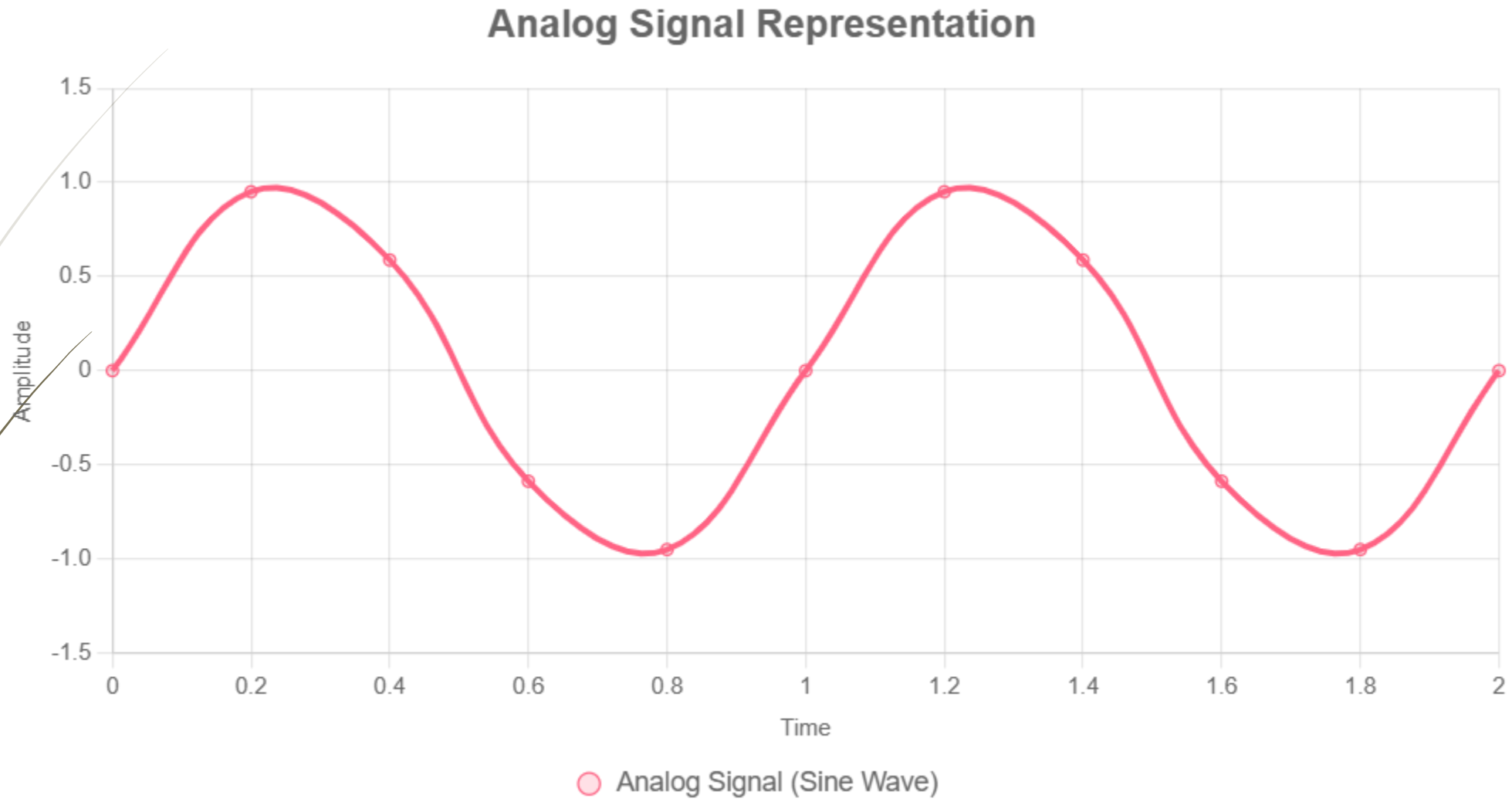
SLO 1.1.1 U

differentiate between analog and digital signals;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Analog vs Digital Signals

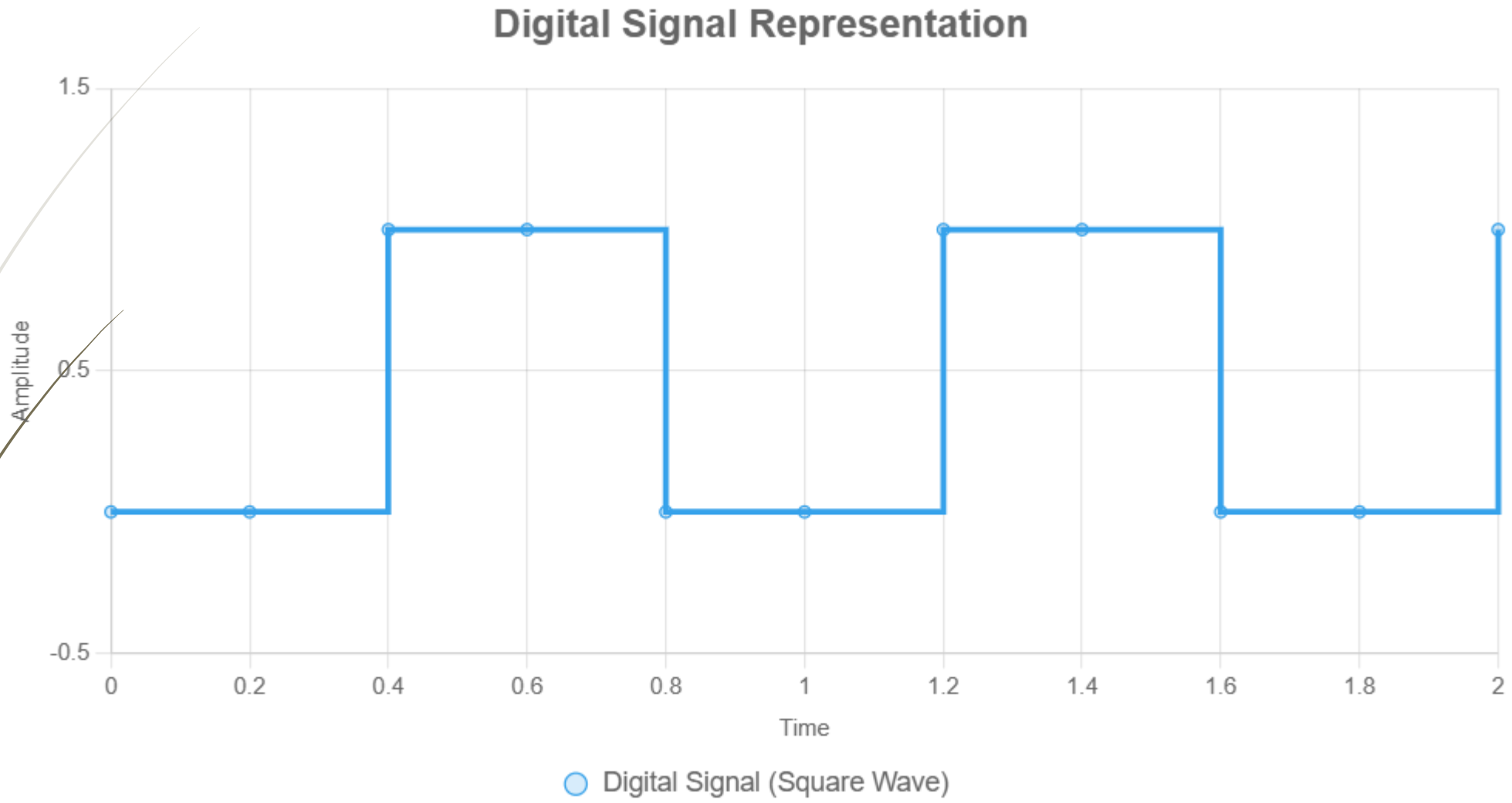
SLO 1.1.1 U



Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Analog vs Digital Signals

SLO 1.1.1 U



Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Analog vs Digital Signals

SLO 1.1.1 U

Feature	Analog Signal	Digital Signal
Definition	A continuous signal that varies smoothly over time.	A discrete signal that has specific values (0 and 1).
Nature	Continuous in both time and amplitude.	Discrete in time and amplitude.
Signal Form	Sine waves or other smooth curves.	Square waves or binary values.
Representation	Represented by voltage, current, or other continuous parameters.	Represented using binary code (bits: 0s and 1s).
Accuracy	Prone to distortion and noise over distance.	Less affected by noise; more accurate over long distances.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Analog vs Digital Signals

SLO 1.1.1 U

Feature	Analog Signal	Digital Signal
Bandwidth	Generally, requires more bandwidth.	Requires less bandwidth compared to analog.
Processing	Harder to process and store.	Easier to process, compress, and store.
Examples	Human voice, analog radio, old TV broadcasts.	Computers, digital phones, CDs, digital TV.

SLO 1.1.2 U

explain the binary data representation using binary pulses, i.e., 0/ low/ off and 1/ high/ on;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Binary Data Representation

SLO 1.1.2 U

- In digital electronics and computing, binary data is represented using two distinct states 0 and 1.
- These are often called binary digits or bits, and they form the foundation of all digital communication and processing.

1. Binary Pulses:

- A **binary pulse** is a basic digital signal that switches between two levels to represent binary values:
  - **0 (Zero) → Low / Off / No voltage or current**
  - **1 (One) → High / On / Presence of voltage or current**
- These states are easily distinguishable by electronic systems, which is why they are ideal for data transmission and storage.

Binary Data Representation

SLO 1.1.2 U

2. Representation in Electrical Terms:

Binary Bit	Electrical Signal	Description
0	Low Voltage (e.g., 0V)	No signal, OFF, Low state
1	High Voltage (e.g., 5V or 3.3V)	Signal present, ON, High state

- In most digital systems (like microprocessors), 0 volts represents binary 0, and a specific positive voltage (such as +5V or +3.3V) represents binary 1.

Binary Data Representation

SLO 1.1.2 U

3. Applications:

- **Data transmission** in digital communication (Wi-Fi, USB, etc.)
- **Storage** in memory chips (RAM, SSDs)
- **Processing** in CPUs using logic gates
- Binary pulses use distinct voltage levels to represent bits: 0 as low/off, and 1 as high/on.
- This simple yet powerful method allows computers and digital devices to process and store all kinds of data—from text to images to video.

1.2 Logic Gates

SLO	Students should be able to	Cognitive Level
1.2.1	define the following terms: a. digital logic, b. logic gates and logic circuits, c. truth table;	R
1.2.2	explain the following logic gates in terms of the number of inputs and outputs using truth tables: a. AND, b. OR, c. NOT, d. NAND, e. NOR, f. Exclusive OR (XOR), g. Exclusive NOR (XNOR);	U
1.2.3	identify logic gates from the truth table;	U
1.2.4	explain the uses of logic gates in digital devices;	U
1.2.5	represent the logic circuits in the form of a truth table;	A

1.2 Logic Gates

SLO	Students should be able to	Cognitive Level
1.2.6	explain the following Boolean identities: a. Identity Law, b. Distributive Law, c. Associative Law, d. Commutative Law, e. Inverse (Complement) Law, f. De Morgan's Theorem, g. Absorption Law;	U
1.2.7	construct a logic circuit for a given real-life problem;	A

SLO 1.2.1 R

define the following terms:

a. digital logic,
b. logic gates and logic circuits,
c. truth table;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Define Terms: Digital Logic

SLO 1.2.1 R

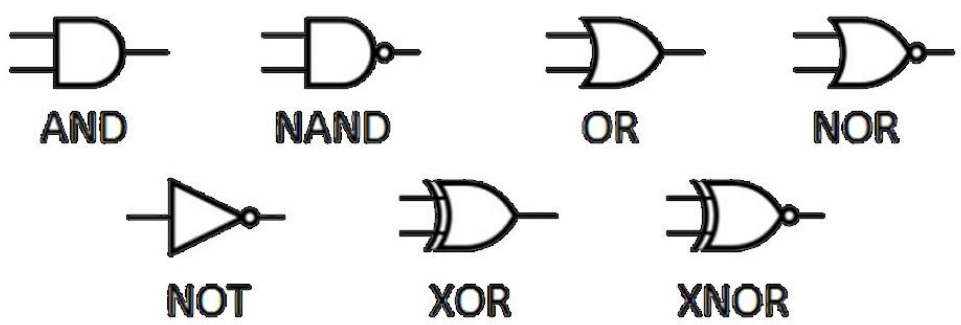
- Digital logic is the foundation of digital systems and computing.
- It involves using binary values (0 and 1) to perform logical operations and decision-making in electronic devices.
- These operations are governed by rules of Boolean algebra and are implemented through electronic components like transistors.
- It controls how digital devices like computers, calculators, and microcontrollers process data.
- Digital logic enables operations like addition, comparison, and data storage.

Define Terms: Logic Gates

SLO 1.2.1 R

- Logic gates are basic building blocks of digital circuits.
- They are electronic devices that perform specific logical operations on one or more binary inputs to produce a single binary output.
- Common types: AND, OR, NOT, NAND, NOR, XOR, XNOR
- Example: An AND gate outputs 1 only if all its inputs are 1.

INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

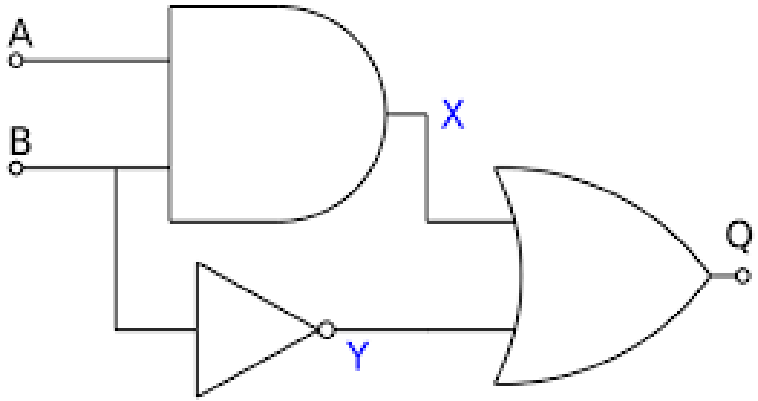


A AND B	$A \cdot B$
A OR B	$A + B$
NOT A	$\overline{A}$
A XOR B	$A \oplus B$

Define Terms: Logic Circuits

SLO 1.2.1 R

- Logic circuits are combinations of logic gates connected together to perform complex operations.
- Combinational circuits (e.g., adders, multiplexers): Output depends only on current inputs.
- Sequential circuits (e.g., flip-flops, counters): Output depends on current inputs and previous states.



A	B	X	Y	Q
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

Define Terms: Truth Table

SLO 1.2.1 R

- A truth table is a chart that shows all possible input combinations to a logic gate or circuit and their corresponding outputs.
- It is used to understand and design logic gates and circuits.
- Each row of the table represents a unique set of input values.
- It helps visualize how a logic gate behaves.

Input A	Input B	Output (A AND B)
0	0	0
0	1	0
1	0	0
1	1	1

SLO 1.2.2 U

explain the following logic gates in terms of the number of inputs and outputs using truth tables:

a. AND,
b. OR,
c. NOT,
d. NAND,
e. NOR,
f. Exclusive OR (XOR),
g. Exclusive NOR (XNOR);

Explain the gate: (a) AND

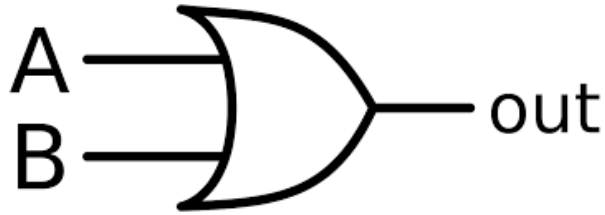
SLO 1.2.2 U

- Inputs: 2 (can be more)
- Output: 1
- Operation: Output is 1 only when all inputs are 1
- Example: A microwave oven only starts when the door is closed and the start button is pressed. Both conditions must be true for the circuit to activate.
- Expression Symbol: $A \cdot B$ or $A\&B$
- Truth Table:

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

SLO 1.2.2 U

Explain the gate: (b) OR

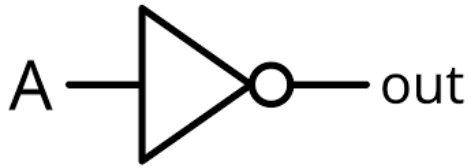


- Inputs: 2 (can be more)
- Output: 1
- Operation: Output is 1 when at least one input is 1
- Example: A car's interior light turns on if either the driver's door or the passenger door is opened.
- Expression Symbol: $A + B$
- Truth Table:

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Explain the gate: (c) NOT

SLO 1.2.2 U

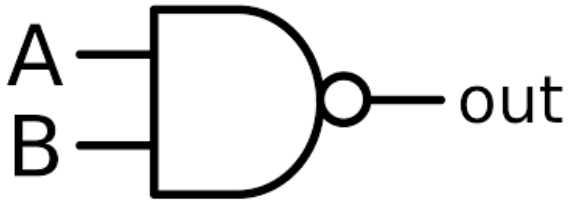


- Inputs: 1
- Output: 1
- Operation: Output is the inverse of the input
- Example: A TV's power LED is off when the system is powered on, and vice versa, using an inverter circuit.
- Expression Symbol: A' or $\overline{A}$ or $\neg A$
- Truth Table:

A	NOT A
0	1
1	0

Explain the gate: (d) NAND (NOT + AND)

SLO 1.2.2 U

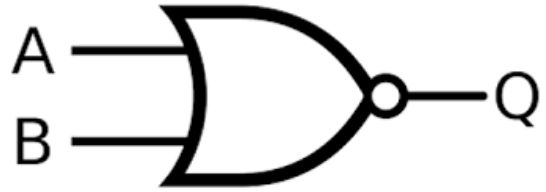


- Inputs: 2 (can be more)
- Output: 1
- Operation: Output is 0 only when all inputs are 1; otherwise, 1
- Example: An alarm system that triggers if a window is not closed and motion is detected inside a house.
- Expression Symbol: $\overline{A \cdot B}$
- Truth Table:

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

SLO 1.2.2 U

Explain the gate: (e) NOR (NOT + OR)



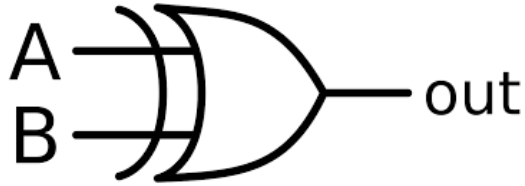
- Inputs: 2 (can be more)
- Output: 1
- Operation: Output is 1 only when all inputs are 0
- Example: A light switch circuit where flipping either switch (upstairs or downstairs) toggles the light, but flipping both does nothing.
- Expression Symbol: $\overline{A + B}$
- Truth Table:

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Explain the gate: (f) Exclusive OR (XOR)

SLO 1.2.2 U

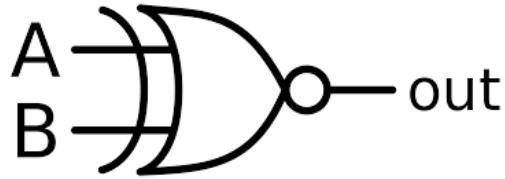


- Inputs: 2
- Output: 1
- Operation: Output is 1 only when inputs are different
- Example: A light switch circuit where flipping either switch (upstairs or downstairs) toggles the light, but flipping both does nothing.
- Expression Symbol: $A \oplus B$
- Truth Table:

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Explain the gate: (g) Exclusive NOR (XNOR)

SLO 1.2.2 U



- Inputs: 2
- Output: 1
- Operation: Output is 1 only when inputs are the same
- Example: A keyless car entry system that unlocks only when the code entered matches the stored code exactly (same HIGH/LOW pattern).
- Expression Symbol: $\overline{A \oplus B}$ or $A \odot B$
- Truth Table:

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.2.3 U

identify logic gates from the truth table;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Identify logic gates from the truth table

SLO 1.2.3 U

A	B	OUT
0	0	1
0	1	1
1	0	1
1	1	0

NAND

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1

OR

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0

XOR

A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	0

NOR

A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

AND

A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	1

XNOR

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Identify logic gates from the truth table

SLO 1.2.3 U

A	B	C	OUT
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

OR

A	B	C	OUT
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NAND

A	B	C	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

AND

A	B	C	OUT
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

NOR

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.2.4 U

explain the uses of logic gates in digital devices;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Uses of Logic Gates in Digital Devices

SLO 1.2.4 U

- Logic gates are the fundamental building blocks of digital electronics.
- They perform basic logical functions on binary inputs to produce a binary output, enabling digital devices to process data and make decisions.

1. Performing Arithmetic Operations

- Logic gates are used in the Arithmetic Logic Unit (ALU) of a computer's CPU to perform calculations like addition, subtraction, multiplication, and division.
- For example, an adder circuit (used for adding numbers) is built using XOR, AND, and OR gates to process binary digits (bits). A half-adder, for instance, uses an XOR gate to compute the sum and an AND gate to compute the carry.
- This is how calculators and computers handle math operations.

Uses of Logic Gates in Digital Devices

SLO 1.2.4 U

2. Data Processing and Decision Making

- Logic gates process inputs to make decisions in digital devices.
- For example, they determine whether a condition is true or false based on input signals.
- In a smartphone, logic gates decide whether to unlock the screen when the correct password is entered (using AND gates to check if all conditions match).
- They also control program flow in software, like deciding which instruction to execute next in a processor.

3. Memory Storage

- Logic gates are used to create memory circuits, such as flip-flops, which store bits of data (0 or 1).
- For example, NAND or NOR gates are combined to form SR latches (Set-Reset latch) or D flip-flops (Data flip-flop), which are the basis of RAM (Random Access Memory) and registers in computers.
- This allows devices to store data temporarily, like saving a phone number in your phone's memory while making a call.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Uses of Logic Gates in Digital Devices

SLO 1.2.4 U

4. Signal Control and Switching

- Logic gates control signals in digital devices, acting like switches to direct data flow.
- For instance, in a traffic light system, AND and OR gates process sensor inputs to decide when to change lights (e.g., green to red when a timer and sensor signal align).
- In a keyboard, logic gates detect which key is pressed and send the correct signal to the computer.

5. Encoding and Decoding Data

- Logic gates are used in encoders and decoders to convert data into formats devices can understand or display.
- For example, in a digital display (like on a calculator), a decoder circuit made of AND, OR, and NOT gates converts binary numbers into signals that light up specific segments of a 7-segment display to show digits like “5” or “9”.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Uses of Logic Gates in Digital Devices

SLO 1.2.4 U

6. Error Detection and Correction

- Logic gates help detect and correct errors in data transmission.
- For example, XOR gates are used in parity check circuits to verify if data sent over a network has been corrupted.
- This ensures reliable communication in devices like Wi-Fi routers or USB drives.

7. Digital Communication

- By combining logic gates, complex circuits like multiplexers, demultiplexers, and counters are built. These are used in devices to select data paths, count events, or manage multiple signals.
- For example, in a digital clock, logic gates in a counter circuit keep track of seconds, minutes, and hours.

SLO 1.2.5 A

represent the logic circuits in the form of a truth table;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.2.5 A

Represent the logic circuits in the form of a truth table

- To represent logic circuits in the form of a truth table, you list all possible input combinations and show the corresponding output for the circuit.
- Each row of the truth table corresponds to one unique combination of inputs, and the output is derived based on the logic gates used in the circuit.
- **Example: Representing a Simple Logic Circuit as a Truth Table**
- Suppose we have a logic circuit with two inputs **A** and **B**, and the output **Y** is:

Y = (A AND B) OR (NOT A)

SLO 1.2.5 A

Represent the logic circuits in the form of a truth table

- **Step 1: List all possible inputs (A and B)**
- Since both A and B are binary, there are $2^2 = 4$ combinations:

A	B
0	0
0	1
1	0
1	1

SLO 1.2.5 A

Represent the logic circuits in the form of a truth table

- **Step 2: Calculate intermediate and final outputs**
- Compute **A AND B**
- Compute **NOT A**
- Compute **Y = (A AND B) OR (NOT A)**

A	B	A AND B	NOT A	Y = (A AND B) OR (NOT A)
0	0	0	1	1
0	1	0	1	1
1	0	0	0	0
1	1	1	0	1

SLO 1.2.6 U

explain the following Boolean identities:

a. Identity Law,
b. Distributive Law,
c. Associative Law,
d. Commutative Law,
e. Inverse (Complement) Law,
f. De Morgan's Theorem,
g. Absorption Law;

Explain Boolean Identities

SLO 1.2.6 U

a. Identity Law

- The identity law states that combining a variable with the identity element (0 or 1) using AND or OR leaves the variable unchanged. This is similar to multiplying by 1 or adding 0 in ordinary algebra.
- For OR:
  - $A+0=A$
  - Because OR with 0 does not change the value of A.
- For AND:
  - $A \cdot 1=A$
  - Because AND with 1 does not change the value of A.
- Example:
  - If $A=1$, then $A+0=1$ and $A \cdot 1=1$.
  - If $A=0$, then $A+0=0$ and $A \cdot 1=0$.

Explain Boolean Identities

SLO 1.2.6 U

b. Distributive Law

- Distributive law allows us to distribute AND over OR and OR over AND, which is useful for expanding or factoring Boolean expressions.
  - AND distributes over OR: $A \cdot (B+C) = (A \cdot B) + (A \cdot C)$
  - OR distributes over AND: $A + (B \cdot C) = (A+B) \cdot (A+C)$
- Example:
  - $A \cdot (B+C)$ means A AND either B OR C. This equals to $(A \cdot B) + (A \cdot C)$ meaning either both A and B are true, or both A and C are true.
  - $A + (B \cdot C)$ means A OR both B AND C true. This is equivalent to $(A+B) \cdot (A+C)$.

Explain Boolean Identities

SLO 1.2.6 U

c. Associative Law

- Associative law states that the way variables are grouped in AND or OR operations does not affect the result.
  - For OR: $(A+B)+C=A+(B+C)$
  - For AND: $(A \cdot B) \cdot C=A \cdot (B \cdot C)$
- Example:
  - Whether you evaluate $(A+B)+C$ or $A+(B+C)$, the output is the same. This allows us to omit parentheses when all operators are the same.

Explain Boolean Identities

SLO 1.2.6 U

d. Commutative Law

- Commutative law states that the order of variables in AND or OR operations does not affect the result.
  - For OR: A+B=B+A
  - For AND: A·B=B·A
- Example: A+B is the same as B+A. Similarly, A·B equals B·A. This property lets us reorder terms freely.

Explain Boolean Identities

SLO 1.2.6 U

e. Inverse (Complement) Law

- Inverse law deals with a variable and its complement (NOT version):
  - OR of a variable and its complement:
    $A+A' = 1$
    Means either A is true or A is false, so the result is always true (1).
  - AND of a variable and its complement:
    $A \cdot A' = 0$
    Means A AND NOT A can never be true simultaneously, so the result is always false (0).
- Example:
  - If $A=1$, then $A+A' = 1+0=1$.
  - If $A=0$, then $A \cdot A' = 0 \cdot 1=0$.

Explain Boolean Identities

SLO 1.2.6 U

f. De Morgan's Theorem

- De Morgan's theorem provides rules to simplify the complement (NOT) of AND and OR expressions by converting them:
  - The complement of an AND is the OR of the complements: $(A \cdot B)' = A' + B'$
  - The complement of an OR is the AND of the complements: $(A+B)' = A' \cdot B'$
- This is very useful when simplifying logic expressions or designing circuits with NAND or NOR gates.
- Example:
  - If A=1 and B=0, then: $(A \cdot B)' = (1 \cdot 0)' = 0' = 1$ And $A' + B' = 0+1=1$ Both sides are equal, confirming the theorem.

Explain Boolean Identities

SLO 1.2.6 U

g. Absorption Law

- Absorption law helps to simplify expressions where a variable and a combination involving that variable appear together.
  - A+(A·B)=A Because if A is true, whole expression is true regardless of B.
  - A·(A+B)=A Because if A is false, whole expression is false regardless of B.
- Example:
  - A+(A·B): If A=1, output is 1. If A=0, then A·B=0, so output = 0. So output = A.
  - A·(A+B): If A=1, then A+B=1, so output = 1. If A=0, output = 0. So output = A.

Explain Boolean Identities

SLO 1.2.6 U

a. Identity Law

- $A+0=A$
- $A \cdot 1=A$

b. Distributive Law

- $A \cdot (B+C)=(A \cdot B)+(A \cdot C)$
- $A+(B \cdot C)=(A+B) \cdot (A+C)$

c. Associative Law

- $(A+B)+C=A+(B+C)$
- $(A \cdot B) \cdot C=A \cdot (B \cdot C)$

d. Commutative Law

- $A+B=B+A$
- $A \cdot B=B \cdot A$

e. Inverse (Complement) Law

- $A+A' =1$
- $A \cdot A' =0$

f. De Morgan's Theorem

- $(A \cdot B)' =A' +B'$
- $(A+B)' =A' \cdot B'$

g. Absorption Law

- $A+(A \cdot B)=A$
- $A \cdot (A+B)=A$

SLO 1.2.7 A

construct a logic circuit for a given real-life problem;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 1: Automatic Light Control

- A room light should turn ON if:
  - It is dark outside OR
  - The motion sensor detects movement inside the room.
- The light should be OFF otherwise.

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 1: Automatic Light Control

- Step 1: Define Inputs and Output

Input	Symbol	Meaning
Darkness	D	1 = Dark, 0 = Light
Motion detected	M	1 = Movement, 0 = No movement

Output	Symbol	Meaning
Light	L	1 = ON, 0 = OFF

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 1: Automatic Light Control

- Step 2: Write the Logical Expression
- Light turns ON if darkness OR motion detected:

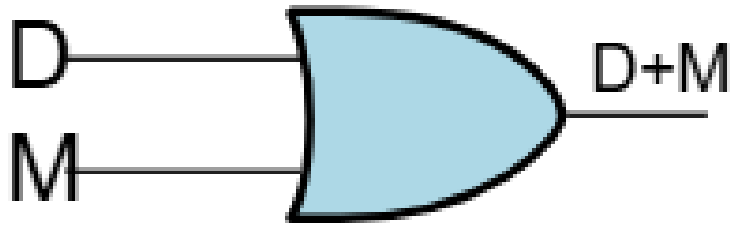
$$L=D+M$$

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 1: Automatic Light Control

- Step 3: Draw the Logic Circuit
- Use an OR gate with inputs D and M.
- The output of the OR gate is connected to the light control.



Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 1: Automatic Light Control

- Step 4: Truth Table

Input D	Input M	Output L = D + M (OR)
0	0	0
0	1	1
1	0	1
1	1	1

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 2: Security Alarm System

- The alarm should sound (activate) if:
  - The door is opened (D = 1), and
  - The security system is armed (A = 1), and
  - The window is opened (W = 1) OR the motion sensor detects movement (M = 1).

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 2: Security Alarm System

- Step 1: Define Inputs and Output

Input	Symbol	Meaning
Door opened	D	1 = Open, 0 = Closed
Security armed	A	1 = Armed, 0 = Disarmed
Window opened	W	1 = Open, 0 = Closed
Motion detected	M	1 = Movement, 0 = No movement

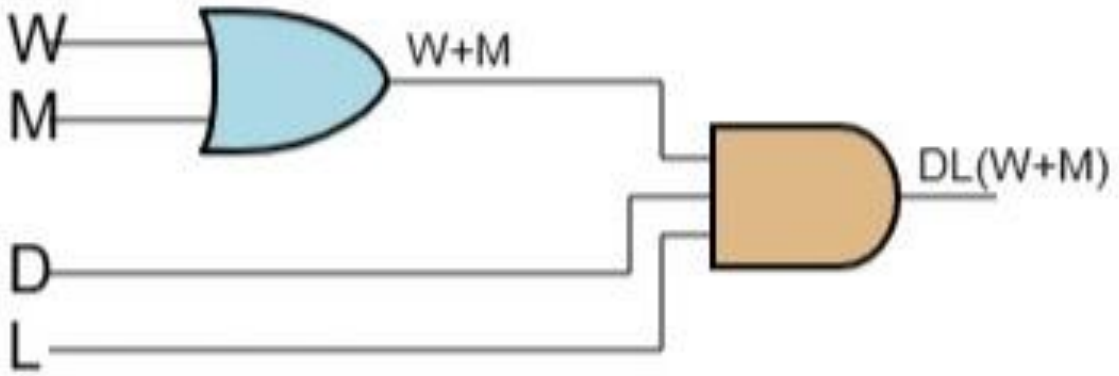
Output	Symbol	Meaning
Alarm	AL	1 = Sound ON, 0 = OFF

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 2: Security Alarm System

- Step 2: Write the Logical Expression
  - Alarm sounds if:
  - Door is opened AND security armed AND (window opened OR motion detected):
  - AL=D·A·(W+M)



Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 2: Security Alarm System

- Step 3: Draw the Logic Circuit Components
- An OR gate with inputs W and M.
- An AND gate with inputs D, A, and output from the OR gate.
- The output of the AND gate is the alarm signal AL.

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example Problem 2: Security Alarm System

- Step 4: Truth Table (Partial for clarity)

D	A	W	M	W + M	AL = D ·A ·(W + M)
0	0	0	0	0	0
1	1	0	0	0	0
1	1	1	0	1	1
1	1	0	1	1	1
1	0	1	1	1	0

Construct a Logic Circuit (Real Life)

SLO 1.2.7 A

Example 3: Vending Machine Coin Validator

- A vending machine dispenses a snack if the correct amount is inserted (e.g., $1, which is two quarters) or if a special employee key is used. This ensures only valid payments or authorized access trigger dispensing.

