# I– KARNAUGH MAP. ①

* If a Boolean Algebraic Expression (Function) is complex ⇒ Then
* The corresponding digital circuit that implement the Function will also be complex.

* Although a truth table for a function is unique ⇒ but
* When that Function is expressed ALGEBRAICALLY ⇒ it may appear in many different forms.

* The K-Map method provides a ⇒
  * Simple straightforward procedure to minimize Boolean Function.
  * it is possible to simplify a logical expression by visual inspection by K-map.

## FORMATION OF K-MAP

Simplified function $F = x + y$.

① There are various techniques :

### I. From Truth Table

Two-variable Function

| $x$ | $y$ | $F$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Plot this function in K-Map $\boxed{F = xy + x\bar{y} + y}$

* The boxes are so arranged that they are ⇒
  * not only physically but also
  * Logically adjacent for Minterms or Maxterms.

Two variable. K-map (2nd order)

| $x \backslash y$ | $\bar{y}$ | $y$ |
|---|---|---|
| $\bar{x}$ | $\bar{x}\bar{y}(0)$ 0 | $\bar{x}y(1)$ 1 |
| $x$ | $x\bar{y}(2)$ 1 | $xy(3)$ 1 |

## II (a). Sum-of-Product form of the function: (SOP):

$$F = xy + x\bar{y} + y.$$

* convert to sum-of-minterms: (Canonical SOP or Standard SOP)

$$F = xy + x\bar{y} + y(x+\bar{x})$$

$$= xy + x\bar{y} + xy + \bar{x}y.$$

$$= \bar{x}y + x\bar{y} + xy \quad \text{(eliminate the repeated terms)}$$

* Now enter the minterms into the corresponding area of the K-map.

| $x \backslash y$ | $\bar{y}$ | $y$ |
|---|---|---|
| $\bar{x}$ | | 1 |
| $x$ | 1 | 1 |

## II (b). Product-of-sum form of the function (POS):-

$$F = (\bar{x}+y)(x+z)(y+z).$$

* convert to product-of-Maxterms:-

$$\bar{x}+y = \bar{x}+y+z\bar{z} = (\bar{x}+y+z)(\bar{x}+y+\bar{z})$$

Similarly we get,

$$F = (x+y+z)(x+\bar{y}+z)(\bar{x}+y+z)(\bar{x}+y+\bar{z}) \quad \text{[By eliminating the repeated terms]}$$

$$= M_0 M_2 M_4 M_5 = \Pi(0,2,4,5)$$

## III (a) Supplied as Sum-of-minterms:

$$F(x,y) = \bar{x}y + x\bar{y} + xy = \sum m(1, 2, 3)$$

## III (b) Supplied as Product-of-Maxterms:-

$F(x,y,z) = \Pi(0,2,4,5) \rightarrow$ insert zeros (0) to corresponding boxes.

## IV. By Inspection of Function:-

$$F = xy + x\bar{y} + y.$$

$$F = (\bar{x}+y)(x+z)(y+z). \longrightarrow$$

| $x \backslash y$ | $\bar{y}$ | $y$ |
|---|---|---|
| $\bar{x}$ | $\bar{x}\bar{y}$ 0 | $\bar{x}y$ 1 |
| $x$ | $x\bar{y}$ 1 | $xy$ 1 |

$\rightarrow y$

$\rightarrow x$

$$F = x + y$$

# S-O-P Form

## Monad

(1) $F(x,y) = \sum m(0,3)$

| $X \backslash Y$ | $\bar{Y}$ 0 | $Y$ 1 |
|---|---|---|
| $\bar{X}$ 0 | 1 | |
| $X$ 1 | | 1 |

$\therefore F(x,y) = \bar{X}\bar{Y} + XY$

## Diad

(2) $F(X,Y) = \sum m(0,1,2)$
$= m_0\ m_1\ m_2$

| $X \backslash Y$ | 0 | 1 |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 1 | 0 |

$P_1 = \bar{Y}$ $\quad P_2 = \bar{X}$

$\therefore F = P_1 + P_2$
$= \bar{X} + \bar{Y}$

## Quad

3) $F(X,Y) = \sum m(0,1,2,3)$

| $X \backslash Y$ | 0 | 1 |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 1 | 1 |

$\rightarrow F(X,Y) = 1$

For a 2-variable K-Map $\Rightarrow$

* a Quad produces a 1

* Because the total no. of variables in the function are = 2 ($N=2$) and The variables which are changing within the Quad is also two (2).

* Similarly, for 3-variable K-Map $\Rightarrow$ a octet produces a 1.

(4) $F(X,Y,Z) = \sum m(2,3,6,7)$

| $X \backslash YZ$ | $\bar{Y}\bar{Z}$ | $\bar{Y}Z$ | $YZ$ | $Y\bar{Z}$ |
|---|---|---|---|---|
| $\bar{X}$ 0 | | | 1 | 1 |
| $X$ 1 | | | 1 | 1 |

$\rightarrow F(X,Y,Z) = Y$

# Three variable K-Map. (Third order K-map).

Simplified Function $F=\bar{z}+x\bar{y}$

## I. Truth table:

$F=\bar{z}x+\bar{z}\bar{x}+x\bar{y}z+x\bar{y}\bar{z}$

→ plot this function in the K-map.

| x | y | z | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

→ plot the Truth table in the K-Map.

| $x \backslash yz$ | $\bar{y}\bar{z}$ | $\bar{y}z$ | $yz$ | $y\bar{z}$ |
|---|---|---|---|---|
| $\bar{x}$ | 0: 1 | 1: | 3: | 2: 1 |
| $x$ | 4: 1 | 5: 1 | 7: | 6: 1 |

⇒ $\bar{z}$

$x\bar{y}$

In general, for a N-variable map if $2^M$ (where, $M=1,2,\ldots N$) logically adjacent boxes are combined then the combination gives rise to a term with $(N-M)$ number of literals.

## II. Sum-of-Product form of the function (SOP):-

$$F=\bar{z}x+\bar{z}\bar{x}+x\bar{y}z+x\bar{y}\bar{z}$$

* convert to Sum-of-Minterms. (Canonical SOP or Standard SOP)

$$F=\bar{z}x(y+\bar{y})+\bar{z}\bar{x}(y+\bar{y})+x\bar{y}z+x\bar{y}\bar{z}$$

$$=\underset{6}{xy\bar{z}}+\underset{4}{x\bar{y}\bar{z}}+\underset{2}{\bar{x}y\bar{z}}+\underset{0}{\bar{x}\bar{y}\bar{z}}+\underset{5}{x\bar{y}z}+\underset{4}{x\bar{y}\bar{z}}$$

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

$$=\Sigma\ 0,2,4,5,6 \quad = \text{Sum-of-Minterms}$$

III. Supplied as Sum-of-Minterms:

$$F(x,y,z) = \sum_m 0, 2, 4, 5, 6$$

IV. By inspection of function:-

$$F = \bar{z}x + \bar{z}\bar{x} + x\bar{y}z + x\bar{y}\bar{z}$$

## FOUR VARIABLE K-MAP (Fourth order K-Map)

Simplified function:

$$F = \bar{y} + \bar{w}\bar{z} + x\bar{z}$$

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

$\bar{w}\bar{z}$

$x\bar{z}$

$\bar{y}$

I. Truth Table:-

| w | x | y | z | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

Given Function:

$$F = w\bar{y} + \bar{w}\bar{y} + \bar{w}x\bar{z} + \bar{w}\bar{x}\bar{z} + x\bar{z}$$

$$= wx\bar{y} + w\bar{x}\bar{y} + \bar{w}\bar{y} + \bar{w}xy\bar{z} + x\bar{z} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}\bar{z}$$

Simplified function

$$= F = \bar{y} + \bar{w}\bar{z} + x\bar{z}$$

## II. Sum-of-Product form of the function (SOP):-

$$F = wx\bar{y} + w\bar{x}\bar{y} + \bar{w}\bar{y} + \bar{w}xy\bar{z} + x\bar{z} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}\bar{z}$$

* Convert to sum-of-Minterms (Sum-of-Standard Product, or (Canonical SOP)

$$F = wx\bar{y}(z+\bar{z}) + w\bar{x}\bar{y}(z+\bar{z}) + \bar{w}\bar{y}(x+\bar{x}) + \bar{w}xy\bar{z}$$
$$+ x\bar{z}(y+\bar{y}) + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}\bar{z}(y+\bar{y}).$$

$$= wx\bar{y}z + wx\bar{y}\bar{z} + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y} + \bar{w}\bar{x}\bar{y}$$
$$+ \bar{w}xy\bar{z} + xy\bar{z} + \underline{x\bar{y}\bar{z}} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}\bar{z}$$

$$= wx\bar{y}z + wx\bar{y}\bar{z} + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}(z+\bar{z}).$$
$$+ \bar{w}\bar{x}\bar{y}(z+\bar{z}) + \bar{w}xy\bar{z} + xy\bar{z}(w+\bar{w}) + \underline{x\bar{y}\bar{z}(w+\bar{w})}$$
$$+ \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}\bar{y}\bar{z}$$

$$= \underset{13}{wx\bar{y}z} + \underset{12}{wx\bar{y}\bar{z}} + \underset{9}{w\bar{x}\bar{y}z} + \underset{8}{w\bar{x}\bar{y}\bar{z}} + \underset{5}{\bar{w}x\bar{y}z} + \underset{4}{\bar{w}x\bar{y}\bar{z}}$$
$$+ \bar{w}\bar{x}\bar{y}z + \bar{w}\bar{x}\bar{y}\bar{z} + \underset{6}{\bar{w}xy\bar{z}} + \underset{14}{wxy\bar{z}} + \underset{6}{\cancel{\bar{w}xy\bar{z}}} + \underset{12}{\cancel{wx\bar{y}\bar{z}}} + \underset{4}{\cancel{\bar{w}x\bar{y}\bar{z}}}$$
$$+ \cancel{\bar{w}x\bar{y}\bar{z}} + \bar{w}\bar{x}y\bar{z}\ (2) + \cancel{\bar{w}\bar{x}\bar{y}\bar{z}}$$

$$= \underset{0}{\bar{w}\bar{x}\bar{y}\bar{z}} + \underset{1}{\bar{w}\bar{x}\bar{y}z} + \underset{4}{\bar{w}x\bar{y}\bar{z}} + \underset{5}{\bar{w}x\bar{y}z} + \underset{6}{\bar{w}xy\bar{z}} + \underset{8}{w\bar{x}\bar{y}\bar{z}}$$
$$+ \underset{9}{w\bar{x}\bar{y}z} + \underset{12}{wx\bar{y}\bar{z}} + \underset{13}{wx\bar{y}z} + \underset{14}{wxy\bar{z}} + \underset{2}{\bar{w}\bar{x}y\bar{z}}$$

$$= \sum m(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14).$$

$\Rightarrow$ Sum-of-Minterms
or
Standard S.O.P
or
Canonical SOP.

