1.Boolean algebra

■ The logical symbol 0 and 1 are used for representing the digital input or output. The symbols "1" and "0" can also be used for a permanently open and closed digital circuit.

■ The digital circuit can be made up of several logic gates. To perform the logical operation with minimum logic gates, a set of rules were invented, known as the **Laws of Boolean Algebra**. These rules are used to reduce the number of logic gates for performing logic operations.

■ It is also known as **Binary algebra** because we only use binary numbers in this. **George Boole** developed the binary algebra in **1854**.

■ Analysis: It is an economical way of describing the function of digital circuitry.

■ Design: Given a desired function, Boolean algebra can be applied to develop a simplified implementation of that function.

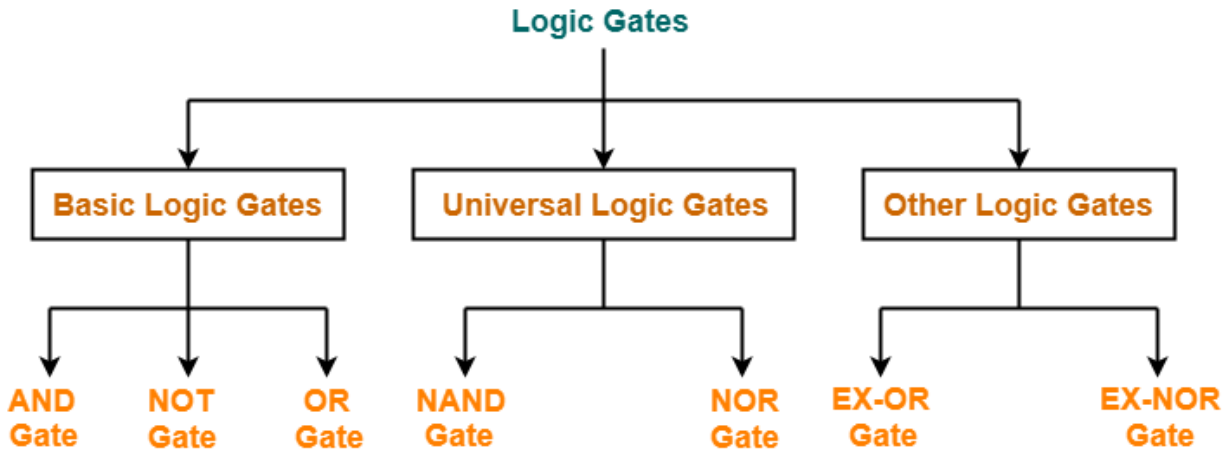
Rules of Boolean algebra

1. Only two values (1 for high and 0 for low) are possible for the variable used in Boolean algebra.
2. The overbar (-) is used for representing the complement variable. So, the complement of variable C is represented as.
3. The plus (+) operator is used to represent the ORing of the variables.
4. The dot(.) operator is used to represent the ANDing of the variables.

2.Logic Gates

- The logic gates are the main structural part of a digital system.
- Logic Gates are a block of hardware that produces signals of binary 1 or 0 when input logic requirements are satisfied.
- Each gate has a distinct graphic symbol, and its operation can be described by means of algebraic expressions.
- The seven basic logic gates includes: AND, OR, XOR, NOT, NAND, NOR, and XNOR.
- The relationship between the input-output binary variables for each gate can be represented in tabular form by a truth table.

- Each gate has one or two binary input variables designated by A and B and one binary output variable designated by x.



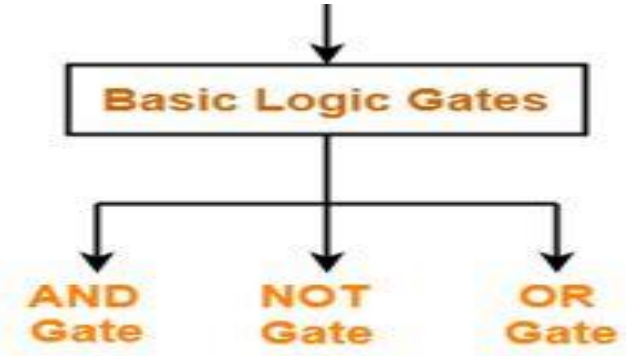
Types of Logic Gates

1.Basic Logic Gates

Basic Logic Gates are the fundamental logic gates using which universal logic gates and other logic gates are constructed.

They have the following properties-

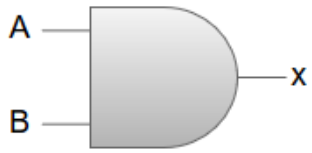
- Basic logic gates are associative in nature.
- Basic logic gates are commutative in nature.



