

PRACTICAL-1

AIM: To Verify the Behavior of Logic Gates using Truth Table and Familiarization with Digital Integrated Circuits

- To verify the behaviour of logic gates using truth table and familiarization with digital integrated circuits:
- A logic gate is a digital gate that allows data to be manipulated. It is used to determine whether or not to pass a signal. Logic gates, on the other hand, govern the flow of information based on a set of rules.
- The logic gates can be classified into the following major types.

i). Basic Logic gates

- a). AND Gate
- b). OR Gate
- c). NOT Gate

ii). Universal Logic gates

- a). NOR Gate
- b). NAND Gate

iii). Derived Logic gates

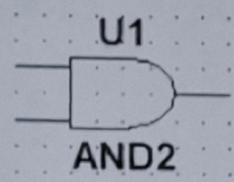
- a). XOR Gate
- b). XNOR Gate

- Let us now discuss each of these types of logic gates in detail one-by-one with its Boolean expression, Truth Table and Logic diagram.

i). AND Gate :-

$$y = A \cdot B$$

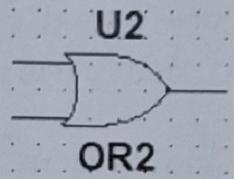
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



ii). OR Gate :-

$$y = A + B$$

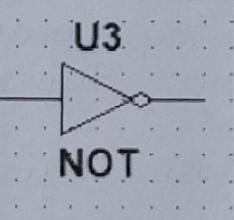
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



iii). NOT Gate :-

$$y = \bar{A}$$

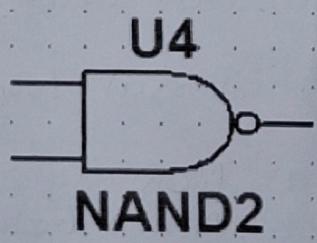
A	B	Y
0	-	1
1	-	0



iv). NAND Gate :-

$$y = \overline{A \cdot B}$$

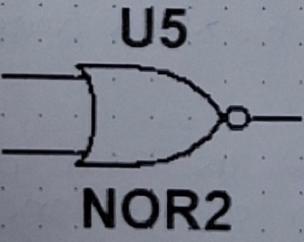
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	1



v). NOR Gate :-

$$y = \overline{A+B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

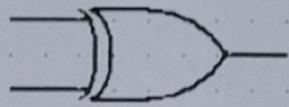


vii). XOR Gate :-

$$y = A \oplus B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

U6



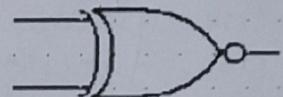
XOR2

viii). XNOR Gate :-

$$y = \overline{(A \oplus B)}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

U7



XNOR2

- Digital integrated circuits (ICs) are used to implement logic gates and other digital functions in a compact form. Here's a basic guide to understanding digital ICs :

i). Ic Packages and Pins :-

- Integrated circuits come in various packages such as DIP, SOIC, and QFP.
- Each Ic has pins that correspond to the inputs and outputs of the internal logic gates.

ii). Common Logic gate ICs :-

- 7400
- 7404
- 7408
- 7432
- 7402
- 7486
- 4000
- 4011
- 4001

- 4040
- 4060

iii). IC Operation and Testing :-

- Connecting IC's are typically connected to a breadboard or PCB with proper wiring to provide power and connect inputs/outputs.
- Testing IC's use a logic probe or multimeter to test the inputs and outputs of IC's to ensure they are functioning correctly.

Conclusion :-

- In these experiment, we had learned about the behaviour of logic gates designing, wire, and operate logic gates such as AND, OR, NOT, NOR, NAND etc. It helps to understand how to implement simple circuits based on a schematic diagram using logic gates in multisim software and also learned about Integrated Circuits (ICs) used to implement logic gates and other digital functions in a compact form.