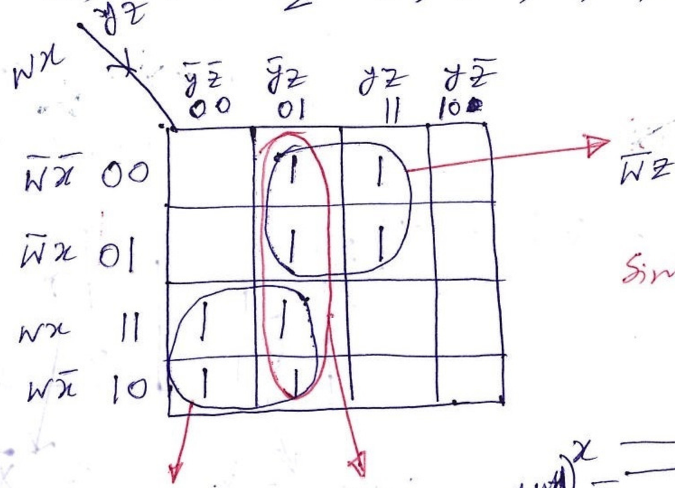
## III. Supplied as Standard SOP :

$$F(w, x, y, z) = \sum_m 0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14$$

## IV. By inspection of function (SOP):

$$F = wx\bar{y} + w\bar{x}\bar{y} + \bar{w}\bar{y} + \bar{w}xy\bar{z} + x\bar{z} + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}\bar{z}.$$

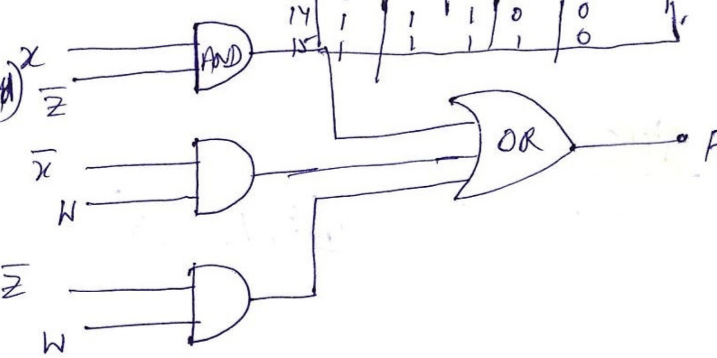
(5) $F = \sum m(1, 3, 5, 7, 8, 9, 12, 13)$




Simplified function:
$F = W\bar{Y} + \bar{W}Z$

| | W | X | Y | Z | F |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 | 0 |
| 12 | 1 | 1 | 0 | 0 | 1 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 1 | 1 | 0 |

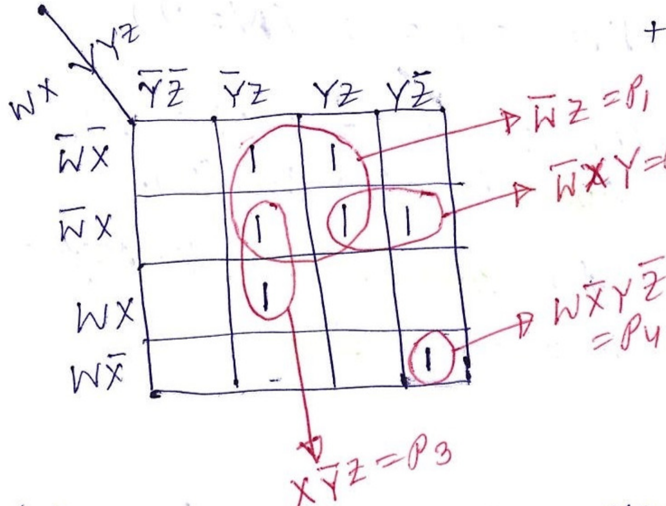
Simplified $F = x\bar{Z} + \bar{x}W + \bar{Z}W$




(6) $F(W, X, Y, Z) = \sum m(1, 3, 5, 6, 7, 10, 13)$

$= \bar{W}\bar{X}\bar{Y}Z + \bar{W}\bar{X}YZ + \bar{W}X\bar{Y}Z + \bar{W}XY\bar{Z}$

$+ \bar{W}XYZ + W\bar{X}Y\bar{Z} + WX\bar{Y}Z$

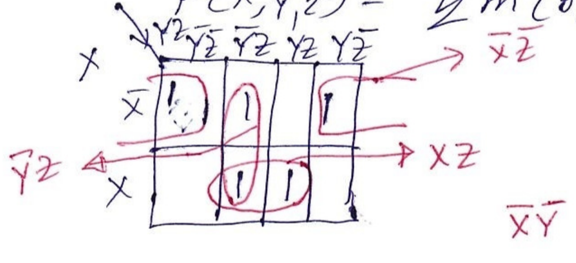


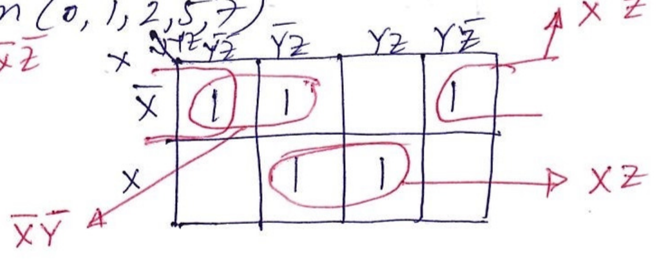

Simplified function:

$F = P_1 + P_2 + P_3 + P_4$

(7) Non-Equal solution: (May not get Unique solution)

$F(X, Y, Z) = \sum m(0, 1, 2, 5, 7)$



## Product-of-Sum Simplification (i.e. POS).

* So far we have ~~ob~~ obtained a minimized Boolean function in SOP form.
* To obtained a minimized Boolean Function in POS form, follow as given below ⇒.
    * Enter the function in K-map.
    * Form groups of adjacent squares of 0s.
    * ~~Simplify~~ Get simplified expression
    * The simplified expression is actually the complement of the function i.e. $\overline{F}$.
    * The complement of $\overline{F}$ gives $\overline{\overline{F}} = F$, which is in POS form.

# Example:

## Truth Table:

pos form obtained by complementing $\overline{F}$

Expressing 0's in (POS) form (directly from K-map) gives F

Expressing 0s in (SOP) form gives $\overline{F}$.

| $F=(\bar{A}+\bar{B})(\bar{C}+\bar{D})(\bar{B}+D)$ | | A | B | C | D | (SOP) $F=\bar{B}\bar{D}+\bar{B}\bar{C}+\bar{A}\bar{C}D$ | (POS) $F=(\bar{A}+\bar{B})(\bar{C}+\bar{D})(\bar{B}+D)$ | $\overline{(SOP)}$ $F'=(B+D)(B+C)(A+C+\bar{D})$ | $F'=AB+CD+B\bar{D}$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 5 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 6 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 7 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 8 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 10 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 11 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 12 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 13 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 14 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 15 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

| AB \ CD | $\bar{C}\bar{D}$ | $\bar{C}D$ | $CD$ | $C\bar{D}$ |
|---|---|---|---|---|
| $\bar{A}\bar{B}$ | 1 | 1 | 0 | 1 |
| $\bar{A}B$ | 0 | 1 | 0 | 0 |
| $AB$ | 0 | 0 | 0 | 0 |
| $A\bar{B}$ | 1 | 1 | 0 | 1 |

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

Simplify the following Boolean Function in

$F(A,B,C,D) = \sum(0,1,2,5,8,9,10)$

(a) Sum-of-products (SOP) and

(b) Product-of-sums (POS) forms.

From the grouping of zeros we get $\Rightarrow$

$\overline{F} = AB + CD + B\bar{D}$

$\overline{(\overline{F})} = F = (\bar{A}+\bar{B})(\bar{C}+\bar{D})(\bar{B}+D) \Rightarrow$ POS representation.

(POS).

# FIVE VARIABLE K- MAP

* Five Variable K-map requires $2^5 = 32$ Squares

## First Representation:- (Method-I)

Fold

| $AB \backslash CDE$ | $\bar C\bar D\bar E$ | $\bar C\bar D E$ | $\bar C DE$ | $\bar C D\bar E$ | $CD\bar E$ | $CDE$ | $C\bar D E$ | $C\bar D\bar E$ |
|---|---|---|---|---|---|---|---|---|
| $\bar A\bar B$ | 0 | 1 | 3 | 2 | 6 | 7 | 5 | 4 |
| $\bar A B$ | 8 | 9 | 11 | 10 | 14 | 15 | 13 | 12 |
| $AB$ | 24 | 25 | 27 | 26 | 30 | 31 | 29 | 28 |
| $A\bar B$ | 16 | 17 | 19 | 18 | 22 | 23 | 21 | 20 |

Fold

* After folding, any two squares that fall one over the other are adjacent.

## Second Representation:- (Method-II).

$\boxed{\bar A = 0}$

| $ABC \backslash DE$ | $\bar D\bar E$ | $\bar D E$ | $DE$ | $D\bar E$ |
|---|---|---|---|---|
| $\bar B\bar C$ | 0 | 1 | 3 | 2 |
| $\bar B C$ | 4 | 5 | 7 | 6 |
| $BC$ | 12 | 13 | 15 | 14 |
| $B\bar C$ | 8 | 9 | 11 | 10 |

$\boxed{A = 1}$

| $ABC \backslash DE$ | $\bar D\bar E$ | $\bar D E$ | $DE$ | $D\bar E$ |
|---|---|---|---|---|
| $\bar B\bar C$ | 16 | 17 | 19 | 18 |
| $\bar B C$ | 20 | 21 | 23 | 22 |
| $BC$ | 28 | 29 | 31 | 30 |
| $B\bar C$ | 24 | 25 | 27 | 26 |

* To visualize the rule for adjacent Squares:
  * Consider two half maps as being one on top of the other
  * Any two squares that fall one over the other are adjacent.

## Ex :

Simplify the Boolean Function :

$$F(A, B, C, D, E) = (0, 2, 4, 6, 9, 13, 21, 23, 25, 29, 31)$$

Method-I. Method-I.

Fold.

| $AB \backslash CDE$ | $\bar{C}\bar{D}\bar{E}$ | $\bar{C}\bar{D}E$ | $\bar{C}DE$ | $\bar{C}D\bar{E}$ | $CD\bar{E}$ | $CDE$ | $C\bar{D}E$ | $C\bar{D}\bar{E}$ |
|---|---|---|---|---|---|---|---|---|
| $\bar{A}\bar{B}$ | 1 | | | 1 | 1 | | | 1 |
| $\bar{A}B$ | 1 | 1 | | | | | 1 | 1 |
| $AB$ | 1 | 1 | | | | 1 | 1 | 1 |
| $A\bar{B}$ | | | | | | 1 | 1 | |

* After folding the Map ⇒
  * We find Three Quad :- They are marked as
    * a, b, and c
* ~~The~~ Each Quad gives three literal terms as follows :-

Quad "a" gives = $ACE$

Quad "b" gives = $\bar{A}\bar{B}\bar{E}$

Quad "c" gives = $B\bar{D}E$

Hence, Simplified function is the logical sum of three product terms ⇒

$$F = ACE + \bar{A}\bar{B}\bar{E} + B\bar{D}E$$

* Same result we may get by Method-II.

