UNIT – I – MINIMIZATION TECHNIQUES AND LOGIC GATES

PART A

1. Prove the Boolean theorems(a) x + x = x ; (b) x + xy = x

(a) x + x = x

x + x = (x + x). 1

= (x + x)(x+x')

= x + xx'

= x + 0

x + x = x

(b) x + xy = x

x + xy = x . 1 +xy

= x(1+y)

= x(y+1)

= x.1

x + xy = x

Define Noise Margin.

It is the maximum noise voltage added to an input signal of a digital circuit that does not cause an undesirable change in the circuit output. It is expressed in volts.

3.State Distributive Law.

A (B+C)=AB+AC

It states that OR in several variable and AND in the result with single variable is equivalent to AND in the result with a single variable with each of the single variables and then OR in the product.

4.What is Prime Implicant?

Each group in a K MAP gives us a product term and summation of all product term gives us a boolean expression. Therefore each product term implies the function and hence each product term is an implicant of the function. All the implicant of the function is the termed as prime implicant.

5.What are Don't Care Terms?

In some logic circuits certain input condition never occur, therefore the output of such inputs are indicated by"X" (or) DON'T CARE OUTPUTS.

$Z=\Sigma_m(1,3)+d\ (5,6,7)$

Here (1,3) are minterms ,d (5,6,7)represent DON'T cares.

6.Apply DE MORGAN Theorem to [(A+B)+C]'

$[(A+B)+C]' = (A + B)\ '\ .\ C'$

$= A'.B'.C'$

7.Simplify the following Boolean expression into one literal. W'X(Z' + YZ)+X(W + Y'Z).

$W'X(Z' + YZ)+X(W + Y'Z) = W'X(Z'+YZ) + X(W+Y'Z)$

$= X\ (W'(Z'+Y) +W+Y'Z)$

$= X\ (W'Z+W+Y+Y'Z)$

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

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

$= X(W +1 +Y)$

$= X$

8.Express the function of Y =A+ B'C Y =A+ B'C

The function has three variables A,B,C. The first term A is missing two variables.

Therefore,

$A = A(B+B')$

$= AB + AB'$

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

$= ABC + ABC' +AB'C +AB'C'$

The second term B'C is missing one variable,

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

Combining all terms,

$Y = A + B'C$

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

Define Minterm & maxterm

- N variables forming an OR term, with each variable being primed or unprimed, provide 2^n possible combinations called minterm.
- N variables forming an AND term, with each variable being primed or unprimed, provide 2^n possible combinations called maxterm.

10.Simplify the following expression X.Y+X(Y+Z)+Y(Y+Z)

= XY+XY+XZ+YY+YZ

=XY+XZ+Y+YZ [Y.Y=Y]

=XY+XZ+Y(1+Z) [1+Z=1]

=XY+XZ+Y

=Y(X+1)+XZ [X+1=1]

=Y+XZ

PART - C

1.Simplify the function using Quine Mckluskey method and verify the saming using K Map for the function F(A,B.C,D) = Σ(1,2,3,5,6,7,9,10,11,13,14,15)

Solution:

Step:1 Minimize the minterms using binary numbers.

Minterm	Binary number
1	0001
2	0010
3	0011
5	0101
7	0111
9	1001
10	1010
11	1011
13	1101
15	1111

Step 2: Group the minterms according to number of 1's

No of 1's	Minterm	Binary Number
1	1	0001
	2	0010
2	3	0011
	5	0101
	9	1001
	10	1010
3	7	0111
	11	1011
	13	1101
4	15	1111

Step 3: Compare each binary number in each group with every term in the adjacent higher group for they differ only by one position. Repeat this step for various cell combinations

2-cell combination

Minterms	Binary number
1,3	00_1 ✓
1,5	0_01 ✓
1,9	_001 ✓
2,3	001_ ✓
2,10	_010 ✓
3,7	0_11 ✓
3,11	_011 ✓
5,7	01_1 ✓
5,13	_101 ✓
9,11	10_1 ✓
9,13	1_01 ✓
10,11	101_ ✓
7,15	_111 ✓

11,15	1_11 ✓
13,15	11_1 ✓

4-cell combination

Minterms	**Binary number**
1,3,5,7	0_ _1 ✓
1,3,9,11	_0_1 ✓
1,9,5,13	_ _01 ✓
2,3,10,11	_01_
3,11,7,15	_ _11 ✓
5,7,13,15	_1_1 ✓
9,11,13,15	1_ _1 ✓

8-cell combination

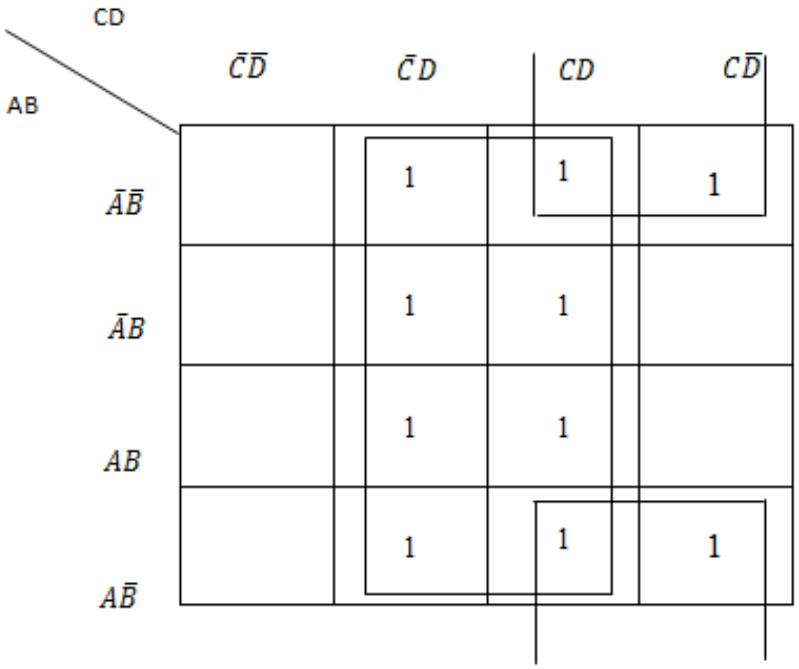
Minterms	**Binary number**
1,3,5,7,9,11,13,15	_ _ _1

Step 4: Form the prime implication table and find the Boolean expression

	1	2	3	5	7	9	10	11	13	15
1,3,5,7,9,11,13,15	*		*	*	*	*		*	*	*
2,3,10,11		*	*				*	*		
	✓	✓		✓	✓	✓	✓		✓	✓

$$F = D + \bar{B}C$$

Step 5: Draw the K-Map and verify the Boolean expression



$$F = D + \bar{B}$$

2. Simplify the function F using Tabulation method, Model- <u>NOV/DEC 2015</u>

$$f(A, B, C, D) = \Sigma m\ (1, 3, 5, 8, 9, 11, 15) + d(2, 13)$$

<u>Solution</u>

<u>Step 1</u>

Grouping of minterms/don't care terms according to number of 1's.

Group	Minterm/ don't care term	Variables A	B	C	D	Check for inclusion in group of 2
1	1	0	0	0	1	✓
	2*	0	0	1	0	✓
	8	1	0	0	0	✓
2	3	0	0	1	1	✓
	5	0	1	0	1	✓
	9	1	0	0	1	✓
	11	1	0	1	1	✓
3	13*	1	1	0	1	✓
4	15	1	1	1	1	✓

Step 2

Grouping of 2 minterms/don't care terms

Group	Minterms/ don't care terms	Variables A	B	C	D	Check for inclusion in group of 4
1	1, 3	0	0	—	1	✓
	1, 5	0	—	0	1	✓
	1, 9	—	0	0	1	✓
	2*, 3	0	0	1	—	
	8, 9	1	0	0	—	
	3, 11	—	0	1	1	✓
	5, 13*	—	1	0	1	✓
2	9, 11	1	0	—	1	✓
	9, 13*	1	—	0	1	✓
3	11, 15	1	—	1	1	✓
	13, 15	1	1	—	1	✓

Step 3

Grouping of 4 minterms/don't care terms

Group	Minterms/ don't care terms	Variables			
		A	*B*	*C*	*D*
1	1, 3, 9, 11	—	0	—	1
	1, 5, 9, 13*	—	—	0	1
	1, 9, 3, 11	—	0	—	1
	1, 9, 5, 13*	—	—	0	1
	9, 11, 13*, 15	1	—	—	1
2	9, 13*, 11, 15	1	—	—	1

Step 4

Prime Implication Table and Boolean Expression

PI terms	Decimal numbers	Minterms/don't care terms								
		1	2*	3	5	8	9	11	13*	15
$\overline{B}D$	1, 3, 9, 11	×		×			×	×		
$\overline{C}D$	1, 5, 9, 13* ✓	×			⊗		×		×	
AD	9, 11, 13*, 15 ✓						×	×	×	⊗
$\overline{A}\overline{B}C$	2*, 3		×	×						
$A\overline{B}C$	8, 9	✓				⊗	×			
					✓	✓				✓

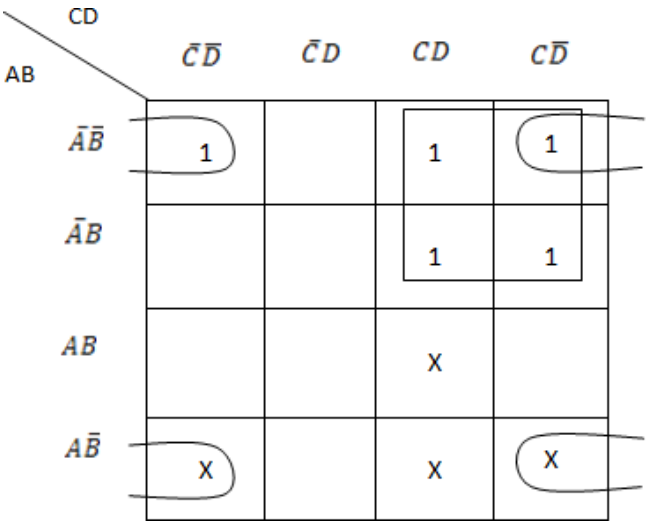
$$f(A, B, C, D) = \overline{B}D + \overline{C}D + AD + A\overline{B}\overline{C}$$

3.Simplify the function in SOP and POS using K-Map

$F = \Sigma(0,2,3,6,7)+d(8,10,11,13,14,15)$

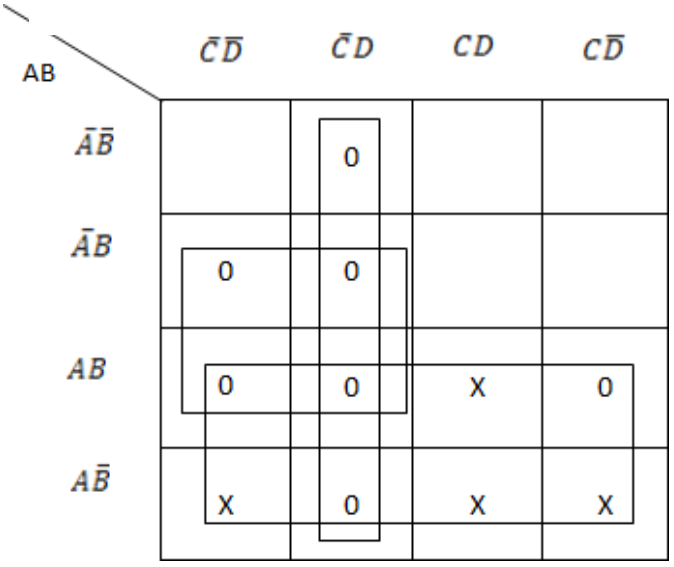
Solution:

(i) Sum of Products (SOP)



$$F = \bar{A}\bar{C} + \bar{}$$

(i) Product of Sum (POS)



$$F = A + B\bar{C} + \bar{C}D$$

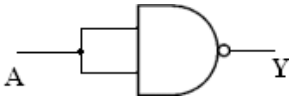
4. Implement basic logic gates using UNIVERSAL gates.

OR, AND and NOT gates are the three basic logic gates as they together can be used to construct the logic circuit for any given Boolean expression. NOR and NAND gates have the property that they individually can be used to hardware-implement a logic circuit corresponding to any given Boolean expression. That is, it is possible to use either

only NAND gates or only NOR gates to implement any Boolean expression. This is so because a combination of NAND gates or a combination of NOR gates can be used to perform functions of any of the basic logic gates. It is for this reason that NAND and NOR gates are universal gates.

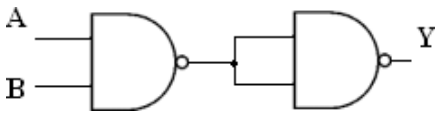
Implementation using NAND gate:

(a) NOT gate



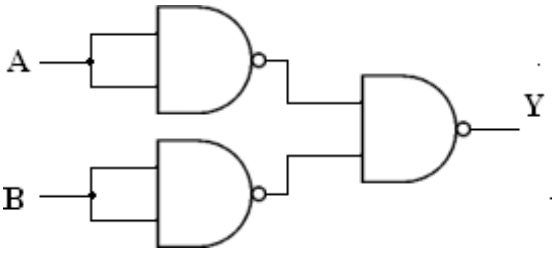
A	Y
0	1
1	0

(b) AND gate



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

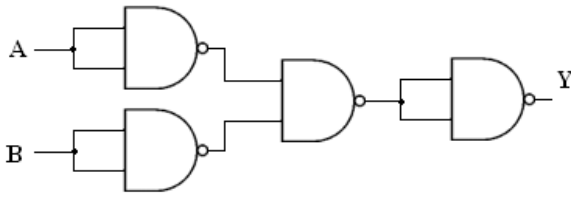
(c) OR gate



A	B	Y
0	0	0
0	1	1
1	0	1

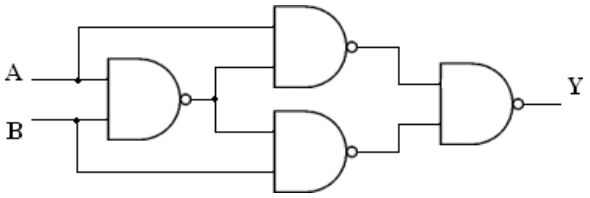
1	1	1

(d) NOR gate



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

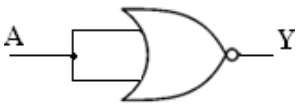
(e)Ex-OR gate



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

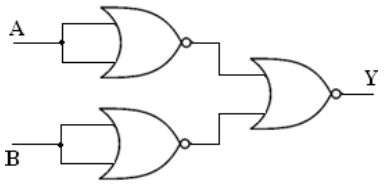
Implementation using NOR gate:

(a) NOT gate



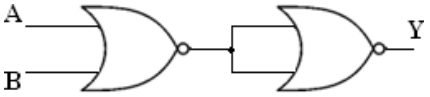
A	Y
0	1
1	0

(b) AND gate



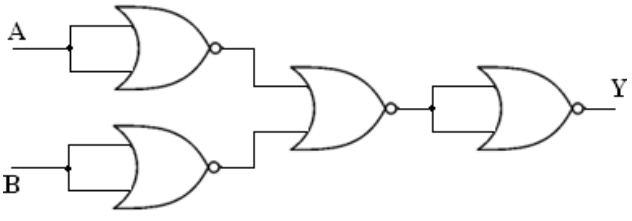
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

(c) OR gate



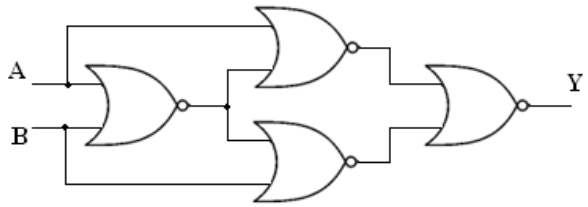
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

(d) NAND gate



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

(e) Ex-NOR gate



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1