Assam Textile Institute

Empowering Textile Education

CLASS = 02

☑ Theory

☐ Practical

COURSE TITLE = TEXTRONICS

COURSE CODE= TT-603

by,

PANKAJ JYOTI DAS

(M.TECH),(ECE)

:BCD or Binary Coded Decimal:

- ***BCD or Binary-Coded Decimal*** is a special kind of representation of decimal number, where each individual digit of a number is converted into its equivalent 4 digit binary number and combining them all to generate a BCD code. For example, $(57)_{10} = (0101\ 0111)_{BCD}$
- But a **BCD** is not a binary representation of a decimal number.

  i.e. $(57)_{10} = (111001)_2$ and

  $(57)_{10} = (0101\ 0111)_{BCD}$

- ***Invalid BCD*** : In the BCD, with four bits we can represent sixteen numbers (0000 to 1111). But in BCD code only first ten of these are used (0000 to 1001). The remaining ***six code*** combinations i.e.(1010 to 1111) are invalid in BCD.

Decimal	Binary
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Valid BCD

Decimal	Binary
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

Invalid BCD

:Binary Addition and Subtraction:

Binary addition:-

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

Binary subtraction:-

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Example :

```
  10010          1011101          10011
+ 1100         + 1000000        + 1111101
-------        ---------        ---------
  11110          10011101         10010000

  10011001       11000011         1001100
+   100111     +   101111       + 1100101
-----------    -----------      ---------
  11000000       11110010         10110001
```

- 1011011 − 10010 = 1001001:

```
  1 0 1 1 0 1 1
−     1 0 0 1 0
-------------
  1 0 0 1 0 0 1
```

- 1010110 − 101010 = 101100:

```
  0   0
  ¼ ¹0 ¼ ¹0 1 1 0
−   1 0 1 0 1 0
-------------
  1 0 1 1 0 0
```

- 100010110 − 1111010 = 10011100:

```
  0 1 1 1 ¹0
  ¼ ¹0 ¹0 ¹0 ¼ ¹0 1 1 0
−     1 1 1 1 0 1 0
-----------------
  1 0 0 1 1 1 0 0
```

- 101101 − 100111 = 110:

```
      0 ¹0
  1 0 ¼ ¼ ¹0 1
− 1 0 0 1 1 1
-------------
        1 1 0
```

- 1000101 − 101100 = 11001:

```
  0 1 1
  ¼ ¹0 ¹0 ¹0 1 0 1
−   1 0 1 1 0 0
-------------
    1 1 0 0 1
```

- 1110110 − 1010111 = 11111:

```
    0 ¹0 1 ¹0 ¹0
  1 ¼ ¼ ¹0 ¼ ¼ ¹0
− 1 0 1 0 1 1 1
---------------
    1 1 1 1 1
```

Questions

Question 1
Calculate the binary numbers:

(a) 11 + 1
(b) 11 + 11
(c) 111 + 11
(d) 111 + 10
(e) 1110 + 111
(f) 1100 + 110
(g) 1111 + 10101
(h) 1100 + 11001
(i) 1011 + 1101
(j) 1110 + 10111
(k) 1110 + 1111
(l) 11111 + 11101

Question 2
Calculate the binary numbers:

(a) 11 – 10
(b) 110 – 10
(c) 1111 – 110
(d) 100 – 10
(e) 100 – 11
(f) 1000 – 11
(g) 1101 – 110
(h) 11011 – 110
(i) 1111 – 111
(j) 110101 – 1010
(k) 11011 – 111
(l) 11110 – 111

Question 3
Solve the following equations, where all numbers, including x, are binary:

(a) $x + 11 = 1101$ $x =$
(b) $x - 10 = 101$ $x =$
(c) $x - 1101 = 11011$ $x =$
(d) $x + 1110 = 10001$ $x =$
(e) $x + 111 = 11110$ $x =$
(f) $x - 1001 = 11101$ $x =$

Question 4
Calculate the binary numbers:

(a) 10 – 1 1
(b) 100 – 1 11
(c) 1000 – 1 111
(d) 10000 – 1 1111

Question 5
Calculate the binary numbers:

(a) 11 + 11
(b) 111 + 111
(c) 1111 + 1111
(d) 11111 + 11111

6) Convert the following numbers into BCD.

i) 92 iii) 127
ii)268 iv) 80

:Digital Logic Gate:

- ***Logic gates*** are the basic building blocks of any digital system. It is an electronic circuit having one or more than one input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as -

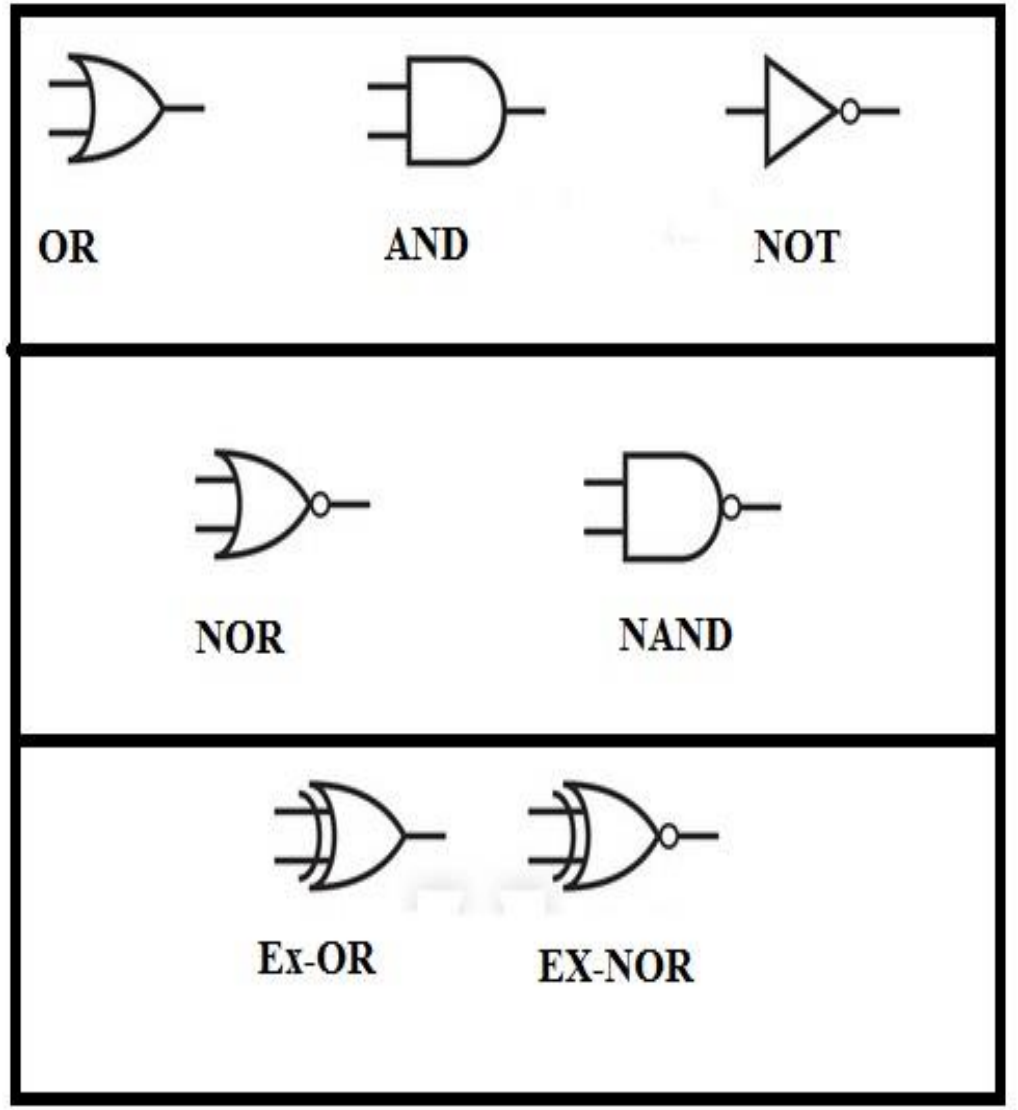
  OR gate, AND gate, NOT gate,

  NOR gate, NAND gate,

  Exclusive-OR gate, Exclusive-NOR gate etc.

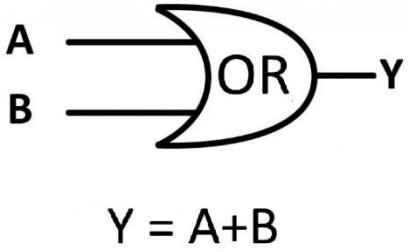
- ***Universal Logic Gate***: A universal gate is a a gate which can implement any Boolean function without use any other gates.

  The ***NAND and NOR*** gates are universal logic gates.



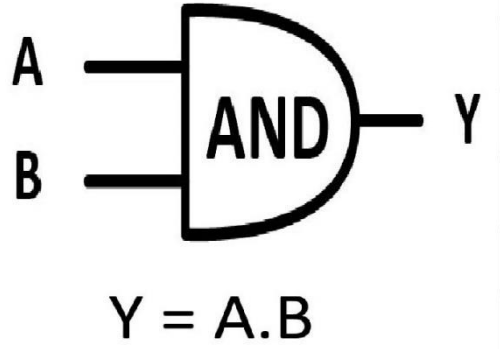
: Truth Table of Logic Gates :

Block Diagram and Truth Table of **OR** *gate.*



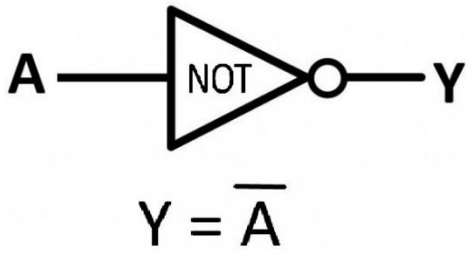
Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Block Diagram and Truth Table of **AND** *gate.*



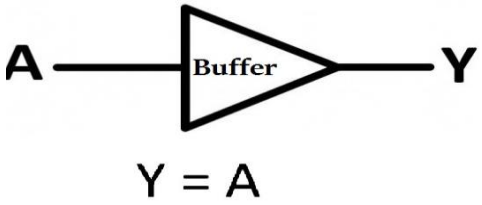
Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Block Diagram and Truth Table of **NOT** *gate.*



Input	Output
A	Y
0	1
1	0

Block Diagram and Truth Table of **BUFFER** *gate.*

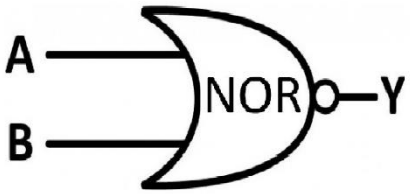


Input	Output
A	Y
0	0
1	1

: Truth Table of Logic Gates :

Block Diagram and Truth Table of **NOR** *gate.*

NOR = OR + NOT

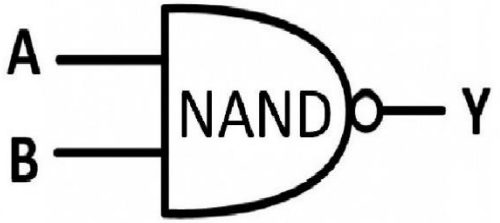


$Y = \overline{A+B}$

Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Block Diagram and Truth Table of **NAND** *gate.*

NAND = AND + NOT



$Y = \overline{A.B}$

Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Block Diagram and Truth Table of **Ex-OR** *gate.*

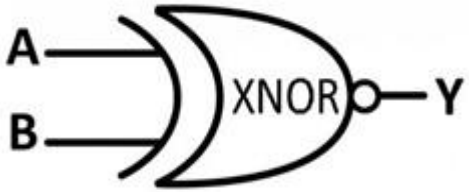
$Y = A\overline{B} + \overline{A}B$

$Y = A \oplus B$

Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Block Diagram and Truth Table of **Ex-NOR** *gate*

Ex-OR + NOT = Ex-NOR



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

$Y = A \oplus B$

$= A \odot B$

Inputs		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1