# SIX VARIABLE K- Map

$F(A, B, C, D, E, F)$

* Six variable Map requirs $2^6 = 64$ squares.

## Method - I.

folding.

| $ABC \backslash DEF$ | $\bar{D}\bar{E}\bar{F}$ | $\bar{D}\bar{E}F$ | $\bar{D}EF$ | $\bar{D}E\bar{F}$ | $DE\bar{F}$ | $DEF$ | $D\bar{E}F$ | $D\bar{E}\bar{F}$ |
|---|---|---|---|---|---|---|---|---|
| $\bar{A}\bar{B}\bar{C}$ | 0 1 | 1 | 3 | 2 | 6 | 7 | 5 | 4 1 |
| $\bar{A}\bar{B}C$ | 8 | 9 | 11 | 10 | 14 | 15 | 13 | 12 |
| $\bar{A}BC$ | 24 | 25 | 27 | 26 | 30 | 31 | 29 | 28 |
| $\bar{A}B\bar{C}$ | 16 | 17 | 19 | 18 | 22 | 23 | 21 | 20 |
| $AB\bar{C}$ | 48 | 49 | 51 | 50 | 54 | 55 | 53 | 52 |
| $ABC$ | 56 | 57 | 59 | 58 | 62 | 63 | 61 | 60 |
| $A\bar{B}C$ | 40 | 41 | 43 | 42 | 46 | 47 | 45 | 44 |
| $A\bar{B}\bar{C}$ | 32 1 | 33 | 35 | 34 | 38 | 39 | 37 | 36 1 |

Folding

Folding

Folding.

* After double folding, any four squares that fall one over the other are adjacent.

## Method - II.

$\boxed{B=0}$

$\boxed{A=0}$

| $ABCD \backslash EF$ | $\bar{E}\bar{F}$ | $\bar{E}F$ | $EF$ | $E\bar{F}$ |
|---|---|---|---|---|
| $\bar{C}\bar{D}$ | 0 | 1 | 3 | 2 |
| $\bar{C}D$ | 4 | 5 | 7 | 6 |
| $CD$ | 12 | 13 | 15 | 14 |
| $C\bar{D}$ | 8 | 9 | 11 | 10 |

$\boxed{B=1}$

A=0

| $ABCD \backslash EF$ | $\bar{E}\bar{F}$ | $\bar{E}F$ | $EF$ | $E\bar{F}$ |
|---|---|---|---|---|
| | 16 | 17 | 19 | 18 |
| | 20 | 21 | 23 | 22 |
| | 28 | 29 | 31 | 30 |
| | 24 | 25 | 27 | 26 |

B=0

$\boxed{A=1}$

| $ABCD \backslash EF$ | | | | |
|---|---|---|---|---|
| $\bar{C}\bar{D}$ | 32 | 33 | 35 | 34 |
| $\bar{C}D$ | 36 | 37 | 39 | 38 |
| $CD$ | 44 | 45 | 47 | 46 |
| $C\bar{D}$ | 40 | 41 | 43 | 42 |

B=1

A=1

| $ABCD \backslash EF$ | | | | |
|---|---|---|---|---|
| | 48 | 49 | 51 | 50 |
| | 52 | 53 | 55 | 54 |
| | 60 | 61 | 63 | 62 |
| | 56 | 57 | 59 | 58 |

* Consider Four Quarter maps as being one on top of the other.
* Any FOUR SQUARES that fall one over the other are ADJACENT.

# Incompletely Specified Functions (Don't Cares)

* completely specified functions : When functional o/ps are 0 or 1 for all possible combinations of ~~boolean functions~~ switching variables.

* Non-essential functions :

  * When certain combinations of i/p logic variables never occur. [Like "BCD Decoder"] 0000 — 1001
    10 → 15 ⇒ Do not occur.

  * When all the combinations of i/p logic variables occur, but certain combinations are Non-essential for the outputs.

In the design of Logic Circuits, Non-essential minterms (maxterms) may be introduced so as to simplify the Logic Circuit. ⇒

⇒ Such non-essential minterms (maxterms) are called Don't Cares (X). terms.

Ex:1 Function:

$$F(A,B,C) = \sum m(0,1,5,7) + \sum X(2,4).$$

| $A \backslash BC$ | $\bar{B}\bar{C}$ | $\bar{B}C$ | $BC$ | $B\bar{C}$ |
|---|---|---|---|---|
| $\bar{A}$ | 1 | 1 | 0 | X |
| $A$ | X | 1 | 1 | 0 |

$\bar{B}$ $AC$ $(\bar{A}+C)$ $(A+\bar{B})$

Simplified function (SOP):

$F = \bar{B} + AC$ (SOP).

$F = (A+\bar{B})(\bar{A}+C)$ (POS).

Ex2. Find out the simplified SOP & POS expressions for

$$F(A,B,C,D) = \Pi M(0,1,4,6,8,14,15) + \Pi X(2,3,9)$$

| $AB \backslash CD$ | $\bar{C}\bar{D}$ | $\bar{C}D$ | $CD$ | $C\bar{D}$ |
|---|---|---|---|---|
| $\bar{A}\bar{B}$ | 0 | 0 | X | X |
| $\bar{A}B$ | 0 | 1 | 1 | 0 |
| $AB$ | 1 | 1 | 0 | 0 |
| $A\bar{B}$ | 0 | X | 1 | 1 |

$\bar{A}BD$ $AB\bar{C}$ $AC$

$F = AC + AB\bar{C} + \bar{A}BD$ (SOP).

| $AB \backslash CD$ | $\bar{C}\bar{D}$ | $\bar{C}D$ | $CD$ | $C\bar{D}$ |
|---|---|---|---|---|
| $\bar{A}\bar{B}$ | 0 | 0 | X | X |
| $\bar{A}B$ | 0 | 1 | 1 | 0 |
| $AB$ | 1 | 1 | 0 | 0 |
| $A\bar{B}$ | 0 | X | 1 | 1 |

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

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

# Eliminating Redundant group :–

After you've finished encircling groups, there's one more thing you should do before writing the simplified Boolean equation : eliminate any group whose 1's are completely overlapped by other groups. $F = \sum m(3, 4, 5, 7, 9, 13, 14, 15)$.

Example :

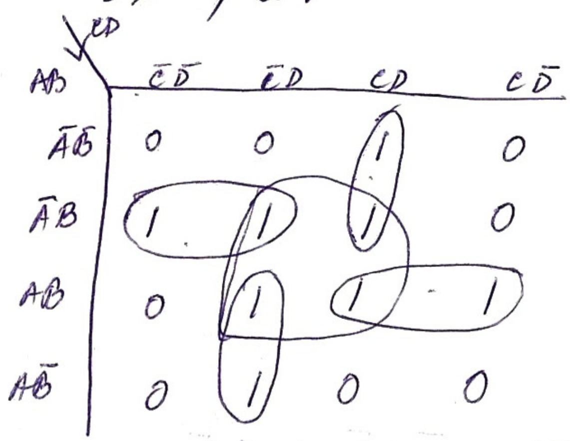
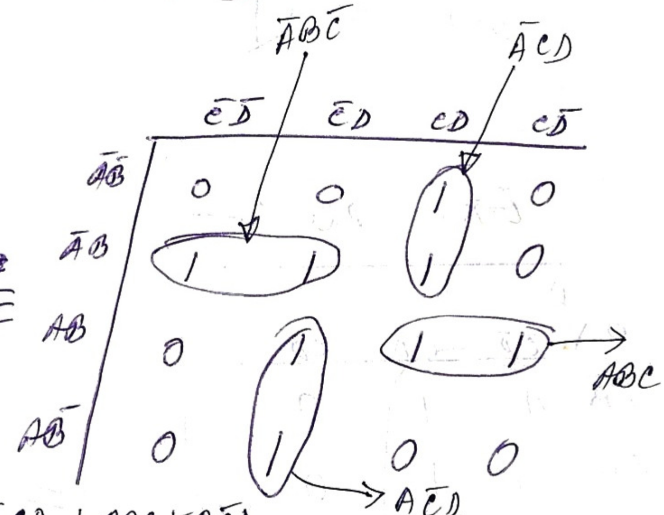



Fig-1

$\equiv$




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

Fig-2

Here, ~~quad~~ after reviewing fig.1 we see that quad is redundant i.e all of its 1s overlapped by other groups. So this quad can be eliminated.

The Boolean equation of fig. 2 will contain one less product than the equation for fig.1. Therefore, fig-2 is the most efficient way to group the 1s.

## PRIME Implicant (Implicate) :

An implicant (Implicate) of a function F is said to be prime implicant (or prime implicate) of F if it is not completely enclosed in a biggers valid group of minterms (or maxterms) in the K-map.

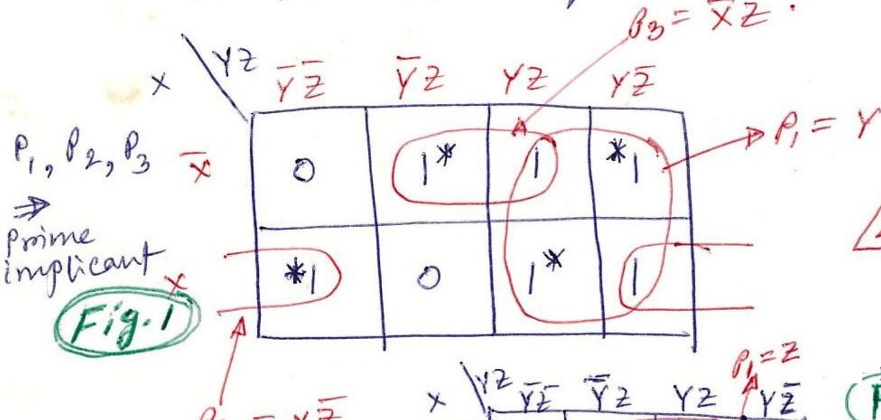



Fig. 1

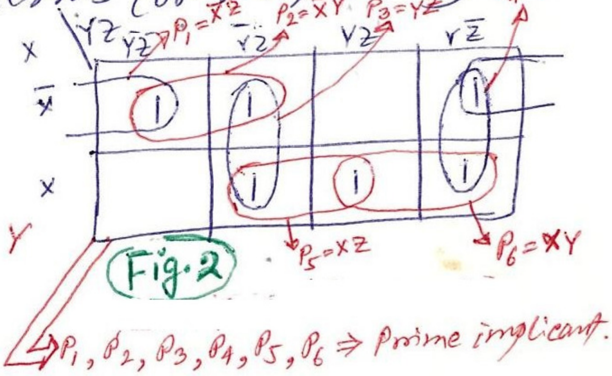



Fig. 2

$P_1, P_2, P_3, P_4, P_5, P_6 \Rightarrow$ Prime implicant.

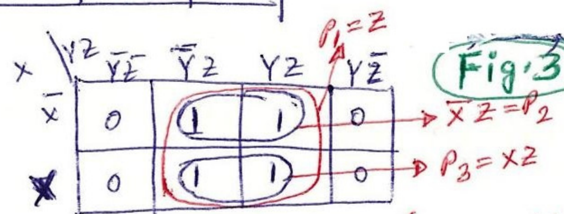



Fig. 3

$P_1 \Rightarrow$ Prime implicant.

$P_2, P_3 \Rightarrow$ Not prime implicant (They are simply implicants)

## Implicant (Implicate) :

**Implicant :** For a Boolean function F expressed in SOP form, a product term "p" is an implicant of the function F, if and only if F=1 ~~for every combination of~~ when p=1.

Hence every (monad), pair (diad), Quad, octet etc. are implicant of the function.

**Implicate :**

For a Boolean function F expressed in POS form, a sum term "s" is an implicate of the function F, if and only if F=0 when S=0.

Hence, every maxterm, monad, diad, Quad etc. are implicate of the function.

## Essential Prime implicant (implicate) :- (EPI)

A prime implicant (implicate) is said to be Essential Prime Implicant (implicate), if and only if the loop for the prime implicant (implicate) contains at least one minterm (maxterm) that is not covered in any other prime implicant (implicate) loop.

Ex:

In Fig. 1 ⇒ $P_1, P_2, P_3$ ⇒ are all EPIs.

In Fig. 2 ⇒ There is no EPI.

In Fig. 3 ⇒ $P_1$ ⇒ is EPI.

## Distinguished Minterm (Maxterm):

Any isolated minterm (maxterm) or any minterm (maxterm) of EPI not covered in any other prime Implicant (Implicate) loop is called Distinguished Minterm (maxterm).

Ex:

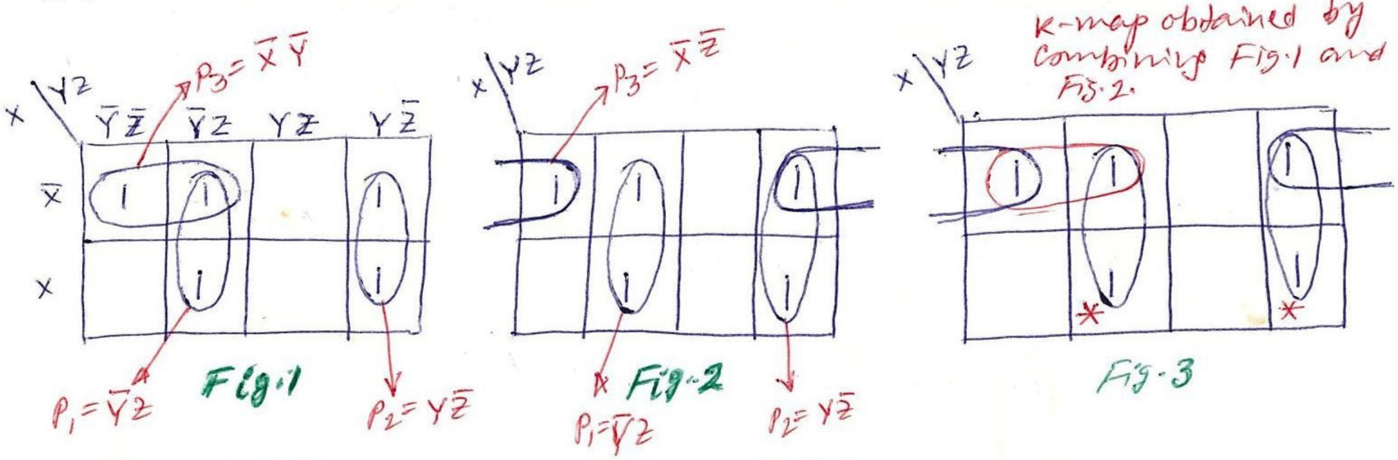
* In fig. 1 ⇒ $m_1$, $m_2$, $m_4$ and $m_7$ ⇒ are Distinguished Minterms.

* Distinguished minterms are marked with stars (*).

## Non-EPI (or Optional Prime Implicant) :

Non-EPIs are those PIs which are used for alternative minimum cover.

Illustration:




$F = \bar{Y}Z + Y\bar{Z} + \bar{X}\bar{Y}$ $\qquad$ $F = \bar{Y}Z + Y\bar{Z} + \bar{X}\bar{Z}$

Here, * $P_1$ and $P_2 \Rightarrow$ EPIs.

and * $\bar{X}\bar{Y}$ and $\bar{X}\bar{Z}$ are Non-EPIs ($P_3$)

* Once one alternative minimum cover has been selected $\Rightarrow$
  * none of the remaining Non-EPIs can be used for minimum cover in that Boolean expression.

## Minimal Cover :

A minimal cover of a Boolean function F consists of all EPIs and some Non-EPIs that cover all the minterm squares of a K-map.

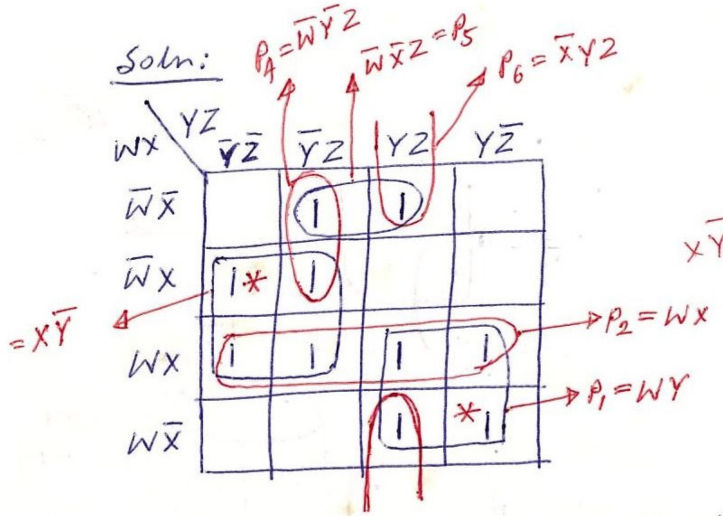
**Conclusion:**
* A minimal solution of a function is not always UNIQUE
* Because the solution may contain some non-EPIs
* And the non-EPIs are not-UNIQUE.

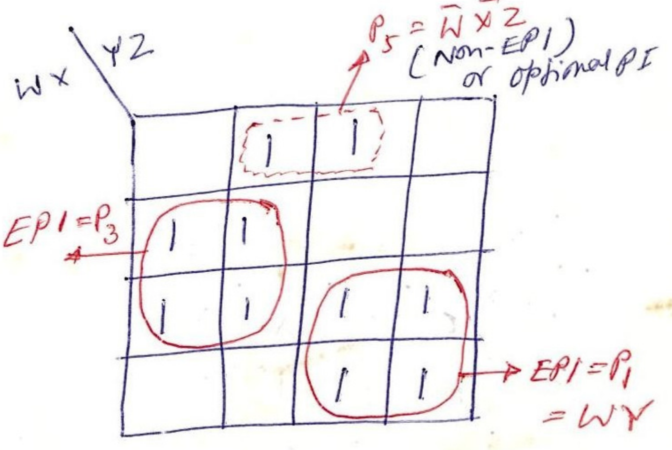
**Problem: 1**

Find out the prime implicants, EPI and minimal SOP expression for the function

$$F(W,X,Y,Z) = \sum(1,3,4,5,10,11,12,13,14,15).$$

Soln:




$EPI = P_1, P_3$, Non-EPI $= P_5$

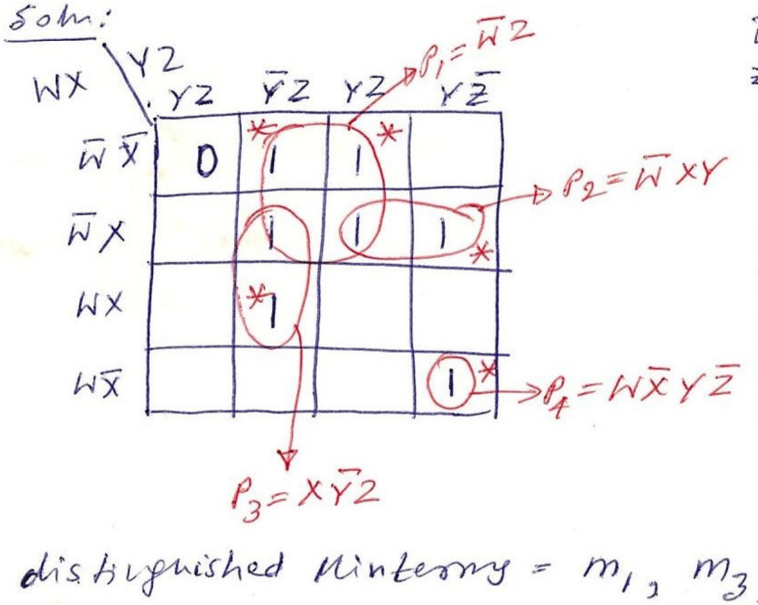
Minimal Form (SOP) $= F = WY + X\bar{Y} + \bar{W}\bar{X}Z$.

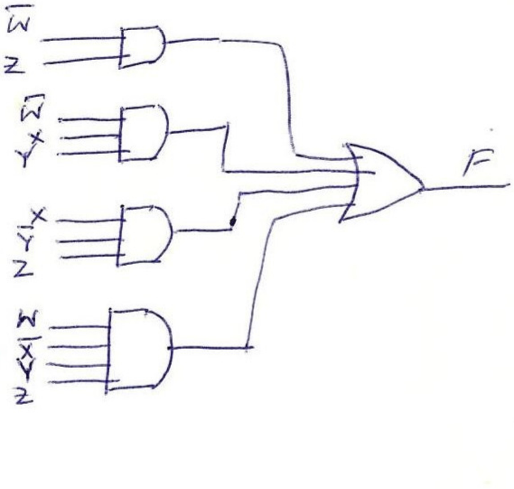
**Problem: 2**

Given $F(W,X,Y,Z) = \sum(1,3,5,6,7,10,13)$

Find the distinguished minterm, the EPI and the minimal solution.

Soln:




distinguished Minterms $= m_1, m_3, m_6, m_{10}, m_{13}$

EPIs = All ~~minterms~~ prime implicant $= P_1, P_2, P_3, P_4$

Minimal solution $= F = P_1 + P_2 + P_3 + P_4 = \bar{W}Z + \bar{W}XY + X\bar{Y}Z + W\bar{X}Y\bar{Z}$

**Problem:** Find out the minimal SOP and POS expression for the given function

$$F(W,x,y,z) = \prod M(0,1,4,6,8,14,15), \ \prod X(2,3,9)$$

Soln

$P_2 = \bar{W}XZ$

| WX \ YZ | $\bar{Y}\bar{Z}$ | $\bar{Y}Z$ | $YZ$ | $Y\bar{Z}$ |
|---|---|---|---|---|
| $\bar{W}\bar{X}$ | 0 | 0 | X | X |
| $\bar{W}X$ | 0 | 1 | 1 | 0 |
| $WX$ | 1 | 1 | 0 | 0 |
| $W\bar{X}$ | 0 | X | 1 | 1 |

$P_1 = \bar{X}Y$

$P_3 = WX\bar{Y}$

Minimal SOP form of the function is

$$F_{sop}(W,X,Y,Z) = \bar{X}Y + WX\bar{Y} + \bar{W}XZ$$

$W+Z$

| WX \ YZ | $\bar{Y}\bar{Z}$ | $\bar{Y}Z$ | $YZ$ | $Y\bar{Z}$ |
|---|---|---|---|---|
| $\bar{W}\bar{X}$ | 0 | 0 | X | X |
| $\bar{W}X$ | 0 | 1 | 1 | 0 |
| $WX$ | 1 | 1 | 0 | 0 |
| $W\bar{X}$ | 0 | X | 1 | 1 |

$(\bar{W}+\bar{X}+\bar{Y})$

$(X+Y)$

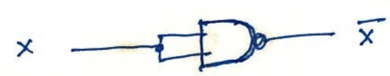
Minimal function in POS form

$$F_{pos} = (W+Z)(X+Y)(\bar{W}+\bar{X}+\bar{Y})$$

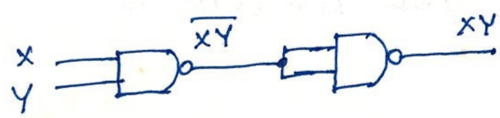
# NAND–NAND implementation of LOGIC CIRCUIT

* (AND - OR - NOT) set of operators combination is said to be functionally complete.
* Because any switching (Boolean) function may be expressed by these operators.

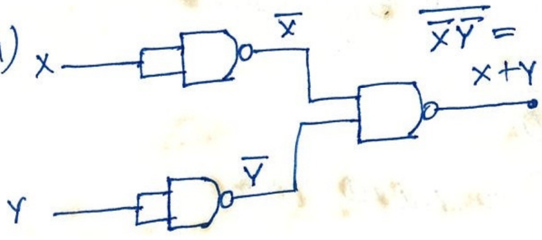
* NAND gate is said to be universal gate ⇒
  * because any digital system can be implement with it.
  * i.e. AND, OR and NOT can be implemented with NAND.

(i) 

NOT.

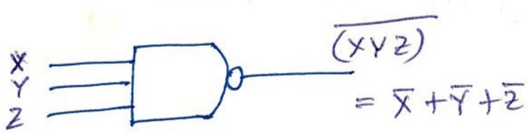
(ii) 

AND

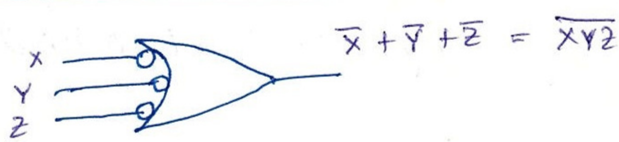
(iii) 

OR

Two equivalent graphic symbol of NAND gate :–




AND-invert
(NAND gate).

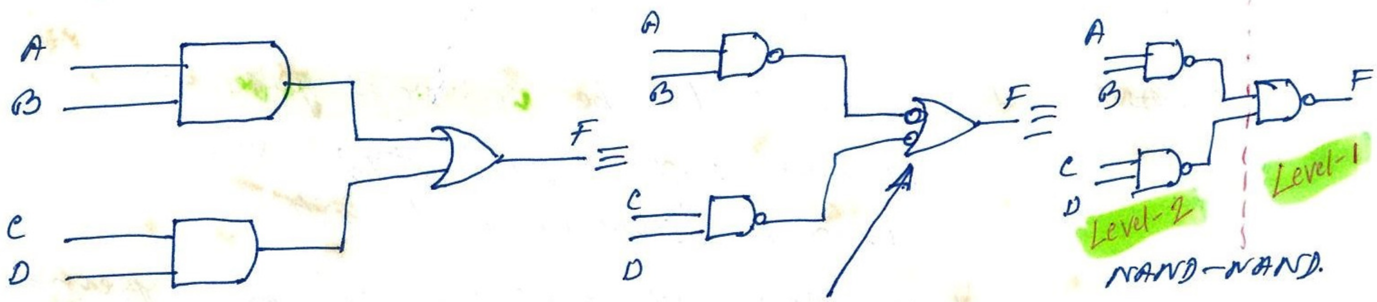



INVERT–OR
(NAND-gate)

# Two level NAND-NAND implementation!

Case-I Function is purely Sum-of-Product Form : $(F = AB + CD)$.
Implementation :

(i) AND-OR to NAND-NAND Conversion:-




AND-OR (a)

NAND — NAND. (Invert-OR graphic symbol). (b)

NAND-NAND. (c).

Fig. 1.

(ii) Direct from Boolean Expression:-

* Express the Boolean function in term of NAND logic

i.e. $F = AB + CD$. Can be expressed as

$$\overline{\overline{F}} = F = \overline{\overline{AB + CD}}$$

$$= \overline{(\overline{AB})(\overline{CD})} \Rightarrow \text{NAND logic}$$

* This can be implemented directly by 2-Level NAND-NAND structure (Fig. 2).

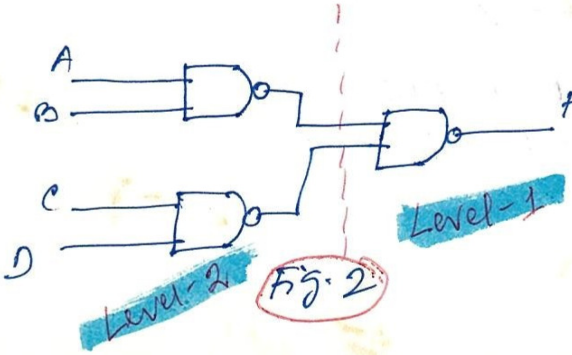



Fig. 2

* Comparing Fig1(a) and Fig1(b) ⇒
    * We see that ⇒
    * Conversion from 2-level AND-OR circuit to NAND-NAND circuit is ⇒
        * just replacement of AND and OR gates by NAND gates.

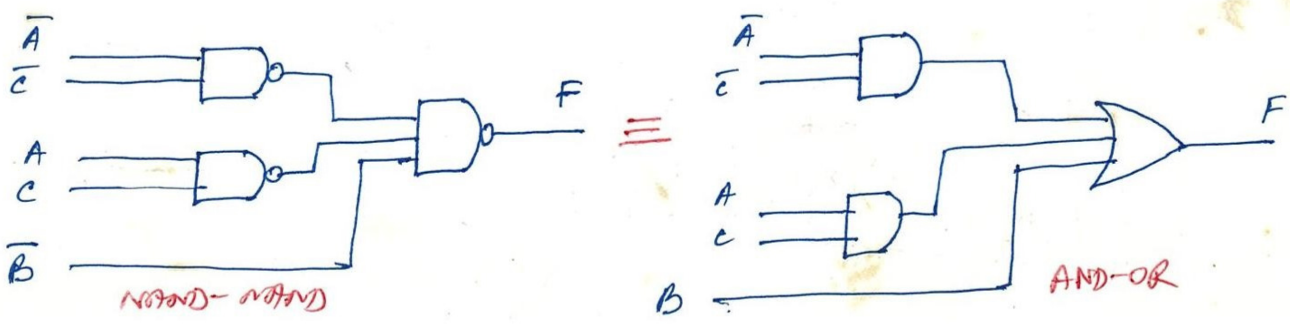
* The reverse is also true ⇒
    * i.e. Conversion from NAND-NAND to AND-OR is done ⇒
        * just by replacing ⇒
            * level-2 NAND by AND
        * and * level-1 NAND by OR.

Case-II Product terms along with terms of single literal.

Problem, $F = \bar{A}\bar{C} + AC + B$ implement with NAND-NAND circuit and AND-OR circuit

Soln:

$$F = \bar{\bar{F}} = \overline{\overline{\bar{A}\bar{C} + AC + B}} = \overline{(\overline{\bar{A}\bar{C}}) \cdot \overline{AC} \cdot \bar{B}}$$



## Steps to design a minimum 2-level NAND-NAND Circuit :

1. Simplify the function and express in SOP form.
2. Assign one NAND gate for each product term in level-2
3. Connect output of all the NAND gate of level-2 to the inputs of another NAND gate in level-1.
4. Complement the single literal terms, if any, and connect to the input of the NAND gate in level-1.