1.3 Karnaugh Map (K-Map)

SLO	Students should be able to	Cognitive Level
1.3.1	simplify two-variable and three-variable Boolean functions using Karnaugh map;	A
1.3.2	convert the given algebraic expression into its Sum of Products (SOP) and Product of Sums (POS) forms;	A

SLO 1.3.1 A

simplify two-variable and three-variable Boolean functions using Karnaugh map;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Karnaugh Map (K-Map)

SLO 1.3.1 A

- Maurice Karnaugh introduced the technique in 1953.
- In many digital circuits and practical problems, we need to find expressions with minimum variables.
- We can minimize Boolean expressions of 2, 3, 4 variables very easily using K-map without using any Boolean algebra theorems.
- It helps to simplify logic into simpler form by organizing grid from truth table values.
- It needs expression are in canonical form (A canonical form is when every term of the expression contains all the variables of the function, either complemented or uncomplemented.)
- K-map can take two forms:
  - Sum of product (SOP)
  - Product of Sum (POS)

Karnaugh Map (K-Map)

SLO 1.3.1 A

1. Two-variable example (Free Shipping)

- An online shop offers Free Shipping (F)
  - if Price ≥ $50 (A) or
  - the buyer is a Loyalty Member (B).
- Variables:
  - A = 1 if price ≥ $50, else 0
  - B = 1 if loyalty member, else 0
- Boolean Equation
  F = A + B

A	B	F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Karnaugh Map (K-Map)

SLO 1.3.1 A

- Sum of Product (SOP)
  - Minterms (where F = 1): rows 1, 2, 3
  - A=0 => A', A=1 => A
  - B=0 => B', B=1 => B
  - Canonical SOP (sum of minterms):
    - $F(A, B) = A'B + AB' + AB$
    - $F(A, B) = m_1 + m_2 + m_3$
    - $F(A, B) = \Sigma m(1, 2, 3)$
    - $F(A, B) = \Sigma(1, 2, 3)$

A	B	F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Decimal	A	B	F=A+B
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Decimal	A	B	Minterm	F=A+B
0	0	0	A' . B'	0
1	0	1	A' . B	1
2	1	0	A . B'	1
3	1	1	A . B	1

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Karnaugh Map (K-Map)

SLO 1.3.1 A

- Product of Sum (POS)
  - Maxterms (where F = 0): rows 0
  - A=0 => A, A=1 => A'
  - B=0 => B, B=1 => B'
  - Canonical POS (product of maxterms):
    - F(A, B)=(A+B) (already minimal)
    - $F(A, B) = M_0$
    - $F(A, B) = \Pi M(0)$
    - $F(A, B) = \Pi(0)$

A	B	F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Decimal	A	B	F=A+B
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Decimal	A	B	Maxterm	F=A+B
0	0	0	A + B	0
1	0	1	A + B'	1
2	1	0	A' + B	1
3	1	1	A' + B'	1

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Karnaugh Map (K-Map)

SLO 1.3.1 A

2) Three-variable example (Security System)

- A home Alarm (F) activates if at least two of these are 1: Door (A), Window (B), Motion (C).
  - A = door sensor
  - B = window sensor
  - C = motion sensor
- Boolean equation
  - F=AB+AC+BC

A	B	C	# of 1s	F
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	2	1
1	0	0	1	0
1	0	1	2	1
1	1	0	2	1
1	1	1	3	1

Karnaugh Map (K-Map)

SLO 1.3.1 A

- Sum of Product (SOP)

Decimal	A	B	C	Minterms	F
0	0	0	0	A' B' C'	0
1	0	0	1	A' B' C	0
2	0	1	0	A' B C'	0
3	0	1	1	A' B C	1
4	1	0	0	A B' C'	0
5	1	0	1	A B' C	1
6	1	1	0	A B C'	1
7	1	1	1	A B C	1

- Canonical SOP:
  - F(A, B, C) =A′BC+AB′C+ABC′+ABC
  - F(A, B, C) = $m_3 + m_5 + m_6 + m_7$
  - F(A, B, C) = Σm(3, 5, 6, 7)
  - F(A, B, C) = Σ(3, 5, 6, 7)

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Karnaugh Map (K-Map)

SLO 1.3.1 A

- Product of Sum (POS)

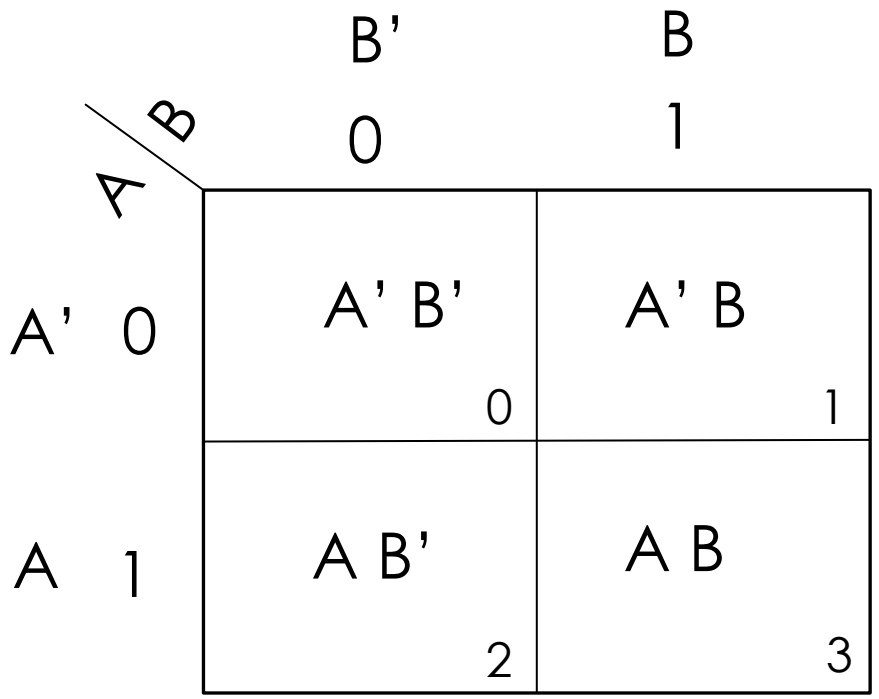
Decimal	A	B	C	Maxterms	F
0	0	0	0	A + B + C	0
1	0	0	1	A + B + C'	0
2	0	1	0	A + B' + C	0
3	0	1	1	A + B' + C'	1
4	1	0	0	A' + B + C	0
5	1	0	1	A' + B + C'	1
6	1	1	0	A' + B' + C	1
7	1	1	1	A' + B' + C'	1

- Canonical POS:
  - F(A, B, C) =(A+B+C)(A+B+C')(A+B'+C)(A'+B+C)
  - $F(A, B, C) = M_0 . M_1 . M_2 . M_4$
  - $F(A, B, C) = \Pi M(0,1,2,4)$
  - $F(A, B, C) = \Pi(0,1,2,4)$

2 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Structure of 2 Variable K-Map
  - Number of cells $2^2=4$.



0 = A'
1 = A

Binary	Decimal
00	0
01	1
10	2
11	3

2 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

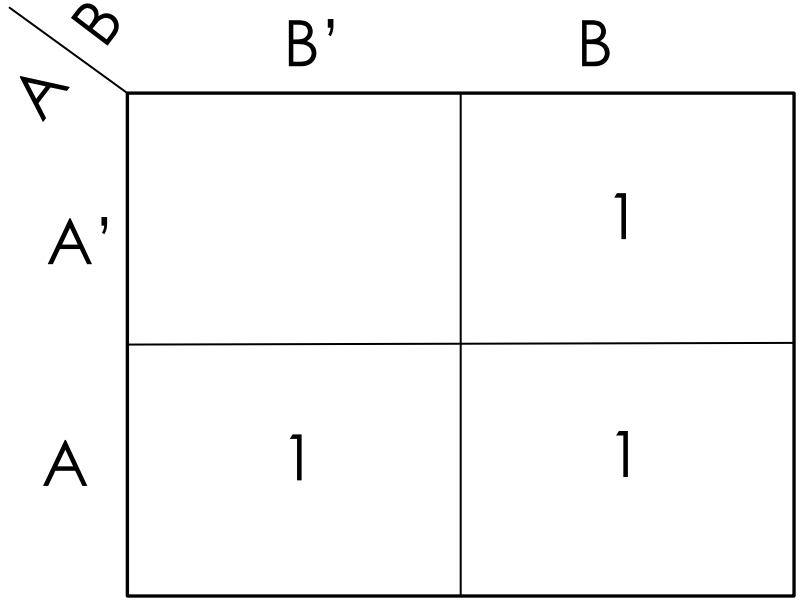
- Rules

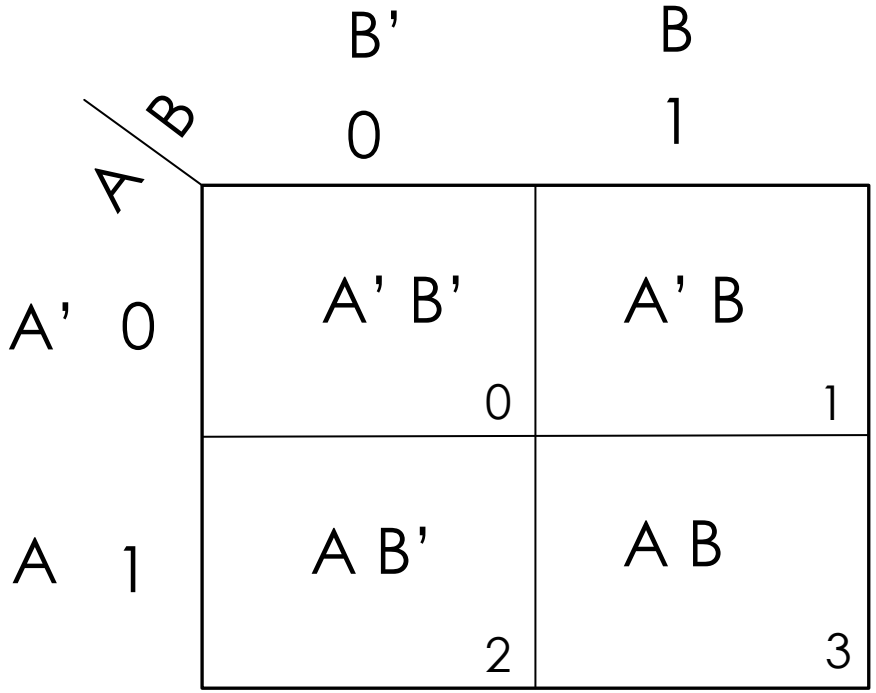
1. Group 1's in powers of 2 → allowed group sizes are 1, 2, or 4.
   - Group of 1 → keep both variables.
   - Group of 2 → one variable eliminated.
   - Group of 4 → all variables eliminated (function = 1).
2. Groups must be rectangular (1×2, 2×1, or 2×2).
3. Wrap-around adjacency allowed
   - Left edge and right edge are adjacent.
   - Top edge and bottom edge are adjacent.
4. Overlap is allowed if it helps form bigger groups.
5. Always make the largest groups possible (maximize simplification).

2 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Fill Up
  - F(A, B)=A′B+AB′+AB

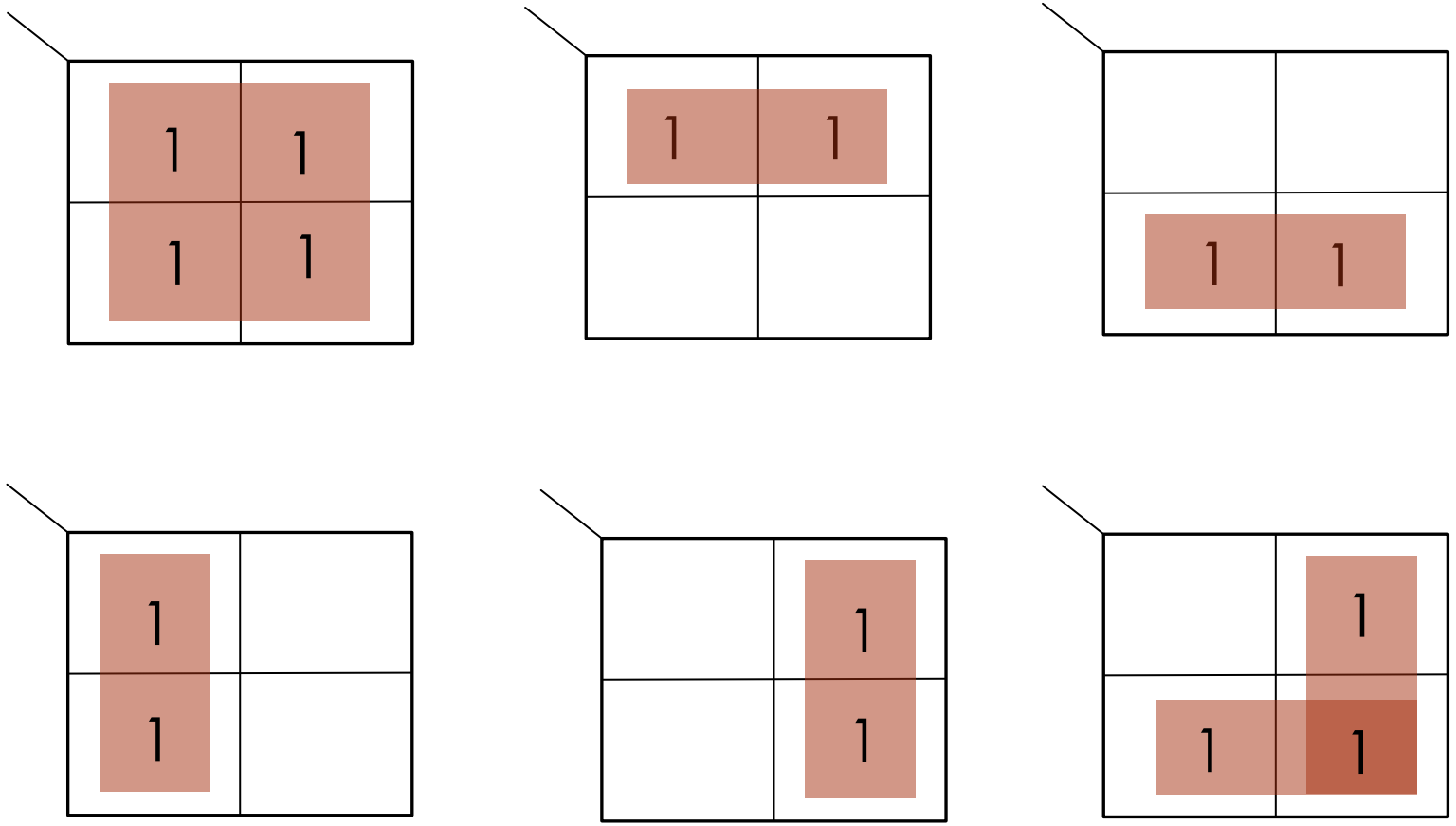




2 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

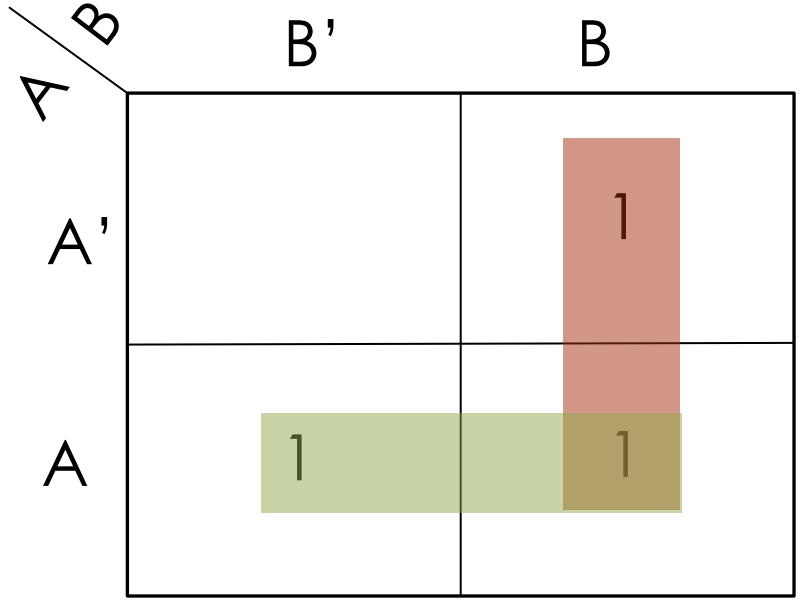
- Grouping



2 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Group Up
  - F(A, B)=A′B+AB′+AB



2 Variable Karnaugh Map (K-Map) SOP

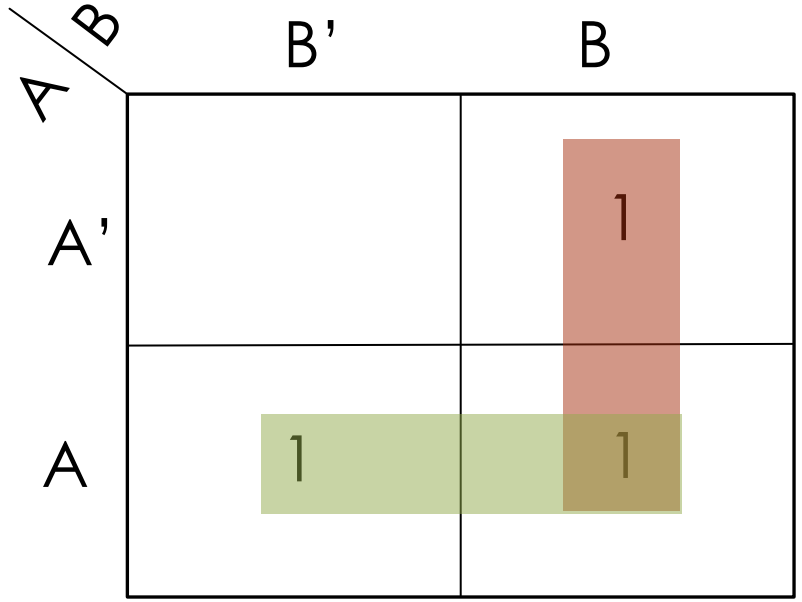
SLO 1.3.1 A

- Simplification outcomes
  - Single 1 (no grouping possible): SOP term = full minterms (e.g., A'B).
  - Pair of 1's in a row/column: eliminates one variable.
    - Example: cells m2 (A=1,B=0) + m3 (A=1,B=1) → group → expression = A.
    - All 4 cells = 1: expression simplifies to 1.
    - No 1's at all: expression simplifies to 0.

