Downloaded from www.studiestoday.com

Unit-IV
Boolean Algebra

Boolean Algebra Chapter: 08

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.

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

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

Truth table:

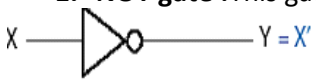
Basic Logic Gates

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 (NOT gate)
2. OR gate
3. AND gate

1. NOT gate : This gate takes one input and gives a single output. The symbol of this logic gate is



This circuit is used to obtain the compliment of a value.

If X = 0, then X' = 1.

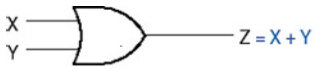
The truth table for NOT gate is :

X	X'
0	1
1	0

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).

Truth Table and circuit diagram for Two Input OR gate is :

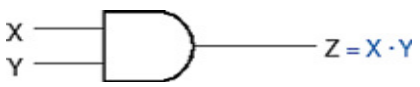
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1



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.

Truth Table and circuit diagram for Two Input AND gate is :

X	Y	F=X.Y
0	0	0
0	1	0
1	0	0
1	1	1



Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

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

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

Basic theorem of Boolean algebra

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 and 1

0+X=X 1+X=1 0.X=01.X=X

2. Indempotence Law X+X=X X.X=X
3. Involution Law (X')' = X
4. Complementarity Law X' + X=1 X. X'=0
5. Commutative Law X+Y=Y+X X.Y=Y.X
6. Associative Law X+(Y+Z)=(X+Y)+Z X(YZ)=(XY)Z
7. Distributive Law X(Y+Z)=XY_XZ X+YZ=(X+Y)(X+Z)
8. Absorption Law X+XY= XX(X+Y)=X

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

Algebraic proof of De Morgan's Theorem (First)

(a+b) + (a'b') = 1	**(a+b)(a'b') = 0.**
First Part (a+b)+(a'b') =(a+b+a')(a+b+b') (Distribution Law) =(1+b)(a+1) (Complement law) =1	Second Part :- (a+b)(a'b') =(a'b')(a+b) (Commutative law) =a'b'a+a'b'b (Distribution Law =0*b'+a'*0 (x*0=0) =0+0 =0

Note: DeMorgan's Second theorem is just the complement of the First Theorem

Minterms and Maxterms

- *Minterm is the product of all the literals* with or without bar within a logical system viz if we have two literals A and B then the possible minterms can be AB,A'B,AB',A'B'.
- *Maxterm is the sum of all the literals* with or without bar within a logical system viz if we have two literals A and B then the possible maxterms can be A+B,A'+B,A+B',A'+B'.

n Variables can be combined to form 2^n minterms or maxterms.

Minterms and Maxterms for Three Binary Variables						
			Minterms		Maxterms	
x	y	Z	Term	Shorthand Notation	Term	Shorthand Notation

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

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

- Sum of Product (SOP) Function: A Boolean function may be represented algebraically from a given truth table by forming a minterm for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.
- Product of Sum (POS) Function: A Boolean function may be represented algebraically from a given truth table by forming a maxterm for each combination of the variables that produces a 0 in the function and then taking the AND of all those terms.

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

For result F(SOP form is)= x'y'z + xy'z' + xyz

For result F(POS form is)= (x+y+z) .(x+y'+z).(x+y'+z').(x'+y+z').(x'+y'+z)

Example: Express the Boolean function $F(A,B,C) = AB + C$ as a sum of minterms.

Step 1 – Each term must contain all variables

$AB = AB(C + C') = ABC + ABC'$

$C = C(A + A') = AC + A'C$

$= AC(B + B') + A'C(B + B')$

$= ABC + AB'C + A'BC + A'B'C$

Step 2 – OR all new terms, eliminating duplicates

$F(A,B,C) = A'B'C + A'BC + AB'C + ABC' + ABC$

$= m_1 + m_3 + m_5 + m_6 + m_7$

$= \Sigma(1, 3, 5, 6, 7)$

Example: Express the Boolean function $F(x,y,z) = x'y + xz$ as a product of maxterms.

Step 1 – Convert the function into OR terms using the distributive law

$F(x,y,z) = (x'y + x)(x'y + z)$

$= (x + x')(y + x)(x' + z)(y + z)$

$= (y + x)(x' + z)(y + z)$

Step 2 – Each term must contain all variables

$y + x = y + x + zz' = (x + y + z)(x + y + z')$

$x' + z = x' + z + yy' = (x' + y + z)(x' + y' + z)$

$y + z = y + z + xx' = (x + y + z)(x' + y + z)$

step 3 – AND all new terms, eliminating duplicates

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

$F(x,y,z) = (x + y + z)(x + y + z')(x' + y + z)(x' + y' + z)$

$= (M_0 M_1 M_4 M_6)$

$= \Pi(0, 1, 4, 6)$

Conversion between Canonical Forms

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 Σ(Sigma) and Π(Pie), and list those numbers missing from the original form.

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. **Algebraic Method**
2. **Karnaugh Map :**

1.Algebric method: This method makes use of Boolean postulates, rules and theorems to simplify the expression.

Example. Simplify AB'CD + A'BCD' +ABCD +ABCD'

Solution-- AB'CD + AB'CD' +ABCD +ABCD'

=AB'C(D+D') +ABC(D+D')

=AB'C.1 + ABC.1 (D+D'=1)

=AC(B'+B)

=AC.1 =AC

2. Using Karnaugh Map :

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

For example:

- Put a 1 in the box for any minterm that appears in the SOP expansion.
- Basic idea is to cover the **largest adjacent** blocks you can whose side length is some power of 2.
- Blocks can wrap around(Map rolling) the edges.
- Redundant groups should be avoided.

Sum Of Products Reduction using K- Map

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

- Pair: Two adjacent 1's makes a pair.*Pair removes one variable.*
- Quad: Four adjacent 1's makes a quad.*Quad removes two variables.*
- Octet: Eight adjacent 1's makes an Octet.*Octet removes three variables.*

Reduction of expression: When moving vertically or horizontally in pair or a quad or an octet it can be observed that only one variable gets changed that can be eliminated directly in the expression.

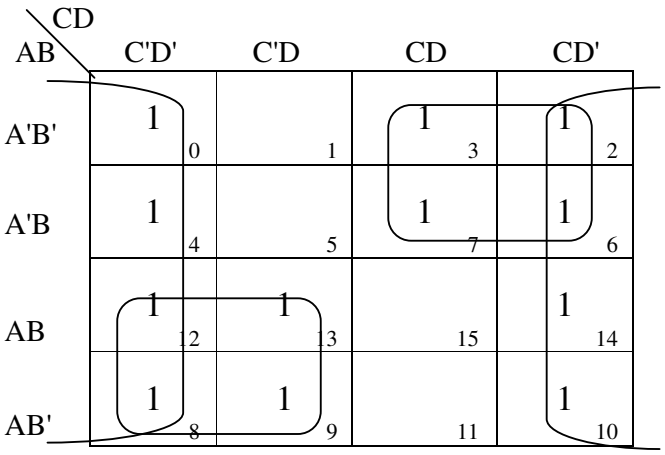
For Example

Q1. Reduce the following Boolean Expression using K-Map:

Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

F(A, B, C, D) = ∑ (0, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14)



There are 1 octet, 2 quads after eliminating the redundant groups.

Octet (m_0, m_2, m_4, m_6, m_8, m_{10}, m_{12}, m_{14}) reduces to D'

Quad (m_2, m_3, m_6, m_7) reduces to A'C

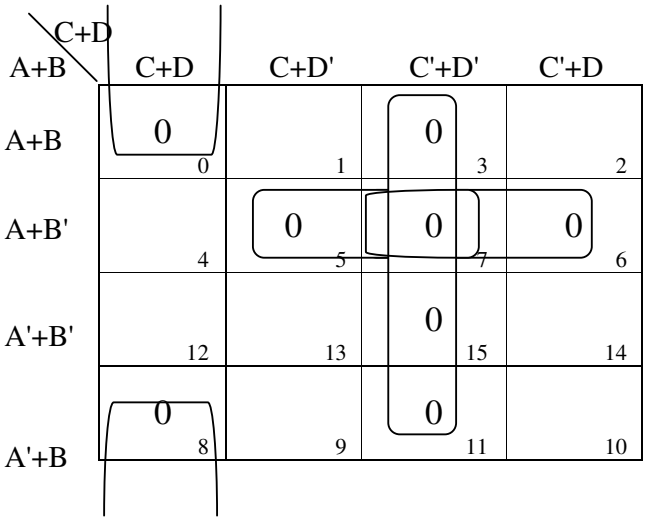
Quad (m_8, m_9, m_{12}, m_{13}) reduces to AC '

Hence, F(A, B, C, D) = D' + A'C + AC ''

Product of Sums Reduction using K- Map

Q1. Reduce the following Boolean Expression using K-Map:

F(A, B, C, D) = π (0, 3, 5, 6, 7, 8, 11, 15)



There are 1 quad and 3 pairs after eliminating the redundant groups.

Quad (M_3, M_7, M_{11}, M_{15}) reduces to C' + D'

Pair (M_5, M_7) reduces to A + B ' + D'

Pair (M_6, M_7) reduces to A + B ' + C'

Pair (M_0, M_8) reduces to B + C + D

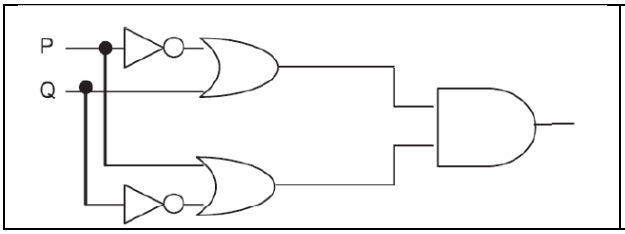
Hence, F(A, B, C, D) = (C' + D') . (A + B ' + D') . (A + B ' + C') . (B + C + D)

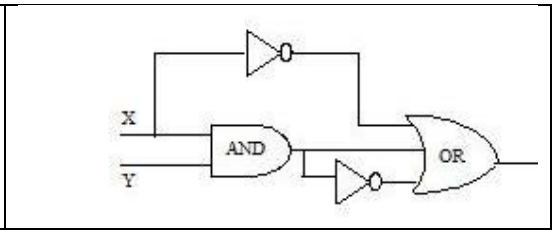
Downloaded from www.studiestoday.com

Downloaded from www.studiestoday.com

2 Marks Questions

1.Write the equivalent Boolean Expression for the following Logic Circuit





2. Draw a Logical Circuit Diagram for the following Boolean expression:

A.(B+C')

3. Prove x'.y'+y.z = x'yz+x'yz'+xyz+x'yz algebraically.

3 Marks Questions

1. If F(a,b,c,d)=∑(0,2,4,5,7,8,10,12,13,15), obtain the simplified form using K-Map.

2. Obtain a simplified form for a boolean expression

F(U,V,W,Z)= π (0,1,3,5,6,7,10,14,15)

Downloaded from www.studiestoday.com