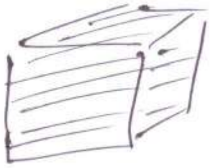
Number System

Number system defines a set of values used to represent quantity.



$$\text{quantity } (q) \Rightarrow \begin{cases} (111001110000)_2 \\ (16340)_8 \\ (7392)_{10} \\ (1CE0)_{16} \end{cases}$$

Number System	Base/Radix	values
Binary	2	0 to $(r-1) = 0$ to $1 \Rightarrow 0, 1$
Octal	8	0 to $r-1 = 7 \Rightarrow 0, 1, 2, 3, 4, 5, 6, 7$
Decimal	10	0 to $10-1 = 9 \Rightarrow 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
Hexadecimal	16	0 to $16-1 = 15 \Rightarrow 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15)$

if $r_1 < r_2$

then $N_1 > N_2$

$r \Rightarrow$ Base or Radix

$N \rightarrow$ Total Number of digits used to represent quantity.

Number System

Weighted System

Ex: Decimal
Binary
octal
BCD etc.

Unweighted System

Ex: Gray code
Excess - 3
etc

Weighted System

$$7392 = 7000 + 300 + 90 + 2$$
$$= 7\times 1000 + 3\times 100 + 9\times 10 + 2\times 1$$
$$= 7\times 10^3 + 3\times 10^2 + 9\times 10^1 + 2\times 10^0$$

Decimal Number System

Addition :

$$\begin{array}{r} 5\ 9\ 2\ 8 \\ +\ 3\ 4\ 7\ 5 \\ \hline 9\ 4\ 0\ 3 \end{array}$$

base

$8+5 = 13 - 10 = 3$

$1+2+7 = 10-10=0$

$1+9+4 = 14-10=4$

$1+5+3 = 9$

Subtraction

$$\begin{array}{r} 8\ 4\ 3\ 5 \\ -\ 7\ 6\ 5\ 7 \\ \hline 0\ 7\ 7\ 8 \end{array}$$

base

$10+5-7=8$

$10+2-5=7$

$10+3-6=7$

Binary Number System

Base	Values	digits
2.	0 to $(r-1)$	
	0 to $(2-1)$	
	0 to 1	$\Rightarrow$ 0, 1

Binary digits (0 and 1) are called 'bits'.

Ex: $(10101.11)_2$ Convert to Decimal.

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

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

$$= 21 + 0.5 + 0.25$$

$$= 21 + 0.75 \Rightarrow (21.75)_{10}$$

Ex: $(21.75)_{10} \Rightarrow (\ ?\ )_2$

$$\begin{array}{r|rl} 2 & 21 & \\ 2 & 10 & 1 \\ 2 & 5 & 0 \\ 2 & 2 & 1 \\ 2 & 1 & 0 \\ & 0 & 1 \end{array} \uparrow$$

$(21)_{10} \Rightarrow (10101)_2$

$0.75 \times 2 = 1.50 \rightarrow 0.50$ 1

$0.50 \times 2 = 1.00 \rightarrow 0.00$ 1 $\downarrow$

$(0.75)_{10} \Rightarrow (0.11)_2$

$(21.75)_{10} \Rightarrow (10101.11)_2$

Q.1 → $(110110)_2 = (54)_{10}$

Q2 $(0.1101)_2 = (0.8125)_{10}$

Q3 $(110.111)_2 = (6.875)_{10}$

Q-4 $(12.0625)_{10} = (1100.0001)_2$

Subtraction

using 1's Complement

Q-1

$$\begin{array}{r} 26 \\ -13 \\ \hline 13 \end{array} \qquad \begin{array}{r} 11010 \\ -\ 1101 \\ \hline 1101 \end{array} \qquad \begin{array}{r} 11010 \\ \xrightarrow{\text{1's Comp.}} +\ 0010 \\ \hline \boxed{1}1100 \\ \hookrightarrow\ +1 \\ \hline 1101 \end{array}$$

$$\begin{array}{r} 4 \\ -48 \\ \hline -44 \end{array} \qquad \begin{array}{r} 000100 \\ -\ 110000 \\ \hline -\ 101100 \end{array} \qquad \begin{array}{r} 000100 \\ \text{1's Comp}\ \ 001111 \\ \hline 010011 \end{array} \rightarrow (19)$$

[No carry indicates negative Ans and in complement form]

Complement → $\overline{010011}$ = 101100 ANS

$$\begin{array}{r} 27 \\ -20 \\ \hline 7 \end{array} \qquad \begin{array}{r} 11011 \\ 10100 \\ \hline 00111 \end{array} \qquad \begin{array}{r} 11011 \\ \text{1's Comp} \rightarrow +\ 01011 \\ \hline 100110 \\ \hookrightarrow\ 1 \\ \hline 00111 \end{array} \qquad \begin{array}{r} 11011 \\ \text{2's Com}\ \ 01100 \\ \hline \boxed{1}00111 \end{array}$$