## Multi level NAND Circuits

### Block Diagram Method :

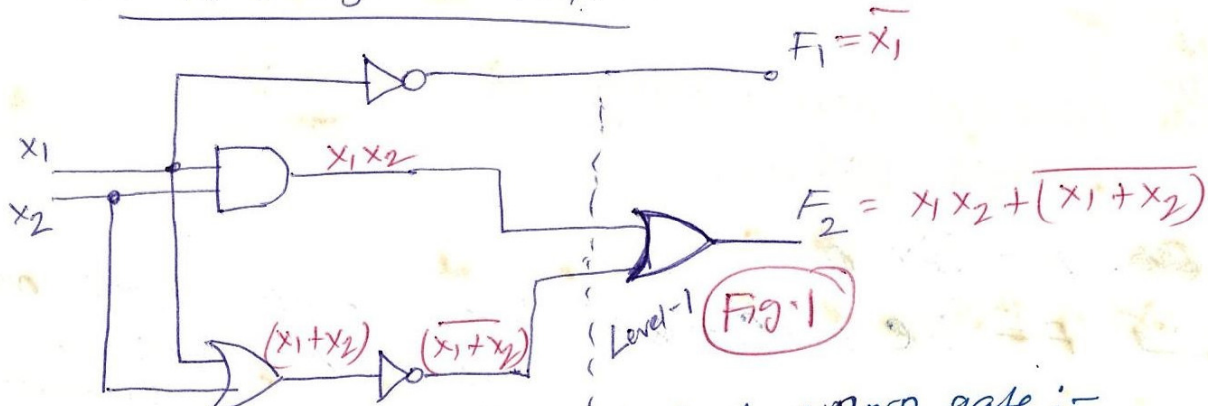



Fig. 1

Replace each gate by its equivalent NAND gate :-

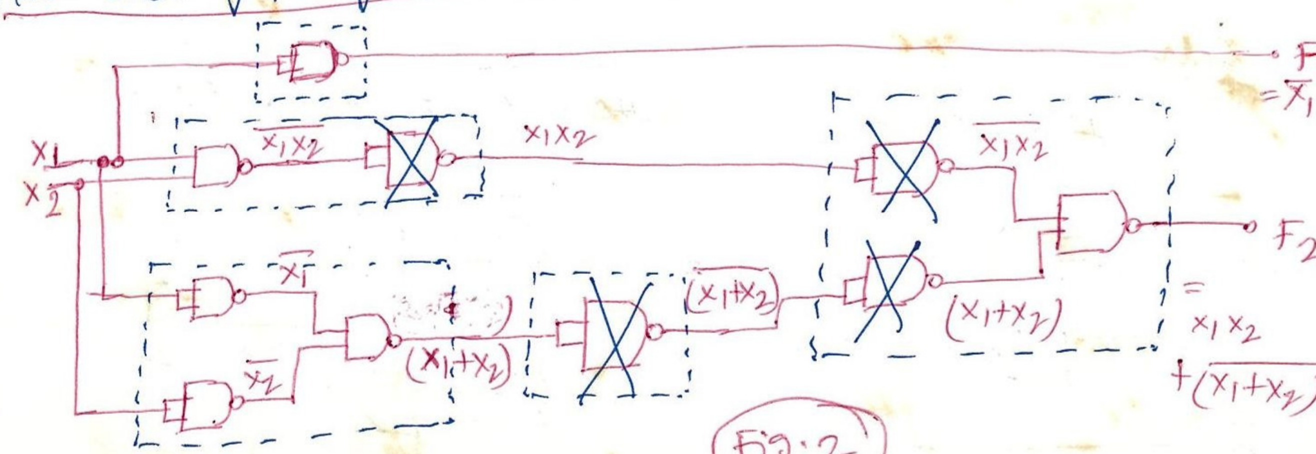



Fig. 2

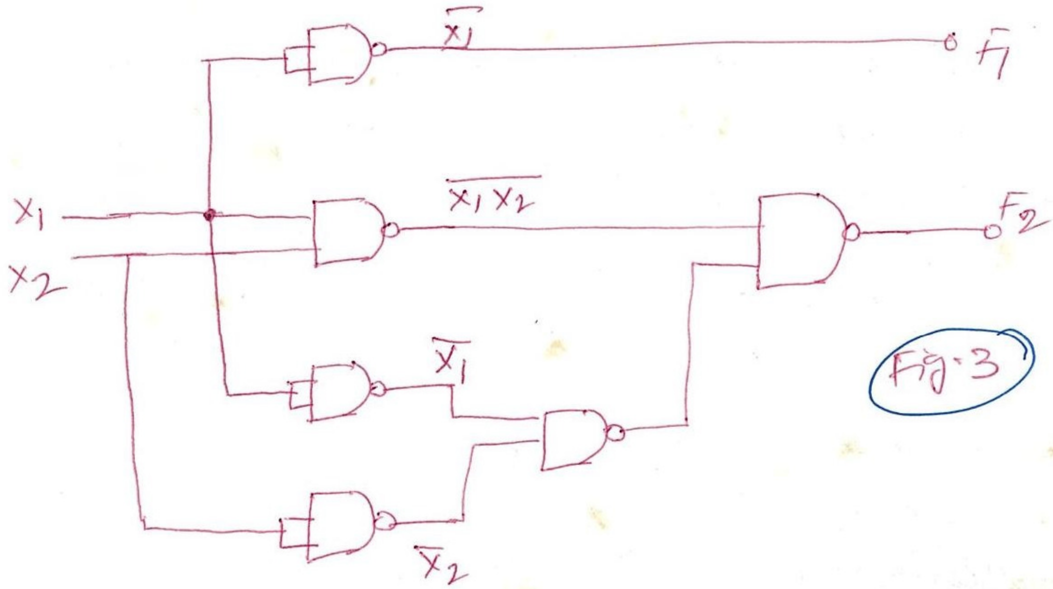
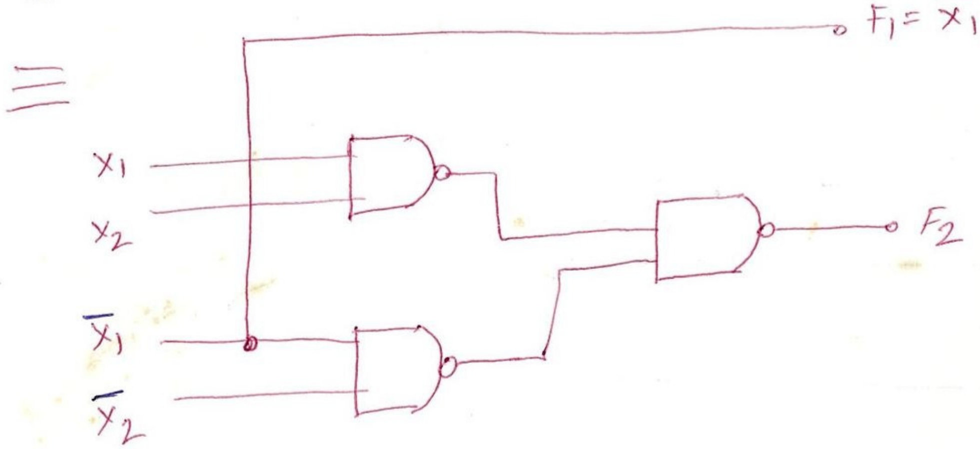



Fig. 3

* if double rail inputs are available
* Then ckt of fig. 3 further be simplified ⇒.




Steps to design a Multilevel NAND - NAND circuit from AND - OR - NOT Circuit :

1. Assign one NAND Gate for each AND Gate.
2. Assign one NAND Gate for each OR Gate.
3. Assign one NAND Gate (single input) for every NOT Gate.
4. If the gate of Level-1 is an AND Gate then put one NAND Gate as Inverter ~~(single input)~~ at the end.
5. If the gate at the Level-1 is an OR gate then no extra NAND gate is required at the end.

6. If NOT Gate is there between two same gates then NOT is to be ignored. Provided any output is not taken from the output of NOT Gate.
7. If NOT Gate is there between two opposite Gates, then an Inverter has to be inserted. Provided no output is taken the output of NOT Gate.
8. Any external inputs to OR Gate are to be complemented while implementing with NAND Gate.

Problem:1 Implement the function by AND-OR-NOT and then convert to NAND-NAND:-

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

Soln: AND-OR-NOT

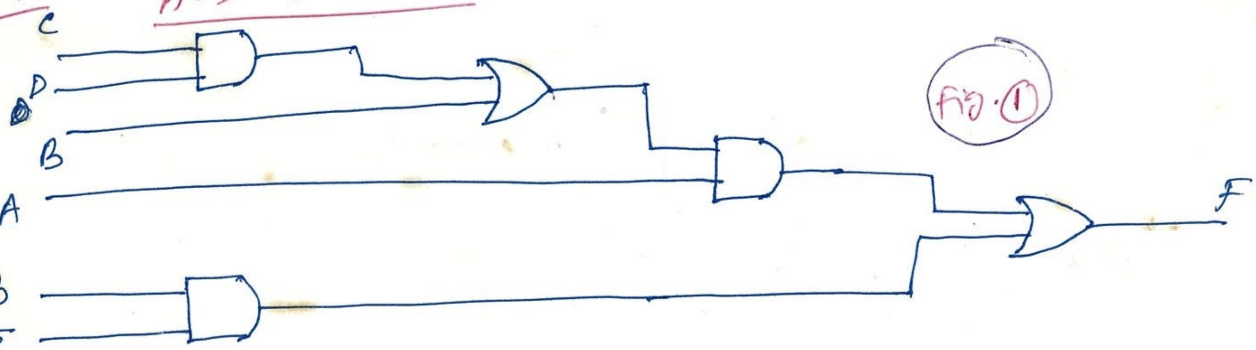
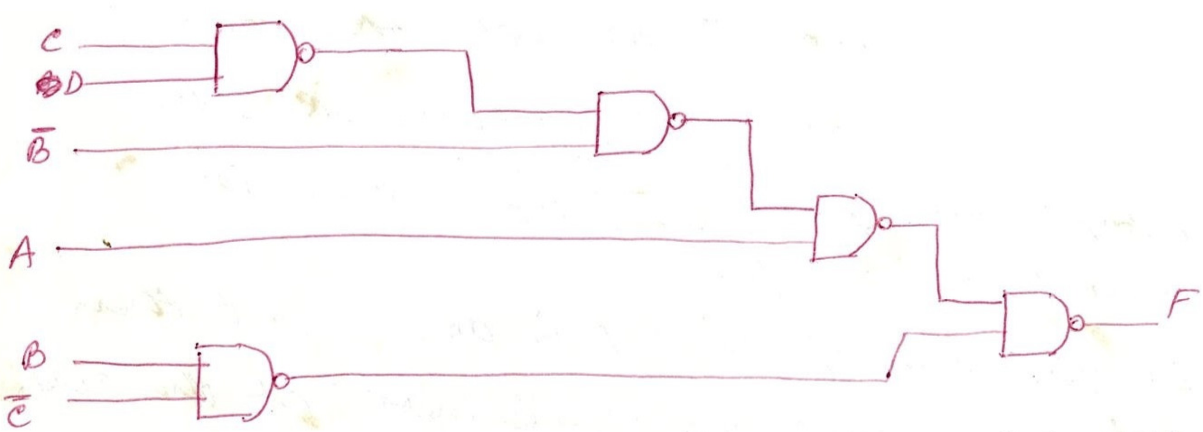



fig. 1

Soln: NAND-NAND implementation from fig 1 ⇒



## Problem. 2

$$F = (A+\bar{B})(CD+E)$$

Implement the function by AND-OR-NOT and NAND-NAND gates:

**Soln:**

AND-OR-NOT:

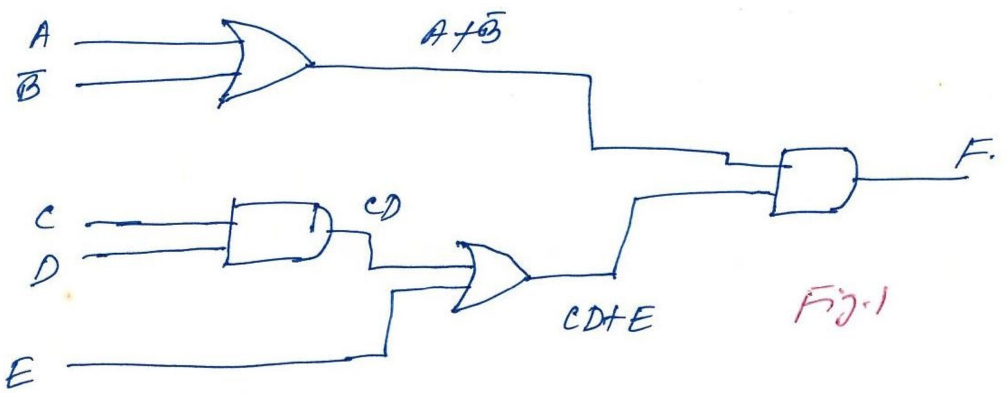



Fig. 1

NAND-NAND implementation.

(i) Direct from AND-OR-NOT ⇒.