2 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Solve
  - F(A, B)=A'B+AB'+AB

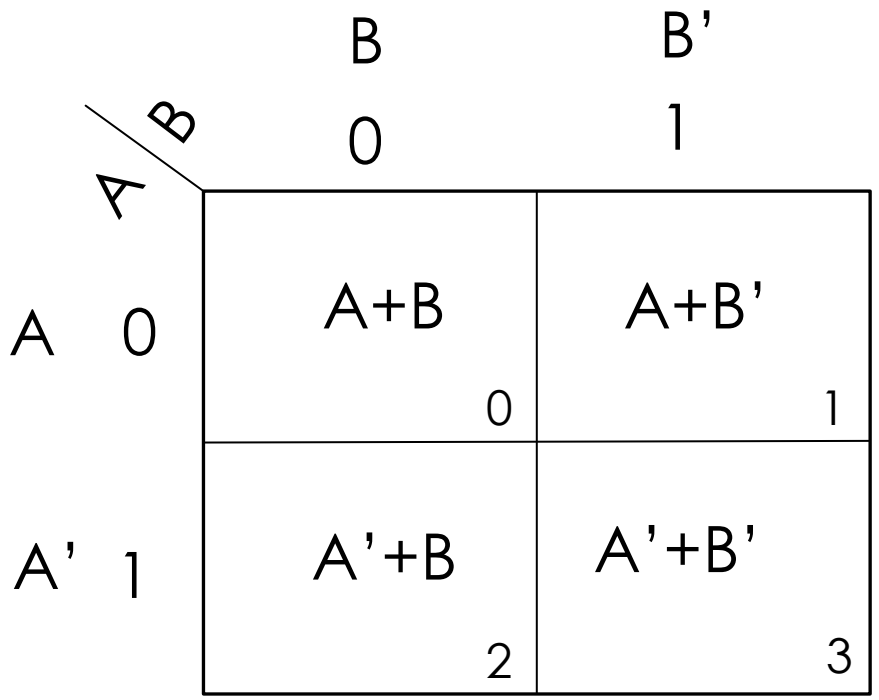


- Solution (Method 1)
  - Common in Green Group: A
  - Common in Red Group: B
  - Result is F(A, B) = A + B

2 Variable Karnaugh Map (K-Map) POS

SLO 1.3.1 A

- Structure of 2 Variable K-Map
  - Number of cells $2^2=4$.
  - Fill with 0 ang group on 0s



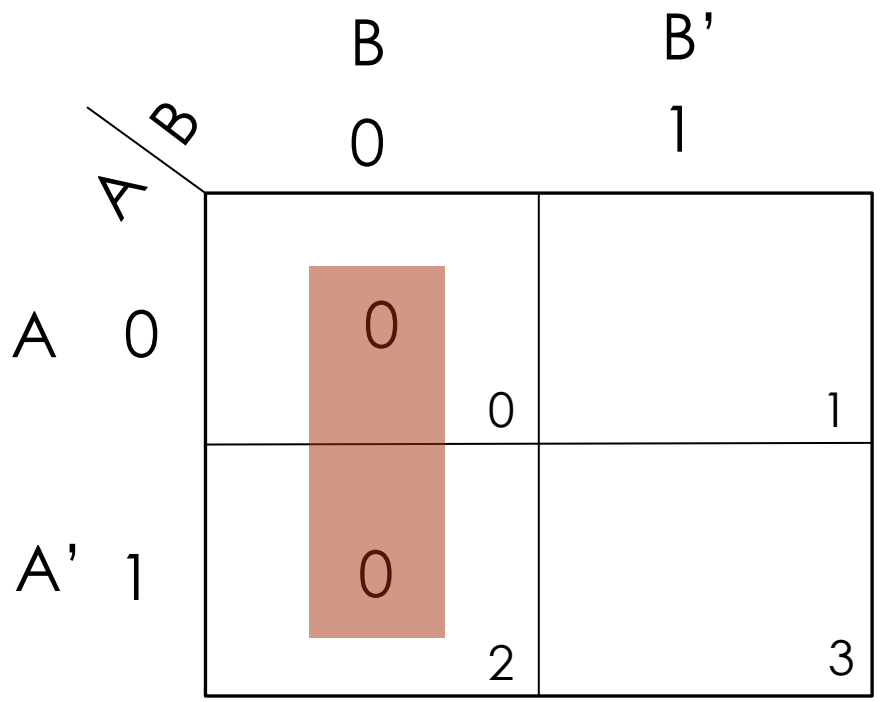
0 = A
1 = A'

Binary	Decimal
00	0
01	1
10	2
11	3

2 Variable Karnaugh Map (K-Map) POS

SLO 1.3.1 A

- F = (A' + B) (A + B)



0 = A
1 = A'

- F = B

2 Variable Karnaugh Map (K-Map)

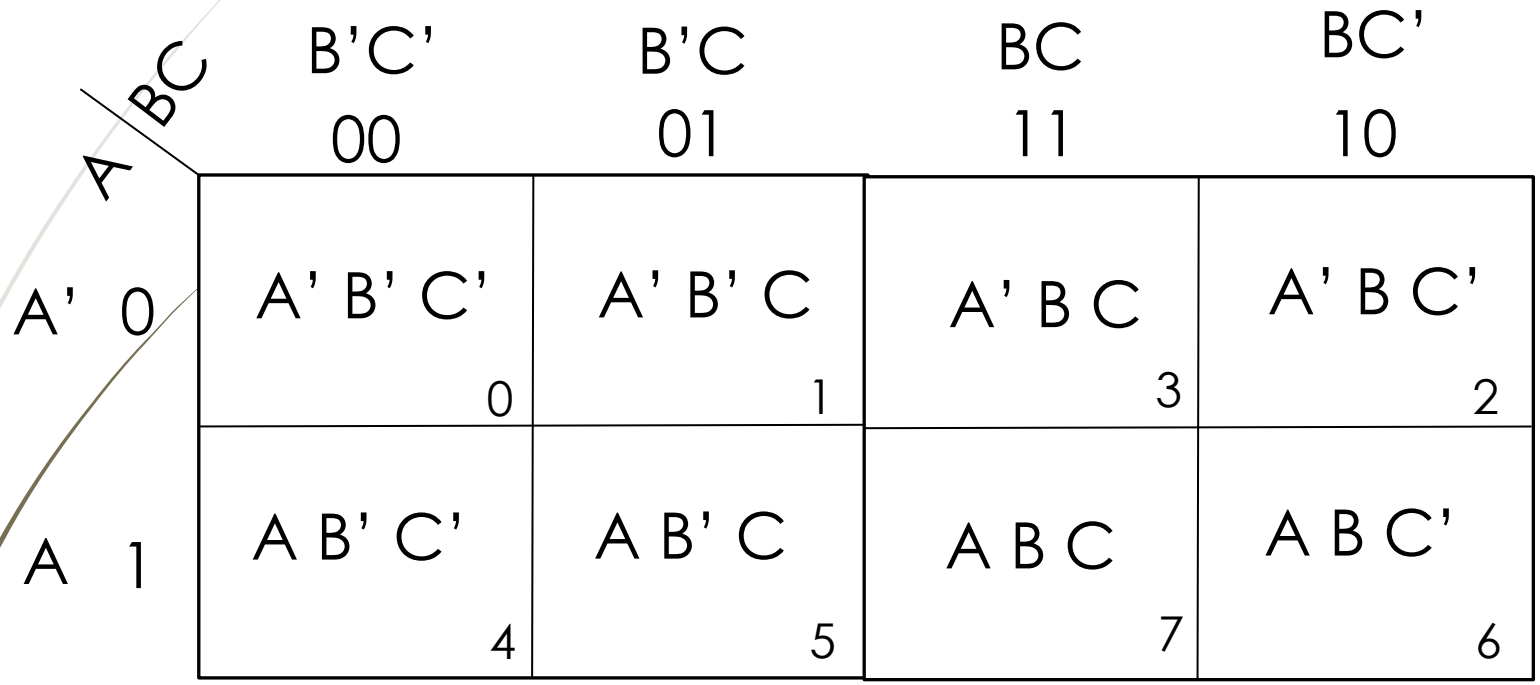
SLO 1.3.1 A

- Practice Equations
  - F(A,B)=AB′+A′B
  - F(A,B)=Σm(0,1,2)
  - F(A,B)=AB+A′B
  - F(A,B)=A′B′+AB
  - F(A,B)=ΠM(2,3)
  - F(A,B)=ΠM(0,1,2,3)

3 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- 3 Variable K-Map Structure



0 = A'
1 = A

Binary	Gray Code
00	00
01	01
10	11
11	10

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

3 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

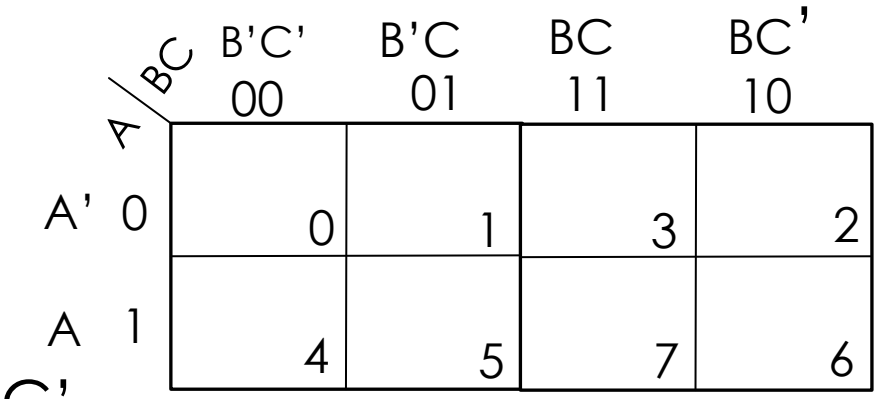
- Rules

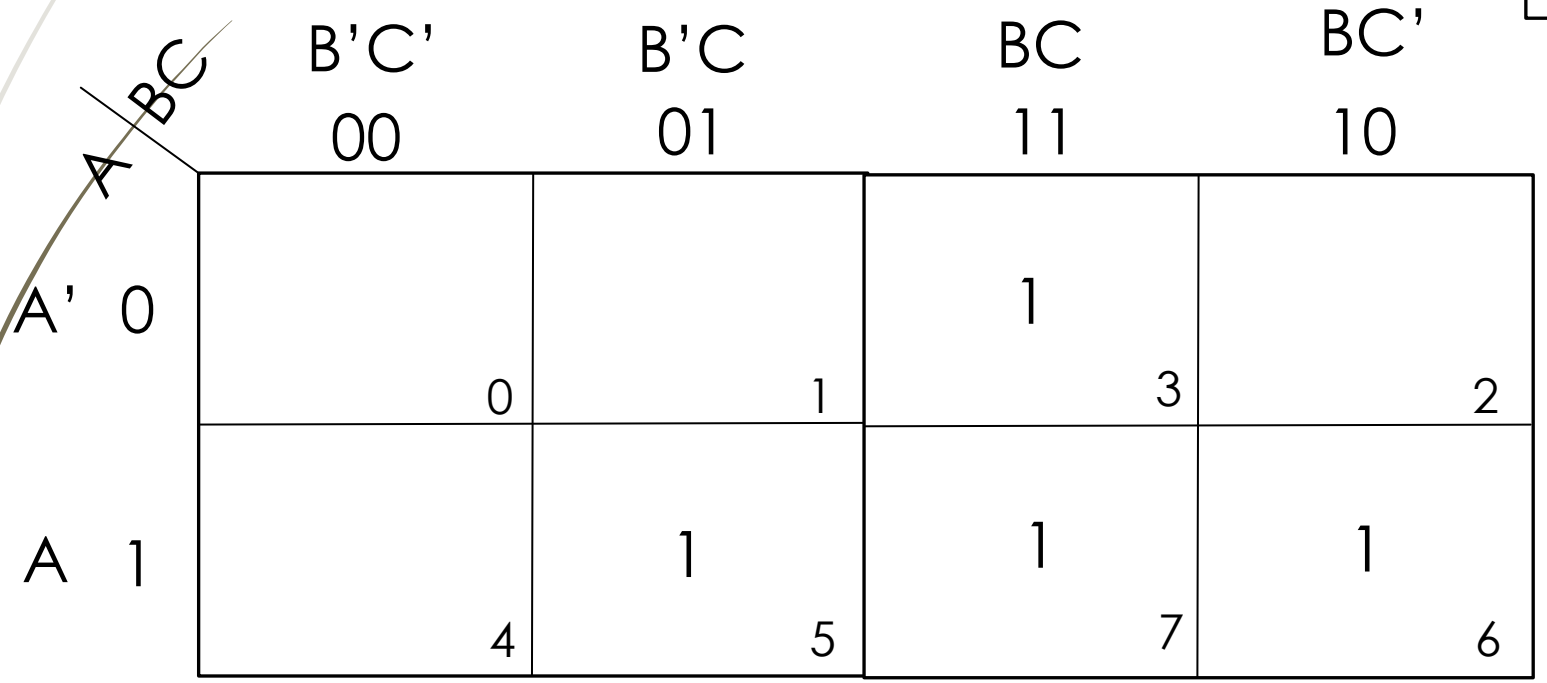
1. Group 1's in powers of 2 → sizes can be 1, 2, 4, or 8.
   - 1 → no simplification (all three variables remain).
   - 2 → one variable eliminated.
   - 4 → two variables eliminated.
   - 8 → all variables eliminated (function = 1).
2. Groups must be rectangular → (1×2, 2×1, 2×2, 4×1).
3. Wrap-around adjacency allowed
   - Leftmost & rightmost columns are adjacent.
   - Top row & bottom row are adjacent.
4. Overlap is allowed if it helps form larger groups.
5. Always make the largest possible groups first.
6. Each group must contain only 1's (no 0's in between).

3 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Fill Up
  - F(A, B, C) = Σm(3, 5, 6, 7)

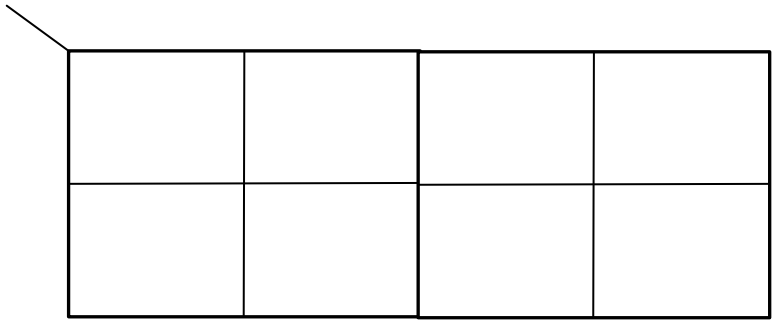


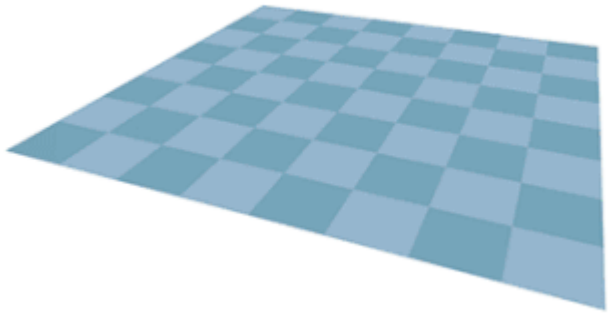


3 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Grouping

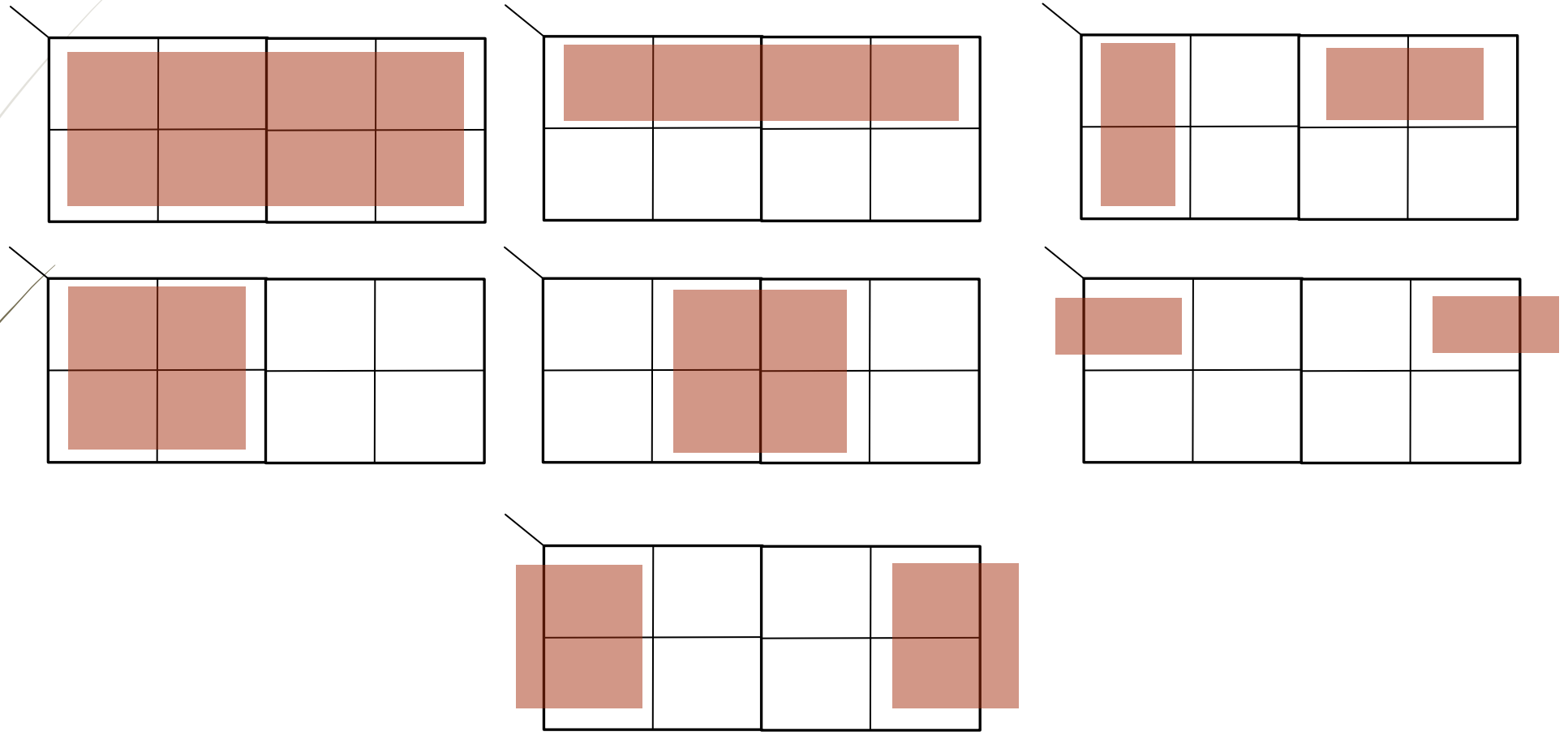




3 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

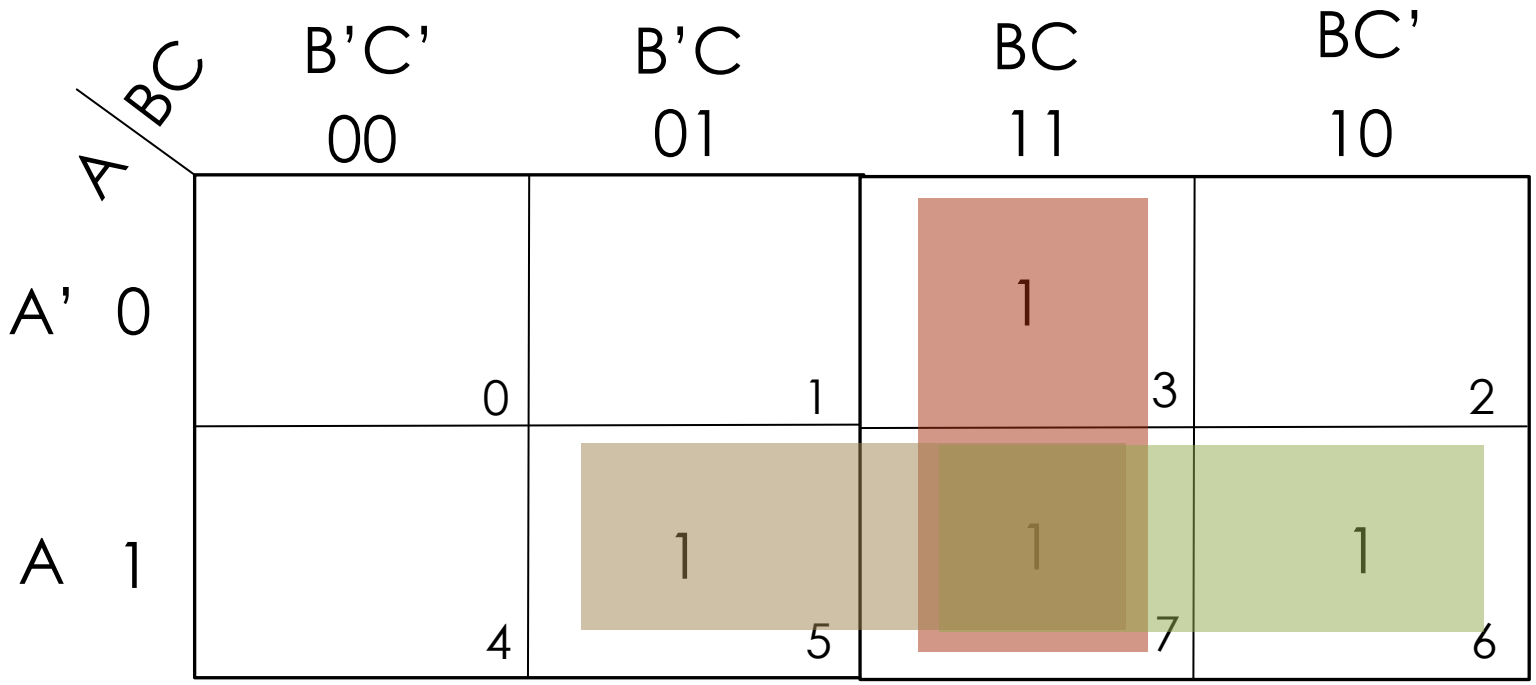
- Grouping



3 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Group Up
  - F(A, B, C) = Σm(3, 5, 6, 7)



3 Variable Karnaugh Map (K-Map) SOP

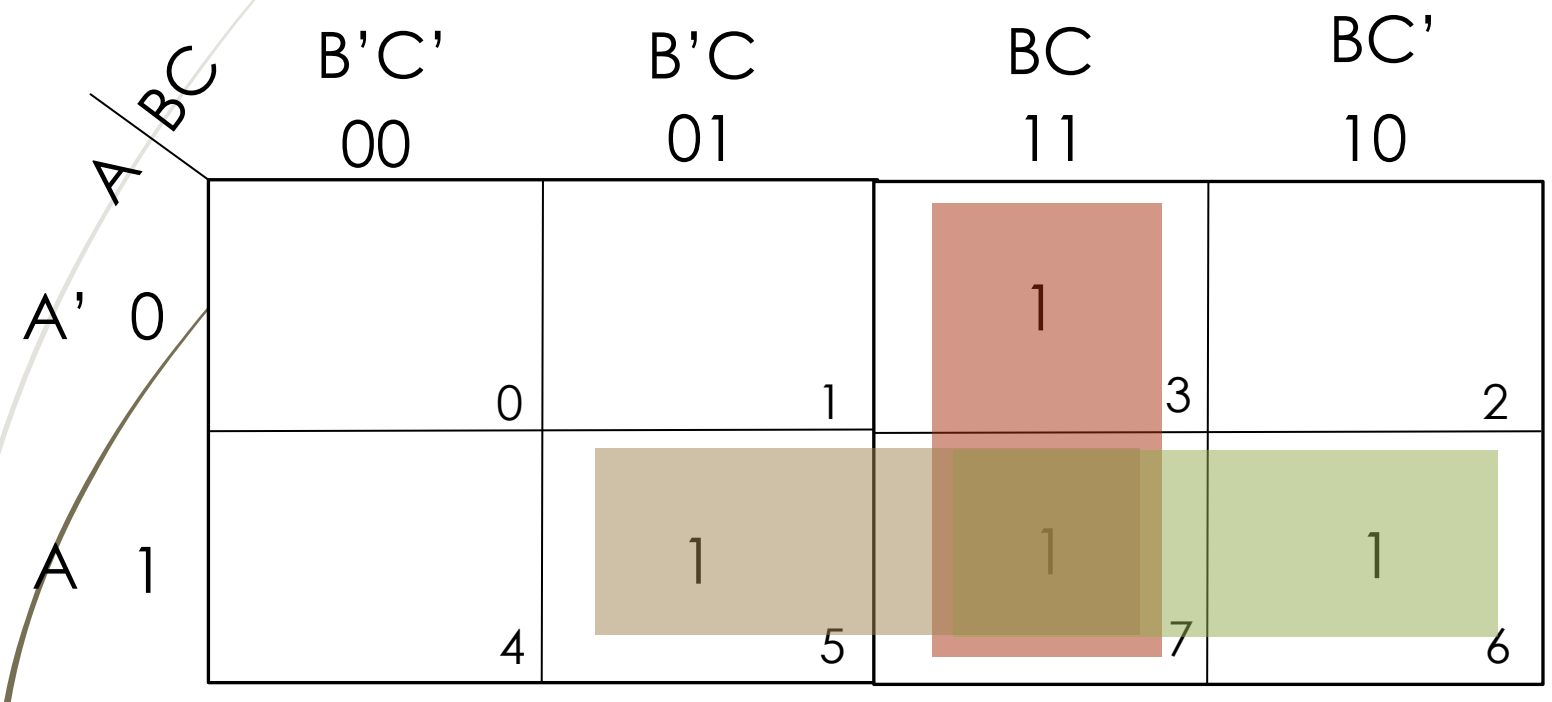
SLO 1.3.1 A

- Simplification outcomes
  - Group of 1 (single cell)→ No simplification (all 3 variables remain).
    - Example: Cell m2=A′BC′.
  - Group of 2 (two adjacent cells)→ Eliminates 1 variable (because it changes within the group).
    - Example: Group m2 (A’BC’) + m3 (A’BC) → expression = A′B.
  - Group of 4 (rectangle of 4 adjacent cells)→ Eliminates 2 variables.
    - Example: Group m4,m5,m6,m7 (entire row A=1) → expression = A.
  - Group of 8 (all cells are 1)→ Eliminates all 3 variables. Expression = 1 (always true).

3 Variable Karnaugh Map (K-Map) SOP

SLO 1.3.1 A

- Solve
  - F(A, B, C) = Σm(3, 5, 6, 7)

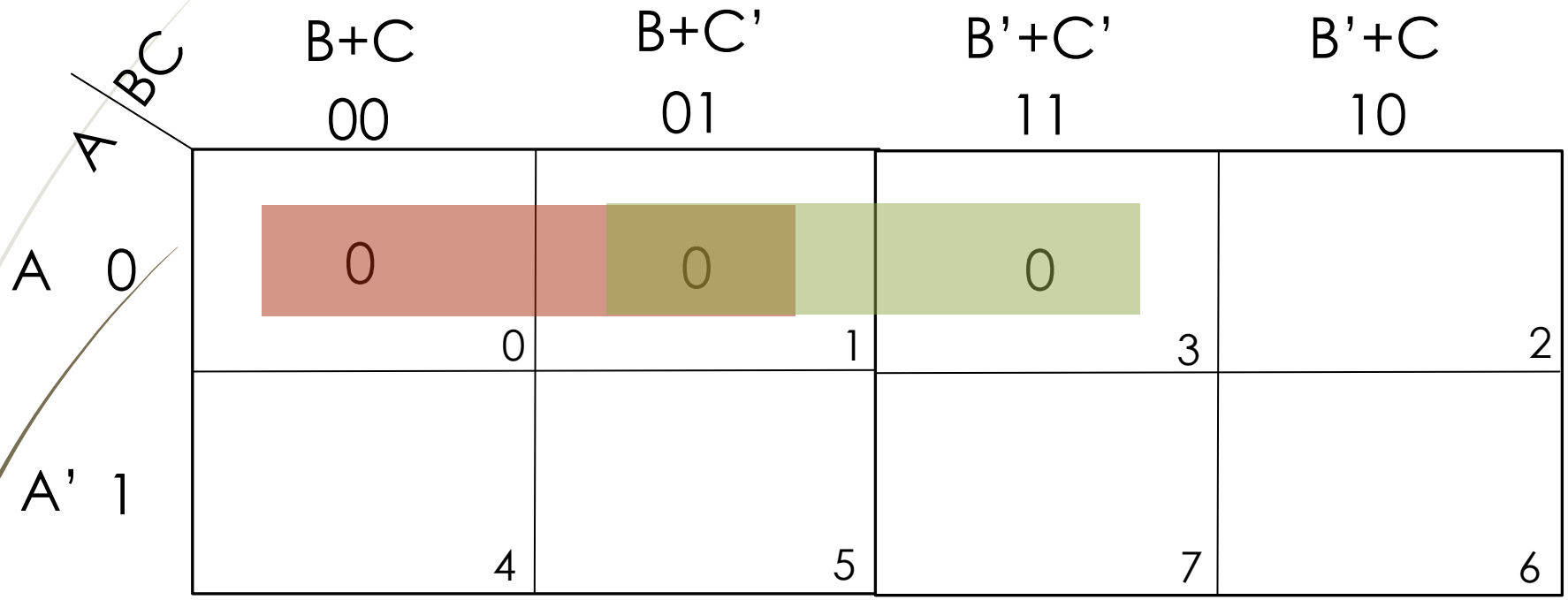


- Solution (Method 1)
  - G1: BC
  - G2: AC
  - G3: AB
- F = BC + AC + AB

3 Variable Karnaugh Map (K-Map) POS

SLO 1.3.1 A

- F(A, B, C) = (A+B+C') (A+B'+C') (A+B+C)



0 = A
1 = A'

- F(A, B, C) = (A+B) (A+C')

3 Variable Karnaugh Map (K-Map)

SLO 1.3.1 A

- Practice Equations
  - F(A,B,C)=A′B′C′+A′BC+AB′C
  - F(A,B,C)=Σm(1,3,5,7)
  - F(A,B,C)=Σm(0,2,4,6)
  - F(A,B,C)=Σm(1,2,3,7)
  - F(A,B,C)=ΠM(0,2,3)
  - F(A,B,C)=ΠM(0,1,6,7)
  - F(A,B,C)=ΠM(0,1,2,3,4,5,6,7)

SLO 1.3.2 A

convert the given algebraic expression into its Sum of Products (SOP) and Product of Sums (POS) forms;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.3.2 A

Convert the given algebraic expression into its SOP and POS forms

SOP for F(A,B,C)=AB′+BC

We must make sure every term has all three variables (A, B, C).

- Step A — Expand each product term
  - For AB′ : Add missing variable C:
    AB′ = AB′(C+C′) = AB′C+AB′C′ → Distributive Law
  - For BC: Add missing variable A:
    BC=BC(A+A′)=ABC+A′BC
- Step B — Collect terms
  F=AB′C+AB′C′+ABC+A′BC

SLO 1.3.2 A

Convert the given algebraic expression into its SOP and POS forms

SOP for F(A,B,C)=AB′+BC

- Step B — Collect terms
  F=AB′C+AB′C′+ABC+A′BC
- Step C — Index minterms
  - Write as minterms Σm().
    - AB′C → A=1, B=0, C=1 → binary 101 → decimal 5
    - AB′C′ → 100 → decimal 4
    - ABC → 111 → decimal 7
    - A′BC → 011 → decimal 3
    - F=Σm(3,4,5,7)

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

SLO 1.3.2 A

Convert the given algebraic expression into its SOP and POS forms

POS for F(A,B,C)=AB′+BC

- Step A — Build truth table quickly

A	B	C	AB'+BC	F
0	0	0	0+0=0	0
0	0	1	0+0=0	0
0	1	0	0+0=0	0
0	1	1	0+0=0	0
1	0	0	1+0=1	1
1	0	1	1+0=1	1
1	1	0	0+0=0	0
1	1	1	0+1=1	1

SLO 1.3.2 A

Convert the given algebraic expression into its SOP and POS forms

POS for F(A,B,C)=AB′+BC

- Step B — Write maxterms for zeros
  - Row 0: (A=0,B=0,C=0) → (A+B+C)
  - Row 1: (0,0,1) → (A+B+C′)
  - Row 2: (0,1,0) → (A+B′+C)
  - Row 6: (1,1,0) → (A′+B′+C)
- Step C — Product of all maxterms
  F=(A+B+C)(A+B+C′)(A+B′+C)(A′+B′+C)
- Step D — Product of all maxterms
  F=ΠM(0,1,2,6)

No	A	B	C	AB'+BC	F	Maxterm
0	0	0	0	0+0=0	0	A+B+C
1	0	0	1	0+0=0	0	A+B+C′
2	0	1	0	0+0=0	0	A+B′+C
3	0	1	1	0+0=0	0	
4	1	0	0	1+0=1	1	
5	1	0	1	1+0=1	1	
6	1	1	0	0+0=0	0	A′+B′+C
7	1	1	1	0+1=1	1	

1.4 Software Development

SLO	Students should be able to	Cognitive Level
1.4.1	explain Software Development Life Cycle (SDLC) and its different phases;	U
1.4.2	compare the phases, advantages and disadvantages of the following software development models: a. waterfall model, b. agile model;	U

SLO 1.4.1 U

explain Software Development Life Cycle (SDLC) and its different phases;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

1. Requirement Analysis

- Purpose:
  - To clearly understand and document what the software must do from the perspective of users, business needs, and stakeholders.
- Activities:
  - Conduct interviews, surveys, and workshops with stakeholders.
  - Gather functional requirements (what the system should do).
  - Gather non-functional requirements (performance, security, usability).
  - Analyze requirements for feasibility and clarity.
  - Prioritize requirements.
  - Create use cases and user stories.
  - Review and get approval on requirements.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

1. Requirement Analysis

- Deliverables:
  - Software Requirement Specification (SRS) document: Detailed description of functional and non-functional requirements.
  - Requirement Traceability Matrix (RTM): Maps requirements to test cases for verification.
- Importance:
  - Provides a foundation for all future phases.
  - Reduces risk of misunderstanding or missing key requirements.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

2. System Design

- Purpose:
  - Translate requirements into a blueprint for construction — defining system architecture and detailed design.
- Activities:
  - Define system architecture (client-server, layered, microservices, etc.).
  - Design databases: ER diagrams, data models.
  - Create module/component design: define how software parts interact.
  - Design user interfaces and user experience flows.
  - Specify hardware and software requirements.
  - Prepare detailed design documents:
    - High-Level Design (HLD): Overview of system modules and interactions.
    - Low-Level Design (LLD): Detailed algorithms, data structures, and interface design.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

2. System Design

- Deliverables:
  - Design Document Specification (DDS).
  - Data Flow Diagrams (DFDs), UML diagrams (class, sequence, activity).
  - Prototype or wireframes for user interfaces.
- Importance:
  - Guides developers in coding.
  - Helps identify potential design flaws early.
  - Enables reuse of components and consistency.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

3. Implementation (Coding)

- Purpose:
  - Actual development of software code following the design documents and standards.
- Activities:
  - Divide the work into tasks and assign to developers.
  - Write source code using appropriate programming languages.
  - Follow coding standards and conventions.
  - Use version control systems (e.g., Git).
  - Perform code reviews and pair programming.
  - Document code and create inline comments.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

3. Implementation (Coding)

- Deliverables:
  - Source code files.
  - Executable software builds.
  - Code documentation.
- Importance:
  - The heart of SDLC, turning design into a working system.
  - Quality code leads to maintainable and bug-free software.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

4. Testing

- Purpose:
  - To verify that the software works as intended and to detect defects before release.
- Activities:
  - Create test plans and test cases based on requirements.
  - Conduct different types of testing:
    - Unit Testing: Test individual components.
    - Integration Testing: Test combined components.
    - System Testing: Test the entire system as a whole.
    - Acceptance Testing: Validate against user requirements.
  - Perform regression testing to check for new bugs after changes.
  - Log defects and track their resolution.
  - Automate tests where possible.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

4. Testing

- Deliverables:
  - Test cases and test scripts.
  - Test reports and defect logs.
  - Acceptance sign-off documents.
- Importance:
  - Ensures software quality, reliability, and user satisfaction.
  - Prevents costly fixes after deployment.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

5. Deployment

- Purpose:
  - Make the software available to users in the production environment.
- Activities:
  - Plan deployment strategy (big bang, phased, parallel).
  - Set up production environment (servers, databases, networks).
  - Install and configure the software.
  - Conduct final validation tests.
  - Train users and provide documentation.
  - Monitor system after deployment for issues.
  - Plan rollback procedures if deployment fails.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

5. Deployment

- Deliverables:
  - Deployed software in production.
  - Deployment manuals and user guides.
  - Training materials.
- Importance:
  - The phase where software delivers real value.
  - Proper deployment ensures minimal disruption and quick adoption.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

6. Maintenance

- Purpose:
  - Support and enhance the software after delivery to keep it effective and up-to-date.
- Activities:
  - Fix bugs discovered post-deployment.
  - Update software to accommodate changes in environment or business needs.
  - Improve performance and usability.
  - Provide technical support.
  - Manage patches and upgrades.
  - Perform periodic reviews and audits.

Software Development Life Cycle (SDLC)

SLO 1.4.1 U

6. Maintenance

- Deliverables:
  - Bug fixes and patches.
  - Updated software versions/releases.
  - Support tickets and resolution reports.
- Importance:
  - Maintains software relevance and reliability over time.
  - Often consumes the largest portion of software lifecycle effort and cost.

