CLASS-XI

Study Notes

Boolean Algebra

Period-1

Introduction to Boolean Logic:

Learning Outcomes

At the end of this chapter, students should be able to:-

- To Develop Boolean Logic.
- Concepts of Binary Valued Quantities.
- Logical Operations
- Basic Logic Gates
- Postulates of Boolean Logic
- Theorems on Boolean Algebra
- Simplification of Boolean Expression
- Draw different Logic Circuits

George Boolie :

Father of Boolean algebra

- Boolean algebra derives its name from the mathematician George Boole (1815-1864) who is considered the "Father of symbolic logic".
- He came up with a type of boolean algebra, the three most basic operations of which were (and still are) AND, OR and NOT.
- It was these three functions that formed the basis of his premise, and were the only operations necessary to perform comparisons or basic mathematical functions.

BOOLEAN ALGEBRA

A variable used in Boolean algebra or Boolean equation can have only one of two variables. The two values are FALSE (0) and TRUE (1)

A Sentence which can be determined to be TRUE or FALSE are called logical statements or truth functions and the results TRUE or FALSE is called Truth values.

Boolean Expression consists of

- Literal: A variable or its complement
- Product term: literals connected by •
- Sum term: literals connected by +

- A truth table is a mathematical table used in logic to computer functional values of logical expressions.

Binary valued Quantities

A binary decision diagram (BDD) is a way to visually represent a boolean function. One application of BDDs is in CAD software and digital circuit analysis where they are an efficient way to represent and manipulate boolean functions. A **binary decision** is a choice between two alternatives, for instance between taking some specific action or not taking it.

- Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values. – In formal logic, these values are "true" and "false." – In digital systems, these values are "on" and "off," 1 and 0, or "high" and "low."
- When we learned numbers like 1, 2, 3, we also then learned how to add, multiply, etc. with them.
- Boolean Algebra covers operations that we can do with 0's and 1's. Computers do these operations ALL THE TIME and they are basic building blocks of computation inside your computer program.

LOGICAL OPERATORS

There are three logical operator, AND, OR and NOT.

These operators are now used in computer construction known as switching circuits.

B = {0, 1} and two binary operators, '+' and '**.**'

The rules of operations: AND, OR and NOT.

AND

x	y	x · y
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

NOT

x	x′
0	1
1	0

AND OPERATOR

The AND operator is a binary operator. This operator operates on two or more variables.

The operation performed by AND operator is called logical multiplication.

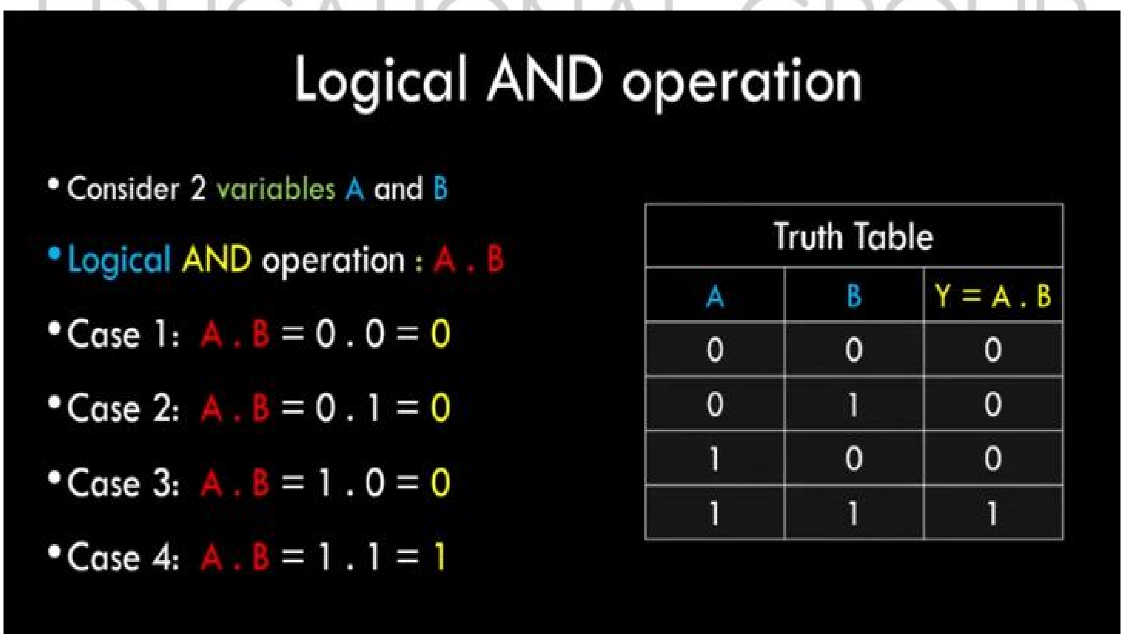
The symbol we use for it is '.'

Example: A . B can be read as A AND B

The Truth table for the AND Operation is :

In Conclusion : The output will be true, when all the elements are high.

If x and Y are two inputs to the gate, then the output will be **R=A . B**



3 Input to AND operator

A	B	C	R
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

OR OPERATOR

The OR operator is a binary operator. This operator operates on two or more variables.

The operation performed by AND operator is called logical multiplication.

