MODULE 3 : BOOLEAN ALGEBRA AND LOGIC GATES

Introduction, NAND and NOR operations, Exclusive –OR and Exclusive – NOR operations, Boolean Algebra Theorems and Properties , Standard SOP and POS form, Reduction of Boolean functions using Algebric method, K - map method (2,3,4 Variable).Variable entered Maps, Quine Mc Cluskey, Mixed Logic Combinational Circuits and multiple output function Basic Digital Circuits: NOT,AND, OR,NAND,NOR,EX-OR,EX-NOR Gates.

CHAPTER – 3

Boolean Algebra & Logic Gates

I] Logic Gates :—

- It is electronic ckt having one or more than one i/p's & only one o/p.
- It is logic ckt.
- It has classified into three categories

Logic Gates
- Basic Gates
  - 1] NOT Gate
  - 2] AND Gate
  - 3] OR Gate
- Universal Gates
  - 1] NAND Gate
  - 2] NOR Gate
- Derived Gates
  - EX-OR Gate
  - EX-NOR Gate

I] Basic Gates :-

1] NOT or Inverter [IC → 7404]

- This logic gate having one i/p & one o/p

symbol

Boolean eqn

$$\boxed{y = \overline{A}}$$

Truth Table

I/P	O/P
A	B
0	1
1	0

2] AND Gate [IC → 7408]

- It performs logical multiplication on its i/p's

Symbol

Boolean Expression

$$\boxed{y = A \cdot B}$$

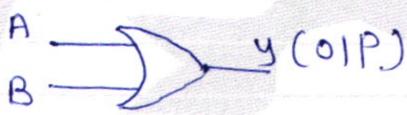
Truth Table

I/P		O/P
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

3] OR Gate

- It performs logical addition on its inputs.

Symbol



Truth Table

I/P		O/P
A	B	y
0	0	0
0	1	1
1	0	1
1	1	1

Boolean Expression

$$\boxed{y = A + B}$$

II] Universal Gates

- NAND & NOR gates are called universal gates because it is possible to implement any boolean expression with the help of NAND or only NOR gates.

1] NAND Gate :- (NOT - AND)

- It is the combination of AND + NOT

Symbol

Bubble represents inversion

Truth Table

I/P		O/P
A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

Boolean Expression

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

- Using NAND gate we can construct AND, OR & NOT gates.

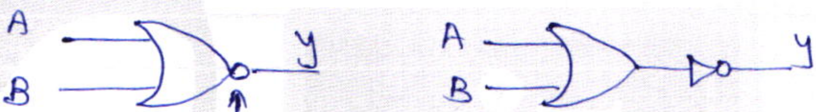
- Truth Table - It shows o/p is (0) low if & only if both i/p's are high.
- It works complete opposite of AND gate

II] NOR gate:

- It is the combination of an OR gate & NOT gate

$$\boxed{OR + NOT = NOR}$$

Symbol



Bubble represents the inversion

Truth Table

I/P		O/P
A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

- It Bolleen Expression

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

- Truth Table :- The O/P of a NOR gate is high (1) if and only if all its inputs are low (0).

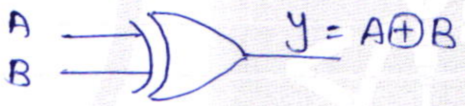
- It works complete opposite of OR gate.

III] Derived Gates (special type of gates)

1] EX-OR gate - (Exclusive-OR gate) (X-OR)

- It can have two or more than two i/p's & one o/p terminal.

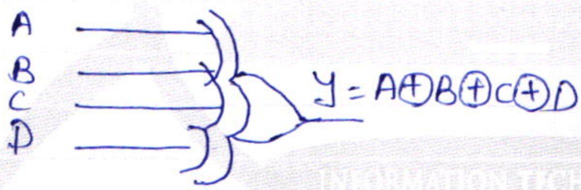
Symbol (2 i/p)



Truth Table

I/P		O/P $A \oplus B$
A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

symbol (4 i/p)



- If the i/p's are same then o/p will be 0 (zero)

  i.e. $\boxed{(A = B) = 0}$

- If i/p's are different then o/p will be 1 (one)

  $\boxed{A \neq B = 1}$

2] EX-NOR Gate (Exclusive-NOR)
(X-NOR)

- Ex-NOR gate is equivalent to an Ex-OR gate followed by a NOT gate

Symbol

Truth Table

I/P		O/P
A	B	y
0	0	1
0	1	0
1	0	0
1	1	1

- If the i/p's are same then o/p will be 1 (one)

i.e. $\boxed{(A=B)=1}$

- If the i/p's are different then o/p will be 0 (zero)

i.e. $\boxed{A \neq B = 0}$

* Boolean Laws

1] Commutative law

1) $A \cdot B = B \cdot A$
2] $A + B = B + A$

2] Associative law

1) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
2) $(A+B)+C = A+(B+C)$

3] Distributive law

$A \cdot (B+C) = AB + AC$

4] AND Law

1) $A \cdot 0 = 0$
2) $A \cdot 1 = A$
3) $A \cdot A = A$
4) $A \cdot \bar{A} = 0$

3

5.] OR Law

1) $A + 0 = A$
2) $A + 1 = 1$
3) $A + A = A$
4) $A + \bar{A} = 1$

6] Inversion Law

$\bar{\bar{A}} = A$

7) Other important law

1) $A + BC = (A+B)(A+C)$
2) $\bar{A} + AB = \bar{A} + B$
3) $\bar{A} + A\bar{B} = \bar{A} + \bar{B}$
4) $A + AB = A$
5) $A + \bar{A}B = A + B$

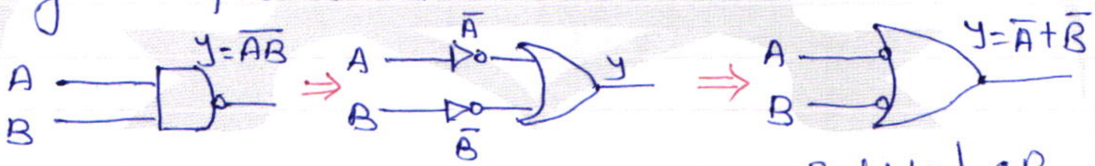
★ De-Morgan's Theorems :—

This theorems are extermely useful in Boolean algebra.

I] Theorem I -

$$\overline{AB} = \bar{A} + \bar{B} \quad (\text{NAND} = \text{Bubbled OR})$$

- It states that the, complement of the product is equal to the addition of complement.



NAND Bubbled OR

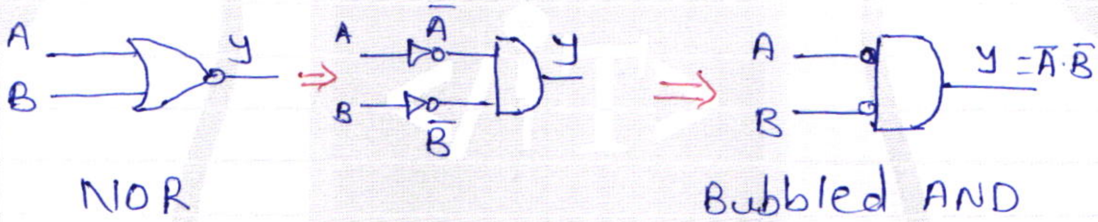
A	B	$\overline{AB}$	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

LHS $\boxed{\overline{AB} = \bar{A} + \bar{B}}$

II] Theorem II

$\overline{A+B} = \bar{A} \cdot \bar{B}$ (NOR = bubbled AND)

- LHS represents a NOR gate with i/p's A & B.
- RHS represents an AND gate with inverted inputs.



Truth Table

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

LHS $\overline{A+B}$ = $\bar{A}\cdot\bar{B}$ RHS

* Reducing the Boolean Expression

Prove the following expression

ex 1] To prove that $(A+B)(A+C) = A+BC$

→ LHS $(A+B)(A+C)$

$= AA + AC + BA + BC$ → Distributive law

But → $AA = A$ → AND law

$= \underline{A+AC} + BA + BC$

$= A(1+C) + BA + BC$ → $1+C = 1$ OR Law

$= A + BA + BC$

$= A(1+B) + BC$ → $1+B = 1$ OR Law

$= A + BC$

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

ex2] simplify $ABCD + A\bar{B}CD$

→ $Y = ABCD + A\bar{B}CD$

$= ACD(B+\bar{B})$ → $B+\bar{B} = 1$ OR Law

$\boxed{Y = ACD}$

ex 3] $A\bar{B} + \bar{A}B + AB + \bar{A}\bar{B}$

$\Rightarrow$ Given $y = \underbrace{A\bar{B}}_{1} + \underbrace{\bar{A}B + AB}_{2} + \underbrace{\bar{A}\bar{B}}_{1}$

$= \bar{B}(A + \bar{A}) + B(\bar{A} + A)$ $\rightarrow A + \bar{A} = 1$ OR Law

$= \bar{B} + B$ $\rightarrow$ $B + \bar{B} = 1$ OR Law

$\therefore \boxed{y = 1}$

4] $A\bar{B}C + \bar{A}BC + ABC$

$\rightarrow$ Given $y = \underbrace{A\bar{B}C}_{1} + \bar{A}BC + \underbrace{ABC}_{1}$

$= AC(\bar{B} + B) + \bar{A}BC$ $\rightarrow \bar{B} + B = 1$ OR Law

$= AC + \bar{A}BC$

$= C(A + \bar{A}B)$ $\rightarrow$ $A + \bar{A}B = A + B$ other law

$= C(A + B)$

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

5] Prove that $AB + ABC + A\bar{B} = A$

$\rightarrow$ LHS $AB + ABC + A\bar{B}$

$= AB(1 + C) + A\bar{B}$ $\rightarrow (1 + C) = 1$ OR Law

$= AB + A\bar{B}$

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

$= A$

$\therefore \boxed{AB + ABC + A\bar{B} = A}$

6] $xy + x(y+z) + y(y+z)$

→ given $= xy + x(y+z) + y(y+z)$

Applying distributive law
$A(B+C) = AB + AC$

$= xy + xy + xz + \underline{y \cdot y} + yz$

→ $yy = y$ AND law

$= \underline{xy + xy} + xz + y + yz$

$xy + xy = xy$

$= xy + xz + \underline{y + yz}$ → $y + yz = y$

$= \underline{xy} + xz + \underline{y}$ → $xy + y = y$

$\boxed{= y + xz}$

7] $xy + xyz + xy\bar{z} + \bar{x}yz$

→ Given
$= xy + xyz + xy\bar{z} + \bar{x}yz$
$= xy(1+z) + xy\bar{z} + \bar{x}yz$ → $(1+\bar{z} = 1)$
$= \underline{xy + xy\bar{z}} + \bar{x}yz$

$= xy + (1+\bar{z}) + \bar{x}yz$ → $1+\bar{z} = 1$

$= xy + \bar{x}yz$ → $x + \bar{x}z = x + z$
$= y(x+z)$
$\boxed{= xy + yz}$

8] $AB(\overline{CD}) + (\overline{B}CD) + (\overline{A}+\overline{C})(B+D)$

$\Rightarrow$ Given $\underline{AB(\overline{CD})} + (\overline{B}CD) + \underline{(\overline{A}+\overline{C})(B+D)}$

Using Demorgan's Theorem

$= AB(\overline{C}+\overline{D}) + (\overline{B}CD) + \overline{A}B + \overline{A}D + \overline{C}B + \overline{C}D$

$= AB\overline{C} + AB\overline{D} + \overline{B}CD + \overline{A}B + \overline{A}D + \overline{C}B + \overline{C}D$

Rearrange the term

$= \underline{AB\overline{C} + B\overline{C}} + \underline{AB\overline{D} + \overline{A}B} + \underline{\overline{B}CD + \overline{C}D} + \overline{A}D$

$= B\overline{C}(A+1) + B(\overline{A}+A\overline{D}) + D(\overline{B}C+\overline{C}) + \overline{A}D$

$\rightarrow A+1 = 1$

$\overline{A}+A\overline{D} = \overline{A}+\overline{D}$

$\overline{B}C+\overline{C} = \overline{B}+\overline{C}$

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

$= \underline{B\overline{C} + B\overline{A} + B\overline{D}} + \underline{D\overline{B} + D\overline{C} + \overline{A}D}$

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

Using Demorgan's Theorem

$= B(\overline{ACD}) + D(\overline{ABC})$

$$\boxed{B(\overline{ACD}) + D(\overline{ABC})}$$

9] $A[B+C(\overline{AC+AB})]$

→ Given $A[B+C(\overline{AC+AB})]$

Demorgan's Theorem

$= A[B+C(\bar{A}+\bar{C})\cdot(\bar{A}+\bar{B})]$

$= A[B+C(\bar{A}\bar{A}+\bar{A}\bar{B}+\bar{C}\bar{A}+\bar{C}\bar{B})]$

$\longrightarrow \bar{A}\bar{A}=\bar{A}$

$= A[B+C(\bar{A}+\bar{A}\bar{B}+\bar{C}\bar{A}+\bar{C}\bar{B})]$

$= A[B+C[\bar{A}(\underline{1+\bar{B}})+\bar{C}\bar{A}+\bar{C}\bar{B}]]$

$\therefore 1+\bar{B}=1$

$= A[B+C(\bar{A}+\bar{C}\bar{A}+\bar{C}\bar{B})]$

$= A[B+C(\bar{A}(\underline{1+\bar{C}})+\bar{C}\bar{B})]$

$\therefore 1+\bar{C}=1$

$= A[B+C(\bar{A}+\bar{C}\bar{B})]$

$= A[B+\bar{A}C+\bar{B}\underline{\bar{C}C}]$ $\therefore \bar{C}\cdot C=0$

$= A[B+\bar{A}C]$

$= AB+A\bar{A}C$ $\therefore A\cdot\bar{A}=0$

$\boxed{=AB}$

10] Prove & Draw the logic diagram

$\bar{A}BC + A\bar{B}C + AB\bar{C} + ABC = AB + BC + AC$

→ LHS $= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

Rearranging the terms

$= \bar{A}BC + ABC + A\bar{B}C + AB\bar{C}$

$= BC(\bar{A} + A) + A\bar{B}C + AB\bar{C}$ → $[A + \bar{A} = 1]$

$= BC + A\bar{B}C + AB\bar{C}$

$= C(B + A\bar{B}) + AB\bar{C}$ → $[A + \bar{A}B = A + B]$

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

$= AC + BC + AB\bar{C}$

Rearrange the terms

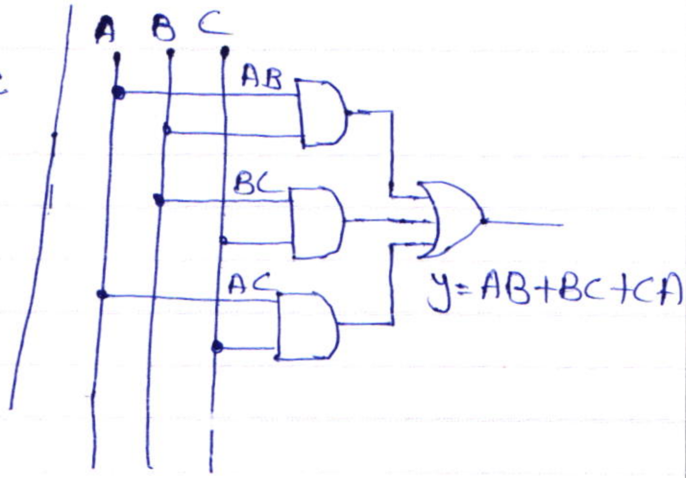
$= AC + AB\bar{C} + BC$

$= A(C + B\bar{C}) + BC$ → $[A + \bar{A}B = A + B]$

$= A(B + C) + BC$

$= AB + AC + BC$

LHS = RHS



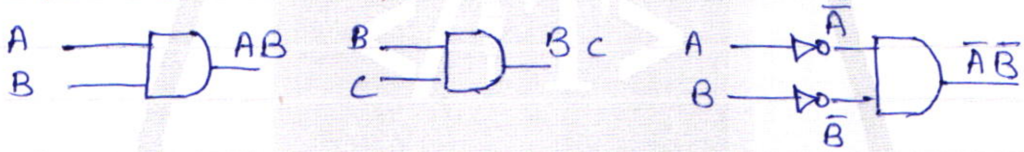
11] Draw the combinational circuit using the basic gates to obtain the following output.

$$Y = AB + BC + \bar{A}\bar{B}$$

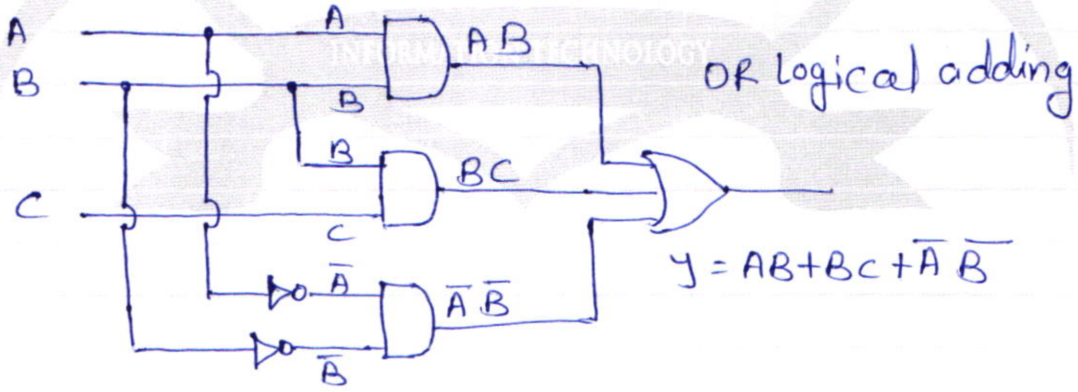
→ This circuit has three inputs

Step I : Implement the three individual terms

Term 1 Term 2 Term 3



Step II : Logically add the O/P of the three circuits



12] simplify using Boolean theorems & draw logic diagram for the following

$$y = \overline{AB}(B+C) + AB(\overline{B}+\overline{C})$$

→ $y = \overline{AB}(B+C) + AB(\overline{B}+\overline{C})$

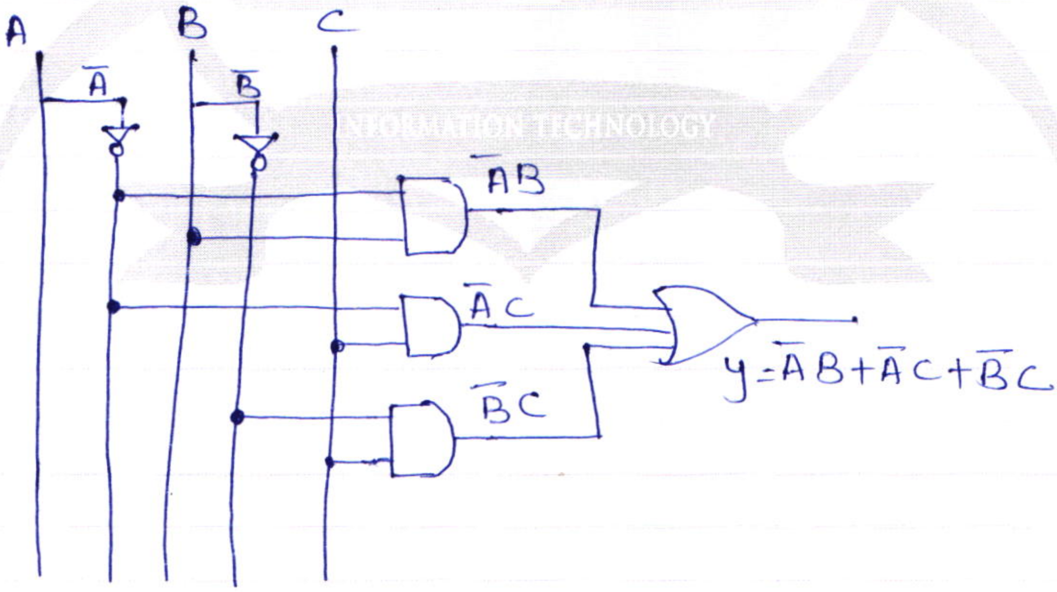
by using Demorgan's theorem → $\left[\begin{array}{l}\overline{AB} = \overline{A}+\overline{B} \\ \overline{B+C} = \overline{B}\cdot\overline{C}\end{array}\right]$

$= (\overline{A}+\overline{B})(B+C) + AB(\overline{B}\cdot\overline{C})$

$= \overline{A}B + \overline{A}C + \underline{\overline{B}B} + \overline{B}C + A\underline{B\overline{B}}\overline{C}$ → $[\overline{B}\cdot B = 0]$

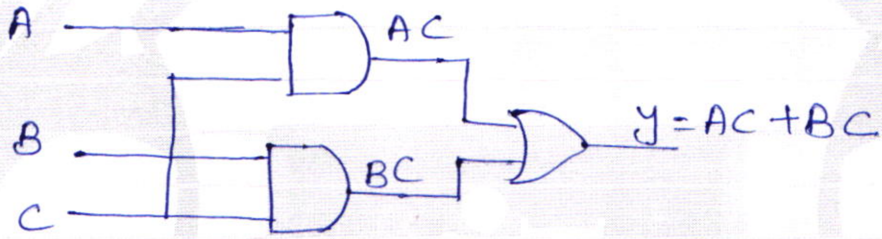
$= \overline{A}B + \overline{A}C + \overline{B}C$

$$\boxed{Ans = \overline{A}B + \overline{A}C + \overline{B}C}$$



* Standard presentation for logical function

- Logic Expression
$$y = AC + BC$$



Here, Input variables are A, B & C & that variables are called.
- literals.

$$\boxed{I/P \rightarrow A, B, C \rightarrow \text{Literals}}$$

1] Sum-of-Products (SOP) Form

2] Product-of-sums (POS) Form

I] sum-of-products Form (SOP)

$$y = \underbrace{AB + BC + AC}$$

sum

product terms

- This type of expression are called as SOP
- Not actual additions or multiplications
- They are OR & AND functions
- A, B & C are called literals or input of combinational circuit.

ex]

$$y = xy + yz + zx$$

$$y = PQR + QRS + RSP$$

$$A = xy + y\bar{z} + z\bar{x}$$

II] Product-of-sums (POS) form

$$y = (A+B) \cdot (B+C) \cdot (A+C)$$

products

sum terms

- This literals are ORed together to form the sum terms & the sum terms are ANDed to get the expression in the POS Form.

* Standard or Canonical SOP & POS Form

- Each terms contains all the i/p variables are called as standard or canonical

ex 1] Canonical SOP

$$y = \underbrace{ABC + A\bar{B}C + \bar{A}BC}$$

Each term contains all i/p variables & each individual term is called minterm

ex 2] Canonical POS

$$y = \underbrace{(A+B) \cdot (A+\bar{B})}$$

Each term contains all i/p variables & each individual term is called maxterm

* Non-standard Forms

- In this form each term may contain one, two or any number of literals.
- It means that, every literals does not appears in each term.
- It is not necessary that each term should contain all the literals.

1] Non-standard SOP Form

$$\boxed{y = AB + AB\bar{C} + \bar{A}BC}$$

2] Non-standard POS Form

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

* Conversion of Forms

I] Non-standard SOP to standard SOP Form

Steps

1] Find the missing literal for each term in the given non-standard SOP expression

2] Then AND this term with the term formed by ORing the missing literal & its compliment.

3] simplify the expression to get the standard SOP expression.

ex1] Convert the expression

$Y = AB + A\bar{C} + BC$ into the standard SOP Form

→ Given $Y = AB + A\bar{C} + BC$

In this eq^n or expression has 3-i/p variables A, B & C

Step I - Find the missing literal in each term

$$\boxed{Y = AB + A\bar{C} + BC}$$

missing $\boxed{C}$ $\boxed{B}$ missing $\boxed{A}$ missing step 2 : AND each term with (missing literal + Its complement)

$$y = \underbrace{AB}_{\text{original product term}} \cdot (C+\bar{C}) + \underbrace{A\bar{C}}_{} \cdot (B+\bar{B}) + \underbrace{BC}_{} \cdot (A+\bar{A})$$

original product term

[missing literal + its complement]

step 3 : simplify the expression to get the std. pos form.

$$y = AB\cdot(C+\bar{C}) + A\bar{C}\cdot(B+\bar{B}) + BC\cdot(A+\bar{A})$$

$$= \underbrace{ABC}_{1} + \underbrace{AB\bar{C} + AB\bar{C}}_{2} + A\bar{B}\bar{C} + \underbrace{ABC}_{1} + \bar{A}BC$$

$= \because$ $ABC + ABC = ABC$, $AB\bar{C} + AB\bar{C} = AB\bar{C}$

$$\boxed{y = ABC + AB\bar{C} + A\bar{B}\bar{C} + \bar{A}BC} \rightarrow \text{std SOP form}$$

Each term contains all the literals.

ex2] convert in standard SOP form
$A\bar{B}C + B\bar{D}$

→ Given $A\bar{B}C + B\bar{D}$

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

$$\underset{\boxed{D}}{\downarrow} \qquad \underset{\boxed{A \,\&\, C}}{\downarrow}$$

missing literals.

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

$$= A\bar{B}CD + A\bar{B}C\bar{D} + [AB\bar{D} + \bar{A}B\bar{D}(C+\bar{C})]$$

$$\boxed{= A\bar{B}CD + A\bar{B}C\bar{D} + ABC\bar{D} + AB\bar{C}\bar{D} + \bar{A}BC\bar{D} + \bar{A}B\bar{C}\bar{D}}$$

Standard sop Form

3] $y = A + B$

→ Given $y = A+B$

Missing literal

$$A + B$$

$$\downarrow \qquad \downarrow$$

missing $\boxed{B}$ $\boxed{A}$ missing

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

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

$\therefore AB + AB = AB$

$$\boxed{= AB + A\bar{B} + \bar{A}B}$$ → Std sop form

II] Conversion from Non-standard POS to standard POS form

Steps

1] Find the missing literal for each term

2] OR each term with (Missing literal & its complement)

3] Simplify the expression to get std. POS

ex 1] Convert the expression

$y = (A+B)(A+C)(B+\bar{C})$ into std POS form

→ Given $y = (A+B)(A+C)(B+\bar{C})$

Step I

$$\underbrace{(A+B)}_{\boxed{C}\text{ missing}} \quad \underbrace{(A+C)}_{\boxed{B}\text{ missing}} \quad \underbrace{(B+\bar{C})}_{\boxed{A}\text{ missing}}$$

Step II : $y = (A+B+C\bar{C})\cdot(A+C+B\cdot\bar{B})\cdot(B+\bar{C}+A\cdot\bar{A})$

missing literal term & its complement

step III simplify the expression

$y = (A+B+C\bar{C}) \cdot (A+C+B \cdot \bar{B})(B+\bar{C}+A \cdot \bar{A})$

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

$= \underbrace{(A+B+C)}_{1}\underbrace{(A+B+\bar{C})}_{2}\underbrace{(A+B+C)}_{1}(A+\bar{B}+C)$

$\underbrace{(A+B+\bar{C})}_{2}(\bar{A}+B+\bar{C})$

$=$ $\because (A+B+C)(A+B+C) = A+B+C$

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

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

This is std POS Form. Each term contains all literals

ex2] Convert the given expression in std POS Form

$f(ABC) = (A+B)(B+C)(A+C)$

$\Rightarrow$ Given expression

$F(ABC) = (A+B)(B+C)(A+C)$

$\downarrow$ C missing $\downarrow$ A missing $\downarrow$ B missing

* Canonical Forms of Boolean Expressions

1] Minterm
2] Maxterm

I] Minterm:

- It is used into the std SOP Form.

ex std SOP

$$y = \underbrace{ABC}_{} + \underbrace{A\bar{B}\bar{C}}_{} + \underbrace{\bar{A}B\bar{C}}_{}$$

Each individual term is Called minterm

II] Maxterm:

- It is used into the std POS Form

ex] std POS

$$\underbrace{(A+B)}_{}\underbrace{(B+A)}_{}$$

Each individual term is Called Maxterm.

Table For Minterm

I/P variables A	B	C	Minterms (SOP)
0	0	0	$\bar{A}\bar{B}\bar{C} = m0$
0	0	1	$\bar{A}\bar{B}C = m1$
0	1	0	$\bar{A}B\bar{C} = m2$
0	1	1	$\bar{A}BC = m3$
1	0	0	$A\bar{B}\bar{C} = m4$
1	0	1	$A\bar{B}C = m5$
1	1	0	$AB\bar{C} = m6$
1	1	1	$ABC = m7$

Table for Maxterm

I/P variables A	B	C	Maxterm (POS)
0	0	0	$A+B+C = M_1$
0	0	1	$A+B+\bar{C} = M_2$
0	1	0	$A+\bar{B}+C = M_3$
0	1	1	$A+\bar{B}+\bar{C} = M_4$
1	0	0	$\bar{A}+B+C = M_5$
1	0	1	$\bar{A}+B+\bar{C} = M_6$
1	1	0	$\bar{A}+\bar{B}+C = M_7$
1	1	1	$\bar{A}+\bar{B}+\bar{C} = M_8$

- Minterm always represents in small letter (m).

- Maxterm always represents in Capital letter (M).

* Representation of logical expressions using minterms & maxterms.

ex 1] $y = ABC + \bar{A}BC + A\bar{B}\bar{C}$ → (SOP)

- Given $\underbrace{ABC}_{m_7} + \underbrace{\bar{A}BC}_{m_3} + \underbrace{A\bar{B}\bar{C}}_{m_4}$ ← corresponding minterms

$y = m_3 + m_4 + m_7$

$\boxed{y = \Sigma m(3, 4, 7)}$ → Summation of Minterm representation

ex 2] $y = (A+\bar{B}+C)(A+B+C)(\bar{A}+\bar{B}+C)$ → (POS)

→ Given $y = \underbrace{(A+\bar{B}+C)}_{M_2}\ \underbrace{(A+B+C)}_{M_0}\ \underbrace{(\bar{A}+\bar{B}+C)}_{M_6}$ → corresponding Maxterms

$y = M_2 \cdot M_0 \cdot M_6$

$\boxed{y = \underbrace{\Pi M}_{\text{Product of maxterm}}(0, 2, 6)}$

* standard sop expression for given truth table

steps

1) From the given truth table, consider only those combinations of i/p's which produce an o/p $y=1$

2] Write the product terms for each combination

3] The product terms should be ORed to get the canonical sop Form.

ex 1]

A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

Ist step - consider those combination who produce o/p 1

IInd step - $y_1 = \bar{A}B$, $y_2 = A\bar{B}$] Boolean expressions in the product Forms

IIIrd step

$$y = y_1 + y_2 = \bar{A}B + A\bar{B}$$

$$y = m_1 + m_2$$

$$\boxed{y = \Sigma m(1,2)}$$

II] To write a standard pos expression for given truthtable

Steps

1] From the given truth table, consider only those o/p which produce 0 (low) $y=0$

2] Write the only maxterms for such combinations

3] AND these maxterms to obtain the expression in standard pos Form

ex] Given truth table

A	B	C	Y	
0	0	0	0	$\rightarrow A+B+C\ (M_0)$
0	0	1	1	
0	1	0	1	
0	1	1	0	$\rightarrow A+\bar{B}+\bar{C}\ (M_3)$
1	0	0	1	
1	0	1	0	$\rightarrow \bar{A}+B+\bar{C}\ (M_5)$
1	1	0	0	$\rightarrow \bar{A}+\bar{B}+C\ (M_6)$
1	1	1	1	

$$y = \underset{M_0}{\underbrace{(A+B+C)}} \cdot \underset{M_3}{\underbrace{(A+\bar{B}+\bar{C})}} \underset{M_5}{\underbrace{(\bar{A}+B+\bar{C})}} \underset{M_6}{\underbrace{(\bar{A}+\bar{B}+C)}}$$

$$y = M_0 \cdot M_3 \cdot M_5 \cdot M_6$$

$$\boxed{y = \pi M(0;3,5,6)}$$

Karnaugh Map [K-Map]

- It contains boxes or cells
- Number of boxes is equal to 2^n

1 i/P variable $2^1 = 2$

A	$\bar{A}$	A
	0	1

2 i/P variable $2^2 = 4$

A \ B	$\bar{B}$	B
$\bar{A}$	$\bar{A}\bar{B}$	$\bar{A}B$
A	$A\bar{B}$	$A.B$

3 i/P variables $2^3 = 8$

C \ AB	00	01	11	10
0				
1				

4 i/P variables $2^4 = 16$

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

- It is important that when we move from one cell to next cell along row or column only one variable in the product term changes.

$00 \rightarrow 01 \rightarrow 11 \rightarrow 10$

* Relation betn a truth-table & k-map

I] 2 - Variables

Truth Table

A	B	y
0	0	0
0	1	0
1	0	0
1	1	1

A \ B	0	1
0	0	0
1	0	1

Inside the cells we have to enter the values of o/p corresponding to different combinations of i/p's

II] 3 - Variables

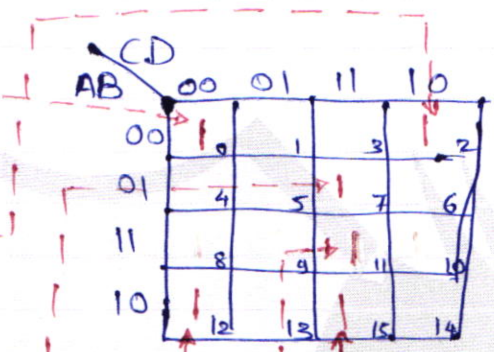
A	B	C	y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

BC \ A	0	1
00	1	0
01	0	1
11	1	1
10	0	0

Inside the boxes we have to enter the values of o/p corresponding to different combinations of A & B & C

III 4 - Variables k-map

A	B	C	D	y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



Inside the boxes we have to enter the values of O/P corresponding of to different combinations of A, B, C & D

* Representation of standard sop Form on k-map

- In the k-map by simply entering 1's in the cells of the k-map corres--ponding to each minterm present in the eqn.

- Remaining boxes are filled with zero (0)

ex] $y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C} + ABC$

- Given expression in the std. sop.
- Each term represents in minterm.
- we have to enter 1's in the cells corresponding to each minterm.

$2^3 = 8$ I/P variables are A, B & C

A \ BC	$\bar{B}\bar{C}$ 00	$\bar{B}C$ 01	BC 11	$B\bar{C}$ 10
$\bar{A}$ 0	1 (0)	1 (1)	(3)	(2)
A 1	1 (4)	(5)	1 (7)	1 (6)

$\bar{A}\bar{B}\bar{C}$ $A\bar{B}\bar{C}$ $\bar{A}\bar{B}C$ ABC $AB\bar{C}$

* Simplification of Boolean expression using K-map

Rules

- It is based on combining or grouping the terms in the adjacent cells (or boxes) of k-map.
- 2 cells of k-map are said to be adjacent if they differ in only one variable.
- Horizontal & vertical cells are adjacent
- cells on left & right or Top & bottom of cells are adjacent cells.
- but the cells connected diagonally are not adjacent ones.
- Group should be as large as possible
- Every must be in at least one group
- overlapping allowed
- wrap around allowed.

1) Pair (2) → It is a group of two adjacent 1's in a k-map
- It eliminate one variable in SOP form

2] Quad (4) → Group of four adjacent 1's on k-map is known as Quad
- It is eliminate two variable

3] octet :— Group of eight adjacent 1's on k-map is known as octet.
- It eliminate three variables.

* Minimization of SOP Expression

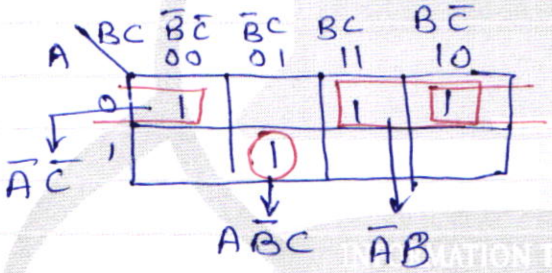
ex 1] A logical expression in std form

$y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C$

minimize it using k-map simplification

→ $y = \underbrace{\bar{A}\bar{B}\bar{C}}_{m_0} + \underbrace{\bar{A}B\bar{C}}_{m_2} + \underbrace{\bar{A}BC}_{m_3} + \underbrace{A\bar{B}C}_{m_5}$

$y = \Sigma m(0, 2, 3, 5)$

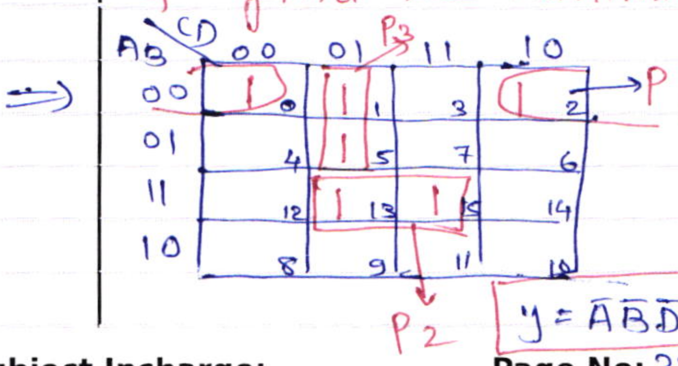


minimized expression in

$$\boxed{y = \bar{A}B + \bar{A}\bar{C} + A\bar{B}C}$$

ex 2] $y = \Sigma m(0, 1, 2, 5, 13, 15)$ Draw the k-map & find the minimized logical expression

⟹



Pair(P_1) → $\bar{A}\bar{B}\bar{D}$

Pair (P_2) → ABD

Pair (P_3) → $\bar{A}\bar{C}D$

$$\boxed{y = \bar{A}\bar{B}\bar{D} + ABD + \bar{A}\bar{C}D}$$

ex 3] $y = \Sigma m(1,3,4,5,7,9,11,13,15)$
this expression using minimum number of gates.

→ $y = \Sigma m(1,3,4,5,7,9,11,13,15)$

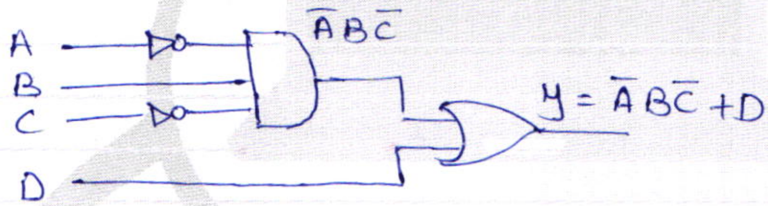
AB \ CD	00	01	11	10
00		1	1	
01	1	1	1	
11		1	1	
10		1	1	

P_1 (octet) $= D$

P_2 (pair) $= \bar{A}B\bar{C}$

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

Realization with minimum gates



ex 4] $y = \Sigma m(1,2,9,10,11,14,15)$
using minimum number of gates

⇒ $y = \Sigma m(1,2,9,10,11,14,15)$

AB \ CD	00	01	11	10
00		1		1
01				
11			1	1
10		1	1	1

→ Group 2 $(\bar{B}C\bar{D})$

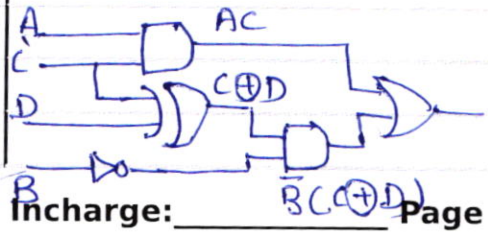
→ Group 3 (AC)

Group 1 $(\bar{B}\bar{C}D)$

$y = \bar{B}\bar{C}D + \bar{B}C\bar{D} + AC$

$= \bar{B}(\underbrace{\bar{C}D + C\bar{D}}_{\text{EX-OR gate}}) + AC$

$y = \bar{B}(C \oplus D) + AC$



Don't Care condition :-

- In such condition the designer has Flexible to assume 0 or 1 as O/P for these combination. This condition is known as don't care condⁿ.
- It is represented by (X) cross mark in the cell.
- (X) mark in cell may be assumed to be 1 or 0 depending upon which one leads to a simpler expression.

ex In the terms of minterms & don't care condⁿ.

$$F(A,B,C,D) = \Sigma m(1,3,7,11,15) + d(0,2,5)$$

AB \ CD	00	01	11	10
00	X	1	1	X
01		X	1	
11			1	
10			1	

$P_1 \rightarrow \bar{A}\bar{B}$

$P_2 \rightarrow CD$

$$y = \bar{A}\bar{B} + CD$$

note :- Every don't care marke need not be considered while grouping.

ex 2] Using k-map simplify

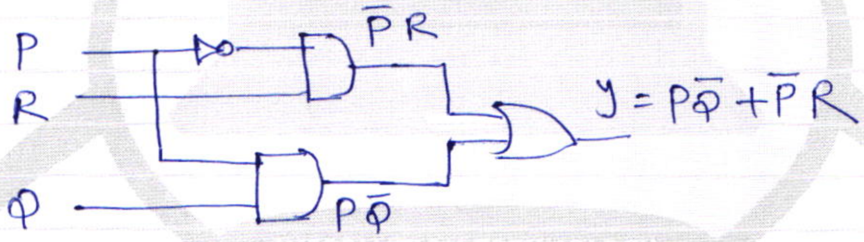
$$F(P,Q,R,S) = \Sigma m(2,3,6,7,8,9,10,11) + d(12)$$

PQ \ RS	00	01	11	10
00			1	1
01			1	1
11	X			
10	1	1	1	1

→ $\bar{P}R$

$P\bar{Q}$

$$y = P\bar{Q} + \bar{P}R$$



Relization of gates.

* NAND-NAND Implimentation

steps

1) simplify the given logical expression & convert it in the SOP form

2] Draw the AND-OR-NOT realization

3) Replace every AND gate by a NAND,
every OR " " bubbled OR
" NOT " " NAND Inverter

4) Draw the ckt using only NAND gates.

ex] Implement the following expression using NAND-NAND logic.

$y = \Sigma m(0, 1, 5)$

→

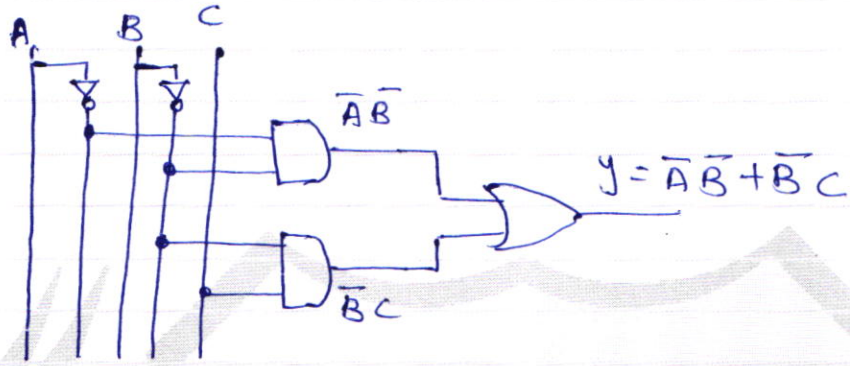
A \ BC	00	01	11	10
0	1	1		
1		1		

⇒ k-map for given expression

$\bar{A}\bar{B}$ $\bar{B}C$

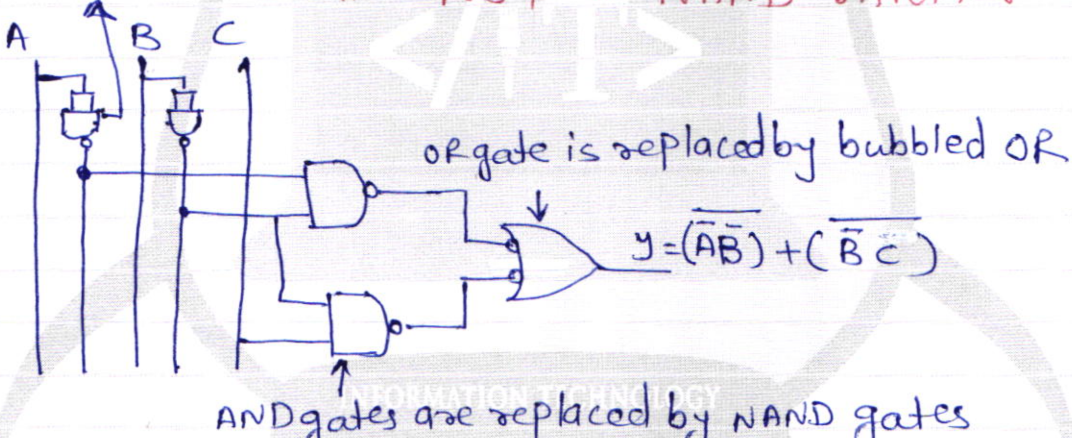
$$\boxed{y = \bar{A}\bar{B} + \bar{B}C}$$

2nd step → Implement using AND-OR-NOT logic gates.

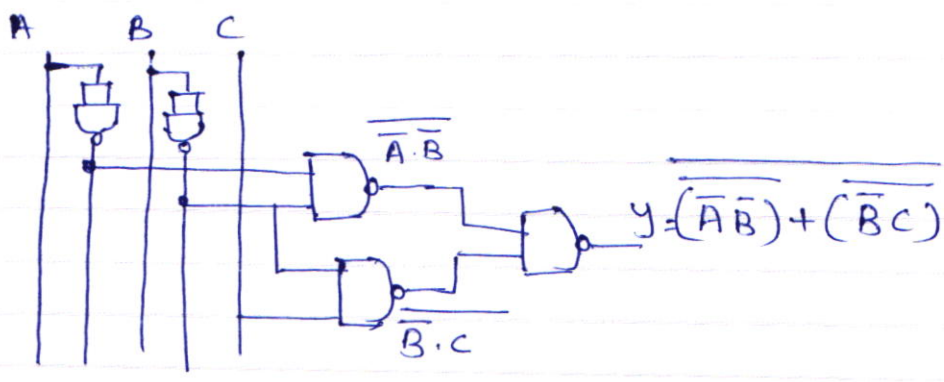


3rd step Replace AND by NAND
" OR by bubbled OR
" NOT " NAND Inverter

NAND Inverters



4th step Draw the ckt using only NAND gates



* NAND-NAND Implimentation

Steps:

1) simplify the given logic expression & convert in the SOP Form

2) Draw the AND-OR-NOT realization

3.) Replace 1) every AND gate by NAND
   2) " OR " " bubbled OR
   3) " NOT " " NAND Inverter

4] Draw the ckt using only NAND gates.

ex] $F(A,B,C,D) = (0,1,2,3,4,7,8,11,12,15)$ using NAND-NAND Implementation

⟹ I^{st} step → k-map

AB \ CD

0 $_{0}$	0 $_{1}$	0 $_{3}$	0 $_{2}$
0 $_{4}$	$_{5}$	0 $_{7}$	$_{6}$
0 $_{12}$	$_{13}$	0 $_{15}$	$_{14}$
0 $_{8}$	$_{9}$	0 $_{11}$	$_{10}$

$A+B$

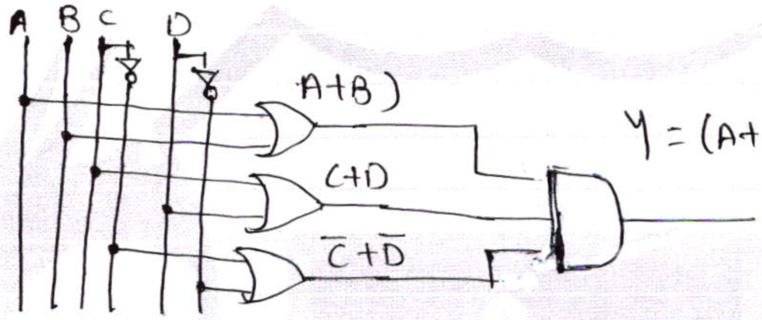
$\bar{C}+\bar{D}$

$C+D$

$$\boxed{F(A,B,C,D) = (C+D)(\bar{C}+\bar{D})(A+B)}$$

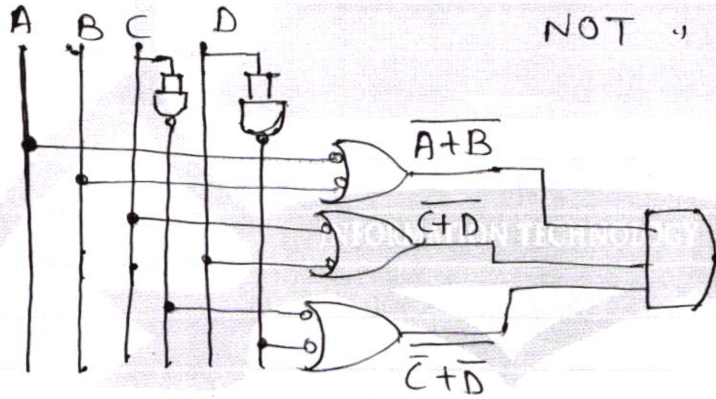
IInd step Make the ckt using AND - OR - NOT

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



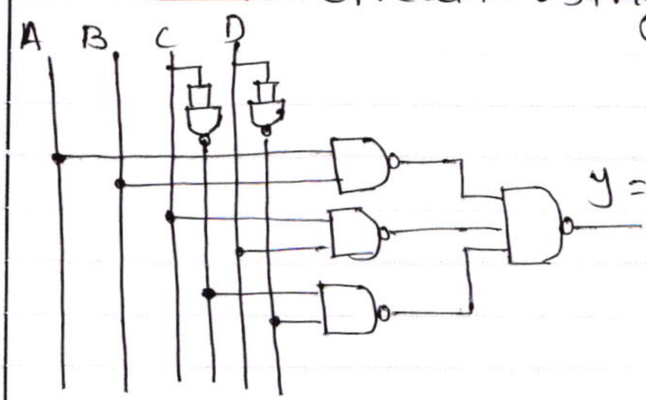
$Y = (A+B)(C+D)(\bar{C}+\bar{D})$

IIIrd step Replace AND by NAND
OR " bubbled OR
NOT " NAND Inverter



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

IVth step Circuit Using only NAND gates



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

* NOR - NOR Implementation :-

Steps

1] Simplify the given logical expression & convert it into product of sum (POS) form

2] Draw the ckt using AND-OR-NOT gates

3] Replace every OR gate by NOR
" AND " " bubbled AND
" NOT " " NOR inverter
Then replace every bubbled AND gate by a NOR gate.

4] Draw the final ckt using only the NOR gates.

ex] Implement the following POS expression using NOR-NOR logic

$F = \pi M(0,1,2,4,6)$

⇒ Step I: Simplify the given fu^n using K-map

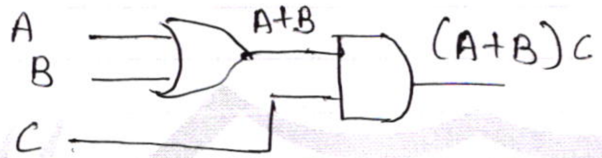
A \ BC	00	01	11	10
0	0	0		0
1	0			0

→ C

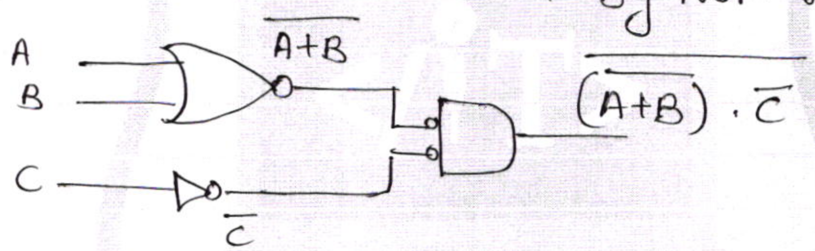
A+B

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

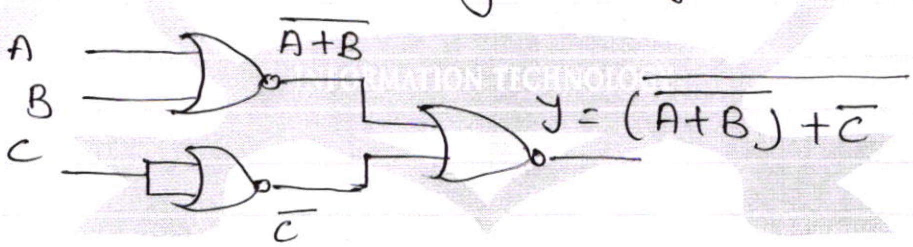
step II Draw ckt by using AND-OR-NOT gates.



step III Replace OR by NOR
AND by bubbled AND
NOT by NOR Inverter



step IV use only NOR gates



* Quine Mc-cluskey

1] It is minimization Technique
2] It is Tabular Method
3] It should have capability of large no. of variables
4] It should ensure minimized expression
5] It should be suitable for computer soln.

Definations

1] Prime Implicant [PI]

It is a group of minterms which can not be combined with any other minterms or groups.

2] Essential Prime Implicant [EPI]

It is prime implicant. It contains at least one minterm which is not contained in any other prime implicant.

- It is based on the concept of Prime Implicants.

★ Steps

1] Arrange the given minterms in an ascending order & make the groups based on the number of ones present in their binary presentations, so, there will be at most n+1 groups. If there are 'n' boolean variables in a boolean funⁿ or n bits in the binary equivalent of minterms

2] Compare the minterms present in successive groups. If there is change in only one bit position, then take the pair of those two minterms & place '_' (underscore) this symbol in the differed bit position & keep the remaining bits as it is

3] Repeat step 2 with newly formed terms till we get all prime Implicants.
Combine the minterm pairs into groups of Four (Quad). & remove the repeated rows

4] Formulate the prime implicant table. It consists of set of rows & columns prime implicants can be placed in rowwise & minterms can be placed in columnwise. Place '1' in the cells corresponding to the minterms that are covered in each prime implicant.

5] Find the Essential prime Implicants [EPI] by observing each column. If minterm is covered only by one prime implicant [PI], then it is essential prime implicants [EPI] will be part of the simplified Boolean fun.

ex 1] Minimize the Logic Fu^n using Q-M method

$F(A, B, C, D) = \Sigma m(1, 3, 7, 9, 10, 11, 13, 15)$

→ step I Group the minterms according to number of 1's

	Group	Minterm	A	B	C	D
one 1's	1	1	0	0	0	1 ✓
Two 1's	2	3	0	0	1	1 ✓
		9	1	0	0	1 ✓
		10	1	0	1	0 ✓
Three 1's	3	7	0	1	1	1 ✓
		11	1	0	1	1 ✓
		13	1	1	0	1 ✓
four 1's	4	15	1	1	1	1 ✓

(Columns A, B, C, D under heading: Binary Representation)

step II: Group the minterms to form the pairs.

Group	Minterms Pair	A	B	C	D	[PI] prime Implicant
1	1 - 3	0	0	–	1	✓
	1 - 9	–	0	0	1	✓
2	3 - 7	0	–	1	1	✓
	3 - 11	–	0	1	1	✓
	9 - 11	1	0	–	1	✓
	9 - 13	1	–	0	1	✓
	10 - 11	1	0	1	–	→ $A\bar{B}C$
3	7 - 15	–	1	1	1	✓
	11 - 15	1	–	1	1	✓
	13 - 15	1	1	–	1	✓

step III Group the minterm to form the Quad

Group	Minterms Quad	A	B	C	D	prime Implicant
1	1 - 3 - 9 - 11	–	0	–	1	$\bar{B}D$
	1 - 9 - 3 - 11	–	0	–	1	
2	3 - 7 - 11 - 15	–	–	1	1	CD
	3 - 11 - 7 - 15	–	–	1	1	
3	9 - 11 - 13 - 15	1	–	–	1	AD
	9 - 13 - 11 - 15	1	–	–	1	

now, remove the repeated rows

Group	Minterms	A B C D	PI
1	1-3-9-11	– 0 – 1	$\bar{B}D$
2	3-7-11-15	– – 1 1	CD
3	9-11-13-15	1 – – 1	AD

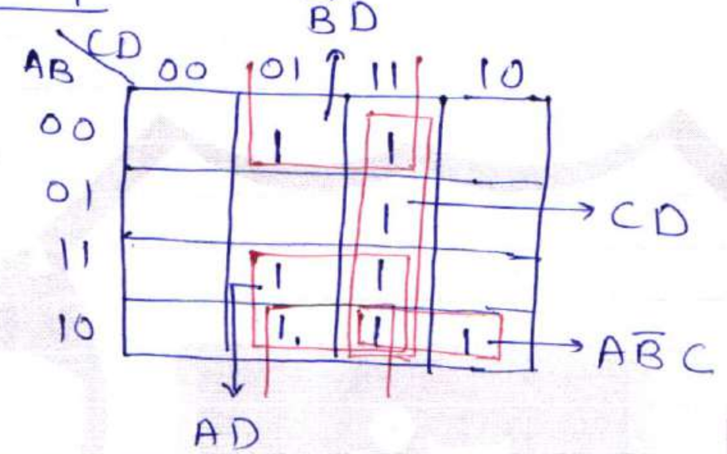
Step IV: prepare the table of prime implicants:

Prime Implicants	Decimal no's	1	3	7	9	10	11	13	15
$A\bar{B}C$	10 - 11					1	1		
$\bar{B}D$	1-3-9-11	1	1		1		1		
CD	3-7-11-15		1	1			1		1
AD	9-11-13-15				1		1	1	1

Essential prime Implicants [EPI]. They cover minterms 1, 3, 7, 9, 10, 11, 13, 15.
encircled numbers are EPI.

$F(A,B,C,D) = A\bar{B}C + \bar{B}D + CD + AD$

Now, we can crosscheck by using K-map



$$\boxed{F(A,B,C,D) = A\bar{B}C + AD + CD + \bar{B}D}$$

ex 2] $F(W,X,Y,Z) = \Sigma m(2,6,8,9,10,11,14,15)$
using Q-M tabular Method

$\Rightarrow$ Step I: Arrange all minterms according to no. of 1's contained & form the groups have no ones, one 1's, two 1's & so on

Group	Minterms	W	X	Y	Z	
1	2	0	0	1	0	one 1's
	8	1	0	0	0	
2	6	0	1	1	0	two 1's
	9	1	0	0	1	
	10	1	0	1	0	
3	11	1	0	1	1	Three 1's
	14	1	1	1	0	
4	15	1	1	1	1	four 1's

(Binary Represⁿ spans W X Y Z)

Step II Combine the minterms into a group of two

Group	Minterm matched Pairs	W	X	Y	Z
1	2 - 6	0	–	1	0 ✓
	2 - 10	–	0	1	0 ✓
	8 - 9	1	0	0	– ✓
	8 - 10	1	0	–	0 ✓
2	6 - 14	–	1	1	0 ✓
	9 - 11	1	0	–	1 ✓
	10 - 11	1	0	1	– ✓
	10 - 14	1	–	1	0 ✓
3	11 - 15	1	–	1	1 ✓
	14 - 15	1	1	1	– ✓

Step III: combination the min term pairs into groups of four (Quad)

Group	Minterm (Group of four)	W	X	Y	Z	PI
1	2-6-10-14	–	–	1	0	$\rightarrow Y\bar{Z}$
	2-10-6-14	–	–	1	0	
	8-9-10-11	1	0	–	–	$\rightarrow W\bar{X}$
	8-10-9-11	1	0	–	–	
2	10-11-14-15	1	–	1	–	$\rightarrow WY$
	10-14-11-15	1	–	1	–	

now, remove the repeated rows

Group	minterms	w	x	y	z	PI
1	2-6-10-14	-	-	1	0	$y\bar{z}$
	8-9-10-11	1	0	-	-	$w\bar{x}$
2	10-11-14-15	1	-	1	-	wy

Prime Implicants are → $y\bar{z}$, $w\bar{x}$, wy

step IV :- The Prime Implicant table is shown below

Prime Implicant	Decimal no's	2	6	8	9	10	11	14	15
$y\bar{z}$	2-6-10-14	1	1			1		1	
$w\bar{x}$	8-9-10-11			1	1	1	1		
wy	10-11-14-15					1	1	1	1

In this example, we got three PI & all the three are essential.

∴ the simplified boolean fu^n is

$$\boxed{F(w,x,y,z) = y\bar{z} + w\bar{x} + wy}$$

Cross check by using k-map

$F(w,x,y,z) = (2,6,8,9,10,11,14,15)$

in SOP form

$wx \backslash yz$	$\bar{y}\bar{z}$ 00	$\bar{y}z$ 01	yz 11	$y\bar{z}$ 10
$\bar{w}\bar{x}$ 00	0	1	3	1 (2)
$\bar{w}x$ 01	4	5	7	1 (6)
wx 11	12	13	1 (15)	1 (14)
$w\bar{x}$ 10	1 (8)	1 (9)	1 (11)	1 (10)

$\rightarrow y\bar{z}$

$\rightarrow wy$

$w\bar{x}$

$$\boxed{F(w,x,y,z) = w\bar{x} + y\bar{z} + wy}$$

ex3] Minimal expression using Q-M method

$F(A,B,C,D) = \sum m(1,5,6,12,13,14) + d(2,4)$

$\Rightarrow$ Step I: Group the minterms according to no. of 1's

Group	Minterm	A	B	C	D	
		Binary Representation				
1	1	0	0	0	1	One 1's
	2* don't care	0	0	1	0	
	4*	0	1	0	0	
2	5	0	1	0	1	Two 1's
	6	0	1	1	0	
	12	1	1	0	0	
2	13	1	1	0	1	Three 1's
	14	1	1	1	0	

Step II: Group the minterms to form Pairs

Group	Minterms Pairs	A	B	C	D	Prime Implicant
1	1 - 5	0	–	0	1	$\bar{A}\bar{C}D$
	2 - 6	0	–	1	0	$\bar{A}C\bar{D}$
	4 - 5	0	1	0	–	✓
	4 - 6	0	1	–	0	✓
	4 - 12	–	1	0	0	✓
2	5 - 13	–	1	0	1	✓
	6 - 14	–	1	1	0	✓
	12 - 13	1	1	0	–	✓
	12 - 14	1	1	–	0	✓

Step III: Group the minterms to form Quad (Four)

Group	Minterm Four	A	B	C	D	Prime Implicant
1	4-5-12-13	–	1	0	–	$B\bar{C}$
	4-6-12-14	–	1	–	0	$B\bar{D}$
	4-12-5-13	–	1	0	–	$B\bar{C}$
	4-12-6-14		1		0	$B\bar{D}$

Step IV: Prime Implicants

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

$\therefore B\bar{C} + B\bar{C} = B\bar{C}$

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

$$\boxed{F(A,B,C,D) = \bar{A}CD + \bar{A}C\bar{D} + B\bar{C} + B\bar{D}}$$

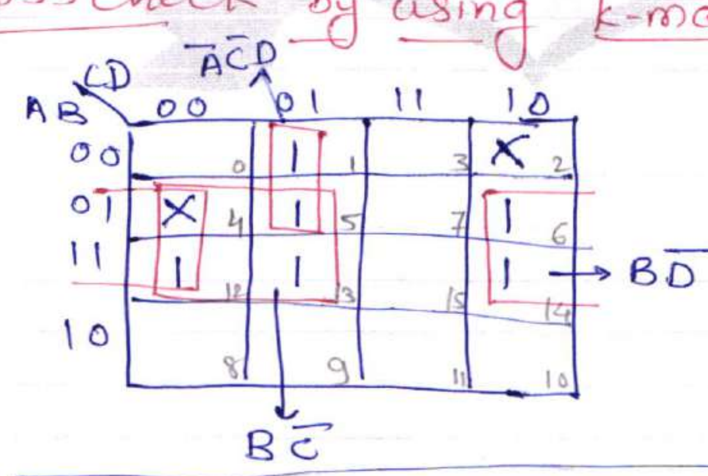
step V : Prepare the table for Prime Implicants

Prime Implicand	minterms	Given Minterms 1	5	6	12	13	14
$\bar{A}\bar{C}D$	1 − 5	1	1				
$\bar{A}C\bar{D}$	2^* − 6			1			
$B\bar{C}$	4^*−5−12−13		1		1	1	
$B\bar{D}$	4^*−6−12−14			1	1		1

$\boxed{F(A,B,C,D) = \bar{A}\bar{C}D + B\bar{C} + B\bar{D}} \rightarrow$ EPI

- Don't Care terms are not listed in this table.
- Encircled the 1's represents the essential prime Implicants [EPI]

Cross check by using k-map



$$\boxed{F(A,B,C,D) = \bar{A}\bar{C}D + B\bar{C} + B\bar{D}}$$