1.1 AND GATE:

The AND gate is an electronic circuit which gives a high output only if all its inputs are high. The AND operation is represented by a dot (.) sign.

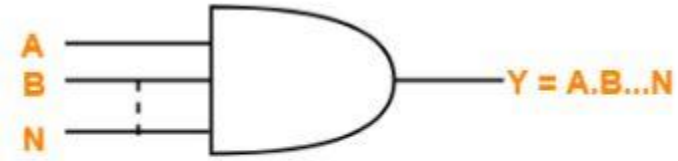
AND Gate:



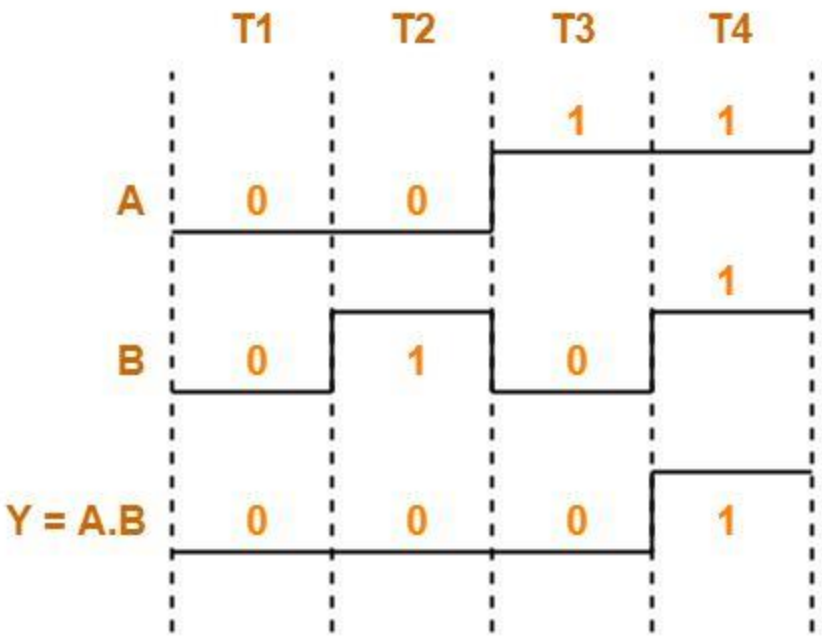
Algebraic Function: x = AB

Truth Table:

A	B	x
0	0	0
0	1	0
1	0	0
1	1	1



N-Input AND Gate

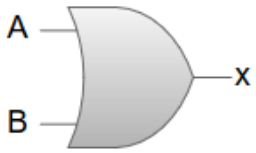


AND Gate Timing Diagram

1.2 OR GATE:

The OR gate is an electronic circuit which gives a high output if one or more of its inputs are high. The operation performed by an OR gate is represented by a plus (+) sign.

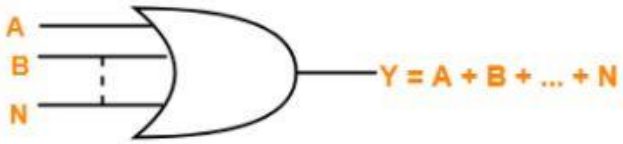
OR Gate:



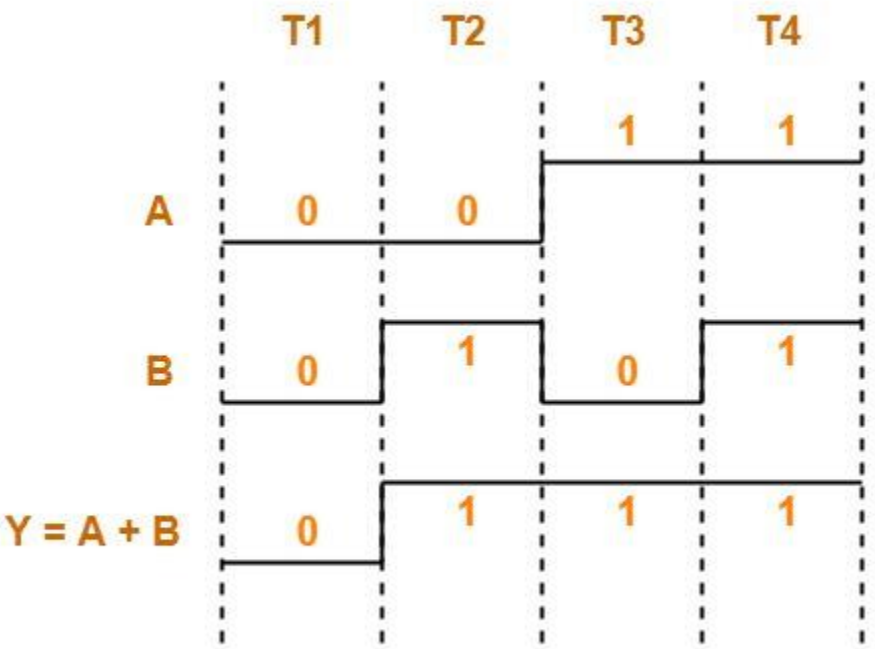
Algebraic Function: x = A + B

Truth Table:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1



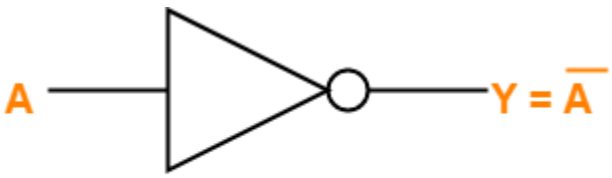
N-Input OR Gate



OR Gate Timing Diagram

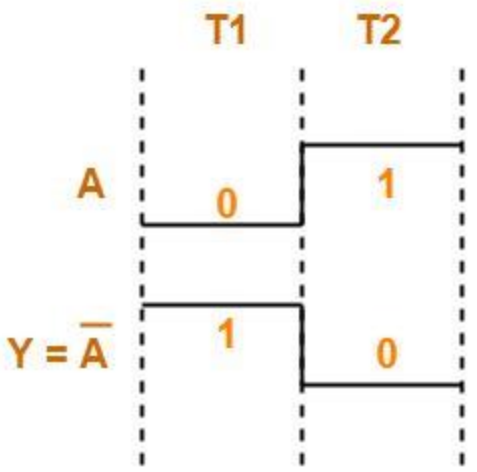
1.3 NOT GATE:

The NOT gate is an electronic circuit which produces an inverted version of the input at its output. It is also known as an **Inverter**.



NOT Gate

A	Y = A'
0	1
1	0



NOT Gate Timing Diagram

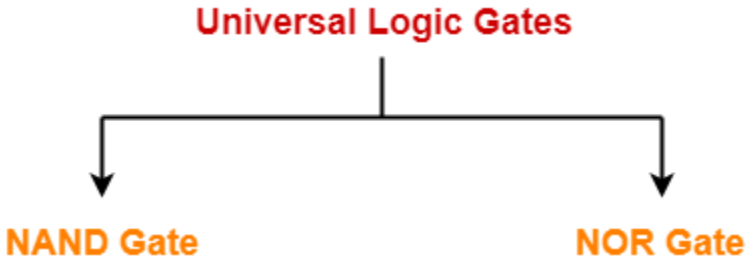
2. Universal logic Gates

They are called as "**Universal Gates**" because-

- They can realize all the binary operations.
- All the basic logic gates can be derived from them.

They have the following properties-

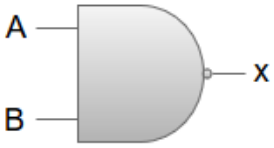
- Universal gates are not associative in nature.
- Universal gates are commutative in nature.



2.1 NAND GATE:

The NOT-AND (NAND) gate which is equal to an AND gate followed by a NOT gate. The NAND gate gives a high output if any of the inputs are low. The NAND gate is represented by a AND gate with a small circle on the output. The small circle represents inversion.

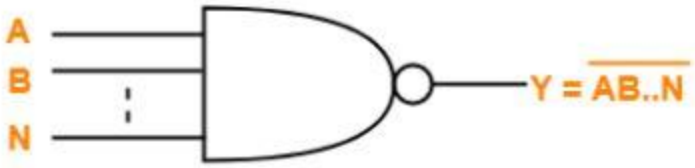
NAND Gate:



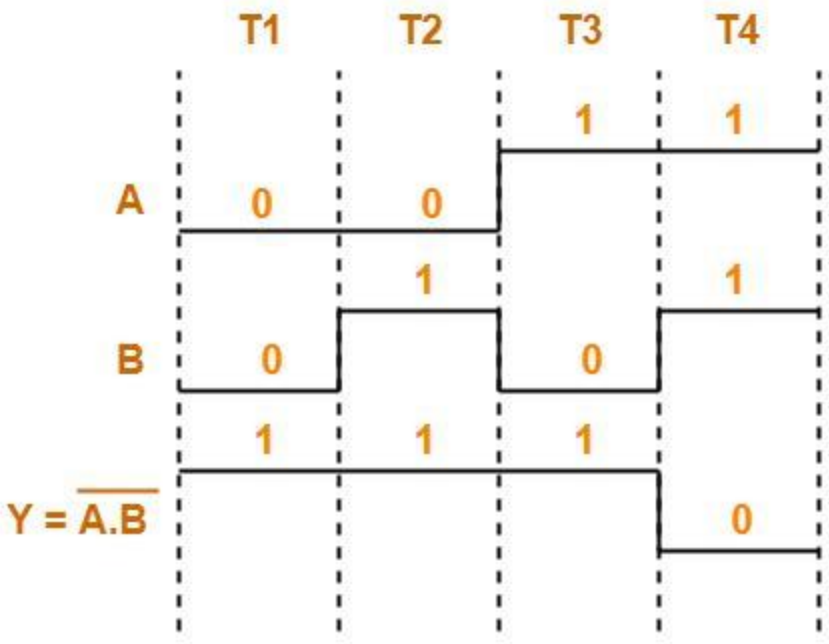
Algebraic Function: x = (AB)'