PRACTICAL-2

AIM: To Verify Demorgan's theorem using Truth Table and Logic Diagram

- To verify Demorgan's theorem using truth table and logic diagram :

i). Boolean Expression :-

a). First Law :

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

b). Second Law :

$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

ii). Truth Table :-

a). First Law :

A	B	$A \cdot B$	$\overline{A \cdot B}$	\overline{A}	\overline{B}	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

b). Second Law :

A	B	$A+B$	$\overline{A+B}$	\overline{A}	\overline{B}	$\overline{A} \cdot \overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

iii). Logic circuit Diagram :-

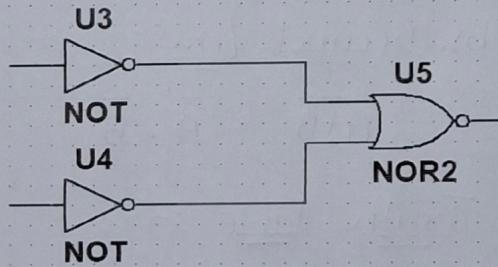
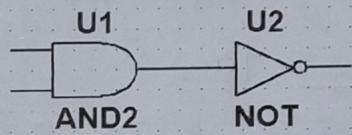
a). First Law Logic Circuit :-

- Left : AND gate followed by a NOT gate for $\bar{A} \cdot \bar{B}$.
- Right : Two NOT gate followed by an OR gate for $\bar{A} + \bar{B}$.

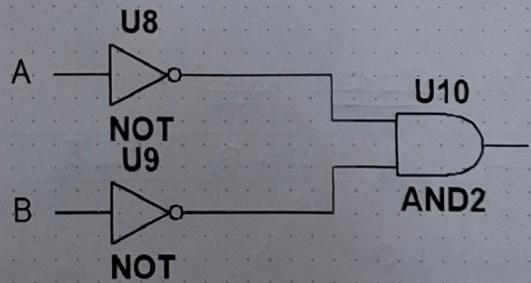
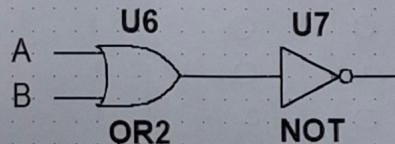
b). Second Law Logic Circuit :-

- Left : OR gate followed by a NOT gate for $\bar{A} + \bar{B}$.
- Right : Two NOT gates followed by an AND gate for $\bar{A} \cdot \bar{B}$.

a). First Law Logic Circuit :-



b). Second Law Logic Circuit :-



Conclusion :-

- By comparing the truth table results and the outputs from the logic circuits , we verify that both De morgan's laws hold true.

PRACTICAL-3

AIM: To Verify Half adder & Full adder Truth Table and Logic Diagram

- To verify the half adder and Full adder truth table and logic diagrams :

i). Boolean Expression :-

a). Half adder :

- Sum (S) : $S = A \oplus B$ (XOR operation)
- Carry (C) : $C = A \cdot B$ (AND operation)

b). Full adder :

- Sum (S) : $S = A \oplus B \oplus C_{in}$
- Carry (C_{out}) : $C_{out} = (A \cdot B) + (C_{in} \cdot (A \oplus B))$

ii). Truth Table :-

a). Half adder :-

A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

b). Full adder :

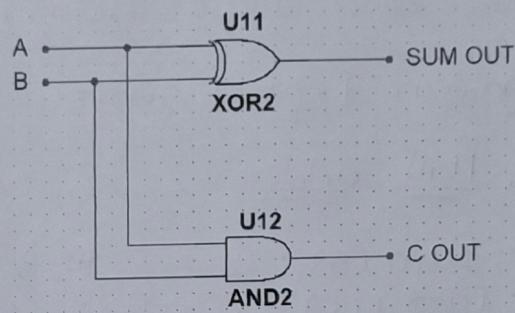
A	B	C_{in}	Sum (S)	Carry (C_{out})
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0