SLO 1.4.2 U

compare the phases, advantages and disadvantages of the following software development models:

a. waterfall model,
b. agile model;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SDLC Model: Waterfall Model

SLO 1.4.2 U

- **Phases**:
  - Requirement Analysis: Gather and document all requirements upfront.
  - System Design: Create architecture and detailed design based on requirements.
  - Implementation (Coding): Develop the software as per the design.
  - Testing: Test the complete software after development.
  - Deployment: Deliver the final product to the user.
  - Maintenance: Perform bug fixes and updates post-deployment.
- Each phase must be completed before moving to the next; it follows a linear, sequential flow.

SDLC Model: Waterfall Model

SLO 1.4.2 U

- **Advantages**:
  - Simple and easy to understand and manage due to its linear approach.
  - Clear milestones and deliverables at each phase.
  - Well-suited for projects with well-defined requirements and low expected changes.
  - Documentation is thorough, which helps in maintenance and knowledge transfer.
  - Easy to measure progress by phase completion.

SDLC Model: Waterfall Model

SLO 1.4.2 U

- **Disadvantages**:
  - Inflexible to requirement changes once the project moves past the initial phases.
  - Late testing means bugs and issues are found late, potentially causing costly fixes.
  - Not ideal for complex or long-term projects with evolving requirements.
  - Customer involvement is minimal after the requirement phase.
  - Risk of delivering a product that does not meet current user needs due to changing market conditions.

SDLC Model: Agile Model

SLO 1.4.2 U

- **Phases** (Iterative and incremental; phases repeat in short cycles called sprints):
  - Concept / Planning: Identify high-level requirements and plan the first sprint.
  - Iteration/Increment (Repeated cycles):
    - Requirement Gathering: For the sprint only.
    - Design: Minimal, just enough for the sprint.
    - Implementation: Develop working software features.
    - Testing: Continuous testing within the sprint.
  - Release: Deliver a usable product increment after each sprint.
  - Review & Feedback: Collect user feedback to adapt the next sprint.
  - Maintenance: Ongoing throughout and after releases.

SDLC Model: Agile Model

SLO 1.4.2 U

- **Advantages**:
  - Highly flexible and adaptive to changing requirements.
  - Frequent deliveries provide early and continuous value to users.
  - Customer involvement throughout development improves satisfaction.
  - Bugs are detected and fixed early due to continuous testing.
  - Encourages collaboration and communication within cross-functional teams.
  - Reduces risk by breaking down development into manageable chunks.

SDLC Model: Agile Model

SLO 1.4.2 U

- **Disadvantages**:
  - Requires high customer involvement and commitment.
  - Less emphasis on comprehensive documentation, which can cause problems later.
  - Managing frequent changes can be challenging and may lead to scope creep.
  - Success depends on the skill and experience of the team.
  - Can be difficult to predict overall project timeline and costs precisely at the start.

SDLC Model: Waterfall vs Agile Model

SLO 1.4.2 U

Aspect	Waterfall Model	Agile Model
Approach	Linear, sequential phases	Iterative, incremental cycles (sprints)
Requirement Handling	Fixed upfront, hard to change	Evolving, flexible, adapts to changes
Customer Involvement	Limited, mostly at the start	Continuous throughout development
Testing	After implementation	Continuous during each iteration
Documentation	Comprehensive, formal	Minimal, just enough
Delivery	One final delivery	Frequent deliveries of working software
Risk Management	Higher risk due to late testing and inflexibility	Lower risk due to incremental releases
Project Size Suitability	Small to medium with stable requirements	Medium to large, complex, evolving projects
Advantages	Simple, clear milestones, well-documented	Flexible, fast feedback, customer-focused
Disadvantages	Inflexible, late bug detection, minimal user involvement	Requires skilled teams, can have scope creep

1.5 Network Topology

SLO	Students should be able to	Cognitive Level
1.5.1	define network topology;	R
1.5.2	explain the following network topologies with diagrams: a. bus, b. ring, c. star, d. tree, e. mesh, f. hybrid;	U
1.5.3	explain the advantages, disadvantages and applications of network topologies mentioned in SLO # 1.5.2;	U
1.5.4	compare scalability and reliability of network topologies;	U

SLO 1.5.1 R

define network topology;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Define Network Topology;

SLO 1.5.1 R

- Network topology refers to the physical or logical layout of how devices (nodes) such as computers, printers, routers, and switches are connected and communicate within a network.
- It describes the structure of a network, including how nodes are arranged and how data flows between them.

SLO 1.5.2 U

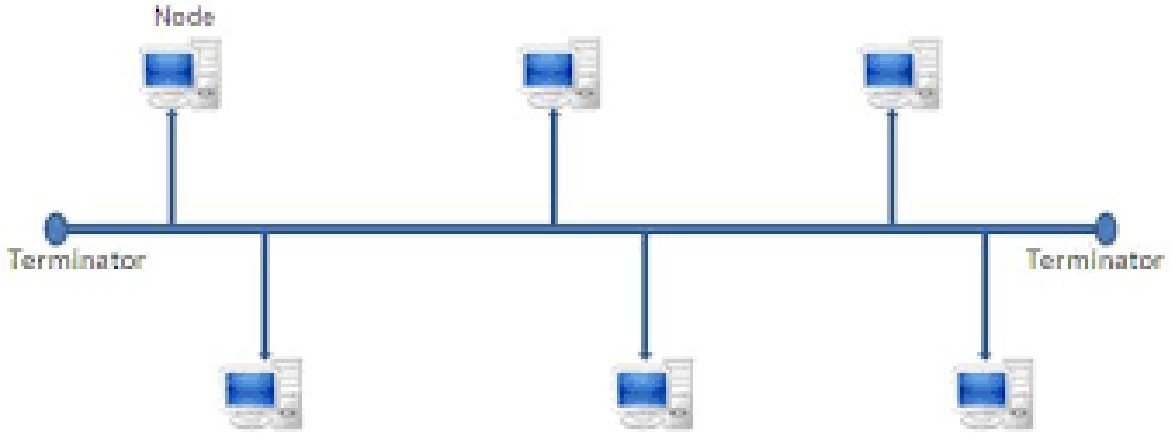
explain the following network topologies with diagrams:

a. bus,
b. ring,
c. star,
d. tree,
e. mesh,
f. hybrid;

Explain Network Topology : Bus

SLO 1.5.2 U

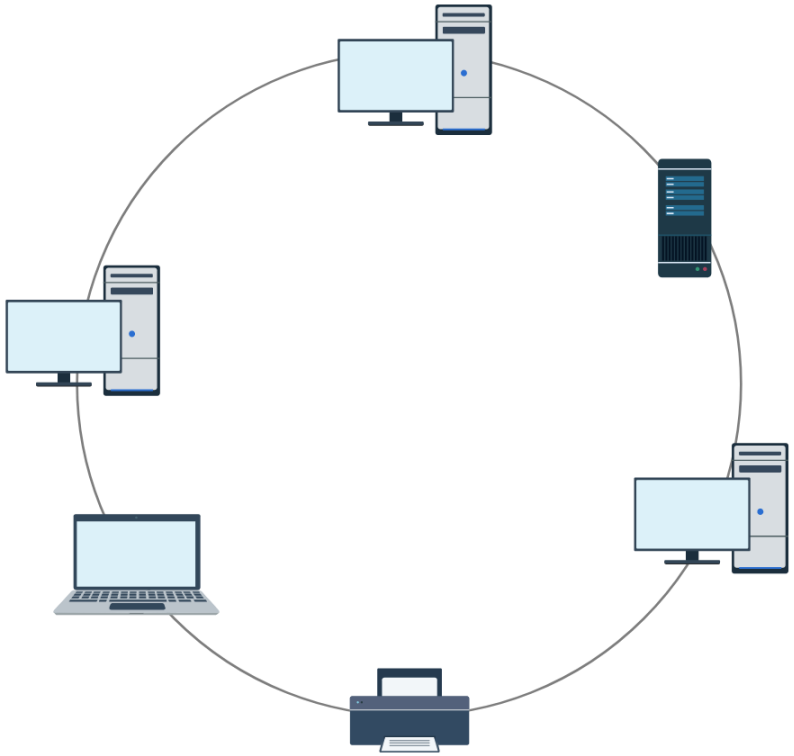
- Bus topology, also known as line topology, is a type of network topology in which all devices in the network are connected by one central RJ-45 network cable or coaxial cable.
- The single cable, where all data is transmitted between devices, is referred to as the bus, backbone, or trunk.



Explain Network Topology : Ring

SLO 1.5.2 U

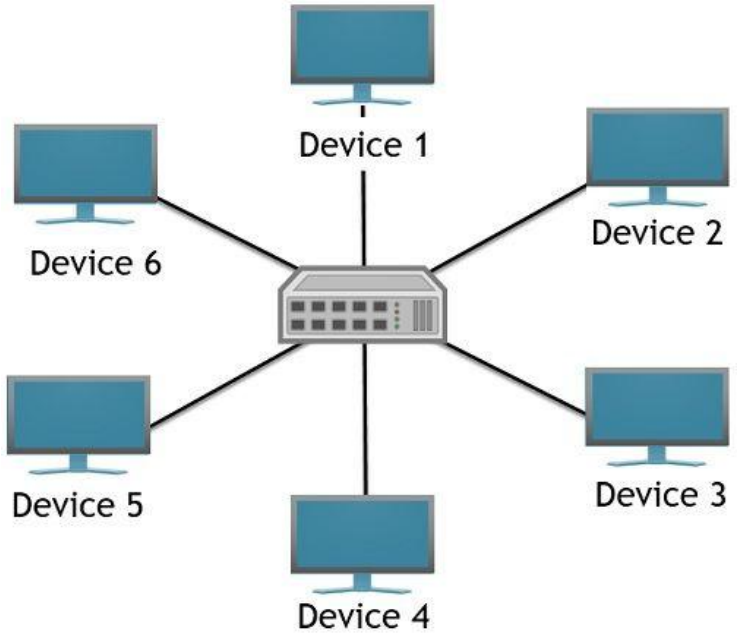
- Ring topology is a type of network topology in which each device is connected to two other devices on either side via an RJ-45 cable or coaxial cable.
- This forms a circular ring of connected devices which gives it its name.
- Data is commonly transferred in one direction along the ring, known as a unidirectional ring.
- The data is forwarded from one device to the next, until it reaches the intended destination.
- In a bidirectional ring, data can travel in either direction.



Explain Network Topology : Star

SLO 1.5.2 U

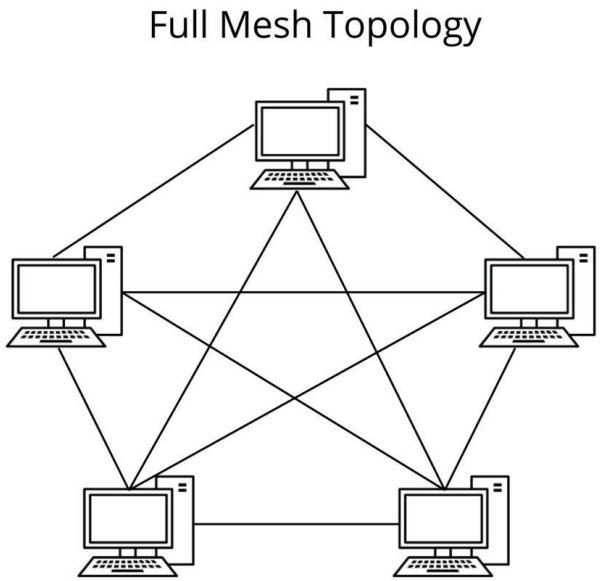
- Star topology is a type of network topology in which every device in the network is individually connected to a central node, known as the switch or hub.
- When represented visually, this topology resembles a star which gives it its name.



Explain Network Topology : Mesh

SLO 1.5.2 U

- Mesh topology is a type of network topology in which all devices in the network are interconnected.
- In a mesh topology, data can be transmitted by routing (sent the shortest distance) and flooding (sent to all devices).

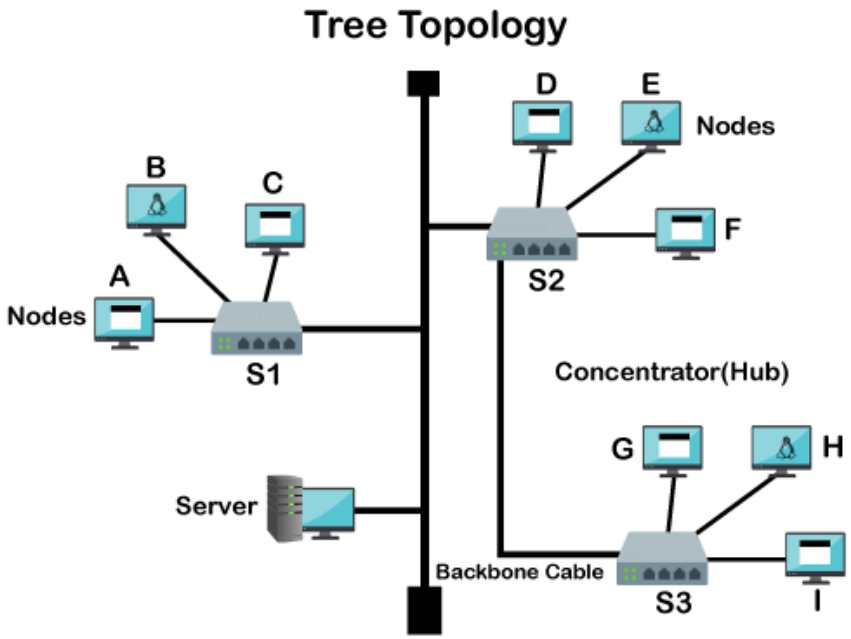


Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Explain Network Topology : Tree

SLO 1.5.2 U

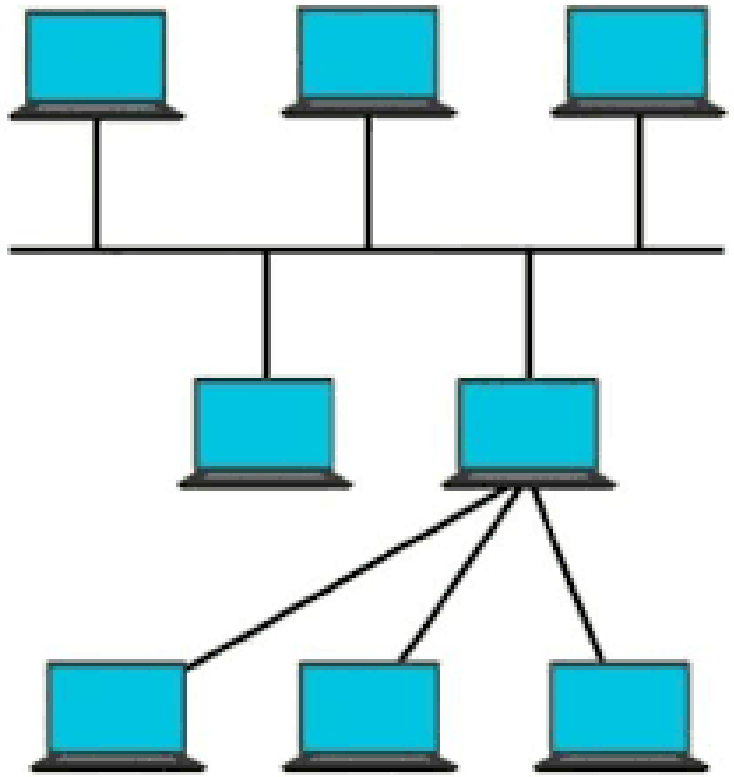
- Tree topology combines the characteristics of bus topology and star topology.
- A tree topology is a type of structure in which all the computers are connected with each other in hierarchical fashion.
- The top-most node in tree topology is known as a root node, and all other nodes are the descendants of the root node.
- There is only one path exists between two nodes for the data transmission. Thus, it forms a parent-child hierarchy.



Explain Network Topology : Hybrid

SLO 1.5.2 U

- A hybrid topology is defined as a network topology that combines two or more different network topologies.
- A hybrid topology can be a combination of bus topology, ring topology and mesh topology.
- The selection of different types of network topologies combined together depends upon the number of computers, their location, and the required performance.
- In the hybrid topology network sections consist of a configuration of different types of network topologies.
- The structure of hybrid topology is more complex but offers various advantages such as flexibility and fault tolerance.



SLO 1.5.3 U

explain the advantages, disadvantages and applications of network topologies mentioned in SLO # 1.5.2;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.5.3 U

Bus Topology

- **Advantages**
  - Works efficiently for small networks
  - Easy and cost-effective to install and add or remove devices
  - Doesn't require as much cabling as alternative topologies
  - If one device fails, other devices are not impacted
- **Disadvantages**
  - If the cable is damaged, the entire network will fail or be split
  - Difficult to troubleshoot problems
  - Very slow and not ideal for larger networks
  - Requires terminators at both ends of the cable to prevent bouncing signals that cause interference
  - Adding more devices and more network traffic decreases the entire network's performance
  - Low security due to all devices receiving the same signal from the source

Bus Topology

SLO 1.5.3 U

- **Applications**
  - Small office networks.
  - Temporary setups for demonstrations or exhibitions.
  - Older Ethernet LANs.

SLO 1.5.3 U

Ring Topology

- **Advantages**
  - Since data flows in one direction, the chance of a packet collision is reduced
  - A network server is not needed to control network connectivity
  - Devices can be added without impacting network performance
  - Easy to identify and isolate single points of failure
  - Better suited for high traffic environments than a bus topology
- **Disadvantages**
  - All data travelling over the network must pass through each device on its way to its destination, which can reduce performance
  - If one device fails, the entire network is impacted
  - Can be difficult to architect the necessary cabling
  - More expensive to implement than a bus topology

SLO 1.5.3 U

Ring Topology

- **Applications**
  - WANs using token ring protocols.
  - Fiber Distributed Data Interface (FDDI).
  - Campus backbone networks (in legacy systems).

