

**YASH SARANG**

**D6AD**

**47**

**DLCOA / Experiment 1**

Aim: To create and implement basic digital operations like AND, OR, NOT, NAND, NOR, EXOR and EXNOR operations

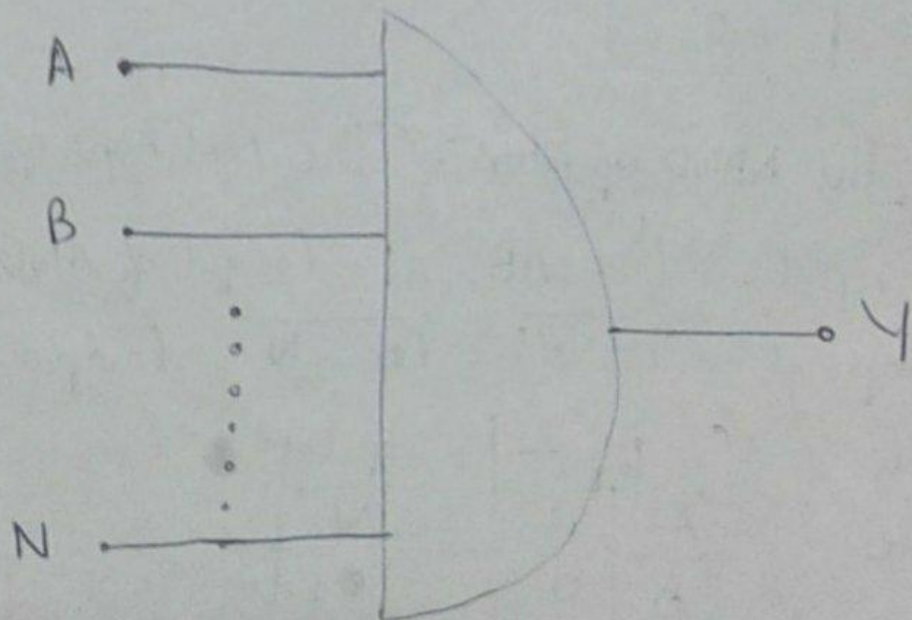
Theory: ① The AND operation: The circuit has 2 or more inputs. Digital signals are applied at the input terminals marked A, B, N the other terminal being ground.

Mathematically,  $Y = A \text{ and } B \text{ and } C \text{ and } N$ .  
 $= A \cdot B \cdot C \cdot N = ABCD : N$ .

• Truth Table

Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

• Diagram.





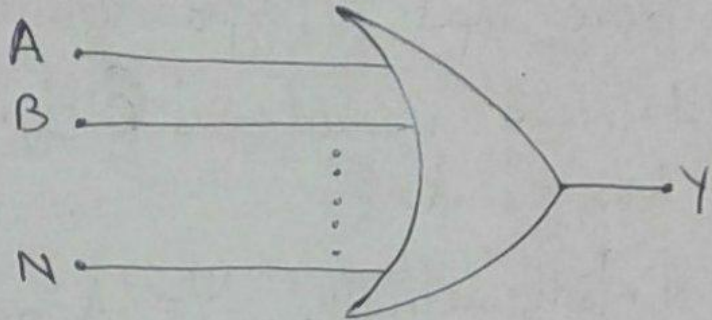
2) The OR operation: The circuit has 2 or more inputs and one output.

Mathematically,  $Y = A \text{ or } B \text{ or } C \text{ or } \dots N$   
 $Y = A + B + C + \dots + N$

Truth table

Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Diagram

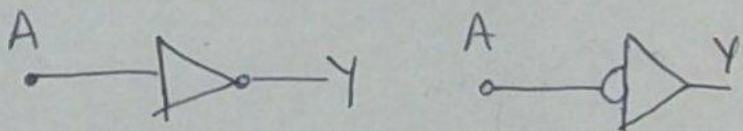


3) The NOT operator: It is also known as inverter. It has one input (A) and one output (Y). Its logic equation is  $Y = \text{NOT } A$ ;  $Y = \bar{A}$ .