1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

iii). Logic Circuit Diagram :-

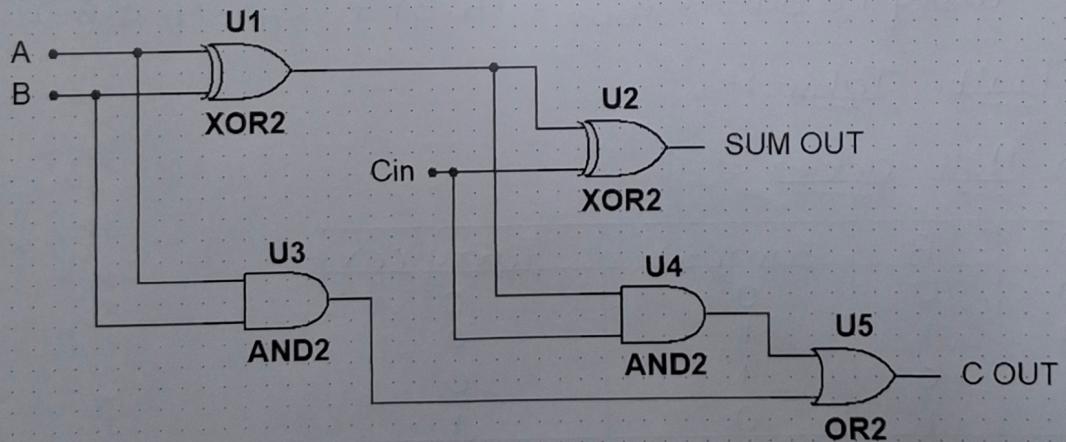
a). Half adder :

- Sum (S) : Use an XOR gate for $A \oplus B$.
- Carry (C) : Use an AND gate for $A \cdot B$.



b). Full adder :

- Sum (S) : $S = A \oplus B \oplus Cin$
- Carry (C -out) : $C_{out} = (A \cdot B) + (Cin \cdot (A \oplus B))$



Conclusion :-

In this experiment, we successfully verified the operations of both the Half adder and full adder circuits using their respective truth tables and logic diagrams.

PRACTICAL-4

AIM: To Verify Half Subtractor & Full Subtractor using Truth Table and Logic Diagram

- To verify Half subtractor and full subtractor using truth table and logic diagram :-

i). Boolean Expression :-

a). Half Subtractor :

- Difference (D) : $D = A \oplus B$ (XOR operation)
- Borrow (B) : $B = \bar{A} \cdot B$ (AND with NOT gate)

b). Full Subtractor :

- Difference (D) : $D = A \oplus B \oplus B_{in}$
- Borrow (B-out) : $B_{out} = \bar{A} \cdot B + (B \oplus A) \cdot B_{in}$

ii). Truth Table :-

a). Half Subtractor :

A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

b). Full Subtractor :

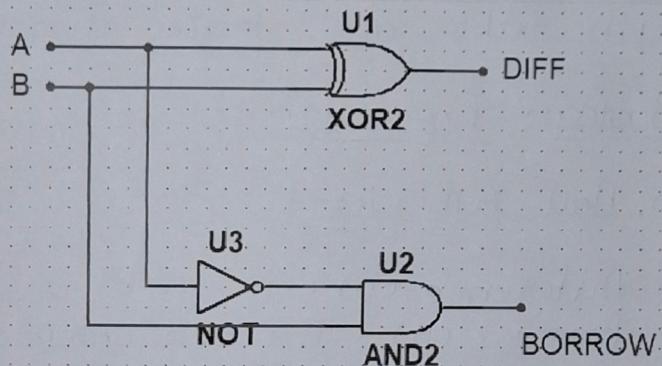
A	B	Bin	D	B-out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0

1	1	0	0	0
1	1	1	1	1

iii). Logic Circuit Diagram :-

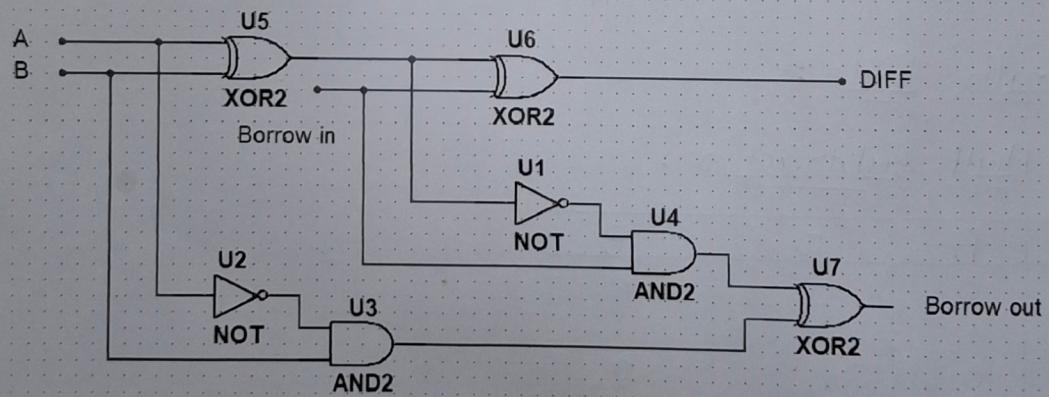
a). Half Subtractor :