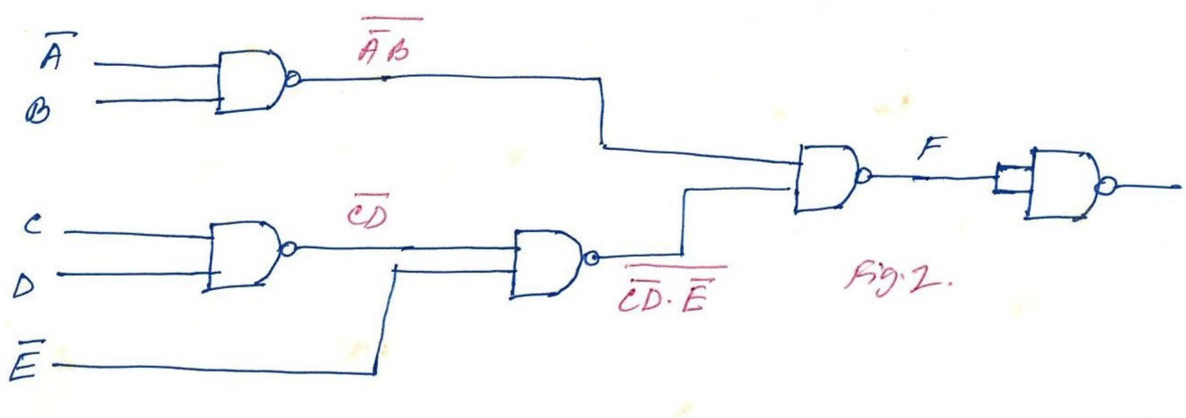



Fig. 2.

(ii) By expressing NAND logic Boolean Expression:

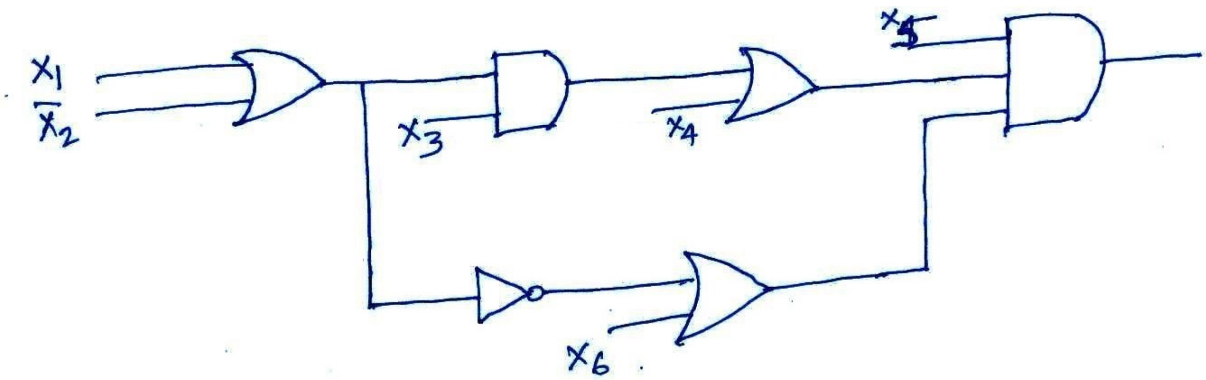
$$F = \overline{\overline{F}} = \overline{\overline{(A+\bar{B})(CD+E)}} = \overline{\overline{(A+\bar{B})} + \overline{(CD+E)}} = \overline{\bar{A}B + \overline{CD}\cdot\bar{E}}$$

$$= \overline{(\bar{A}B)}\;\overline{(\overline{CD}\cdot\bar{E})} \Rightarrow \text{NAND logic Expression.}$$

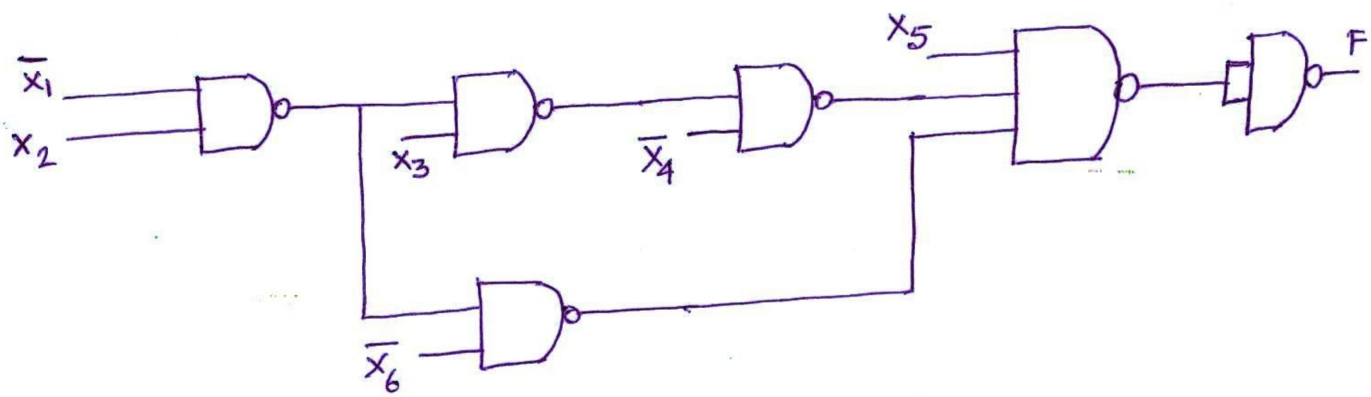
By inspection of this expression we will get the NAND circuit of fig. 2.

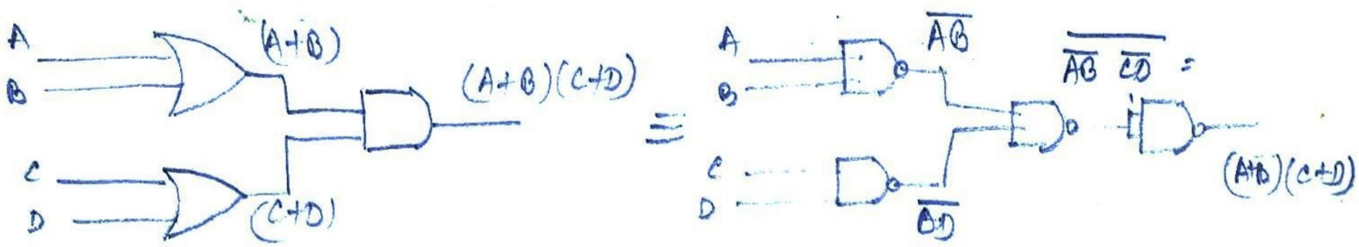
## Problem 3

Implement the given AND-OR-NOT circuit with NAND gates.




**Solution:** By inspection of AND-OR-NOT circuit ⇒.

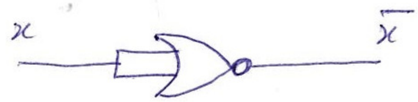


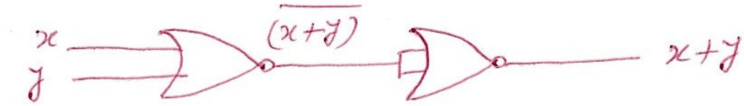
# NOR Implementation of LOGIC CIRCUIT

* NOR gate is another universal gate.
* NOR implementation is just the Dual of the NAND implementation.
* Hence all procedures and rules for NOR logic are the dual of the corresponding procedures and rules developed for NAND logic implementation.

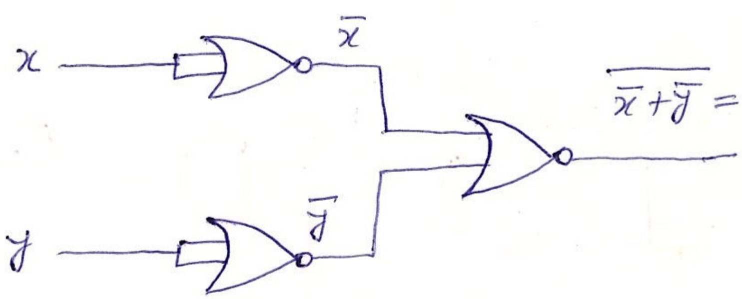
NOT (inverter) ⇒



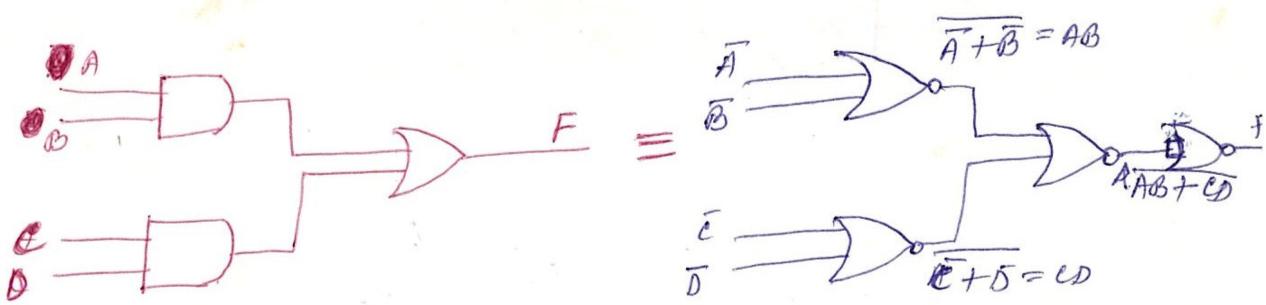

OR ⇒




AND ⇒




**Ex:**




$F = AB + CD$

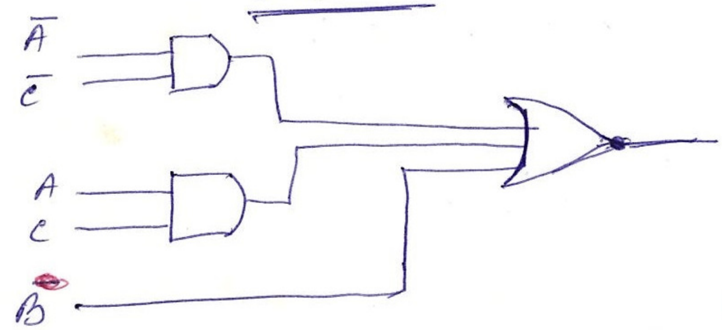
$\therefore F = AB + CD.$

**Ex:**

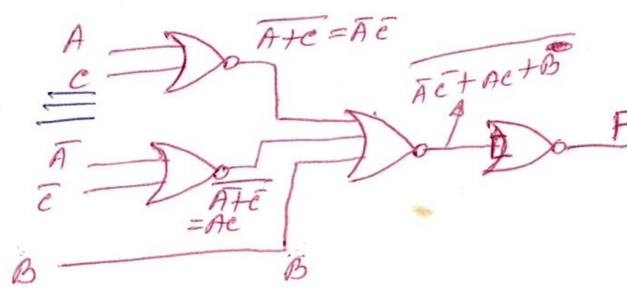
$F = \bar{A}\bar{C} + AC + B$ implement with NOR Circuit and AND-OR Circuit

**Soln:**

AND-OR.



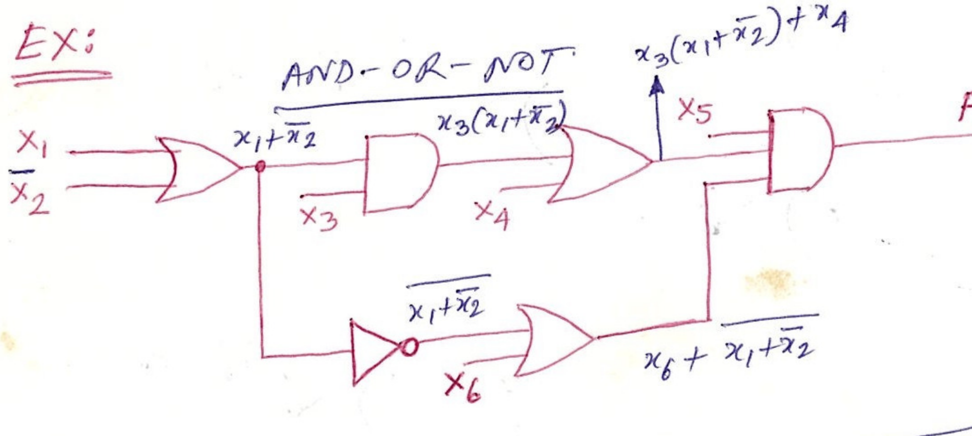

NOR.




$\therefore F = \bar{A}\bar{C} + AC + B$

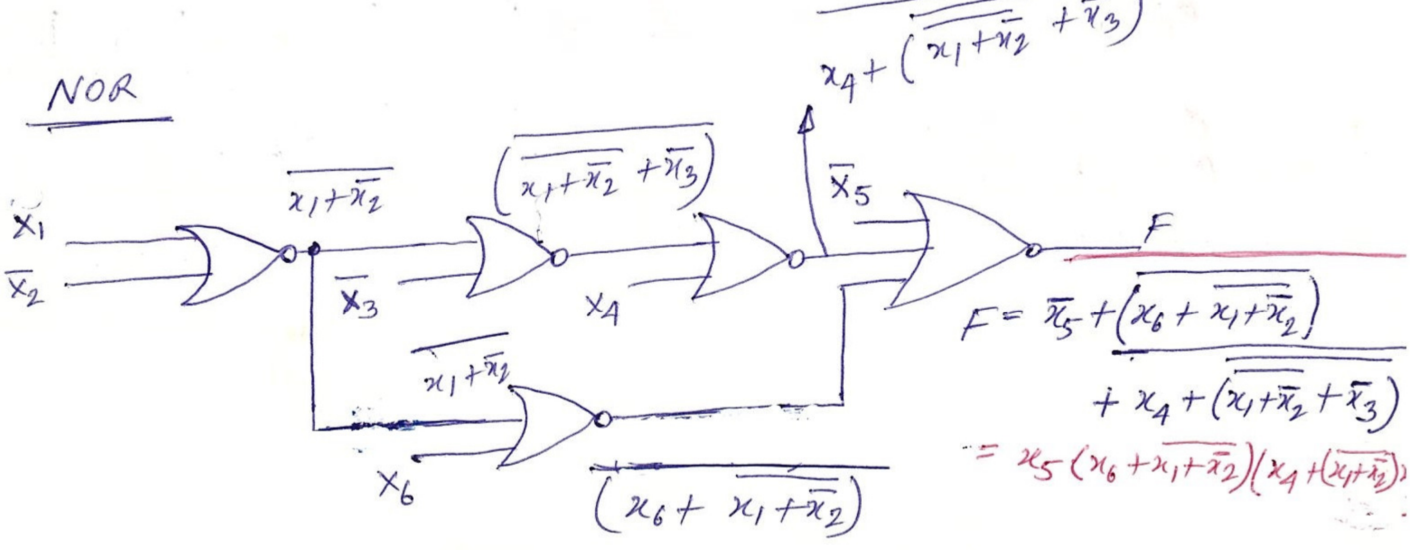
**Ex:**

AND-OR-NOT




$F = x_5(x_6 + \overline{x_1+\bar{x}_2})$
$[x_4 + x_3(x_1+\bar{x}_2)]$

NOR




$F = \overline{\bar{x}_5 + \overline{(x_6 + \overline{x_1+\bar{x}_2})}}$
$\overline{+ x_4 + (\overline{\overline{x_1+\bar{x}_2}+\bar{x}_3})}$
$= x_5(x_6+\overline{x_1+\bar{x}_2})(x_4+(\overline{x_1+\bar{x}_2})$

Problem:

Realize the cost effective 2-level digital circuit for the function $F = \sum m(1,2,3,9,10,11)$ using AND and OR gates.

$F = \bar{X} + \bar{Y}\bar{Z}$

$F = \bar{X}(Y+Z)$

$\bar{F} = \overline{\bar{X} \cdot (Y+Z)}$

$\bar{F} = \overline{\bar{X}} + \overline{Y+Z}$

$= X + \bar{Y}\bar{Z}$

Soln:

| WX \ YZ | $\bar{Y}\bar{Z}$ | $\bar{Y}Z$ | $YZ$ | $Y\bar{Z}$ |
|---|---|---|---|---|
| $\bar{W}\bar{X}$ | 0 | 1 | 1 | 1 |
| $\bar{W}X$ | 0 | 0 | 0 | 0 |
| $WX$ | 0 | 0 | 0 | 0 |
| $W\bar{X}$ | 0 | 1 | 1 | 1 |

$\bar{X}Z$ $\bar{X}Y$

Simplified function in SOP form

$F_{SOP} = \bar{X}Z + \bar{X}Y$

| WX \ YZ | $\bar{Y}\bar{Z}$ | $\bar{Y}Z$ | $YZ$ | $Y\bar{Z}$ |
|---|---|---|---|---|
| $\bar{W}\bar{X}$ | 0 | 1 | 1 | 1 |
| $\bar{W}X$ | 0 | 0 | 0 | 0 |
| $WX$ | 0 | 0 | 0 | 0 |
| $W\bar{X}$ | 0 | 1 | 1 | 1 |

$\bar{X}$

$(Y+Z)$

Simplified function in POS form

$F_{POS} = \bar{X} \cdot (Y+Z)$

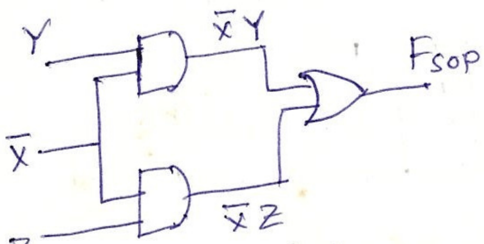



Fig. 1(a)

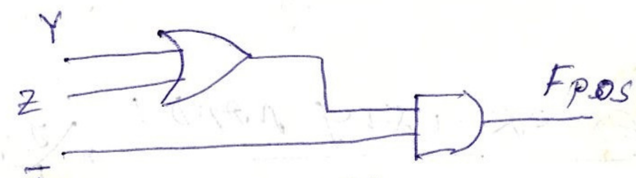



Fig. 1(b)

$F_{SOP} = F_{POS}$

Comparing Fig. 1(a) and Fig. 1(b) we see that $\Rightarrow$

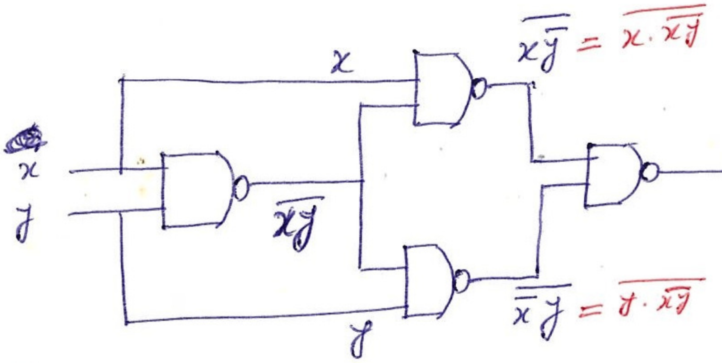
* The circuit of fig 1(b) is cost effective.

# Implementation of XOR with NAND & NOR.

**XOR with NAND:** $F = \bar{x}y + x\bar{y}$, $F = \bar{\bar{F}} = \overline{\overline{\bar{x}y + x\bar{y}}} = \overline{\overline{\bar{x}y}\cdot\overline{x\bar{y}}}$

$$= \overline{\overline{y\cdot\overline{xy}}\;\overline{x\,\overline{xy}}}$$

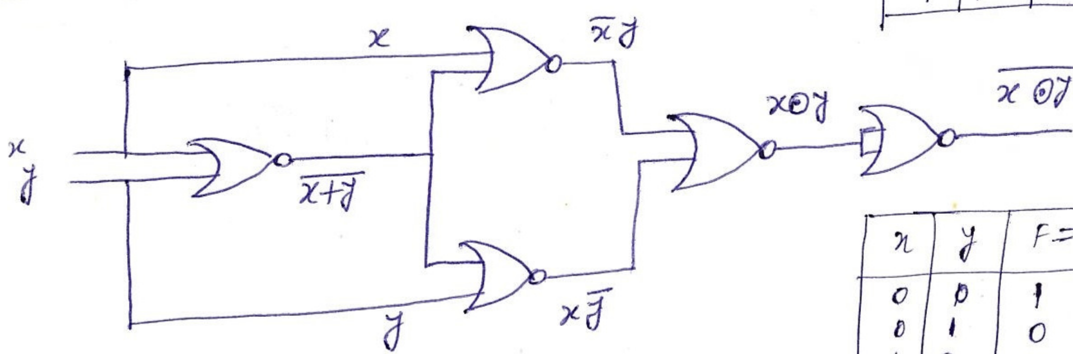
$[\because y\cdot\overline{xy} = y(\bar{x}+\bar{y}) = \bar{x}y + y\bar{y} = \bar{x}y]$




$$F = \overline{\overline{x\bar{y}}\cdot\overline{\bar{x}y}} = x\bar{y} + \bar{x}y = x \oplus y.$$

| $x$ | $y$ | $F = x \oplus y = \overline{x \odot y}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

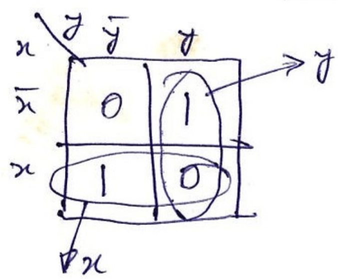
**XOR with NOR.**




| $x$ | $y$ | $F = x \odot y$ | $= \overline{x \oplus y}$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

## EX-OR using NAND:

$F = \bar{x}y + x\bar{y}$




$$F = x(\overline{xy}) + y(\overline{xy})$$

$$\bar{\bar{F}} = \overline{\overline{x(\overline{xy}) + y(\overline{xy})}} = \overline{\overline{x(\overline{xy})}\cdot\overline{y(\overline{xy})}}$$