00111 ANS.

~~discart~~ drop carry

Possible Representations

Sign Magnitudes	1's Complement	2's Complement
000 = +0	000 = +0	000 = +0
001 = +1	001 = +1	001 = +1
010 = +2	010 = +2	010 = +2
011 = +3	011 = +3	011 = +3
100 = −0	100 = −3	100 = −4
101 = −1	101 = −2	101 = −3
110 = −2	110 = −1	110 = −2
111 = −3	111 = −0	111 = −1

In 1's complement these two zero's one is positive and another is negative.

Octal to Binary $(5272)_8 = (?)_2$

5 2 7 2

101 010 111 010

$(5272)_8 = (101010111010)_2$

Hexa to Binary $(763)_H = (?)_2$

7 6 3

0111 0110 0011

$\Rightarrow 763_H = (011101100011)_2$

~~Binary to Hexa decimal~~

~~$(101100.001)_2$~~

Binary to Hexadecimal $\underline{1}\,\underline{0010}\,\underline{1101}\,\underline{0101}$

$$\underbrace{0001}_{1}\ \underbrace{0010}_{2}\ \underbrace{1101}_{\substack{13\\D}}\ \underbrace{0101}_{5} = (12D5)_H$$

$(1011000001)_2 = (2C1)_H$

Boolean Algebra

Boolean algebra is used to express the effects that various digital circuits have on logic inputs, and to manipulate logic variable for the purpose of the determination of the best method for performing a given circuit function.

Boolean algebra differs in a major way from ordinary algebra in that Boolean constants and variables are allowed to have only two possible values 0 or 1.

In Boolean algebra there are only three basic operations:

1. Logical Addition or OR operation
2. Logical Multiplication or AND operation
3. Logical Complementation or NOT operation

Boolean Algebra Law

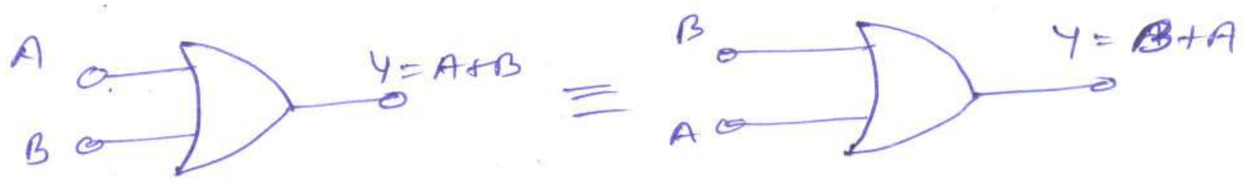
Three of the basic laws of Boolean algebra are the same as in ordinary algebra. These are:

1. Commutative law
2. Associative law
3. Distributive law

1. Commulative Law

These laws of addition and multiplication say that the order in which the variables are ORed or ANDed makes no difference as the same output is arrived at either way. These laws of addition and multiplication for two variables are written algebraically as below:

$$\boxed{\begin{aligned} A+B &= B+A \\ A \cdot B &= B \cdot A \end{aligned}}$$



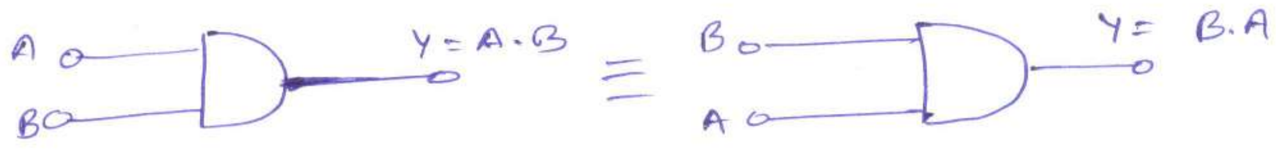


Fig: Application of Commulative law of addition and Multiplication

Figures illustrates the commutative law as applied to the OR gate and the AND gate respectively.

2. Associative Law

These laws of addition and multiplication say that in the ORing or ANDing of several variables, grouping of the variables is immaterial and the results obtained are the same. These laws of addition and multiplication for three variables are written algebraically as below:

$$\boxed{A + (B + C) = (A + B) + C}$$

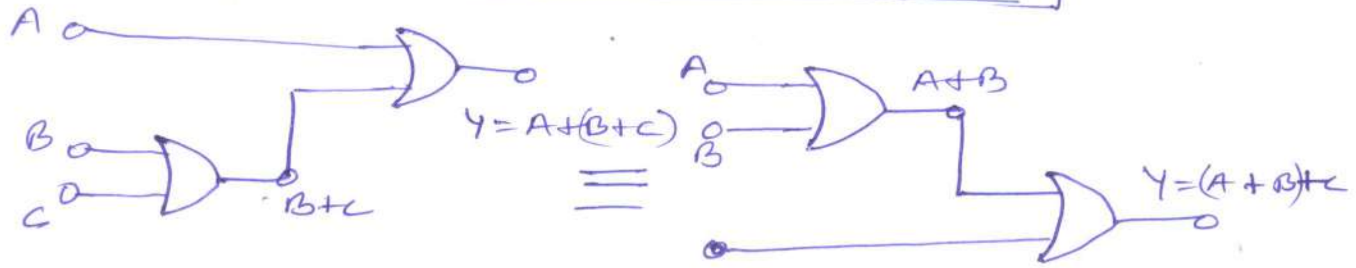


Fig: Application of Associative law of Addition

$$\boxed{A \cdot (B \cdot C) = (A \cdot B) \cdot C}$$

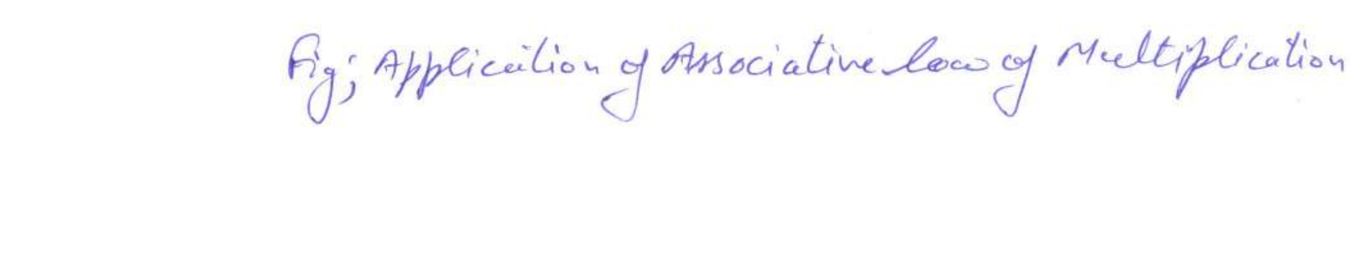


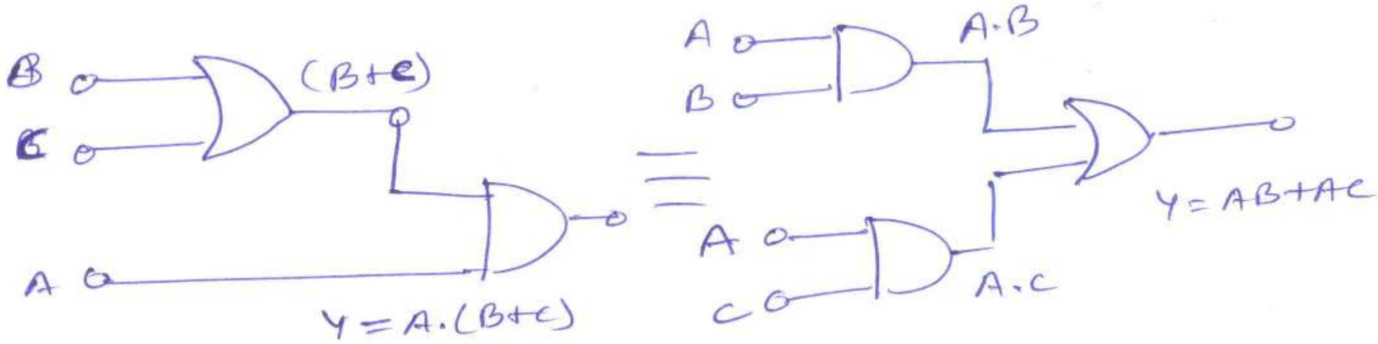
Fig: Application of Associative law of Multiplication

3. Distributive Law

This states that ORing several variables and ANDing the result with a single variable is equivalent to ANDing the single variable with each of several variables and then ORing the products. This law is written algebraically as below

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

This law in terms of gate implementation is illustrated in given figure.



Fig; Application of Distributive law.

Boolean Algebra Rules

Basic rules that are useful in manipulation and simplification of Boolean expressions are tabulated below.

Table: Boolean Algebra Basic Rules

Rule No.	Rule	Name of Rule
1	$A + 0 = A$	OR Rules
2	$A + 1 = 1$	
3	$A + A = A$	
4	$A + A' = 1$	
5	$A . 0 = 0$	AND Rules
6	$A . 1 = A$	
7	$A . A = A$	
8	$A . A' = 0$	
9	$A'' = A$	Complementation Rules
10	$A + AB = A$	Absorptive Rules
11	$A + A'B = A + B$	
12	$(A + B)(A + C) = A + BC$	

Proof of Rule 10 $\quad A+AB = A$

$LHS = A+AB$

$= A(1+B) \quad \because 1+B=1$

$= A = RHS$

Proof of Rule 11 $\quad A+\bar{A}B = A+B$

$LHS = A+\bar{A}B = (A+AB)+\bar{A}B \quad \because A = A+AB$

$= AA+AB+A\bar{A}+\bar{A}B$

$= A(A+\bar{A})+B(A+\bar{A}) \quad \because A+\bar{A}=1$

$= A+B = RHS$

Proof of Rule 12

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

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

$= AA+AC+AB+BC$

$= A+AC+AB+BC$

$= A(1+C)+AB+BC$

$= A+AB+BC$

$= A(1+B)+BC$

$= A+BC$

$= RHS$

De-Morgan's Theorems

These theorems are extremely useful in simplification of expressions in which a sum or product of variables is inverted.

First Demorgan's Theorem

The complement of the sum of two or more variables is equal to the product of the complements of the variables. In other way, the complement of two or more variables ORed is the same as the AND of the complements of each individual variables as shown in figure

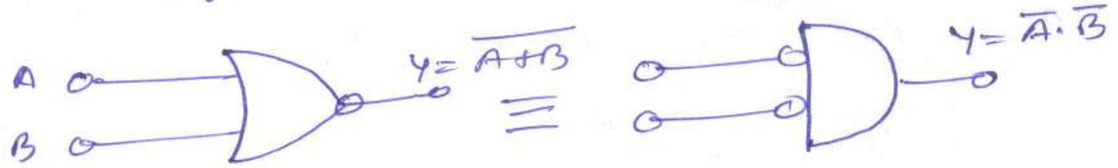


Figure illustrate the first Demorgan's theorem and the expression for it is as.

$$\boxed{\overline{A+B} = \overline{A} . \overline{B}}$$

A	B	$\overline{A+B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

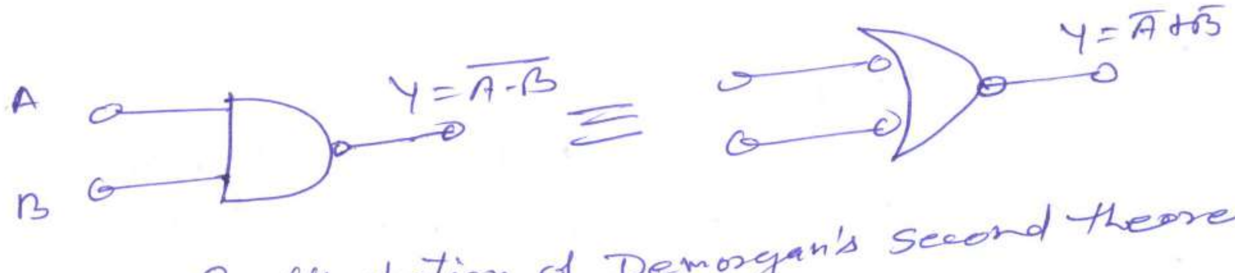
Second D-morgan's Theorem

The complement of the product of two or more variables is equal to the sum of the complements of the variables.

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

In other way, the complement of two or more variables ANDed is the same as the OR of the complements of each ~~ind~~ individual variables.

This theorem is illustrated by the gate equivalant circuit and truth table respectively.



Fig; Illustration of Demorgan's Second theorem

A	B	$\overline{A \cdot B}$	$\overline{A} + \overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Example: De-Morganization of given expression.

$$Y = \overline{(\bar{A}+C).(B+\bar{D})}$$

Step 1 Break the bar over the expression and change the sign (. to +).

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

Step 2 Break the bar again

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

Step 3 Cancel the double inversion.

$$\boxed{Y = A.\bar{C} + \bar{B}.D}$$

Duality Theorem: Each Boolean expression has its 'dual' which is as true as the original expression. The dual of a given Boolean expression can be obtained by following procedure ~~given~~

1. Change each OR(+) sign to an AND (.) sign
2. Change each AND (.) sign to an OR (+) sign
3. Complement any 0 or 1 appearing in the expression

Relation		Dual Relation
$A+0=A$	$\rightarrow$	$A.1=A$
$A+\bar{A}=1$	$\rightarrow$	$A.\bar{A}=0$
$A(B+C)=AB+AC$	$\rightarrow$	$A+(B.C)=(A+B).(A+C)$
$A+AB=A$	$\rightarrow$	$A(A+B)=A$