The symbol we use for it is ‘ + ’

Example: X . Y can be read as X OR Y

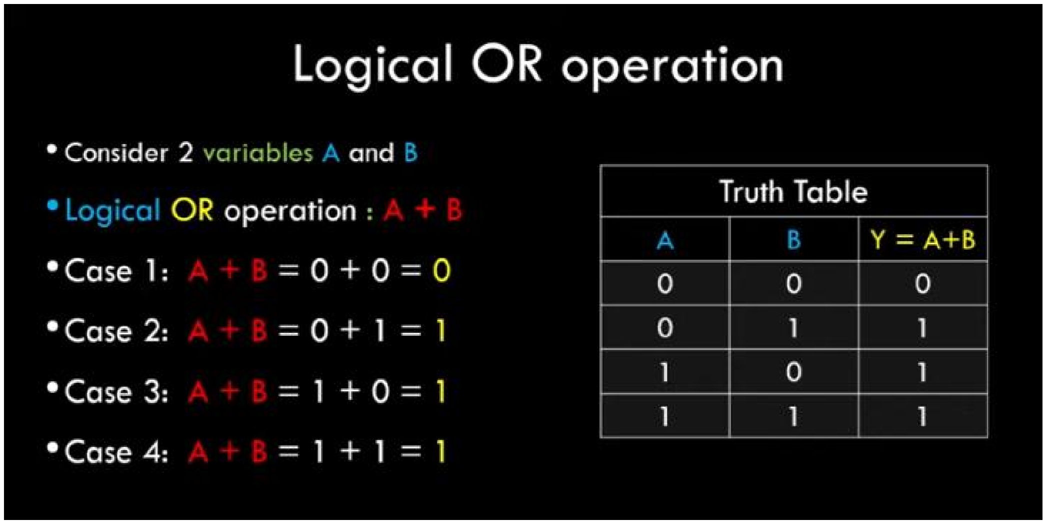
The Truth table for the OR Operation is :

OR

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

In Conclusion : The output will be true, when any one elements is high.

If X and Y are two inputs to the gate, then the output will be **Y=X + Y**



3 Input to OR operator

X	Y	Z	R
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Y = X + Y + Z

NOT OPERATOR

The Not operator is a unary operator. This operator operates on single variable.

NOT

x	x'
0	1
1	0

The operation performed by Not operator is called complementation.

The symbol we use for it is bar.

X' means complementation of X

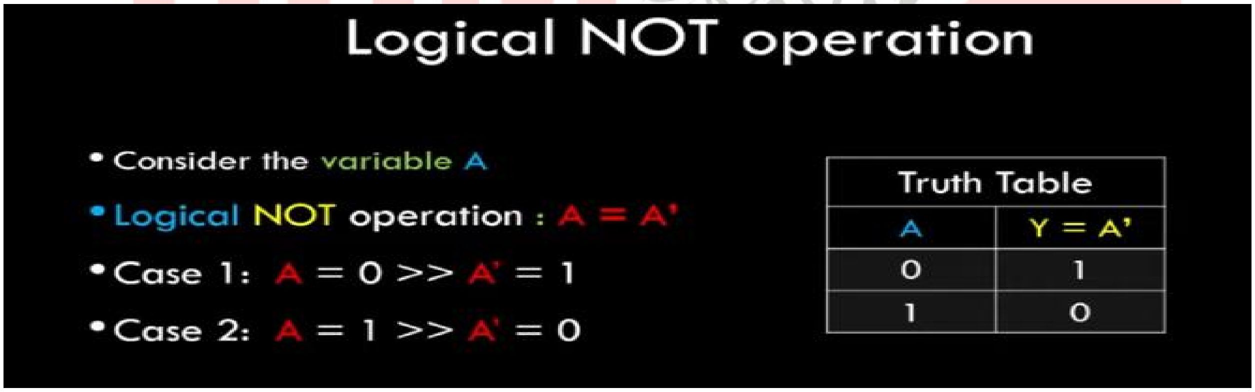
The Truth table and the Venn diagram for the NOT operator is:

TRUTH TABLE

Truth table is a table that contains all possible values of logical variables/statements in a Boolean expression.

No. of possible combination = 2^n, where n=number of variables used in a Boolean expression.

If the output of Boolean expression is always True or 1 is called **Tautology**. If the output of Boolean expression is always False or 0 is called **Fallacy**



Draw the Truth Table of XY + Z

X	Y	Z	XY	XY+Z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

Using Boolean Logic: Verify using truth table that

(X + Y)' = X' . Y' for each X , Y in [0,1]

X	Y	X + Y	(X + Y)'	X'	Y'	X' . Y'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

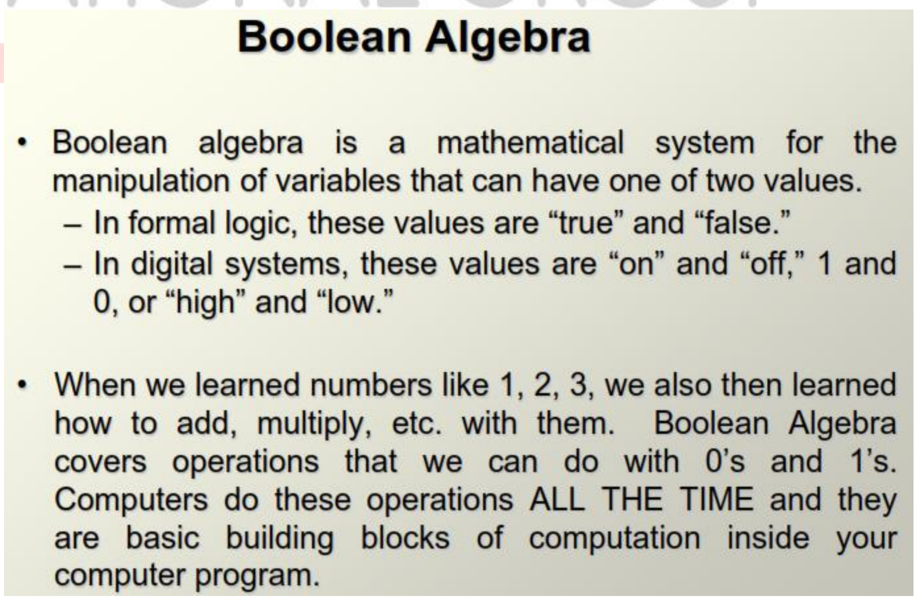
Period-2

Basic Logic Gates(AND, OR, NOT), Postulates of Boolean Logic, Principles of Duality

Learning Outcomes

At the end of this chapter, students should be able to:-

- Basic Logic Gates
- Postulates of Boolean Logic
- Principles of Duality



Boolean Algebra

- Boolean algebra is a mathematical system for the manipulation of variables that can have one of two values.
  - In formal logic, these values are "true" and "false."
  - In digital systems, these values are "on" and "off," 1 and 0, or "high" and "low."
- When we learned numbers like 1, 2, 3, we also then learned how to add, multiply, etc. with them. Boolean Algebra covers operations that we can do with 0's and 1's. Computers do these operations ALL THE TIME and they are basic building blocks of computation inside your computer program.

A collection of individual logic gates connect with each other and produce a logic design known as a Logic Circuit

- The following are the types of logic circuits:
  - Decision making
  - Memory
  - A gate has two or more binary inputs and single output.

LOGIC GATES

A gate is an digital circuit which operates on one or more signals and produce single output.

Gates are digital circuits because the input and output signals are denoted by either 1 (high voltage) or 0 (low voltage).

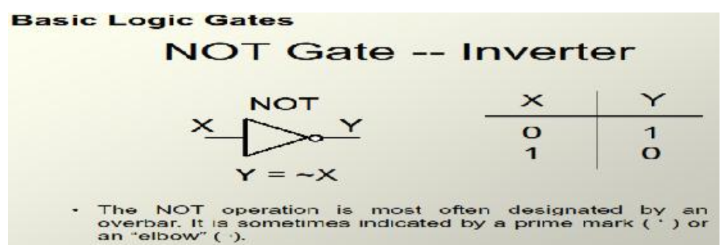
There are three basic gates and are:

1. AND Gate
2. OR Gate
3. NOT Gate

NOT GATE

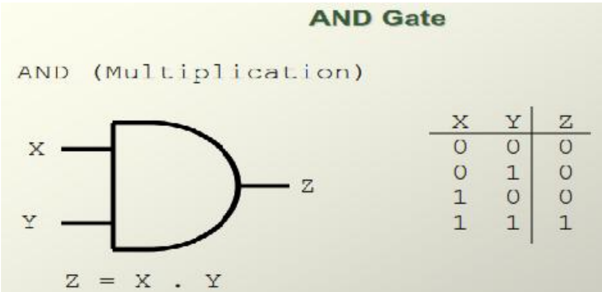
- The simplest form of a digital logic circuit is the inverter or the NOT gate
- It consists of one input and one output and the input can only be binary numbers namely; 0 and 1

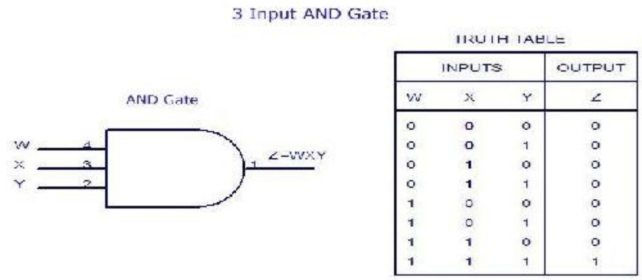
the truth table for NOT Gate:



AND Gate

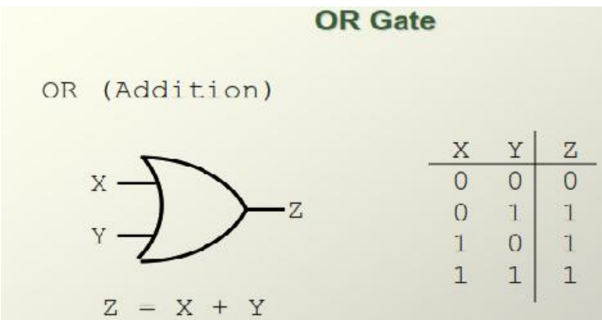
- The AND gate is a logic circuit that has two or more inputs and a single output
- The operation of the gate is such that the output of the gate is a binary 1 if and only if all inputs are binary 1
- Similarly, if any one or more inputs are binary 0, the output will be binary 0.

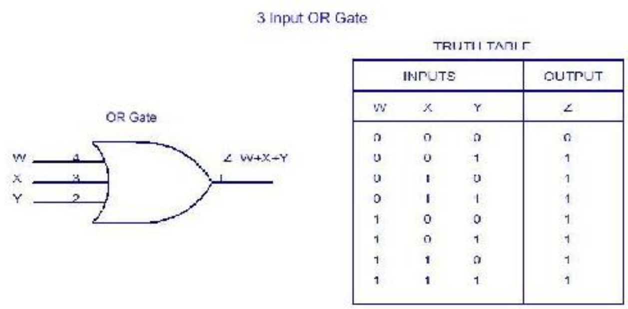




OR Gate

- The OR gate is another basic logic gate
- Like the AND gate, it can have two or more inputs and a single output
- The operation of OR gate is such that the output is a binary 1 if any one or all inputs are binary 1 and the output is binary 0 only when all the inputs are binary 0





BASIC POSTULATES OF BOOLEAN LOGIC

I. If X != 0, then X = 1; and if X != 1 the X = 0

II. OR Relations (Logical Addition)

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

III. AND Relations (Logical Multiplication)

$0 . 0 = 0$
$0 . 1 = 0$
$1 . 0 = 0$
$1 . 1 = 1$

IV. Complement Rules :

0 ‘ = 1

1’ = 0

PRINCIPLES OF DUALITY

This states that starting with a Boolean Relation , another Boolean Relation can be derived by :

1. Changing each OR sign (+) to an AND sign (.)
2. Changing each AND sign (.) to an OR sign (+)
3. Replacing each 0 by 1 and Each 1 by 0

Example:

(A + 0) . (A. 1. A') = (A. 1) + (A + 0 + A')

X + X' Y = X . (X' + Y)

(A + B) . (A' + B) = (A . B) + (A' . B)

Assignments

Write down the Duals for the following Expression :

1. X' Y' + X' Y + Z
2. X Y Z' + X' Y' + X' . 0
3. (X' Y Z' + Y') . (X + Z) + X . Y'
4. X (Y' + Z') + X Y' . X
5. X Y' (Z + Y Z') + Z' . Y
6. A [(B' + C) + C'] . A . (B + 0)

Period-3

Theorems on Boolean Algebra

Learning Outcomes

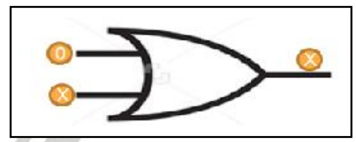
At the end of this chapter, students should be able to:-

- Learn the concepts of different theorems
- Use of theorems in Truth Table
- Principles of Duality

1. Properties of 0 :

(a) 0 + X = X AND 0.X =0

Circuit Design :



The Truth table for the above expression is as follows:

0	X	R
0	0	0
0	1	1

Where R signifies the output.

Truth Table for 0 + X = X

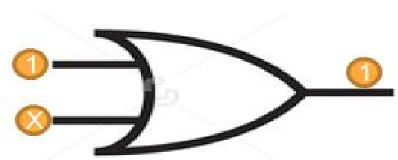
As X can have values either 0 and 1, both the values ORed with 0 Produces the same output as X.

The same is also true for AND Gate.

2. Properties of 1 :

(a) 1 + X = 1 AND 1.X =X

Circuit Design :



1	X	R
1	0	1
1	1	1

The Truth table for the above expression is as follows:

Where R signifies the output.

Truth Table for 1 + X = 1 is as above.

As X can have values either 0 and 1, both the values ORed with 1 Produces the same output as X.

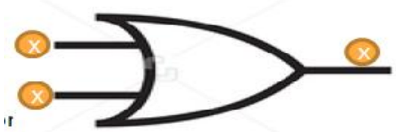
The same is also true for AND Gate.

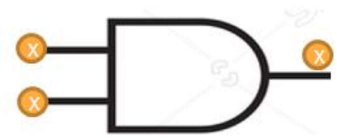
3. Indempotence Law :

This Law states that :

(a) X + X = X (B) X . X = X

Circuit Diagram :





Truth Table

X	X	X+X
0	0	0
1	1	1

X	X	X.X
0	0	0
1	1	1

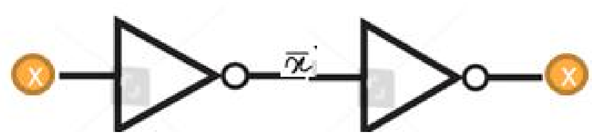
Hence proved.

4. Involution Law :

This Law states that :

(a) $\overline{(\overline{x})}$ = X

Circuit Diagram :



Truth Table

X	X’	(X’)’
0	1	0
1	0	1

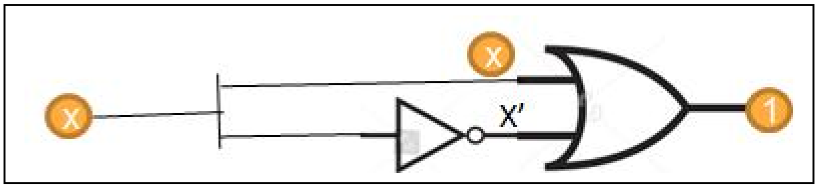
Hence proved.

5. Complementarity Law:

This Law states that :

(a) X + X' = 1 (b) X . X' = 0

Circuit Diagram :



The Truth table for the above expression is as follows:

X	X'	X+X'
0	1	1
1	0	1

Hence proved.

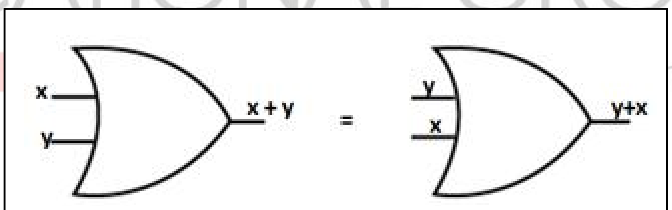
The same is also true for AND Gate.

6. Commutative Law :

This Law states that :

(a) X + Y = Y + X (b) X . Y = Y . X

Circuit Diagram :



The Truth table for the above expression is as follows:

X	X	X+Y	Y+X
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Hence proved.

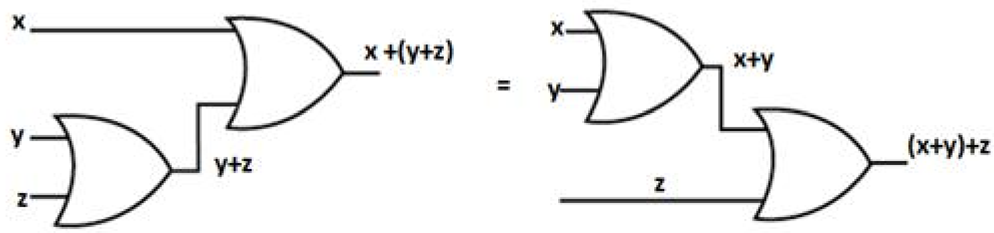
The same is also true for Other Gate.

7. Associative Law :

This Law states that :

(a) X + (Y + Z) = (X + Y) + Z (b) X . (Y . Z) = (X . Y) . Z

Circuit Diagram :



The Truth table for the above expression is as follows:

X	Y	Z	Y+Z	X+(Y+Z)	X+Y	(X+Y)+Z
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

Hence proved.

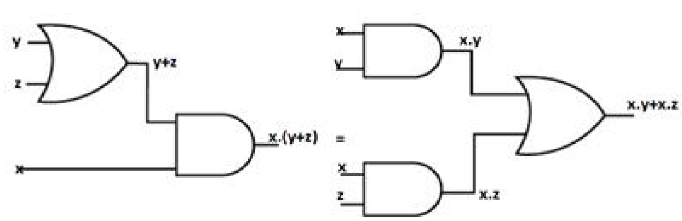
The same is also true for Other Gate.

8. Distributive Law :

This Law states that :

(a) X . (Y + Z) = X .Y + X.Z (b) X + (Y . Z) = X . Y +X . Z

Circuit Diagram :



The Truth table for the above expression is as follows:

X	Y	Z	Y+Z	X.(Y+Z)	X.Y	X.Z	XY+XZ
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Hence proved.

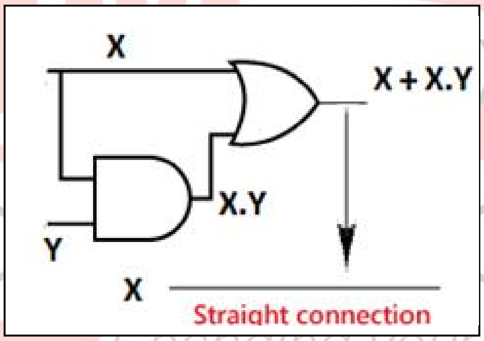
The same is also true for Other Gate.

9. Absorption Law :

This Law states that :

(a) X + X .Y = X (b) X . (X + Y) = X

Circuit Diagram :



The Truth table for the above expression is as follows:

X	Y	X.Y	X+XY
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

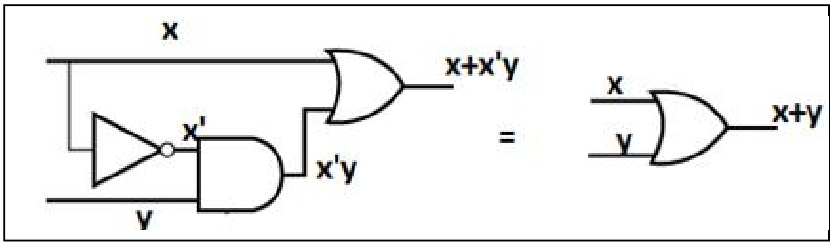
Hence proved.

The same is also true for Other Gate.

10. Third Distribution Law :

This Law states that : X + X' .Y = X + Y

Circuit Diagram :



The Truth table for the above expression is as follows:

X	Y	X'	X'Y	X+X'Y	X+Y
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Hence proved.

Rules of Boolean Algebra

1. Properties of 0 :	(a)	0 + X = X	(b) 0 . X = 0
2. Properties of 1 :	(a)	1 + X = 1	(b) 1 . X = X
3. Indempotence Law :	(a)	X + X = X	(b) X . X = X
4. Involution Law :		(X')' = X	
5. Complementarity Law:	(a)	X + X' = 1	(b) X . X' = 0
6. Commutative Law :	(a)	X + Y = Y + X	(b) X . Y = Y . X
7. Associative Law :	(a)	X + (Y + Z) = (X + Y) + Z	(b) X . (Y . Z) = (X . Y) . Z
8. Distributive Law :	(a)	X . (Y + Z) = X .Y + X.Z	(b) X + (Y . Z) = X . Y +X . Z
9. Absorption Law :	(a)	X + X .Y = X	(b) X . (X + Y) = X
10. Third Distribution Law :		X + X' .Y = X + Y	

Period-4

De-Morgan's Theorem & More on Logic gates

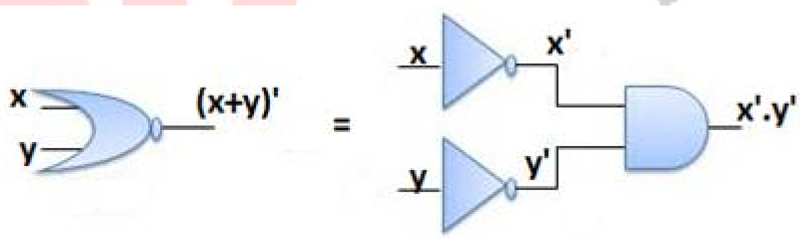
Learning Outcomes

At the end of this chapter, students should be able to:-

- Learn the concepts of De-Morgan's theorems
- Learn different other Logic Gates
-

DE-MORGAN'S 1ST THEOREM

It States That, $(X + Y)' = X' . Y'$

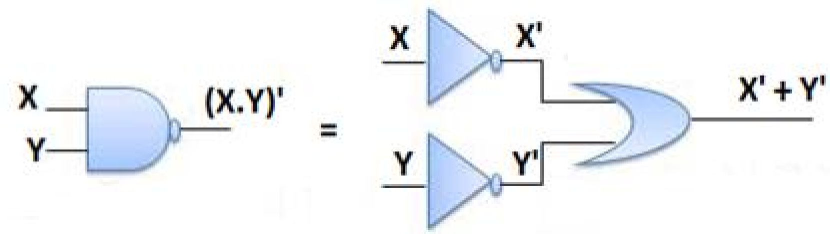


The Truth table for the above expression is as follows:

X	Y	X+Y	(X+Y)'	X'	Y'	X'.Y'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Hence proved.

DE-MORGAN'S 2ND THEOREM **It States That, (X . Y) ' = X' + Y'**



The Truth table for the above expression is as follows:

X	Y	X.Y	(X.Y)'	X'	Y'	X'+Y'
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Hence proved.

DE-MORGAN'S 1ST THEOREM BY Algebraic Proof :

It States That, (X + Y) ' = X' . Y'

Proof:

To prove this Theorem, we need to recall the Complimentarity Laws, Which State that

X + X' = 1 AND X . X' = 0

i.e., a logical variable/expression when added with its complement produces the output as 1 and when multiplied with its compliment produces the output as 0.

Let us assume that, P = X + Y, Where P, X, Y are logical/Boolean variables, then according to Complimentarity Laws,

P + P' = 1 and P . P' = 0

That means, if P, X, Y are Boolean variables, then this Complimentarity Law must hold for variable P too.

In other words, $P = X + Y \Rightarrow P' = (X + Y)' = X' . Y'$, Then

$P + P' = 1 \Rightarrow (X + Y) + (X + Y)' = 1 \Rightarrow (X + Y) + (X' . Y') = 1$ [$X + X' = 1$]

$P . P' = 0 \Rightarrow (X + Y) . (X + Y)' = 0 \Rightarrow (X + Y) . (X' . Y') = 0$ [$X . X' = 0$]

Let us 1st Prove that,

$(X + Y) + (X' . Y') = 1$

L.H.S.

$(X + Y) + (X' . Y')$

$= ((X + Y) + X') . ((X + Y) + Y')$ [$X + YZ = (X + Y) . (X + Z)$]

$= (X + X' + Y) . (X + Y + Y')$

$= (1 + Y) . (X + 1)$ [$X + X' = 1$]

$= (1 + Y) . (1 + X)$ [$X + Y = Y + X$]

$= 1.1.$ [$1 + X = 1$]

$= 1$

Hence, Proved.

Let us Prove the second part that,

$(X + Y) . (X' . Y') = 0$

L.H.S.

$(X + Y) . (X' . Y')$

$= (X' . Y') . (X + Y)$ [$X . Y = Y . X$]

$= (X . X' . Y) + (X . Y . Y')$ [$X .(Y+Z) = X.Y + X.Z$]

$= (0 . Y) + (X . 0)$ [$X . X' = 0$]

$= 0 . 0$

$= 0$

Hence, Proved.

Hence, 1st theorem on Boolean Algebra is Proved.

DE-MORGAN'S 2nd THEOREM BY Algebraic Proof :

It States That, $(X . Y)' = X' + Y'$

Proof:

To prove this Theorem, we need to recall the Complimentarity Laws, Which State that

$X + X' = 1$ AND $X . X' = 0$

i.e., a logical variable/expression when added with its complement produces the output as 1 and when multiplied with its compliment produces the output as 0.

Let us assume that, $P = X . Y$, Where P, X, Y are logical/Boolean variables, then according to Complimentarity Laws,

$P + P' = 1$ and $P . P' = 0$

That means, if P, X, Y are Boolean variables, then this Complimentarity Law must hold for variable P too.

In other words, $P = X . Y \Rightarrow P' = (X . Y)' = X' + Y'$, Then

$P + P' = 1 \Rightarrow (X . Y) + (X' + Y') = 1$

$P . P' = 0 \Rightarrow (X . Y) . (X' + Y') = 0$

Let us 1st Prove that,

$(X . Y) . (X' + Y') = 1$

L.H.S.

$(X . Y) + (X' + Y')$

$= (X' + Y') + X . Y)$ $\quad [X + Y = Y + X]$

$= (X' + Y' + X) . (X' + Y' + Y)$ $\quad [X + YZ = (X + Y) . (X + Z)]$

$= (X + X' + Y') . (Y + Y' + X)$

$= (1 + Y') . (1 + X)$ $\quad [X + X' = 1]$

$= 1.1.$ $\quad [1 + X = 1]$

$= 1$

Hence, Proved.

Let us Prove THE SECOND PART that,

$(X . Y) . (X' + Y') = 0$

L.H.S.

$(X . Y) . (X' + Y')$

$= (X . Y . X' + X . Y . Y')$ $[X + (Y+Z) = X.Y + X.Z]$

$= (X . X' . Y') + (X . Y . Y')$

$= (0 . Y') + (X . 0)$ $[X . X' = 0]$

$= 0 + 0$

$= 0$

Hence, Proved.

Hence, 2nd theorem on Boolean Algebra is Proved.

More on Logic Gates

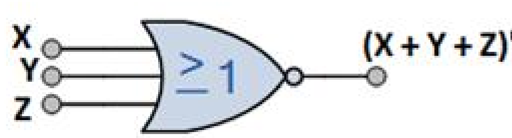
NOR Gate :

This gate has two or more input signals and gives out one output signal, the output will be true or high, when all the inputs to the gate are Low.

Circuit Design:

2-input "OR" gate plus a "NOT" gate

2-input NOR Gate



3-input NOR Gate

Truth Table:

This gate has two or more input signals and gives out one output signal, the output will be true or high, when all the inputs to the gate are Low.

X	Y	X + Y	(X+Y)’
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

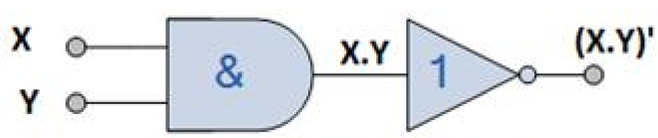
Truth Table FOR THREE INPUT NOR GATE:

X	Y	Z	X+Y+Z	(X+Y+Z)’
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

NAND Gate :

This gate has two or more input signals and gives out one output signal, the output will be false or low, when all the inputs to the gate are high.

Circuit Design:



2-input "AND" gate plus a "NOT" gate

2-input NAND Gate

3-input NAND Gate

X	Y	X . Y	(X.Y)’
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Truth Table FOR THREE INPUT NAND GATE:

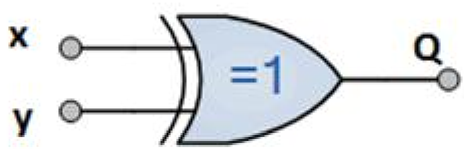
X	Y	Z	X.Y.Z	(X.Y.Z)’
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

XOR (Exclusive OR Gate) :

This gate has two or more input signals and gives out one output signal, the output will be high or produces 1 , for odd number of 1’s.

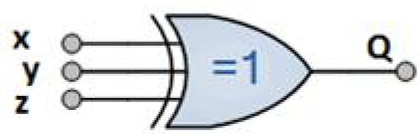
If X and Y are two input to XOR Gate, then output is represented as X $\oplus$ Y

Circuit Design:



2-input Ex-OR Gate

Boolean Expression Q = $x \oplus y$



3-input Ex-OR Gate

Boolean Expression Q = $X \oplus Y \oplus Z$

Truth Table

X	Y	X + Y	(X $\oplus$ Y)
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	0

Truth table for three input Exclusive-OR is as follows.

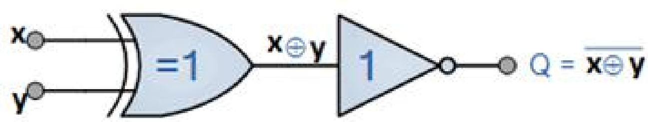
X	Y	Z	X+Y+Z	(X ⊕ Y ⊕ Z)
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

XNOR (Exclusive NOR Gate) :

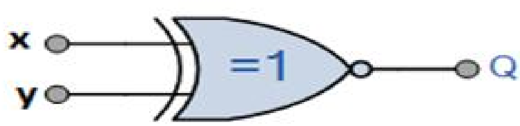
This gate has two or more input signals and gives out one output signal, the output will be high or produces 1 , for even number of 1's.

If X and Y are two input to XOR Gate, then output is represented as (X ⊕ Y)'

Circuit Design:

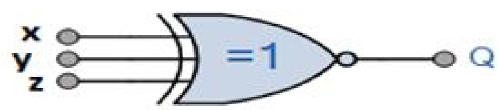


2-input "Ex-OR" gate plus a "NOT" gate



2-input Ex-NOR Gate

Boolean Expression Q = $\overline{x \oplus y}$



3-input Ex-NOR Gate

Boolean Expression Q = $\overline{x \oplus y \oplus z}$

Truth Table

X	Y	X + Y	(X ⊕ Y)'
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	1

Truth table for three input Exclusive-OR is as follows.

X	Y	Z	X+Y+Z	$(X \oplus Y \oplus Z)'$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0

Period-5

Drawing Logic Circuits

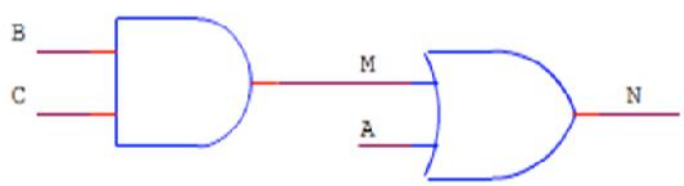
Learning Outcomes

At the end of this chapter, students should be able to:-

- Draw different Logic Circuit

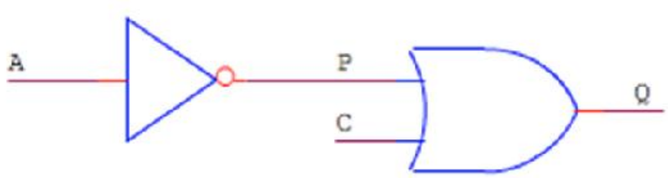
Write down the logical expression for the following Logic Circuit

1



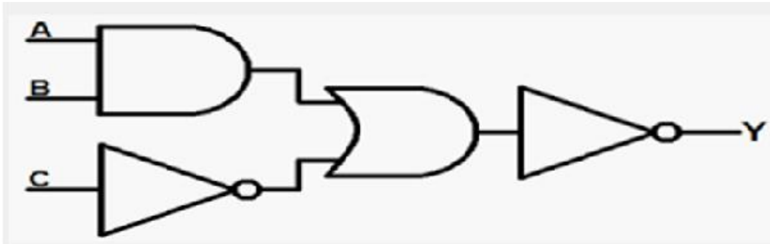
N = A + B.C

2



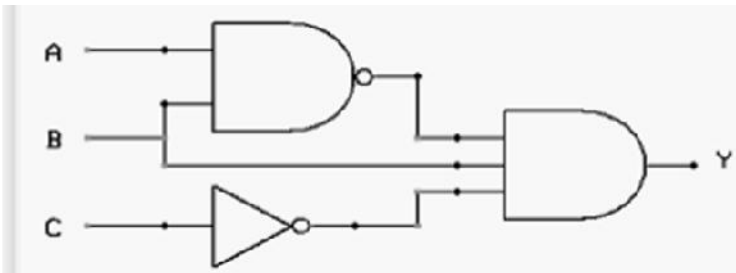
Q = A’ . C

3



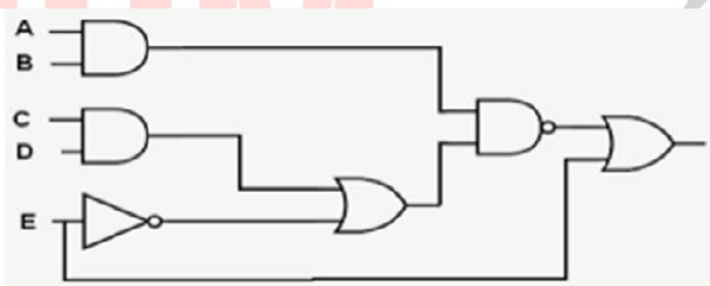
Y = (A.B+C')'

4



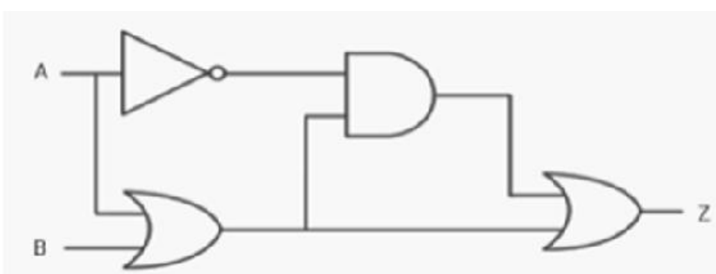
Y = (A.B)'.B.C'

5



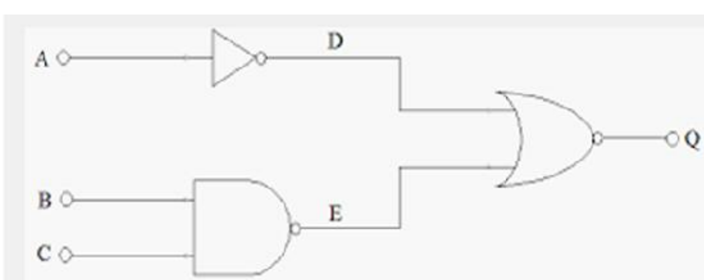
= [AB .(CD+E')]'+E'

6



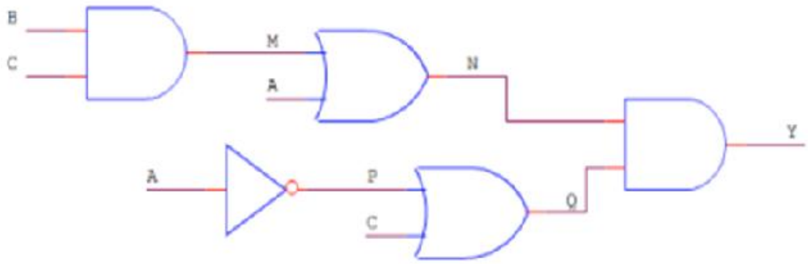
Z = (A'.A+B)+(A+B)

7



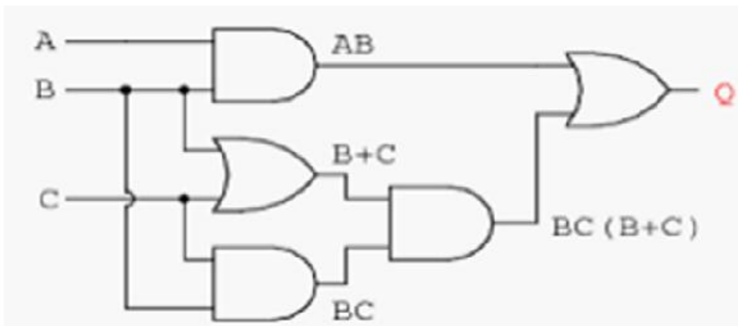
Q = (A' + (B.C)')'

8



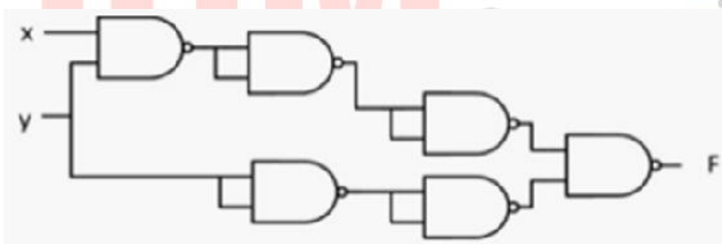
Y = (A+BC) . (A’+C)

9



Q=AB+[BC(B+C)]

10



Try it in your own

Period-6

Simplification of Boolean Expression

Learning Outcomes

At the end of this chapter, students should be able to:-

- Solve the Complex Boolean expression

1. $X + X . Y = X$

LHS

$X + X . Y$

$= X . 1 + X . Y$ [$X . 1 = X$]

$= X . (1 + Y)$

$= X . 1$ [$1 + Y = 1$]

$= X$ (PROVED)

2. $X + X' . Y = X + Y$

LHS

$X + X' . Y$

$= (X + X') . (X + Y)$ [$X + Y.Z = (X + Y) . (X + Z)$]

$= 1 . (X + Y)$ [$X + X' = 1$]

$= X + Y$ [$1 . X = X$]

(PROVED)

3. $[(X + Y)' + (X + Y)']' = X + Y$

LHS

$[(X + Y)' + (X + Y)']'$

Let $Z = (X + Y)'$

$= (Z + Z)'$ [$X + X = X$]

$= Z'$

$= [(X + Y)']'$

IF $T = X + Y$

$= (T')'$ [$(X')' = X$]

$= T$

$= X + Y$ (Proved)

4. $(A' + B') . (A + B) = A' . B + A . B'$

LHS

$(A' + B') . (A + B)$

$= (A' + B') . A + (A' + B') . B$ [$X . (Y + Z) = X.Y + X.Z$]

$= A . (A' + B') + B . (A' + B')$ [$X . Y = Y . X$]

$= A . A' + A . B' + A' . B + B . B'$

$= 0 + A . B' + A' . B + 0$ [$X . X' = 0$]

$= A' . B + A . B'$

5. Prove algebraically, X. Y + X'Z + Y.Z = X.Y + X'.Z

L.H.S. = X Y + X' Z + Y Z

= X Y + X' Z + 1. Y Z

= X Y + X' Z + (X + X') Y Z [X + X' = 1]

= X Y + X' Z + X Y Z + X' Y Z

= X Y + X Y Z + X' Z + X' Y Z

= X Y (1 + Z) + X' . Z (1 + Y)

= X Y . 1 + X' Z · 1 [1 + X = 1]

= X Y + X' Z = R.H.S. (Proved)

6. (A+ B).(A' + C) = (A+ B+ C).(A+ B+ C').(A' + B+C).(A' + B' + C) ALGEBRAICALLY.

(A + B) . (A' + C) = (A + B + C) . (A . 1 + B + C') . (A' + B+ C) . (A' + B' + C')

LHS (A + B) . (A' + C)

= (A + B+ O) . (A' + C + O)

= (A+ B + C.C') . (A' + C + B.B') (∵ C . C' = 0)

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

(∵ X + YZ =(X+ Y)(X+ Z))

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

= RHS.

7. Prove algebraically X Y' Z' + X Y' Z + X' Y Z + X' Y Z' + X Y' Z' + X Y' Z = X' Y + X Y'

L.H.S.

X Y' Z' + X Y' Z + X' Y Z + X' Y Z' + X Y' Z' + X Y' Z

= X Y' (Z' + Z) + X' Y (Z + Z') + X Y' (Z' + Z)

= X Y' + X' Y + X Y' [x + x' = 1]

= X' Y + X Y' (Proved)

8. Prove algebraically, X . Y + X‘ Z + Y . Z = X . Y + X‘ . Z

X Y + X‘ Z + Y Z = X Y + X‘ Z

L.H.S. = X Y + X‘ Z + Y Z

= X Y + X‘ Z + 1. Y Z

= X Y + X’ Z + (X + X‘) Y Z

= X Y + X‘ Z + X Y Z + X‘ Y Z

= X Y + X Y Z + X‘ Z + X‘ Y Z

= X Y (1 + Z) + X‘ Z (1 + Y)

= X Y . 1 + X‘ Z . 1

= X Y + X‘ Z = R.H.S

****