- Difference (D) : Use an XOR gate for $A \oplus B$.
- Borrow (B) : Use a NOT gate to invert A and an AND gate for $\bar{A} \cdot B$.



b). Full Subtractor :

- Difference (D) : Use two XOR gates for $A \oplus B$
- Borrow (B-out) : Use two AND and one OR gate



Conclusion :-

In this experiment, we successfully verified the operations of both the half and full subtractor circuits using their respective truth tables and logic diagrams.

PRACTICAL-5

AIM: To Verify 4×1 Multiplexer using Truth Table and Logic Diagram

- To Verify 4×1 multiplexer using truth table and Logic diagram :

A 4×1 multiplexer (MUX) is a combinational circuit that selects one of four input signals and forwards it to the output based on two select lines.

- Boolean Expression :-

$$Y = \bar{S_1} \bar{S_0} I_0 + \bar{S_1} S_0 I_1 + S_1 \bar{S_0} I_2 + S_1 S_0 I_3$$

where,

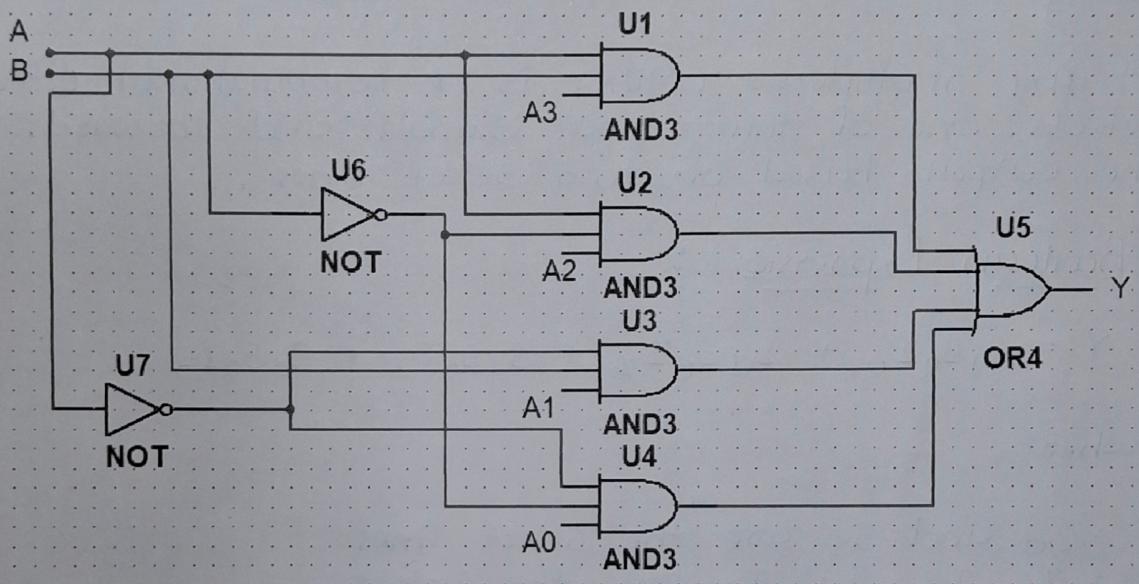
- S_0 and S_1 are the select lines
- I_0, I_1, I_2, I_3 are the inputs.

- Truth Table :-

S_1	S_0	I_0	I_1	I_2	I_3	Y
0	0	0	1	0	0	1
0	0	1	0	1	1	1
0	1	0	1	0	1	1
0	1	0	1	1	0	0
1	0	0	1	0	1	0
1	0	0	1	1	1	0
1	1	1	0	0	0	1
1	1	1	0	1	1	0

iii). Logic Diagram :-

- The logic diagram for a 4×1 multiplexer consists of : AND and OR gate.
- Here, we will use 4 AND gates to combine each input with the corresponding select lines. and we will use a single OR gate to combine the outputs of all four AND gates.