Truth table

Input	Output
A	Y
0	1
1	0

Diagram:



4) The NAND operator: The Not AND operator can be described

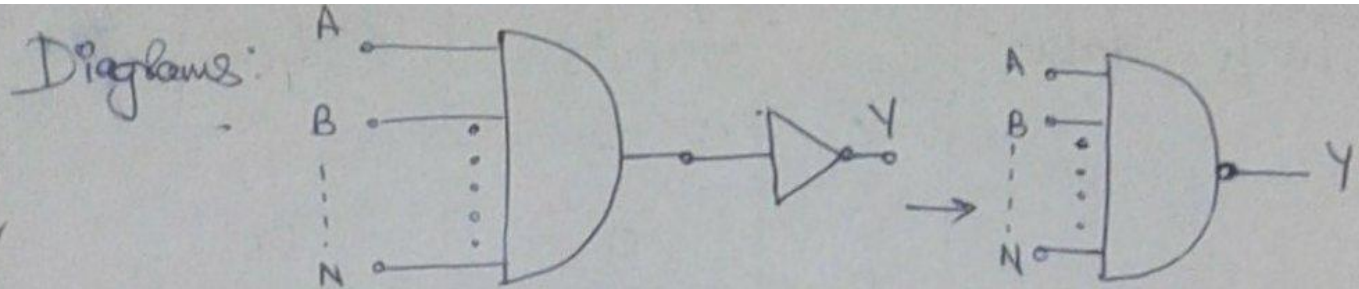
Step 1:  $Y' = AB \dots N$  (output of AND gate)

Step 2:  $Y = \overline{Y'} = \overline{AB \dots N}$  (output of NAND gate)

Truth table

Input		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0





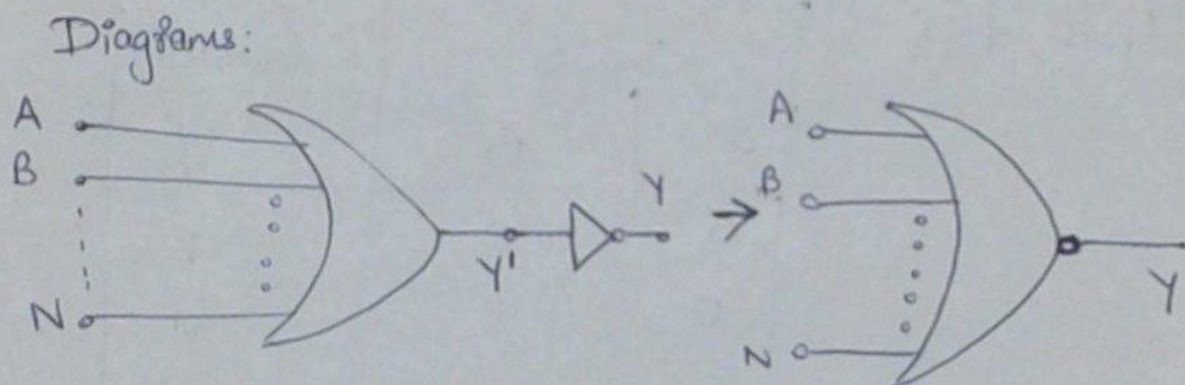
5) The NOR operator: The Not OR operator has more than 2 or more inputs and 1 output.

$$Y' = A + B + \dots + N \quad (\text{output of OR gate})$$

$$Y = \overline{Y'} = \overline{A + B + \dots + N} \quad (\text{output of NOR gate})$$

Truth table:

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



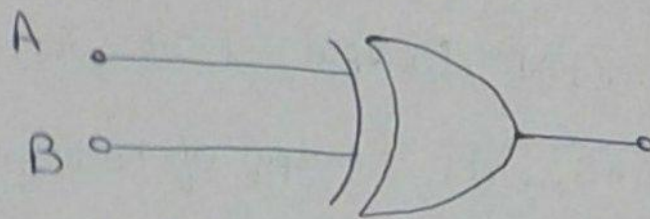
6) The EXOR: It is not a basic operation but can be performed using the basic gates AND, OR & NOT. Its logical equation is written as:  $Y = A \text{ EXOR } B$   
 $Y = A \oplus B.$



Truth Table:

Inputs		Outputs
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

• Diagram:



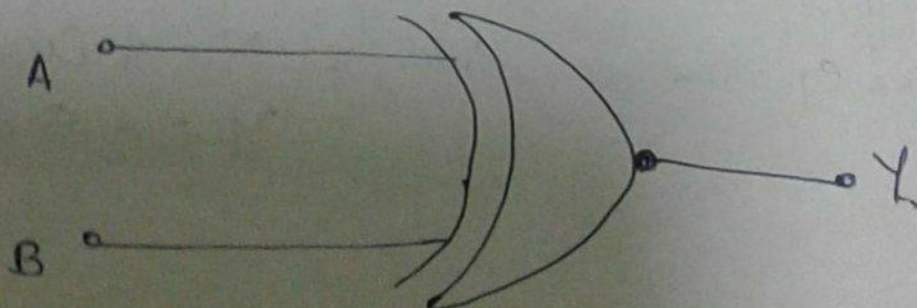
7) The EXNOR operator:

Similar to the EXOR operator. Referred to as the coincidence operator because it produces output of 1 only when its inputs coincide.

Truth table

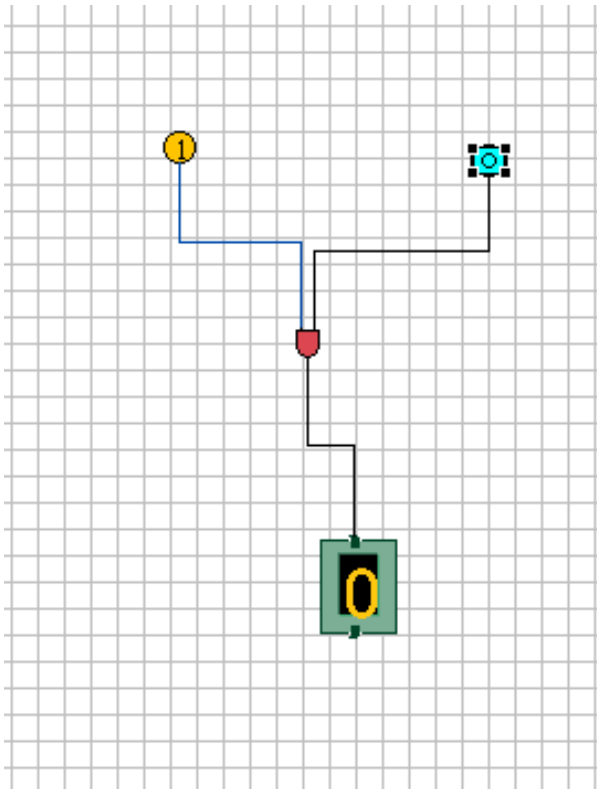
Inputs		Outputs
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Diagram:

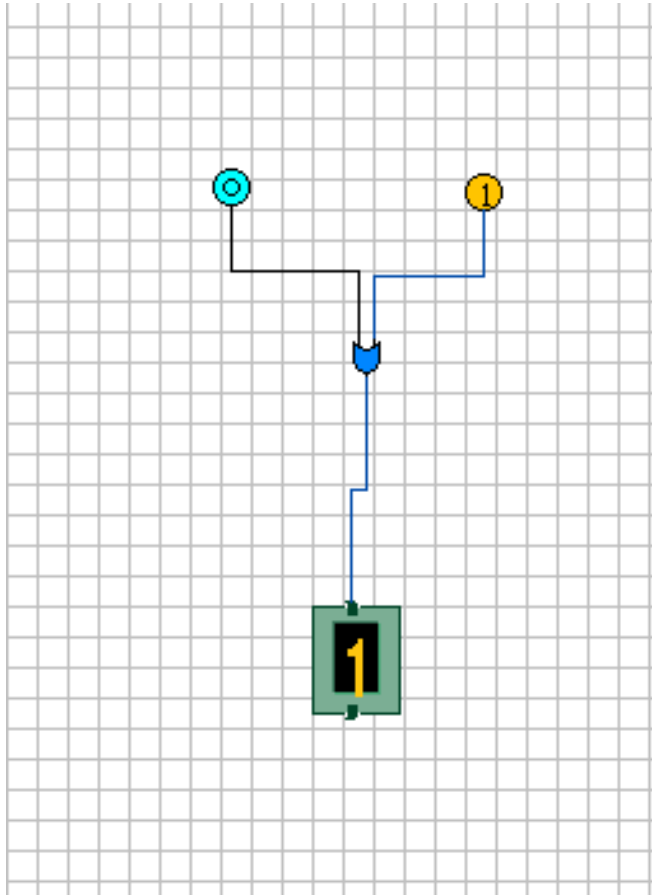


Conclusion: We have successfully created and implemented all the gates including AND, OR, NOT, NAND, EXOR & EXNOR.