Truth Table:

A	B	x
0	0	1
0	1	1
1	0	1
1	1	0



N-Input NAND Gate

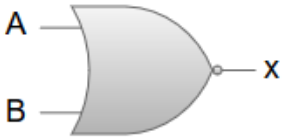


NAND Gate Timing Diagram

2.2 NOR GATE:

The NOT-OR (NOR) gate which is equal to an OR gate followed by a NOT gate. The NOR gate gives a low output if any of the inputs are high. The NOR gate is represented by an OR gate with a small circle on the output. The small circle represents inversion.

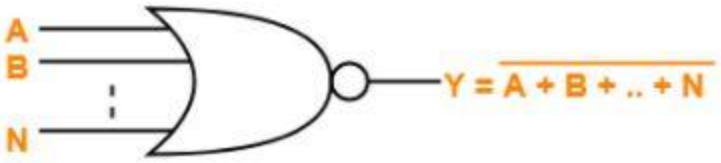
NOR Gate:



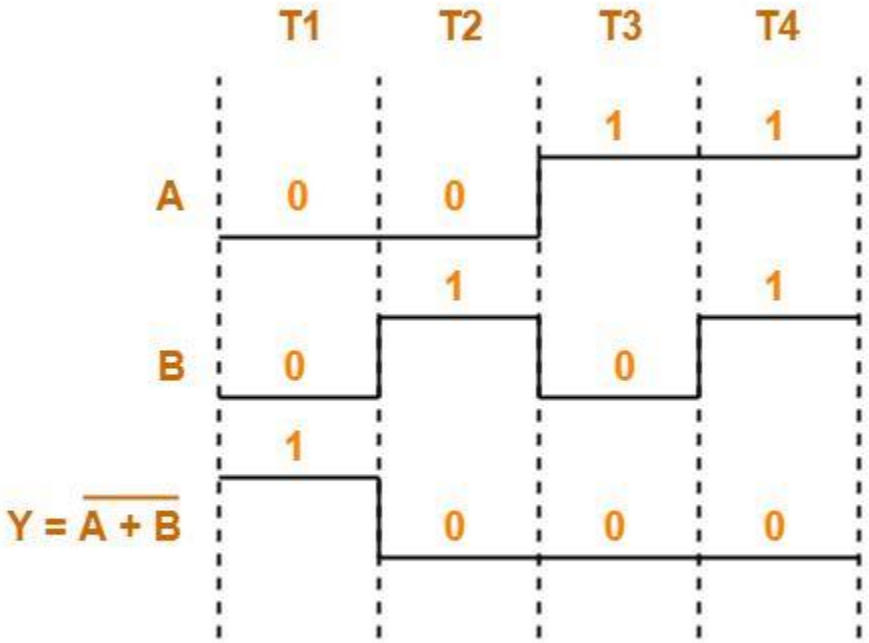
Algebraic Function: x = (A+B)'

Truth Table:

A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

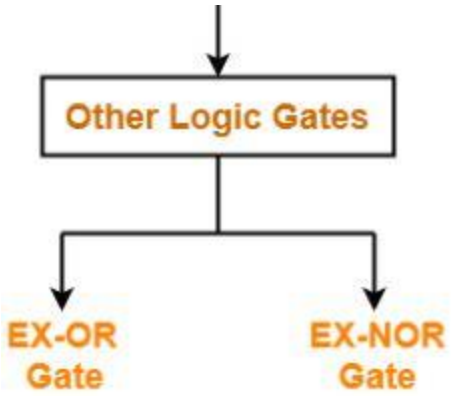


N-Input NOR Gate



NOR Gate Timing Diagram

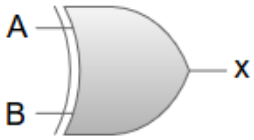
3. Other Logic Gates



3.1 XOR GATE:

The 'Exclusive-OR' gate is a circuit which will give a high output if one of its inputs is high but not both of them. The XOR operation is represented by an encircled plus sign.

XOR Gate:



Algebraic Function: $x = A \oplus B$

or

$x = A'B + AB'$

Truth Table:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0