Star Topology

SLO 1.5.3 U

- **Advantages**
  - Limits the impact of a single point of failure because each device is isolated by its relationship to the switch
  - Adding or removing devices to the network is simple and doesn't disrupt the network
  - High-performance as no data collisions can occur
  - Fault detection is easy
  - Each device only requires one port to connect to the switch
- **Disadvantages**
  - Requires more cabling and is more expensive than some alternatives
  - If the switch fails, all the connected devices are disabled
  - The switch requires more resources and maintenance
  - Performance is dependent on the switch

Star Topology

SLO 1.5.3 U

- **Applications**
  - Modern Ethernet LANs.
  - Corporate networks and server-client models.
  - Home networking.

Mesh Topology

SLO 1.5.3 U

- **Advantages**
  - Multiple devices can transmit data at the same time, allowing for high amounts of traffic
  - If one device fails, data transmission is not impacted in the rest of the network
  - Adding devices to the network does not disrupt data transmission
  - Troubleshooting is easier than with alternative topologies
- **Disadvantages**
  - Network installation and maintenance is time and resource intensive
  - High power requirement due to all the devices needing to remain active all the time
  - Requires a large amount of cables and ports
  - The potential for a large amount of redundant connections increases costs and reduces efficiency

Mesh Topology

SLO 1.5.3 U

- **Applications**
  - Military communication systems.
  - Banking systems and data centers.
  - Internet backbone and advanced wireless networks (e.g., Zigbee, IoT).

Tree Topology

SLO 1.5.3 U

- **Advantages**
  - Extension of bus and star topologies.
  - Expansion of nodes is possible and easy.
  - Easily managed and maintained.
  - Error detection is easily done.
- **Disadvantages**
  - Heavily cabled.
  - Costly.
  - If more nodes are added maintenance is difficult.
  - Central hub fails, network fails.

Tree Topology

SLO 1.5.3 U

- **Applications**
  - Large corporate and educational institution networks.
  - ISP infrastructures.
  - Departmental and campus-wide networks.

SLO 1.5.3 U

Hybrid Topology

- **Advantages**
  - Combines strengths of different topologies.
  - Flexible, scalable, and robust.
  - Can be customized to suit complex network needs.
- **Disadvantages**
  - Design and implementation are complex.
  - Costly to maintain due to varied hardware.
- **Applications**
  - Large enterprises with diverse networking needs.
  - Telecommunication networks.
  - Modern data centers and multi-site organizations.

SLO 1.5.4 U

compare scalability and reliability of network topologies;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Scalability and Reliability of Network Topologies

SLO 1.5.4 U

- 1. Bus Topology

Aspect	Description
Scalability	**Low** – Adding more devices can lead to signal degradation and performance issues.
Reliability	**Low** – A failure in the central bus cable disables the entire network.

Scalability and Reliability of Network Topologies

SLO 1.5.4 U

- 2. Ring Topology

Aspect	Description
Scalability	**Moderate** – Adding devices requires breaking the ring temporarily, which affects the network.
Reliability	**Low to Moderate** – One device or cable failure can disrupt the whole ring unless it's a dual ring system.

Scalability and Reliability of Network Topologies

SLO 1.5.4 U

- 3. Star Topology

Aspect	Description
Scalability	**High** – Devices can be easily added without affecting the network.
Reliability	**Moderate** – Individual device failure does not affect the network, but hub/switch failure causes total shutdown.

Scalability and Reliability of Network Topologies

SLO 1.5.4 U

- 4. Tree Topology

Aspect	Description
Scalability	**High** – Well-suited for expanding large hierarchical networks.
Reliability	**Moderate** – Failure in main/root node or backbone affects entire branches.

Scalability and Reliability of Network Topologies

SLO 1.5.4 U

- 5. Mesh Topology

Aspect	Description
Scalability	**Low to Moderate** – Adding devices requires many additional connections in full mesh; complex in large networks.
Reliability	**Very High** – Multiple redundant paths ensure network stays functional even if several nodes fail.

Scalability and Reliability of Network Topologies

SLO 1.5.4 U

- 6. Hybrid Topology

Aspect	Description
Scalability	**Very High** – Combines scalable parts like star and tree; highly adaptable.
Reliability	**High** – Can be designed for fault tolerance by combining resilient topologies.

Scalability and Reliability of Network Topologies

SLO 1.5.4 U

- Summary

Topology	Scalability	Reliability
Bus	Low	Low
Ring	Moderate	Low to Moderate
Star	High	Moderate
Tree	High	Moderate
Mesh	Low to Moderate	Very High
Hybrid	Very High	High

1.6 Cloud Computing

SLO	Students should be able to	Cognitive Level
1.6.1	explain cloud computing, its characteristics, and impact on modern IT infrastructure;	U
1.6.2	explain the following cloud computing service models with examples: a. Software as a Service (SaaS), b. Platform as a Service (PaaS), c. Infrastructure as a Service (IaaS);	U
1.6.3	explain the following cloud computing types: a. public cloud, b. private cloud, c. community cloud, d. hybrid cloud;	U

SLO 1.6.1 U

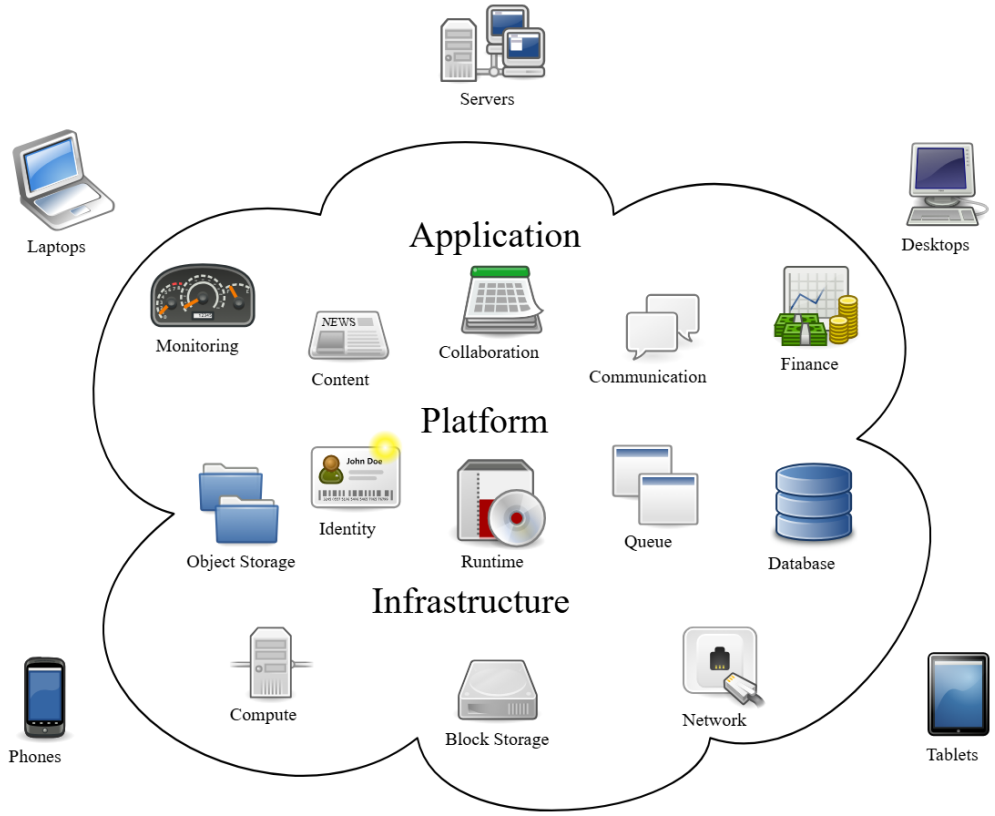
explain cloud computing, its characteristics, and impact on modern IT infrastructure;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Cloud Computing

SLO 1.6.1 U

- Cloud computing is the delivery of computing services such as servers, storage, databases, networking, software, and analytics over the internet ("the cloud") instead of using local servers or personal devices.
- Users can access these resources on-demand, pay only for what they use, and scale them according to their needs.



Characteristics of Cloud Computing

SLO 1.6.1 U

1. On-Demand Self-Service
   - Users can provision resources (e.g., storage, servers) automatically, without human intervention from the provider.
2. Broad Network Access
   - Services are available over the internet and can be accessed from anywhere via devices like laptops, smartphones, and tablets.
3. Resource Pooling
   - Cloud providers use a multi-tenant model to serve multiple users with dynamically assigned physical and virtual resources.

Characteristics of Cloud Computing

SLO 1.6.1 U

4. Rapid Elasticity
   - Resources can be scaled up or down automatically and quickly to meet changing demands.
5. Measured Service
   - Resource usage is monitored, controlled, and reported for transparency and pay-per-use pricing.
6. High Availability
   - Cloud services are designed to ensure continuous uptime and failover support.
7. Security
   - Built-in firewalls, encryption, authentication, and compliance tools are commonly provided.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Impact on Modern IT Infrastructure

SLO 1.6.1 U

1. Cost Efficiency
   - Reduces the need for investing in physical hardware and maintenance.
   - Moves IT spending from capital expenditure (CapEx) to operational expenditure (OpEx).
2. Scalability and Flexibility
   - Businesses can easily adjust computing resources to meet changing workloads without major investments.
3. Faster Deployment
   - Applications and services can be deployed in minutes rather than weeks or months.

Impact on Modern IT Infrastructure

SLO 1.6.1 U

4. Global Reach
   - Companies can deploy applications and services in data centers worldwide, ensuring fast access and redundancy.
5. Improved Collaboration
   - Cloud platforms support real-time collaboration and remote work environments, crucial for modern, distributed teams.
6. Disaster Recovery and Backup
   - Cloud services often include automated data backup, replication, and disaster recovery features.
7. Innovation Enablement
   - Cloud enables faster experimentation with emerging technologies like AI, big data analytics, and IoT due to ready-to-use platforms and tools.

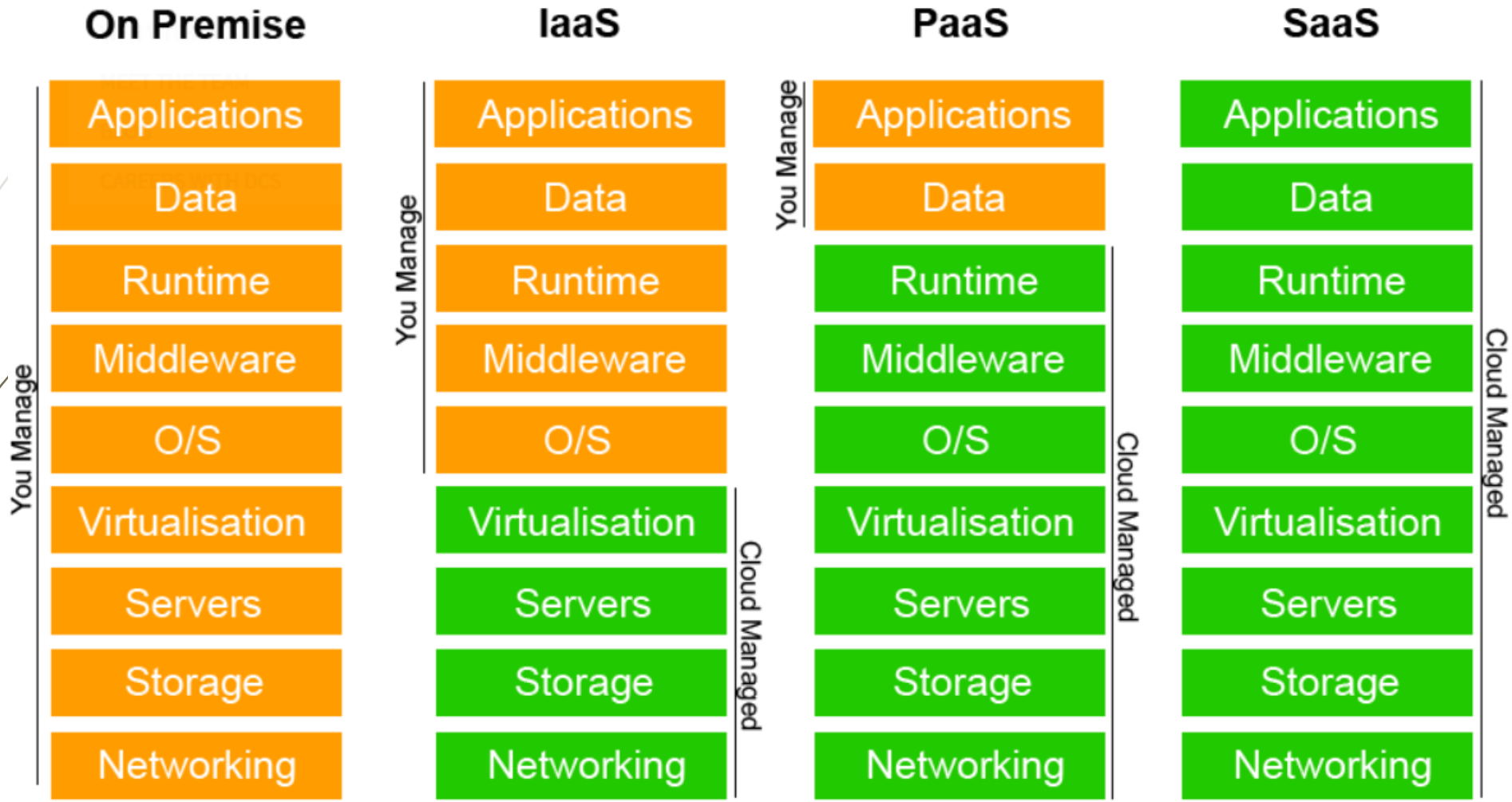
SLO 1.6.2 U

explain the following cloud computing service models with examples:

a. Software as a Service (SaaS),
b. Platform as a Service (PaaS),
c. Infrastructure as a Service (IaaS);

Cloud Computing Model

SLO 1.6.2 U



Cloud Computing Model : IaaS

SLO 1.6.2 U

c. Infrastructure as a Service (IaaS)

- **Architecture**: Virtualized hardware resources are provided via hypervisors (e.g., Xen, KVM, VMware ESXi) running on physical servers in data centers.
- **Resources Provided**:
  - Compute (virtual machines or bare metal)
  - Storage (block storage, object storage)
  - Networking (virtual private clouds, load balancers, firewalls)
  - Other services: IP addresses, DNS, VPN
- **User Control**: Users have full control over operating systems, installed software, middleware, security policies, and configurations.
- **Provisioning**: Resources are provisioned through web portals, APIs, or command-line tools.
- **Scalability**: Users can scale compute and storage on demand, often with automation scripts or orchestration tools.

Cloud Computing Model : IaaS

SLO 1.6.2 U

c. Infrastructure as a Service (IaaS)

- **Security**: Users are responsible for OS and application security, while providers secure the physical infrastructure.
- **Networking**: Virtual networks can be created with subnets, routing tables, security groups, and VPN tunnels.
- **Storage Types**:
  - Block storage (e.g., Amazon EBS) for VM disks
  - Object storage (e.g., Amazon S3) for unstructured data
  - File storage (e.g., Amazon EFS) for shared file systems
- **Examples** of Technologies:
  - Hypervisors: VMware ESXi, Microsoft Hyper-V, KVM
  - Virtual Machine Managers and APIs: OpenStack, AWS EC2, Azure Virtual Machines
  - Software-defined Networking (SDN) and Network Functions Virtualization (NFV)

Cloud Computing Model : PaaS

SLO 1.6.2 U

b. Platform as a Service (PaaS)

- **Architecture**: PaaS abstracts the infrastructure layer but exposes development platforms including runtime environments, middleware, and databases.
- **Components Provided**:
  - Runtime environment: Java, .NET, Python, Node.js, etc.
  - Middleware: Message queues, caching layers, integration tools.
  - Databases: Managed SQL/NoSQL instances.
  - Development tools: Source control, CI/CD pipelines, testing frameworks.
- **Developer Control**: Developers focus on writing and deploying application code without managing servers, OS patches, or storage.
- **Scalability & Load Balancing**: Built-in auto-scaling and load balancing managed by the platform.
- **Integration**: Provides APIs and connectors for third-party services and on-premise systems.

Cloud Computing Model : PaaS

SLO 1.6.2 U

b. Platform as a Service (PaaS)

- **Security**: Includes environment isolation, role-based access control, and encryption.
- **Deployment Models**: Supports containers (Docker, Kubernetes) and serverless functions.
- **Examples** of Technologies:
  - Containers and orchestration: Docker, Kubernetes
  - Managed databases: Azure SQL Database, Amazon RDS
  - Development pipelines: Jenkins, GitLab CI/CD integrated with the platform

Cloud Computing Model : SaaS

SLO 1.6.2 U

a. Software as a Service (SaaS)

- **Architecture**: SaaS applications are hosted centrally on cloud servers and accessed via the internet, usually through web browsers or thin clients. Multi-tenancy is common, meaning a single instance of the software serves multiple users with data isolation.
- **Deployment**: Delivered over HTTP/HTTPS protocols. Often designed with REST APIs, microservices, or serverless backends.
- **Data Management**: Data is stored in cloud databases (SQL or NoSQL), managed by the SaaS provider. Users have limited control over backend infrastructure.
- **Security**: Providers implement authentication (OAuth, SAML), encryption (TLS in transit, AES at rest), and compliance (GDPR, HIPAA).

Cloud Computing Model : SaaS

SLO 1.6.2 U

a. Software as a Service (SaaS)

- **Customization**: Limited user customization (themes, settings) but core codebase is shared.
- **Examples** of Underlying Technologies:
  - Web servers: Apache, Nginx
  - Application frameworks: Django, Ruby on Rails, Node.js
  - Databases: PostgreSQL, MongoDB, DynamoDB
  - Cloud providers: AWS, Azure, Google Cloud

SLO 1.6.3 U

explain the following cloud computing types:

a. public cloud,
b. private cloud,
c. community cloud,
d. hybrid cloud;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Cloud Computing Types: Public Cloud

SLO 1.6.3 U

a. Public Cloud

- Definition: A public cloud is a cloud infrastructure that is owned and operated by third-party cloud service providers and made available to the general public or large industry groups over the internet.
- Characteristics:
  - Resources (servers, storage) are shared among multiple users (multi-tenant).
  - Pay-as-you-go pricing model.
  - Scalable and flexible.
  - Managed entirely by the cloud provider.
- Examples:
  - Amazon Web Services (AWS)
  - Microsoft Azure
  - Google Cloud Platform (GCP)

Cloud Computing Types: Public Cloud

SLO 1.6.3 U

a. Public Cloud

- Advantages:
  - Cost-effective (no need to invest in hardware).
  - Easy to scale resources up or down.
  - No maintenance responsibility for users.
- Disadvantages:
  - Less control over security and data privacy.
  - Possible compliance challenges for sensitive data.

Cloud Computing Types: Private Cloud

SLO 1.6.3 U

b. Private Cloud

- Definition: A private cloud is a cloud infrastructure operated solely for a single organization, either managed internally or by a third party, and hosted on-premises or off-premises.
- Characteristics:
  - Dedicated resources for one organization.
  - Greater control over security, data, and compliance.
  - Can be customized to specific organizational needs.
- Examples:
  - VMware vSphere private cloud
  - OpenStack private cloud
  - Microsoft Azure Stack

Cloud Computing Types: Private Cloud

SLO 1.6.3 U

b. Private Cloud

- Advantages:
  - Enhanced security and privacy.
  - More control over infrastructure and policies.
  - Can meet strict regulatory requirements.
- Disadvantages:
  - Higher costs due to dedicated hardware and management.
  - Requires IT expertise to maintain.
  - Less scalability compared to public cloud.

Cloud Computing Types: Community Cloud

SLO 1.6.3 U

c. Community Cloud

- Definition: A community cloud is a cloud infrastructure shared by several organizations with common concerns (security, compliance, policy) and jointly managed by them or a third party.
- Characteristics:
  - Shared infrastructure among organizations with similar interests.
  - Supports collaborative projects or shared services.
  - Can be on-premises or hosted by a third party.
- Examples:
  - Government agencies sharing cloud infrastructure.
  - Healthcare organizations collaborating on patient data systems.

Cloud Computing Types: Community Cloud

SLO 1.6.3 U

c. Community Cloud

- Advantages:
  - Shared costs among organizations.
  - Tailored to meet specific community needs and compliance.
  - Improved collaboration.
- Disadvantages:
  - Limited scalability compared to public cloud.
  - Potential conflicts in management and governance.

Cloud Computing Types: Hybrid Cloud

SLO 1.6.3 U

d. Hybrid Cloud

- Definition: A hybrid cloud combines two or more cloud types (public, private, or community), allowing data and applications to be shared between them.
- Characteristics:
  - Offers flexibility to run workloads in the most appropriate environment.
  - Enables data and application portability.
  - Often integrates on-premises infrastructure with cloud services.
- Examples:
  - A company uses a private cloud for sensitive data and public cloud for less critical workloads.
  - Disaster recovery setups where backup is stored in the public cloud.

Cloud Computing Types: Hybrid Cloud

SLO 1.6.3 U

d. Hybrid Cloud

- Advantages:
  - Balances scalability, cost, and security.
  - Allows gradual migration to cloud environments.
  - Enhances business continuity and disaster recovery.
- Disadvantages:
  - Complex to manage and integrate multiple environments.
  - Security risks if not properly configured.
  - Requires expertise to orchestrate seamless operation.

1.7 Understanding Cybersecurity Fundamentals

SLO	Students should be able to	Cognitive Level
1.7.1	describe cybersecurity and its importance;	U
1.7.2	explain the following cybersecurity fundamentals: a. confidentiality, b. integrity, c. availability;	U

SLO 1.7.1 U

describe cybersecurity and its importance;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Cybersecurity

SLO 1.7.1 U

- Cybersecurity refers to the practice of protecting computer systems, networks, software, and data from digital attacks, unauthorized access, damage, or theft.
- It encompasses technologies, processes, and controls designed to safeguard the confidentiality, integrity, and availability of information.

Cybersecurity: Importance

SLO 1.7.1 U

1. Protection of Sensitive Data
   - Prevents unauthorized access to personal, financial, and business data.
   - Safeguards customer information and intellectual property.
2. Maintaining Privacy
   - Ensures personal and organizational data is not exposed or misused.
   - Helps comply with privacy regulations (GDPR, HIPAA, etc.).
3. Ensuring Business Continuity
   - Protects against cyberattacks that can disrupt operations (e.g., ransomware).
   - Minimizes downtime and financial loss.
4. Safeguarding Reputation
   - Prevents data breaches that damage customer trust and brand image.
   - Helps maintain credibility in competitive markets.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Cybersecurity: Importance

SLO 1.7.1 U

5. Preventing Financial Loss
   - Avoids costs related to data breaches, legal penalties, and remediation.
   - Reduces risk of fraud, theft, and cyber extortion.
6. Protecting National Security
   - Shields critical infrastructure and government systems from cyber warfare and espionage.
   - Supports the security of communication, defense, and public services.
7. Supporting Safe Innovation
   - Enables the safe adoption of emerging technologies (cloud, IoT, AI).
   - Encourages secure digital transformation and business growth.

SLO 1.7.2 U

explain the following cybersecurity fundamentals:

a. confidentiality,
b. integrity,
c. availability;

Cybersecurity Fundamentals: Confidentiality

SLO 1.7.2 U

a. Confidentiality

- **Definition**: Confidentiality ensures that sensitive information is accessed only by authorized individuals or systems and is protected from unauthorized disclosure.
- **Purpose**:
  - Prevents data leaks or exposure to unauthorized parties.
  - Maintains privacy of personal and organizational information.
- **Methods to Ensure Confidentiality**:
  - Encryption (data at rest and in transit)
  - Access controls and permissions
  - Authentication mechanisms (passwords, biometrics)
  - Network security (firewalls, VPNs)

Cybersecurity Fundamentals: Integrity

SLO 1.7.2 U

b. Integrity

- **Definition**: Integrity guarantees that data is accurate, complete, and unaltered during storage, processing, and transmission, except by authorized parties.
- **Purpose**:
  - Prevents unauthorized modification or deletion of data.
  - Ensures trustworthiness and reliability of information.
- **Methods to Ensure Integrity**:
  - Hash functions and checksums
  - Digital signatures
  - Access controls and audit trails
  - Version control and backups

Cybersecurity Fundamentals: Availability

SLO 1.7.2 U

c. Availability

- **Definition**: Availability ensures that information and resources are accessible and usable by authorized users when needed.
- **Purpose**:
  - Prevents downtime or denial of service that disrupts operations.
  - Ensures continuous access to critical systems and data.
- **Methods to Ensure Availability**:
  - Redundant systems and backups
  - Load balancing and failover mechanisms
  - Regular maintenance and patching
  - Protection against DoS (Denial of Service) attacks

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

1.8 Common Cybersecurity Threats

SLO	Students should be able to	Cognitive Level
1.8.1	explain the following cybersecurity threats and their impact: a. malware, b. phishing, c. pharming, d. Denial of Service (DoS), e. Distributed Denial of Service (DDoS), f. ransomware;	U
1.8.2	recommend effective measures to safeguard against cybersecurity threats to enhance digital security;	E
1.8.3	design a multi-layered security strategy for a given situation that integrates preventive, detective, and corrective measures to mitigate the risks associated with DoS, DDoS, and ransomware attacks;	C
1.8.4	create a cybersecurity awareness campaign highlighting common threats and effective prevention methods for your community;	FA

SLO 1.8.1 U

explain the following cybersecurity threats and their impact:

a. malware,
b. phishing,
c. pharming,
d. Denial of Service (DoS),
e. Distributed Denial of Service (DDoS),
f. ransomware;

Cybersecurity Threats and Their Impact

SLO 1.8.1 U

a. Malware

- **Definition**: Malware (malicious software) is any software designed to harm, exploit, or damage computers, networks, or data.
- **Types**: Viruses, worms, Trojans, spyware, adware, ransomware.
- **Impact**:
  - Data theft or destruction.
  - System slowdown or crashes.
  - Unauthorized access/control of systems.
  - Financial loss and reputational damage.

Cybersecurity Threats and Their Impact

SLO 1.8.1 U

b. Phishing

- **Definition**: Phishing is a social engineering attack where attackers impersonate trustworthy entities (e.g., banks, companies) to trick victims into revealing sensitive information like passwords or credit card numbers.
- **Impact**:
  - Identity theft and fraud.
  - Unauthorized access to accounts.
  - Financial loss.
  - Spread of malware via malicious links or attachments.

Cybersecurity Threats and Their Impact

SLO 1.8.1 U

c. Pharming

- **Definition**: Pharming redirects users from legitimate websites to fraudulent ones without their knowledge, usually by exploiting vulnerabilities in DNS servers or infecting user computers.
- **Impact**:
  - Theft of login credentials and personal information.
  - Financial fraud.
  - Loss of user trust in legitimate websites.

Cybersecurity Threats and Their Impact

SLO 1.8.1 U

d. Denial of Service (DoS)

- **Definition**: A DoS attack aims to make a system, network, or website unavailable to users by overwhelming it with excessive traffic or requests.
- **Impact**:
  - Service downtime and disruption.
  - Loss of revenue and productivity.
  - Damage to brand reputation.

Cybersecurity Threats and Their Impact

SLO 1.8.1 U

e. Distributed Denial of Service (DDoS)

- **Definition**: A DDoS attack is similar to DoS but launched from multiple compromised devices (botnets) simultaneously, making it harder to block.
- **Impact**:
  - Extended and more severe service outages.
  - Greater financial and operational damage.
  - Difficulty in mitigation.

Cybersecurity Threats and Their Impact

SLO 1.8.1 U

f. Ransomware

- **Definition**: Ransomware encrypts a victim's files or system and demands a ransom payment for decryption keys.
- **Impact**:
  - Loss of critical data access.
  - Significant financial cost to recover or pay ransom.
  - Disruption of operations and services.
  - Potential legal and compliance issues.

SLO 1.8.2 E

recommend effective measures to safeguard against cybersecurity threats to enhance digital security;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Measures to Safeguard against Cybersecurity

SLO 1.8.2 E

1. Use Strong Authentication
   - Implement multi-factor authentication (MFA) combining passwords with tokens, biometrics, or SMS codes.
   - Enforce strong, unique passwords and change them regularly.
2. Keep Software Updated
   - Regularly update operating systems, applications, and security software to patch vulnerabilities.
   - Enable automatic updates wherever possible.
3. Install and Maintain Antivirus/Antimalware
   - Use reputable antivirus software to detect and remove malware.
   - Schedule regular scans and real-time protection.
4. Educate and Train Users
   - Conduct awareness programs about phishing, social engineering, and safe internet practices.
   - Teach users how to identify suspicious emails and websites.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Measures to Safeguard against Cybersecurity

SLO 1.8.2 E

5. Implement Firewalls and Intrusion Detection/Prevention Systems
   - Use network firewalls to control incoming and outgoing traffic.
   - Deploy IDS/IPS to monitor and block malicious activities.
6. Encrypt Data
   - Encrypt sensitive data both at rest and in transit.
   - Use secure protocols like HTTPS, TLS, and VPNs.
7. Backup Data Regularly
   - Maintain regular, secure backups of critical data.
   - Test backups to ensure data can be restored quickly in case of ransomware or data loss.
8. Limit Access and Privileges
   5. Follow the principle of least privilege (PoLP), granting users only the access needed.
   6. Use role-based access control (RBAC) and regularly review permissions.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Measures to Safeguard against Cybersecurity

SLO 1.8.2 E

9. Secure Network Infrastructure
   - Segment networks to limit the spread of attacks.
   - Use strong Wi-Fi encryption (WPA3) and change default router passwords.
10. Monitor and Respond
   - Continuously monitor systems and networks for suspicious activities.
   - Have an incident response plan to act quickly on breaches.
11. Use Secure Software Development Practices
   - Incorporate security into the software development life cycle (SDLC).
   - Perform regular code reviews, vulnerability assessments, and penetration testing.
12. Legal and Compliance
   - Stay compliant with relevant cybersecurity regulations (GDPR, HIPAA, PCI DSS).
   - Conduct regular audits and risk assessments.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.8.3 C

design a multi-layered security strategy for a given situation that integrates preventive, detective, and corrective measures to mitigate the risks associated with DoS, DDoS, and ransomware attacks;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Design a Multi-layered Security Strategy

SLO 1.8.3 C

Scenario:

- A mid-sized online business wants to protect its web services and data from DoS/DDoS and ransomware threats.

Design a Multi-layered Security Strategy

SLO 1.8.3 C

1. Preventive Measures

- Aim: Stop or reduce attack impact before it happens.
- Network Protection:
  - Deploy firewalls and next-generation firewalls (NGFW) with DoS/DDoS mitigation capabilities.
  - Use rate limiting and traffic filtering to block excessive requests.
  - Employ Content Delivery Networks (CDNs) and Web Application Firewalls (WAFs) to absorb traffic spikes and filter malicious payloads.
- Endpoint Protection:
  - Install and update anti-malware and anti-ransomware solutions on all endpoints.
  - Use application whitelisting to prevent unauthorized software execution.
- Access Control:
  - Implement strong authentication (MFA) and limit admin privileges.
  - Harden network devices and disable unused ports and services.
- Patch Management:
  - Keep all software and firmware updated to close vulnerabilities attackers exploit.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Design a Multi-layered Security Strategy

SLO 1.8.3 C

2. Detective Measures

- Aim: Identify and alert on suspicious or malicious activity quickly.
- Intrusion Detection Systems (IDS) / Intrusion Prevention Systems (IPS):
  - Monitor network traffic for signatures or anomalies related to DoS/DDoS and ransomware.
  - Automatically block or alert on suspicious traffic.
- Network Traffic Analysis:
  - Use tools to analyze traffic patterns, detecting abnormal spikes or unusual connection attempts.
- Endpoint Detection and Response (EDR):
  - Continuously monitor endpoints for ransomware-like behaviors (e.g., rapid file encryption).
- Log Monitoring & SIEM:
  - Centralize logs from firewalls, servers, and applications.
  - Use Security Information and Event Management (SIEM) tools to correlate events and trigger alerts.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Design a Multi-layered Security Strategy

SLO 1.8.3 C

3. Corrective Measures

- Aim: Respond to and recover from attacks to minimize damage.
- Incident Response Plan:
  - Have a documented and tested plan specific to DoS/DDoS and ransomware incidents.
  - Define roles, communication channels, and steps to contain and mitigate attacks.
- Traffic Diversion & Blackholing:
  - Use scrubbing centers or traffic diversion services during DDoS to filter out malicious traffic.
- Data Backup & Recovery:
  - Maintain frequent, offline, and immutable backups of critical data.
  - Test backup restoration procedures regularly to ensure ransomware recovery readiness.
- System Isolation:
  - Quickly isolate infected endpoints or network segments to prevent ransomware spread.
- Patch and Harden Post-Incident:
  - Analyze attack vectors and immediately patch or mitigate vulnerabilities exploited.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.8.4 FA

create a cybersecurity awareness campaign highlighting common threats and effective prevention methods for your community;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Cybersecurity Awareness Campaign

SLO 1.8.4 FA

- To raise awareness about common cybersecurity threats, promote safe digital habits, and foster a security-conscious culture within the organization.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

1.9 Data Encryption

SLO	Students should be able to	Cognitive Level
1.9.1	describe encryption and its importance;	U
1.9.2	relate the terms cryptography and encryption;	U
1.9.3	compare the following types of encryption techniques in terms of speed, key management, and use cases: a. symmetric, b. asymmetric;	U

SLO 1.9.1 U

describe encryption and its importance;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Encryption and its Importance

SLO 1.9.1 U

- Encryption is the process of converting readable data (plaintext) into an unreadable format (ciphertext) using algorithms and keys.
- This transformation ensures that only authorized parties who have the correct decryption key can convert the data back into its original form.

Encryption and its Importance

SLO 1.9.1 U

1. Data Confidentiality: Ensures sensitive information like personal data, financial details, and trade secrets cannot be accessed by unauthorized users.
2. Data Integrity: Helps detect if data has been altered or tampered with during transmission or storage.
3. Secure Communication: Enables safe transmission of information over insecure networks (e.g., internet, Wi-Fi).
4. Authentication: Supports verifying identities and preventing impersonation through digital signatures and certificates.
5. Compliance: Helps organizations meet legal and regulatory requirements (e.g., GDPR, HIPAA) for protecting sensitive data.
6. Protect Against Cyber Threats: Mitigates risks from hacking, eavesdropping, and data breaches.

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

SLO 1.9.2 U

relate the terms cryptography and encryption;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Relate the terms Cryptography and Encryption;

SLO 1.9.2 U

Cryptography

- Definition: Cryptography is the science and study of techniques for securing communication and data from adversaries. It encompasses a broad set of methods for ensuring confidentiality, integrity, authentication, and non-repudiation.
- Scope: Includes various techniques such as encryption, decryption, hashing, digital signatures, key exchange, and more.

Relate the terms Cryptography and Encryption;

SLO 1.9.2 U

Encryption

- Definition: Encryption is a specific process within cryptography that transforms readable data (plaintext) into an unreadable format (ciphertext) to protect it from unauthorized access.
- Role: Encryption is one of the fundamental tools or methods used in cryptography to achieve data confidentiality.

SLO 1.9.3 U

compare the following types of encryption techniques in terms of speed, key management, and use cases:

a. symmetric,
b. asymmetric;

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Compare Types of Encryption

SLO 1.9.3 U

- Symmetric Encryption: Same key is used for encryption and decryption (e.g., AES, DES).
- Asymmetric Encryption: Uses a pair of keys (public and private). Public key encrypts, private key decrypts (e.g., RSA, ECC).

Compare Types of Encryption

SLO 1.9.3 U

Criteria	Symmetric Encryption	Asymmetric Encryption
Speed	Very fast; suitable for encrypting large amounts of data	Slower; computationally intensive due to complex algorithms
Key Management	Single shared secret key used for both encryption and decryption; key distribution can be challenging and risky if not secure	Uses a pair of keys (public and private); public key can be openly shared, private key kept secret, simplifying secure key distribution
Use Cases	- Encrypting bulk data (files, disks, databases)	

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903

Prepared By: Naseer Ahmad (Nusrat Jahan Boys College) 0333-9790903