Conclusion :-

- The 4×1 multiplexer was successfully verified using the truth table and logic diagram. The MUX selects one of four input signals based on the two select lines and forward it to the output. The experiment confirms the correct working of the multiplexer as per its truth table and logic implementation.

PRACTICAL-6

AIM: To Verify NAND & NOR as Universal Gate using Truth Table and Logic Diagram

- To verify NAND and NOR as universal gate using truth table and logic diagram :
- The NAND gate can be used to create any basic logic gate (AND, OR, NOT) and any combinational logic circuit.
- The NOR gate, like NAND is a universal gate that can be used to create the basic logic gates (AND, OR, NOT) and any combinational logic circuit.

i). Boolean Expression :-

a). NAND :-

i). NOT :

$$A' = A \text{ NAND } A$$

b). NOR :-

i). NOT :

$$A' = A \text{ NOR } A$$

ii). AND :

$$A \cdot B = (A \text{ NAND } B)'$$

ii). AND :

$$A \cdot B = (A' \text{ NOR } B)'$$

iii). OR :

$$A + B = (A' \text{ NAND } B)'$$

iii). OR :

$$A + B = (A \text{ NOR } B)'$$

ii). Truth Table :-

a). NAND :

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

b). NOR :-

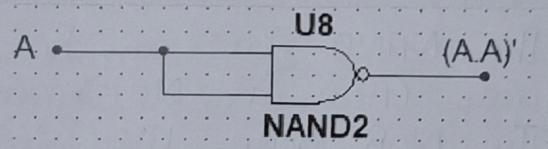
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

iii). Logic circuit diagram :-

a). NAND :-

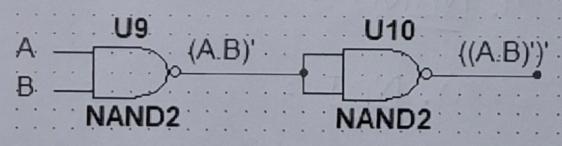
i). NOT using NAND gate :

A	A.A	$\bar{A} \cdot A$	A'
0	0	1	1
1	1	0	0



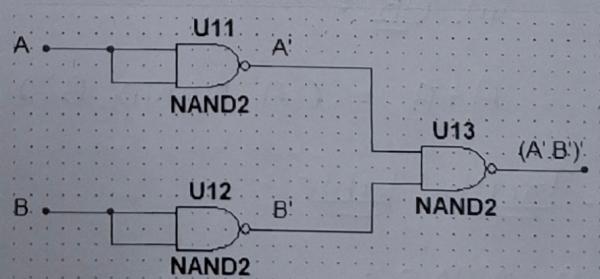
ii). AND using NAND gate :

A	B	A.B	$\bar{A} \cdot B$	$\bar{A} \cdot \bar{B}$	AB
0	0	0	1	0	0
0	1	0	1	0	0
1	0	0	1	0	0
1	1	1	0	1	1



iii). OR using NAND gate :

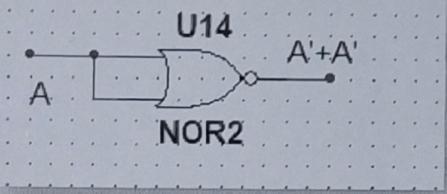
A	B	\bar{A}	\bar{B}	$\bar{A} \cdot \bar{B}$	$\bar{A} \cdot \bar{B}$	A+B
0	0	1	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	0	1	1



b). NOR :-

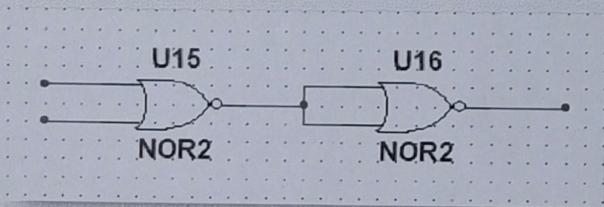
i. NOT using NOR gate :

A	$A' + A'$	$A + A$
0	1	0
1	0	1



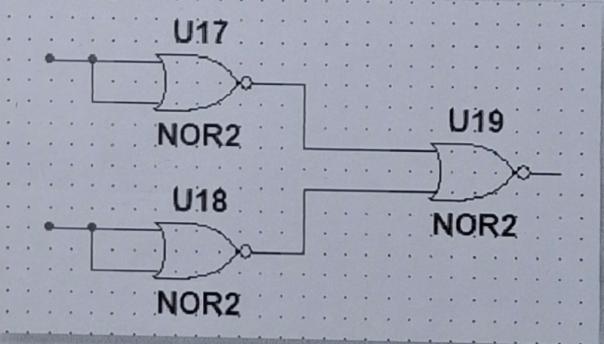
ii. AND using NOR gate :

A	B	$\bar{A} \cdot A$	$\bar{B} \cdot B$	$\bar{A} \cdot \bar{B}$	$\bar{A} \cdot B$
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	0	1



iii. OR using NOR gate :

A	B	A'	B'	$\bar{A}' \cdot \bar{B}'$
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1



Conclusion :-

In these experiment, we had verify NAND and NOR as universal gate using its truth table and logic diagram.

PRACTICAL-7

AIM: To Verify 1×4 Demultiplexer using Truth Table and Logic Diagram

- To verify 1×4 Demultiplexer using truth table and logic diagram :
- A 1×4 demultiplexer is a digital logic device that takes a single input and routes it to one of several outputs based on the value of the control signals. To verify the functionality of a 1×4 demultiplexer, you can use a truth table and a logic diagram.

ii). Boolean Expression :-

$$\bar{S}_1 \bar{S}_0 A + \bar{S}_1 S_0 A + S_1 \bar{S}_0 A + S_1 S_0 A$$

iii). Truth Table :-

INPUTS		OUTPUTS			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	A
0	1	0	0	A	0
1	0	0	A	0	0
1	1	A	0	0	0

The Logical expression of the term Y is as follows :

$$Y_0 = \bar{S}_1 \bar{S}_0 A$$

$$Y_1 = \bar{S}_1 S_0 A$$

$$Y_2 = S_1 \bar{S}_0 A$$

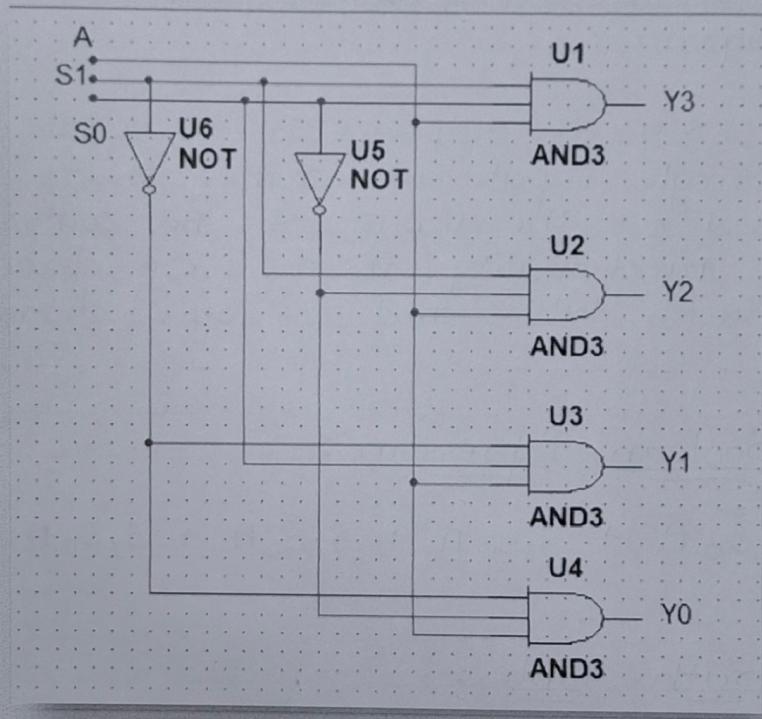
$$Y_3 = S_1 S_0 A$$

iii). Logic circuit Diagram :-

• Logic diagram components we will use are :

- i). AND gates
- ii). NOT gates

Here, we will use four AND gates and two NOT gates.



Conclusion :-

In these experiment, we had done a verification of a 1-to-4 or 1×4 demultiplexer using truth tables and logic diagrams which provides a clear understanding of its functionality and operation.