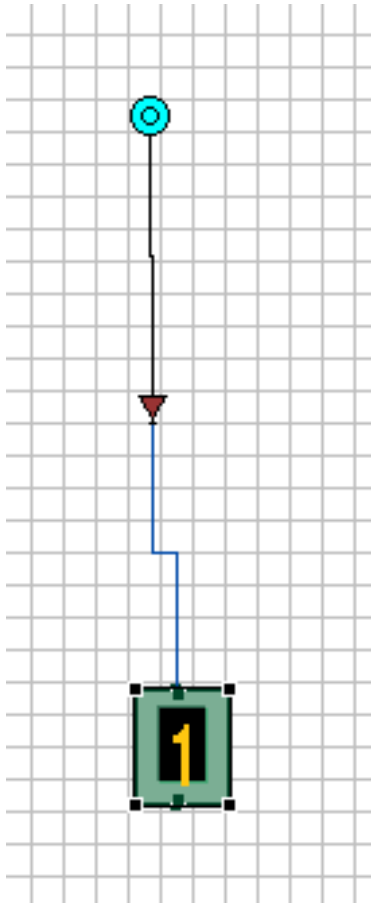
## And Gate:



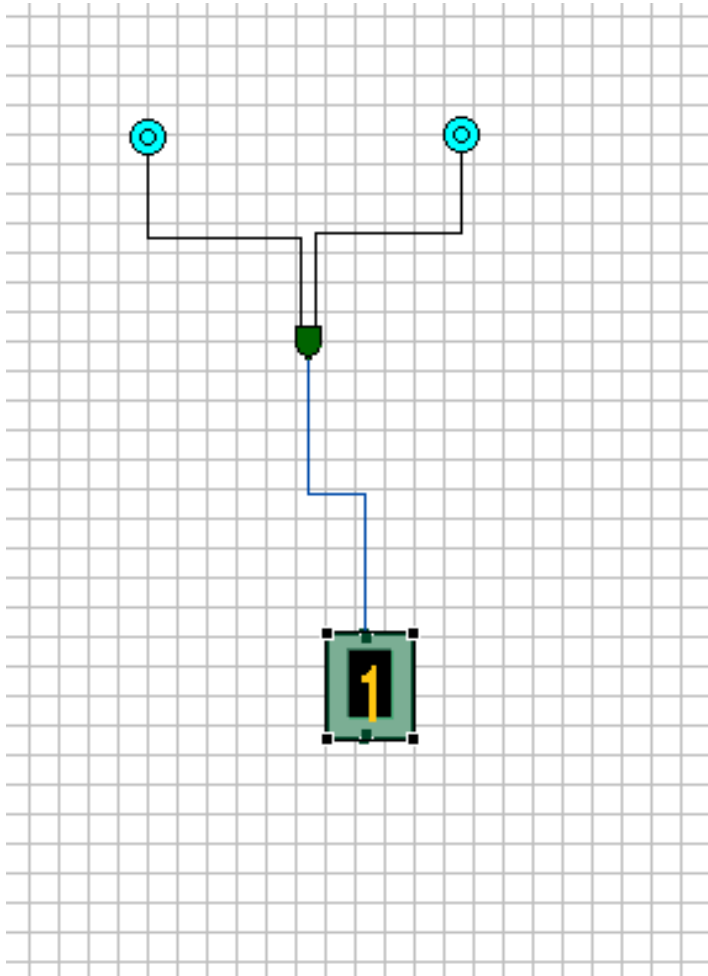
## Or Gate:



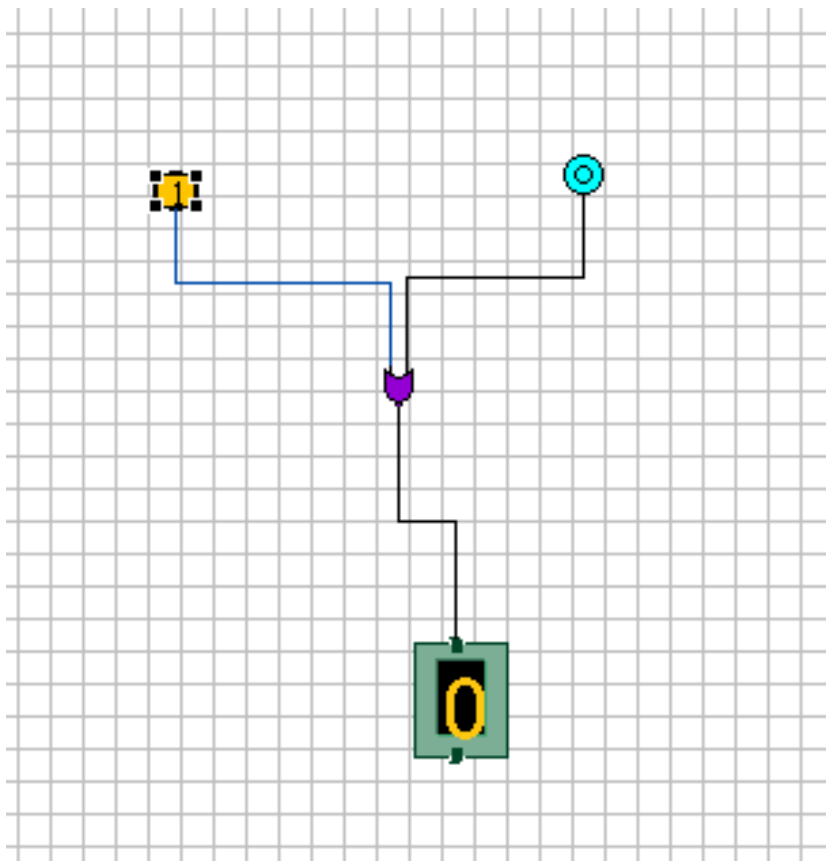
## Not Gate:



## Nand Gate:

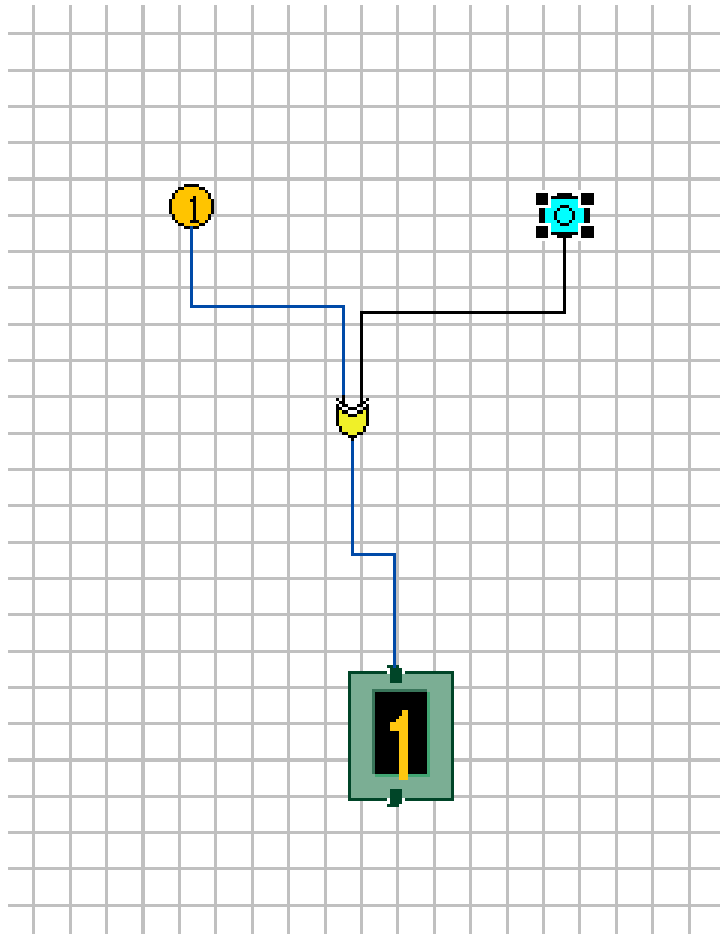


## Nor Gate:





## Exor Gate:



## Exnor Gate:

