LOGIC GATES

INTRODUCTION :

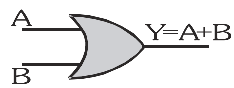
- A logic gate is a digital circuit which is based on certain logical relationship between the input and the output voltages of the circuit.
- The logic gates are built using the semiconductor diodes and transistors.
- Each logic gate is represented by its characteristic symbol.
- The operation of a logic gate is indicated in a table, known as truth table. This table contains all possible combinations of inputs and the corresponding outputs.
- A logic gate is also represented by a Boolean algebraic expression. Boolean algebra is a method of writing logical equations showing how an output depends upon the combination of inputs. Boolean algebra was invented by George Boole.

BASIC LOGIC GATES

There are three basic logic gates. They are (1) OR gate (2) AND gate, and (3) NOT gate

- **The OR gate** :- The output of an OR gate attains the state 1 if one or more inputs attain the state 1.

Logic symbol of OR gate



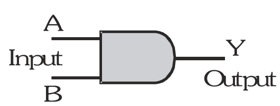
The Boolean expression of OR gate is Y = A + B, read as Y equals A 'OR' B.

Truth table of a two-input OR gate

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

- **The AND gate** :- The output of an AND gate attains the state 1 if and only if all the inputs are in state 1.

Logic symbol of AND gate



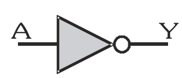
The Boolean expression of AND gate is Y = A.B
It is read as Y equals A 'AND' B

Truth table of a two-input AND gate

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

- **The NOT gate** : The output of a NOT gate attains the state 1 if and only if the input does not attain the state 1.

Logic symbol of NOT gate



The Boolean expression is Y = $\overline{A}$, read as Y equals NOT A.

Truth table of NOT gate

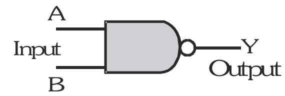
A	Y
0	1
1	0

COMBINATION OF GATES :

The three basis gates (OR, AND and NOT) when connected in various combinations give us logic gates such as NAND, NOR gates, which are the universal building blocks of digital circuits.

☯ **The NAND gate :**

Logic symbol of NAND gate



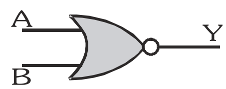
The Boolean expression of NAND gate is $Y = \overline{A.B}$

Truth table of a NAND gate

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

☯ **The NOR gate :**

Logic symbol of NOR gate



The Boolean expression of NOR gate is $Y = \overline{A + B}$

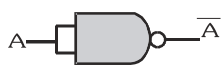
Truth table of a NOR gate

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

UNIVERSAL GATES :

The NAND or NOR gate is the universal building block of all digital circuits. Repeated use of NAND gates (or NOR gates) gives other gates. Therefore, any digital system can be achieved entirely from NAND or NOR gates. We shall show how the repeated use of NAND (and NOR) gates will gives us different gates.

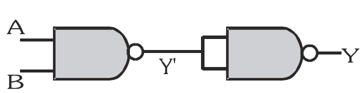
☯ **The NOT gate from a NAND gate :-** When all the inputs of a NAND gate are connected together, as shown in the figure, we obtain a NOT gate



Truth table of a single input NAND gate

A	B= (A)	Y
0	0	1
1	1	0

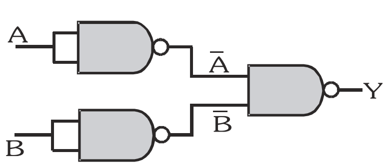
☯ **The AND gate from a NAND gates :-** If a NAND gate is followed by a NOT gate (i.e., a single input NAND gate), the resulting circuit is an AND gate as shown in figure and truth table given show how an AND gate has been obtained from NAND gates.



Truth table

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

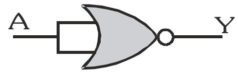
☯ **The OR gate from NAND gates :-** If we invert the inputs A and B and then apply them to the NAND gate, the resulting circuit is an OR gate.



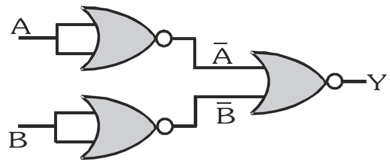
Truth table

A	B	$\overline{A}$	$\overline{B}$	Y
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

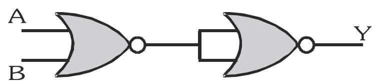
- **The NOT gate from NOR gates :-** When all the inputs of a NOR gate are connected together as shown in the figure, we obtain a NOT gate



- **The AND gate from NOR gates :-** If we invert the inputs A and B and then apply them to the NOR gate, the resulting circuit is an AND gate.



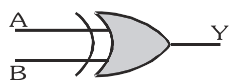
- **The OR gate from NOR gate :-** If a NOR gate is followed by a single input NOR gate (NOT gate), the resulting circuit is an OR gate.



XOR AND XNOR GATES :

- **The Exclusive - OR gate (XOR gate):-** The output of a two-input XOR gate attains the state 1 if one and only one input attains the state 1.

  Logic symbol of XOR gate



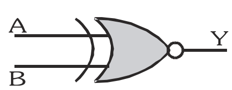
  The Boolean expression of XOR gate is $Y = A.\overline{B} + \overline{A}.B$ or $Y = A \oplus B$

  Truth table of a XOR gate

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

- **Exclusive - NOR gate (XNOR gate):-** The output is in state 1 when its both inputs are the same that is, both 0 or both 1.

  Logic symbol of XNOR gate



  The Boolean expression of XNOR gate is $Y = A.B + \overline{A}.\overline{B}$ or $Y = \overline{A \oplus B}$ or $A \odot B$

  Truth table of a XNOR gate

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

LAWS OF BOOLEAN ALGEBRA

Basic OR, AND, and NOT operations are given below :

OR	AND	NOT
$A + 0 = A$	$A . 0 = 0$	$A + \overline{A} = 1$
$A + 1 = 1$	$A . 1 = A$	$A . \overline{A} = 0$
$A + A = A$	$A . A = A$	$\overline{\overline{A}} . A = A$

Boolean algebra obeys commutative, associative and distributive laws as given below :

- **Commutative laws :**

  $A + B = B + A$;

  $A.B = B.A$

- **Associative laws :**

  $A + (B + C) = (A + B) + C$

  $A. (B . C) = (A. B) . C$

- **Distributive laws :**

  $A. (B + C) = A.B + A.C$

- **Some other useful identities :**

  (i) $A + AB = A$

  (ii) $A . (A + B) = A$

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

  (iv) $A. (\overline{A} + B) = A.B$

  (v) $A + (B.C) = (A + B). (A + C)$

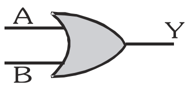
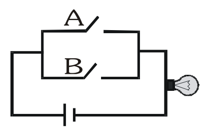
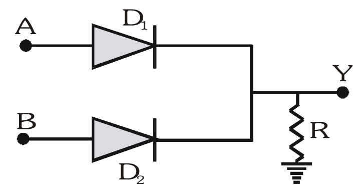
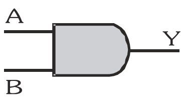
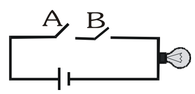
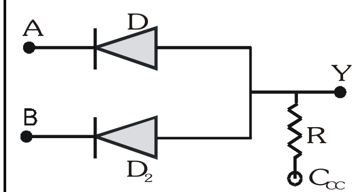
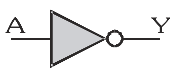
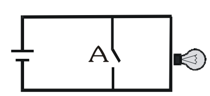
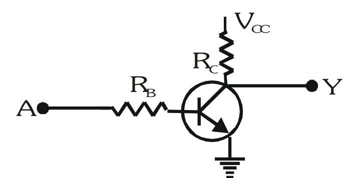
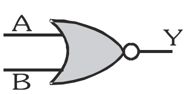
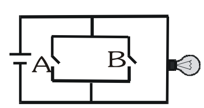
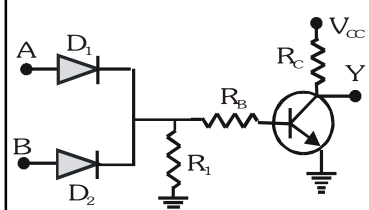
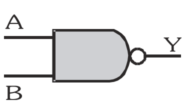
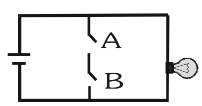
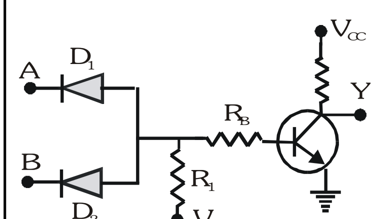
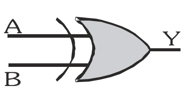
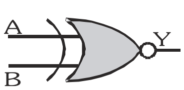
  (vi) $(\overline{A} + B).(A + C) = \overline{A}.C + B.A + B.C$

- **De Morgan's theorem :**

  First theorem : $\overline{A + B} = \overline{A}.\overline{B}$

  Second theorem : $\overline{A.B} = \overline{A} + \overline{B}$

SUMMARY OF LOGIC GATES

Names	Symbol	Boolean Expression	Truth table	Electrical analogue	Circuit diagram (Practical Realisation)
OR		Y = A + B	A B Y 0 0 0 0 1 1 1 0 1 1 1 1		
AND		Y = A. B	A B Y 0 0 0 0 1 0 1 0 0 1 1 1		
NOT or Inverter		$Y = \overline{A}$	A Y 0 1 1 0		
NOR (OR +NOT)		$Y = \overline{A + B}$	A B Y 0 0 1 0 1 0 1 0 0 1 1 0		
NAND (AND+NOT)		$Y = \overline{A.B}$	A B Y 0 0 1 0 1 1 1 0 1 1 1 0		
XOR (Exclusive OR)		$Y = A \oplus B$ or $Y = \overline{A}.B + A\overline{B}$	A B Y 0 0 0 0 1 1 1 0 1 1 1 0		
XNOR (Exclusive NOR)		$Y = A \odot B$ or $Y = A.B + \overline{A}.\overline{B}$ or $Y = \overline{A \oplus B}$	A B Y 0 0 1 0 1 0 1 0 0 1 1 1		

NUMBER SYSTEMS

Decimal Number system

The base of this system is 10 and in this system 10 numbers [0,1,2,3,4,5,6,7,8,9] are used.

Ex. 1396, 210.75 are decimal numbers.

Binary Number System

The base of this system is 2 and in this system 2 numbers (0 and 1) are used.

Ex. 1001, 1101.011 are Binary numbers.

Binary to decimal conversion

We can write any decimal number in following form

$2365.75 = 2000 + 300 + 60 + 5 + 0.7 + 0.05$

$$= 2 \quad 1000 + 3 \quad 100 + 6 \quad 10 + 5 \quad 1 + 7 \quad \frac{1}{10} + 5 \quad \frac{1}{100}$$

$$= 2 \quad 10^3 + 3 \quad 10^2 + 6 \quad 10^1 + 5 \quad 10^0 + 7 \quad 10^{-1} + 5 \quad 10^{-2}$$

Similarly we can write any binary number in following form

$$10101.11 = 1 \quad 2^4 + 0 \quad 2^3 + 1 \quad 2^2 + 0 \quad 2^1 + 1 \quad 2^0 + 1 \quad 2^{-1} + 1 \quad 2^{-2}$$

$$= 1 \quad 16 + 0 \quad 8 + 1 \quad 4 + 0 \quad 2 + 1 \quad 1 + 1 \quad \frac{1}{2} + 1 \quad \frac{1}{4}$$

$$= 16 + 4 + 1 + \frac{1}{2} + \frac{1}{4} = 21.75$$

Ex.1 Convert binary number 1011.01 into decimal number.

$$1011.01 = 1 \quad 2^3 + 0 \quad 2^2 + 1 \quad 2^1 + 1 \quad 2^0 + 0 \quad 2^{-1} + 1 \quad 2^{-2}$$

$$= 8 + 2 + 1 + \frac{1}{4} = 11.25$$

Ex.2 Convert binary number 1000101.101 into decimal number.

$$1000101.101 \equiv 1 \quad 2^6 + 0 \quad 2^5 + 0 \quad 2^4 + 0 \quad 2^3 + 1 \quad 2^2 + 0 \quad 2^1 + 1 \quad 2^0 + 1 \quad 2^{-1} + 0 \quad 2^{-2} + 1 \quad 2^{-3}$$

$$= 64 + 4 + 1 + \frac{1}{2} + \frac{1}{8} = 69 + 0.5 + 0.125 = 69.625$$

Question for Practise : Convert the following binary numbers into decimal numbers –

(a) 101 (b) 110.001 (c) 11111 (d) 1011.11

Ans. : (a) 5 (b) 6.125 (c) 31 (d) 11.75

DECIMAL TO BINARY CONVERSION

You should remember this table for decimal to binary conversion

2^{-3}	2^{-2}	2^{-1}	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}
0.125	0.25	0.5	1	2	4	8	16	32	64	128	256	512	1024

Ex.3 Convert the decimal number 25 into its binary equivalent

Sol. $25 = 16 + 8 + 1 \equiv 2^4 + 2^3 + 2^0 = 1 \quad 2^4 + 1 \quad 2^3 + 0 \quad 2^2 + 0 \quad 2^1 + 1 \quad 2^0$ so $(25)_{10} \equiv (11001)_2$

Ex.4 Convert 69 into its binary equivalent

Sol. $69 = 64 + 4 + 1 = 1 \quad 2^6 + 1 \quad 2^2 + 1 \quad 2^0 \Rightarrow (69)_{10} \equiv (1000101)_2$

Ex.5 Convert 13.5 into its binary equivalent

$13.5 = 8 + 4 + 1 + 0.5 = 1 \quad 2^3 + 1 \quad 2^2 + 0 \quad 2^1 + 1 \quad 2^0 + 1 \quad 2^1 \Rightarrow (13.5)_{10} = (1101.1)_2$

Question for Practise : Convert the following decimal numbers into binary numbers

(a) 6 (b) 65 (c) 106 (d) 268 (e) 8.125

Ans. (a) 110 (b) 1000001 (c) 1101010 (d) 100001100 (e) 1000.001