# Boolean Algebra

Boolean algebra is an algebra that deals with Boolean values(TRUE and FALSE) .

**Truth table:**
Truth table is a table, which represents all the possible values of logical variables/statements along with all the possible results of given combinations of values.

**Logical Operators**:
Logical operators are derived from the Boolean algebra, which is the mathematical representation of the concepts without going into the meaning of the concepts.

**1. NOT Operator**—Operates on single variable. It gives the complement value of variable.

| **X** | $\bar{X}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

**2. OR Operator** -It is a binary operator and denotes logical Addition operation and is represented by ”+”symbol

0 + 0 =0
0 + 1 =1
1 + 0 =1
1 + 1 =1

| X | Y | X+Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**3. AND Operator** – AND Operator performs logical multiplications and symbol is (.)dot.

0.0=0
0.1=0
1.0=0
1.1=1

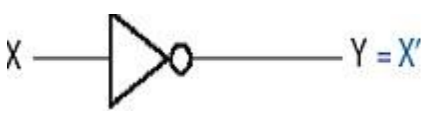
Truth table:

| X | Y | X.Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Basic Logic Gates**
A gate is simply an electronic circuit, which operates on one or more signals to produce an output signal. Gates are digital circuits because the input and output signals are either low (0) or high (1). Gates also called logic circuits.
There are three types of logic gates:

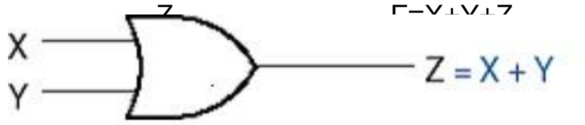
1. Inverter (NOTgate)
2. ORgate
3. AND gate

**1. NOTgate** **:** This gate takes one input and gives a single output. The symbol of this logic gates is

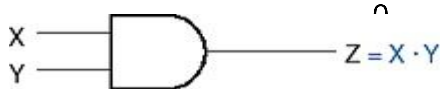


This circuit is used to obtain the compliment of a value.
If X = 0, then X' = 1.

**2. OR gate** : The OR gate has two or more input signals but only one output signal if any of the input signal is 1(high) the output signal is 1(high).
The circuit diagram of two inputs OR gate is:-




**3.** AND gate:- The AND gate have two or more than two input signals and produce an output signal. When all the inputs are 1(High) then the output is 1 otherwise output is 0 only.




**<u>Basic postulates of BooleanAlgebra:</u>**

Boolean algebra consists of fundamental laws that are based on theorem of Boolean algebra. These fundamental laws are known as basic postulates of Boolean algebra. These postulates are:-
If X ≠ 0 then x=1 and If X≠1 thenx=0

I OR relations(logicaladdition)

II AND relations (logicalmultiplication)

III Complement Rules

**Principle of Duality:-**
This principle states that we can derive a Boolean relation from another Boolean relation by performing simple steps. The steps are:-

1. Changing each AND(.) with an OR(+)sign
2. Changing each OR(+) with an AND(.)sign
3. Replace each 0 with 1 and each 1 with0

e.g
0+0=0 then dual is 1.1=1
1+0=1 then dual is 0.1=0

**<u>Basic theorem of Boolean algebra</u>**
Basic postulates of Boolean algebra are used to define basic theorems of Boolean algebra that provides all the tools necessary for manipulating Boolean expression.

| | |
|---|---|
| 1. Properties of 0 and1<br>(a) 0+X=X<br>(b) 1+X=1<br>(c)0.X=0<br>(d)1.X=X<br>2. IndempotentLaw<br>(a) X+X=X<br>(b) X.X=X<br>3. InvolutionLaw<br>$\overline{(X)} = \overline{X}$<br>4. ComplementyLaw<br>(a) X +X=1X.$\overline{X}$=0<br>5. CommutativeLaw<br>(a) X+Y=Y+X<br>(b) X.Y=Y.X | 6. AssociativeLaw<br>(a) X+(Y+Z)=(X+Y)+Z<br>(b) X(YZ)=(XY)Z<br><br>7. DistributiveLaw<br>(a) X(Y+Z)=XY_XZ<br>(b) X=YZ=(X+Y)(X+Z)<br><br>8. AbsorptionLaw<br>(a) X+XY=X<br>(b) X(X+Y)=X |

## Demorgan's Theorem

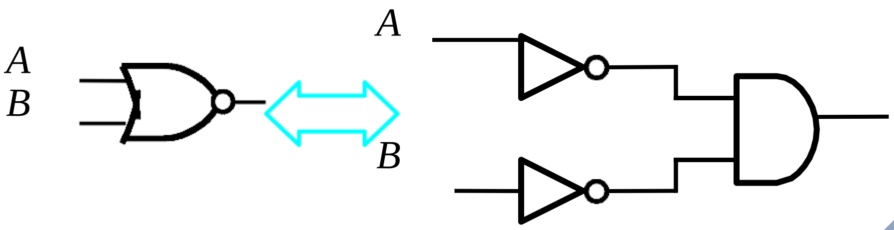
A mathematician named DeMorgan developed a pair of important rules regarding group complementation in Boolean algebra.

**Demorgan's First Theorem:**

This rule states that the compliment of OR of two operands is same as the AND of the compliments of those operands.

Mathematically it can be written as:-

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

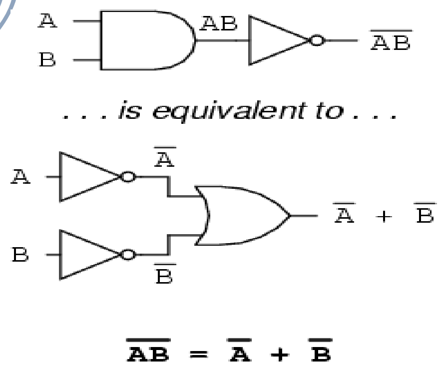



**Demorgan's Second Theorem:**

This rule states that the compliment of AND of two operands is same as the OR of the compliments of those operands.

Mathematically it can be written as:-

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




**Derivation of Boolean expression:-**

**Minterms and Maxterms**

- Consider two binary variables *x* and *y* combined with an AND operation.

  x'y', x'y, xy',xy

Each of these four AND terms represents one of the distinct areas in the Venn diagram and is called a *mintermor standard product*.

- Consider two binary variables *x* and *y* combined with an ORoperation.

  x'+ y', x'+ y, x + y', x +y

Each of these four OR terms represents one of the distinct areas in the Venn diagram and is called a *maxtermor standard sum*.

- *n*Variables can be combined to form $2^n$ minterms ormaxterms.

| **Minterms and Maxterms for Three Binary Variables** | | | | | | |
|---|---|---|---|---|---|---|
| | | | **Minterms** | | **Maxterms** | |
| **x** | **y** | **Z** | **Term** | **Designation** | **Term** | **Designation** |
| 0 | 0 | 0 | x'y'z' | $m_0$ | x+y+z | $M_0$ |
| 0 | 0 | 1 | x'y'z | $m_1$ | x+y+z' | $M_1$ |
| 0 | 1 | 0 | x'yz' | $m_2$ | x+y'+z | $M_2$ |
| 0 | 1 | 1 | x'yz | $m_3$ | x+y'+z' | $M_3$ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | xy'z' | $m_4$ | x'+y+z | $M_4$ |
| 1 | 0 | 1 | xy'z | $m_5$ | x'+y+z' | $M_5$ |
| 1 | 1 | 0 | xyz' | $m_6$ | x'+y'+z | $M_6$ |
| 1 | 1 | 1 | xyz | $m_7$ | x'+y'+z' | $M_7$ |

**Conversion between CanonicalForms**

The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function. This is because the original function is expressed by those minterms that make the function equal to 1, whereas its complement is a 1 for those minterms that the function is 0.

Example : $F(A,B,C) = \Sigma(0, 2, 4, 6,7)$

$F'(A,B,C) = \Sigma(1, 3, 5) = m_1 + m_3 + m_5$

Take the complement of *F'* by DeMorgan's theorem to obtain *F* in a different form:

$F(A,B,C) = (m_1 + m_3 + m_5)' = (m_1' \cdot m_3' \cdot m_5') = M_1M_3M_5 = \Pi(1, 3, 5)$

- To convert from one canonical form to the other, interchange the symbols Σand Π, and list those numbers missing from the originalform.

**Minimization of Boolean expressions:-**

After obtaining SOP and POS expressions, the next step is to simplify the Boolean expression. There are two methods of simplification of Boolean expressions.

1. **AlgebraicMethod**
2. **Karnaugh Map:**

**1. Algebricmethod:**This method makes use of Boolean postulates, rules and theorems to simplifythe expression.

**Example..1**.Simplify $A\bar{B}C\bar{D} + A\bar{B}CD + ABC\bar{D} + ABCD$

Solution-- $A\bar{B}C\bar{D} + A\bar{B}CD + ABC\bar{D} + ABCD$

$=A\bar{B}C(\bar{D}+D) + ABC(\bar{D}+D)$

$=A\bar{B}C.1 + ABC.1$ $(\bar{D}+D=1)$

$=AC(\bar{B}+B)$

$=AC.1 = AC$

**2. Using KarnaughMap:**

A Karnaugh map is graphical display of the fundamental products in a truth table.

- Forexample,thefirstK-maphererepresents$xy+xy=$ $x(y + \bar{y})=$ $x$ . (since y+y'=1)
- ThesecondK-map,similarly,shows$xy+x\bar{y}$

$= (x + x)y = y$

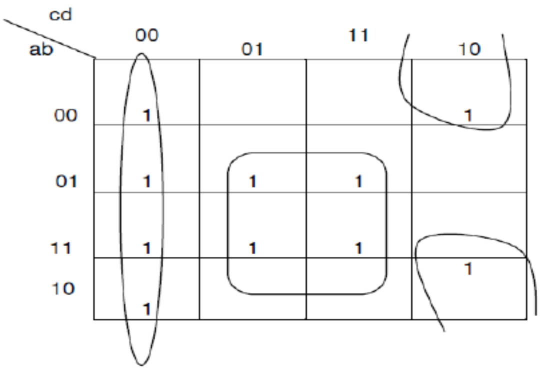
Remember, group together adjacent cells of 1s, to form largest possible rectangles of sizes that are powers of2.

- Notice that you can overlap the blocks ifnecessary.

For reducing the expression first mark Octet, Quad, Pair then single.

- Pair: Two adjacent 1's makes a pair.
- Quad: Four adjacent 1's makes aquad.
- Octet: Eight adjacent 1's makes anOctet.
- Pair removes one variable.
- Quad removes twovariables.
  - Octet removes three variables.

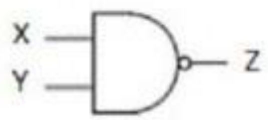
***1***. F(a,b,c,d)=∑(0,2,4,5,7,8,10,12,13,15)





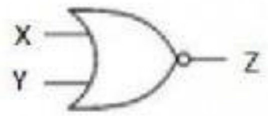

**Universal logic gates :-**A universal logic gate is one which can implement any Boolean function {0,1}, without the use of any other logic gate. The NAND and NOR gates are called universal logic gates because all the other gates can be created using these two.

**NAND GATE:-**It is the NOT AND gate. The inverter bubble reverses the output of the AND gate logic, giving a 0 output only when all inputs are 1.



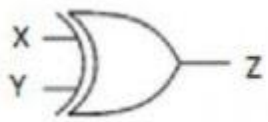

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**NOR GATE:-**It is the NOT – OR gate. The inverter bubble reverses the output of the OR gate logic giving 1 as the output only if all inputs are 0. It can have more than 1 input.



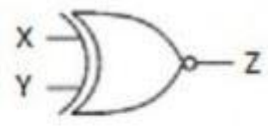

| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**XOR GATE:-**It is the 'Exclusive – OR' gate. The output is 1 only when one of the inputs is 1. And 0 if the inputs are the same.




| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**XNOR GATE:-**It is the 'Exclusive – NOR' gate is a combination of XOR gate followed by an inverter. The output is 1 only when both the inputs (even high) are the same and false if the inputs are different.




| X | Y | Z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |